Book Series in Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity

VOLUME I

MODEL-BASED AND SIGNAL-BASED INVERSE METHODS

EDITORS: ARIOSTO B. JORGE, CARLA T.M. ANFLOR,
GUILHERME F. GOMES, and SERGIO H.S. CARNEIRO

University of Brasilia (UnB)
Post-Graduate Program - Integrity of Engineering Materials

Book Series in Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity

Volume I

Model-based and Signal-Based Inverse Methods

Book Series Editors

Ariosto Bretanha Jorge (leading editor)
Carla Tatiana Mota Anflor
Guilherme Ferreira Gomes
Sérgio Henrique da Silva Carneiro

This Book Series is an initiative of the Post-Graduate Program - Integrity of Engineering Materials (PPG-Integridade) of University of Brasilia (UnB)

With the incentive, encouragement and support from

- Latin American Journal of Solids and Structures (LAJSS) (www.lajss.org)
- Brazilian Society of Mechanical Sciences and Engineering (ABCM) (www.abcm.org.br)
- Brazilian Association of Computational Methods in Engineering (ABMEC) (www.abmec.org.br).

Brasília – DF
Open access digital book published by
University of Brasilia Central Library (BCE/UnB)
2022

Cover page pictures, clockwise from top left: a SAAB Gripen (Brazilian version, monoplace); an EMBRAER's KC-390 transport aircraft, a Brazilian-manufactured Riachuelo-class submarine, a nuclear powerplant at Angra dos Reis, Brazil; an offshore platform operated by PETROBRAS; and a generator in a wind power plant in the northeast region of Brazil.

Several engineering fields of application, such as aerospace, naval, nuclear, energy, mechanical, civil, etc, may take advantage of the technology, innovation and research in the areas discussed in this book.

All pictures in the cover page are public domain.

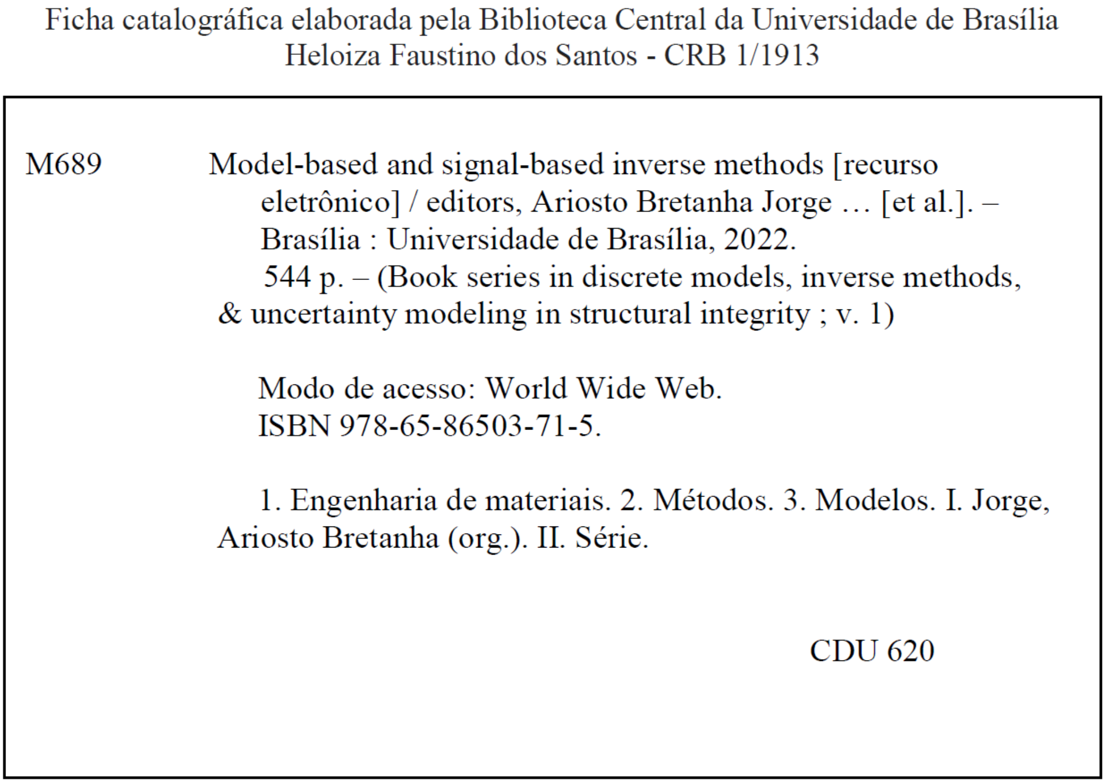
Jorge, Ariosto B.; Anflor, Carla T. M.; Gomes, Guilherme F.; Carneiro, Sergio H. S. (Eds.): Model-based and Signal-Based Inverse Methods [1st Edition], UnB, Brasilia, DF, Brazil, 2022. Volume I in Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity. ISBN 978-65-86503-71-5
DOI: https://doi.org/10.4322/978-65-86503-71-5

Edited as an initiative of: Universidade de Brasília
Faculdade do Gama (FGA) / Faculdade de Tecnologia (FT)
Programa de Pós-Graduação em Integridade de Materiais da Engenharia

Address:
Universidade de Brasília, Campus do Gama
Área Especial de Indústria e Projeção – Setor Leste
Prédio UED (Unidade de Ensino e Docência) - 1º Andar
CEP: 72.444-240, Gama - DF, Brasil

Contact: Phone: (+55) (61) 3107-8903 WebSite: http://www.pgintegridade.unb.br/
E-mail pgintegr@unb.br; ctanflor@gmail.com; ariosto.b.jorge@gmail.com

Cataloging-in-Publication (CIP) Data:
Prepared by the librarian at University of Brasilia Central Library (BCE/UnB).



Ficha catalográfica elaborada pela Biblioteca Central da Universidade de Brasília
Heloiza Faustino dos Santos - CRB 1/1913

M689 Model-based and signal-based inverse methods [recurso eletrônico] / editors, Ariosto Bretanha Jorge … [et al.]. – Brasília : Universidade de Brasília, 2022.
544 p. – (Book series in discrete models, inverse methods, & uncertainty modeling in structural integrity ; v. 1)

Modo de acesso: World Wide Web.
ISBN 978-65-86503-71-5.

1. Engenharia de materiais. 2. Métodos. 3. Modelos. I. Jorge, Ariosto Bretanha (org.). II. Série.

CDU 620

Foreword from FGA / UnB

The engineering sector drives and enables the development of a country. The formation of an engineer allows a technical capacity to evaluate, plan, design, suggest and apply all possible techniques in search of the best construction of a technological equipment. Currently, the engineer must be more and more prepared to solve existing problems in various sectors of society. It is through it that societies grow in search of progress.

The recognition of engineering and the training of new professionals increases every year in Brazil. In the 2000s, the University of Brasília (UnB) went through an expansion process, resulting in the implementation of the new UnB Engineering Campus in the city of Gama (UnB-Gama, FGA). Five new undergraduate courses were created: Aerospace Engineering, Automotive Engineering, Electronic Engineering, Energy Engineering and Software Engineering. The UnB Gama Campus project converges to increase the education level of the Brazilian population, especially in the five areas of engineering activity, all in line with current national public policies, aimed at expanding the population's access to quality higher education in the country.

Following the high quality teaching line, the UnB-Gama campus has the Graduate Program in Integrity of Engineering Materials (PPG-Integridade). The program has the following lines of research: Dynamics and Vibrations, Fatigue, Structural Materials, Biomaterials, Structure Fluid Interaction and Numerical Simulation of the Mechanical Behavior of Materials. This book series is an initiative of PPG-Integridade - UnB, organized as a collaborative work involving researchers, engineers, scholars, from several institutions, universities, industry, recognized both nationally and internationally.

Beside the high technical quality and relevance of the topics covered in the books, this series will enable an essential internationalization of the research currently developed within the University of Brasília. Several authors from different countries also contributed to these books, enabling greater interaction between national and international research groups. This internationalization raises the level of academic education for new professionals in the field of engineering, in addition to more advanced scientific research and technological development.

Additionally, this book series features a strong contribution from the industrial sector. Several professionals from different companies collaborated with the writing of some chapters in the three volumes that make up this series. These initiatives are of great strategic importance, as they allow the grouping of different technical capabilities. On the part of companies in the sector, with knowledge of market demands, and on the part of universities, by adding the technical-scientific knowledge of their team of researchers to the improvement of innovative products and services.

This book should be appreciated by anyone in need of knowledge of Materials Integrity. The completeness of Discrete Modeling and Inverse Methods theory combined with the Uncertainty Modeling in Structural Integrity makes these books mandatory for everybody aiming at Direct and Inverse Problems, including model-based and signal-based inverse problems.

Prof. Dr. Sandro A.P. Haddad, Director
UnB-Gama campus (https://fga.unb.br/)

Foreword from LAJSS

Book Series in Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity

Book Series editors: Ariosto B. Jorge, Carla T.M. Anflor, Guilherme F. Gomes, and Sergio H.S. Carneiro

This book series represents a commendable effort in compiling the latest developments on three important Engineering subjects: discrete modeling, inverse methods, and uncertainty structural integrity. Although academic publications on these subjects are plenty, this book series may be the first time that these modern topics are compiled together, grouped in volumes, and made available for the community.

The application of numerical or analytical techniques to model complex Engineering problems, fed by experimental data, usually translated in the form of stochastic information collected from the problem in hand, is much closer to real-world situations than the conventional solution of PDEs. Moreover, inverse problems are becoming almost as common as direct problems, given the need in the industry to maintain current processes working efficiently, as well as to create new solutions based on the immense amount of information available digitally these days. On top of all this, deterministic analysis is slowly giving space to statistically driven structural analysis, delivering upper and lower bound solutions which help immensely the analyst in the decision-making process.

All these trends have been topics of investigation for decades, and in recent years the application of these methods in the industry proves that they have achieved the necessary maturity to be definitely incorporated into the roster of modern Engineering tools. The present book series fulfills its role by collecting and organizing these topics, found otherwise scattered in the literature and not always accessible to industry.

Moreover, many of the chapters compiled in these books present ongoing research topics conducted by capable fellows from academia and research institutes. They contain novel contributions to several investigation fields and constitute therefore a useful source of bibliographical reference and results repository.

The Latin American Journal of Solids and Structures (LAJSS) is honored in supporting the publication of this book series, for it contributes academically and carries technologically significant content in the field of structural mechanics.

On behalf of LAJSS,

Prof. Dr. Marcílio Alves (USP), Editor-in-Chief
Prof. Dr. Rogério J. Marczak (UFRGS), Assoc. Editor
Prof. Dr. Pablo A. Muñoz-Rojas (UDESC), Assoc. Editor
Prof. Dr. Marco L. Bittencourt (Unicamp), Assoc. Editor

Latin American Journal of Solids and Structures (LAJSS)
(www.lajss.org)

Foreword from ABCM

The Brazilian Society of Mechanical Sciences and Engineering – ABCM welcomes enthusiastically the publication of the Book Series in Models, Inverse Methods & Uncertainty Modeling in Structural Integrity.

The initiative, undertaken by Prof. Ariosto B. Jorge, Dr. Carla T.M. Anflor, Dr. Guilherme F. Gomes and Dr. Sergio H. S. Carneiro, with the support of the University of Brasília, is received by the scientific community as a valuable contribution to the dissemination of knowledge encompassing the large number of topics covered in the three volumes of the series.

These topics have been judiciously selected to encompass comprehensively the theoretical aspects, modeling techniques and numerical methods related to Structural Integrity, and are presented in a large collection of chapters authored by renowned experts, from both academia and industry. We gladly realize that many members of ABCM have contributed as authors.

Besides the comprehensive and well-articulated content, one distinguishing characteristic of this book series is that it has been conceived to serve both for educational purposes at graduate level and as an information source for researchers and engineering practitioners, which amplifies, to a large extent, its utility. Another relevant feature is that the material is intended to be available to the public in electronic format at no cost, which highlights the generosity of the authors and editors and their commitment to the most fundamental academic principles.

On behalf of the scientific community of the field of Mechanical Sciences and Engineering, ABCM acknowledges the editors and authors of the present book series for their contribution to the progress of Engineering research and education.

Prof. Dr. Domingos Alves Rade
President of ABCM

On behalf of

ABCM
Associação Brasileira de Engenharia e Ciências Mecânicas

Brazilian Society of Mechanical Sciences and Engineering (ABCM)
(www.abcm.org.br)

Foreword from ABMEC

The whole range of topics related to Direct & Inverse Problems and Modeling of Uncertainties is substantially associated with the needs of the mechanical, civil, aeronautical/aerospace, nuclear, and naval/oceanic industries. Indeed, they play a core role in industrial renewal, contributing to productivity and competitiveness. Especially taking Brazil into account, this book series, conceived as a comprehensive one that covers these important topics, is very welcome.

These themes are also among the main interests of the Brazilian Association of Computational Methods in Engineering, ABMEC. ABMEC is concerned with the application of numerical methods and digital computers to the solution of engineering problems. Its mission is to promote, foster, and organize activities encompassing the development and use of such computational methods in Brazil.

We are fortunate to have the opportunity to support this book series as a collaborative work that intends to involve scholars from different institutions and researchers from industry, with national and international relevance. We sincerely believe that this work will provide a common forum for discussion, education, and research information transfer between the several subjects concerning computational methods in engineering.

Our congratulations to the editors, professors Ariosto Bretanha Jorge, Carla Tatiana Mota Anflor, Sergio Henrique da Silva Carneiro, Guilherme Ferreira Gomes for this important contribution to the Brazilian engineering.

Prof. Dr. Felício Bruzzi Barros
President of ABMEC

On behalf of

abmec
Associação Brasileira de Métodos
Computacionais em Engenharia

Brazilian Association of Computational
Methods in Engineering (ABMEC)
(www.abmec.org.br)

Acknowledgements

Acknowledgements from the Book Series editors

This book series is an initiative of the Graduate Program in Integrity of Engineering Materials (PPG-Integridade) at the University of Brasilia (UnB), Brazil (www.pgintegridade.unb.br).

The editors would like to thank PPG-Integridade and UnB for the initiative, incentive and support for this Book Series project.

The book series is organized as a collaborative work involving researchers, engineers, scholars, engaged in research, development and applications in the related areas, affiliated to several institutions, universities, industry, and recognized both nationally and internationally.

The editors are grateful and would like to show their appreciation to all the co-authors of the book chapters, for their participation, dedication, and support.

The book series is published as a digital version, with ISBN provided by UnB, and DOI for each chapter, provided by the Latin American Journal of Solids and Structures (LAJSS) (www.lajss.org). The scope of the Book series is in the broad areas of interest of LAJSS, and also of the Brazilian Society of Mechanical Sciences and Engineering (ABCM) (www.abcm.org.br) and the Brazilian Association of Computational Methods in Engineering (ABMEC) (www.abmec.org.br). For increased visibility, these three institutions are encouraging the divulgation of the Book Series project in their websites.

The editors would like to express their appreciation to LAJSS, ABCM and ABMEC, for their incentive, encouragement and support for this Book Series project.

Ariosto, Carla, Guilherme, Sergio

Brasilia, February 1st, 2022.

A personal dedication from the Book Series leading editor

In my point of view, this Book Series project represents the culmination of a dedicated academic career, in many aspects intrinsically related to the different research topics and areas covered along the three Volumes of the Book Series. I would like to thank all my co-editors of the Book Series, all the co-authors of the book chapters, and also all the researchers, scholars, students with whom I had the opportunity to share collaborative work throughout my many years along this academic career. I've enjoyed learning a lot from you all!

To you, my sincere thank you!

I would like also to dedicate this project to my wife Daisy, for her love, understanding, and unconditional support, throughout my entire academic career, and to my daughter Elisa and my son Luís Paulo, for their love and support.

To you, my true love and deepest appreciation!

Ariosto

Brasilia, February 1st, 2022.

Table of Contents

Chapter 1. Introduction to Optimization and Identification Techniques for Model-Based and Signal-Based Inverse Problems

Chapter details

Chapter DOI:

https://doi.org/10.4322/978-65-86503-71-5.c01

Chapter suggested citation / reference style:

Jorge, Ariosto B., et al. (2022). "Introduction to Optimization and Identification Techniques for Model-Based and Signal-Based Inverse Problems". *In* Jorge, Ariosto B., et al. (Eds.) *Model-Based and Signal-Based Inverse Methods*, Vol. I, UnB, Brasilia, DF, Brazil, pp. 1–7. Book series in Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity.

P.S.: DOI may be included at the end of citation, for completeness.

Book details

Book: Model-based and Signal-Based Inverse Methods

Edited by: Jorge, Ariosto B., Anflor, Carla T. M., Gomes, Guilherme F., & Carneiro, Sergio H. S.

Volume I of Book Series in:

Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity

Published by: UnB City: Brasilia, DF, Brazil Year: 2022

DOI: https://doi.org/10.4322/978-65-86503-71-5

Introduction to Optimization and Identification Techniques for Model-Based and Signal-Based Inverse Problems

Ariosto Bretanha Jorge[1a*], Carla Tatiana Mota Anflor[1b], Guilherme Ferreira Gomes[2] and Sérgio Henrique da Silva Carneiro[1c]

[1]Post-Graduate Program - Integrity of Engineering Materials, University of Brasilia, Brazil. Book series editors. E-mail: ariosto.b.jorge@gmail.com, anflor@unb.br, shscarneiro@unb.br
[2] Mechanical Engineering Institute, Federal University of Itajubá, Itajubá, Brazil. Book series editor. E-mail: guilhermefergom@unifei.edu.br

*Corresponding author. Book series leading editor.

Abstract

This chapter presents an overview of the Book Series in Direct Methods, Inverse Methods and Uncertainty Modeling, with focus on its Volume I: Model-Based and Signal-Based Inverse Methods, and includes an introduction to the different topics in Optimization and Identification Techniques comprising the several chapters included in this Volume I of the Book series.

1 Book Series in Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity: overview

The Book series in “Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity” is an initiative of the Post-Graduate Program - Integrity of Engineering Materials (PPG-Integridade) of University of Brasilia (UnB), organized as a collaborative work involving researchers, engineers, scholars, from several institutions, universities, industry, recognized both nationally and internationally.

This book series is an activity related to the Research, Development & Innovation (R,D&I) Project at UnB, titled: “Technological Demonstration Platform for Inverse Methods and Uncertainty Modeling Integrity of Structures and Components”, available at the UnB Central Library (in Portuguese) (Jorge, 2020).

The Book Series project is comprised by three Volumes:

- Volume I – Model-based and Signal-Based Inverse Methods
- Volume II – Fundamental Concepts and Models for the Direct Problem
    - Part I - Material Modeling
    - Part II - Discrete Modeling
- Volume III – Uncertainty Modeling: Fundamental Concepts and Models

The different book chapters were elaborated encompassing the relevant project topics, including chapters covering:

- Fundamentals, including topics such as: basic principles, concepts & foundations, for the Direct & Inverse Problems (including model-based and signal-based inverse methods), and for the Modeling of Uncertainties;
- Special Topics, Applications, and Research Review, including topics such as: research review, state-of-the-art & future trend topics, for the Direct & Inverse Problems (including model-based and signal-based inverse methods), and for the Modeling of Uncertainties.

The different book chapters were prepared as a collaborative work by researchers, engineers, scholars, involved in research, development and applications in the related areas.

The research areas of interest throughout the book chapters include:

- Modeling of the inverse problem, monitoring & diagnosis / prognosis: models and methods for inverse problems, optimization methods (including techniques such as; multi-objective optimization, topology optimization, evolutionary optimization), Wavelets, Kalman Filter (KF), Particle Filter (PF), Machine Learning (ML), Artificial Intelligence (AI), Data Science (DS), for applications such as Structural Health Monitoring (SHM) (including impedance-based and Lamb Wave-based techniques), Health & Usage Monitoring Systems (HUMS);

- Modeling of the direct problem: mechanics of materials (including metallic materials, composites), structures (including civil, mechanical naval, aeronautical structures) machinery design and mechanical components, fracture mechanics, impact, fatigue, damage tolerance, integrity, mechanical vibrations, dynamics of structures, computational mechanics, including mathematical methods and numerical methods for discrete modeling for continuum mechanics (such as Finite Element Methods (FEM), Boundary Element Methods (BEM), Mesh-Free Methods (MFM));

- Probabilistic methods and modeling of uncertainties: probabilistic methods in engineering, Design of Experiments (DOE), Response Surface Methods (RSM), Risk & Reliability (including structural and system reliability), Uncertainty Modeling (UM) & Uncertainty Quantification (UQ), Bayesian Approaches (BA), stochastic Finite Element approaches (Stochastic FEM, Spectral FEM, Polynomial Chaos), sthochastic optimization, meta-modeling (including techniques such as Surrogate Models (SM), Reduced Order Models (ROM)), model Verification & Validation (V&V).

The different models, methods and approaches presented throughout the several chapters in the three Volumes of this Book Series are intended as an introductory presentation of some possibilities of methods that could be used in problems related to integrity of structures and components, and maybe even extended to other engineering areas, as appropriate. The list of models is not unique, and is neither comprehensive nor exhaustive, and the reader is encouraged to look for different possibilities of methods that may be applicable to the particular engineering problem at hand.

A common aphorism, often presented as *"All models are wrong, but some are useful"*, is usually considered to be applicable to scientific models in general, and to statistical models in particular. The aphorism recognizes that statistical or scientific models always fall short of the complexities of reality but can still be of use. The aphorism is generally attributed to the statistician George E. P. Box, although the underlying concept predates Box's writings.

The following sections present an introduction to Model-Based and Signal-Based Inverse Methods, with emphasis on the different topics in Optimization and Identification Techniques comprising the several chapters included in this Volume I of the Book Series, as well as their connection and relationship with regard to the whole setting of methods and models.

2 Volume I: Model-Based and Signal-Based Inverse Methods – context

The detection, localization, classification and identification of parameters and/or material properties, related to the integrity of structures and components, with and without defects or damages, involves the modeling of inverse problems, as well as an adequate modeling and quantification of the uncertainties involved in the problem.

The formulation of the direct problem, of the inverse problem, and the related uncertainties modeling, needed for an adequate description of the structure and/or the mechanical component, and of its potential defects or damages, involves multidisciplinary modeling techniques, whose understanding and proper application transcends the field of integrity and damage tolerance, being able to serve as a basis for applications, in other contexts or fields.

Among the application problems of interest for inverse methods, one can cite Structural health monitoring (SHM) and Health and Usage Monitoring Systems (HUMS).

The monitoring of structural integrity (SHM) is a competitive technique for damage detection and identification, wherein information is collected online, and compared with an existing database for an undamaged (“healthy”) structure. From this comparison, real-time information on the presence of damages can be obtained, including their localization, size, propagation speed, and, ultimately, the remaining operational life of the structural component.

The monitoring of mechanical components (HUMS) is a technique which is being used to follow / accompany the state of the integrity of mechanical systems / components (Health) and to monitor the appearance of indicators of the presence of damage (usage) in dynamic systems, such as rotating components (in engines, for example) and in gearboxes (in mechanical transmission systems, for example). In this case, the comparison is made between vibration signals from the healthy components (accumulated historical data) and from the components being monitored, to identify significant discrepancies in the vibration signals, which could be correlated to specific / particular / known damages.

The scientific challenge of the modeling of inverse problems, as well as of the adequate modeling and quantification of the related uncertainties, in a problem of integrity of structures and components, involves several aspects:

- The modeling of the problems (direct problem, inverse problem, uncertainties) needs to be done, whenever possible, by using more than one technique, for each case being described, in order to implement, for the particular problem at hand, model techiques which are independent, complementary, and/or redundant. Whenever possible, more than one model should be used, for redundancy and/or comparative analysis, especially in the case of unavailability of prior data for the healthy structure and/or component.

  The techniques used for inverse methods may involve:

  i. Optimization techniques, based on multiobjective optimization models, using classical optimization techniques (such as Sequential Quadratic Programming (SQP), BFGS, etc), or evolutionary optimization techniques (such as Genetic Algorithms (GA), Differential Evolution (DE), etc);

  ii. Identification techniques based on Artificial Intelligence, Machine Learning, Pattern Recognition, Data Science, etc, models (such as identification models based on Artificial Neural Networks (ANN), for example;

  iii. Models based on the Wavelet Transform (continuous Wavelet Transform (CWT), discrete Wavelet transform (DWT), with different types and sizes of the Wavelet window, for example);

  iv. Stochastic models (such as Kalman Filter (KF), Extended Kalman Filter (EKF), Extended Information Filter (EIF), Particle Filter (PF), Least Squares (LS), etc).

- In several situations, the direct models to be implemented may involve different problem physics (Multi-physics Modeling), and multiple scales (Multi-scale Modeling). In such cases, the description for the direct problem may involve coarser global models, and more detailed local models;
- The computational simulations and the experimental / laboratory tests must take into account the additional challenge of properly simulating / representing the local behavior of a complex structure, in the regions of interest, where the defect of damage is expected to be, or is expected to appear. For example, Fracture Mechanics (FM) problems and Damage Tolerance (DT) problems cannot be properly represented by reduced-scale models, as the damaged region must be represented using full-scale models. In these cases, the computational simulations (and also the experimental / laboratory tests) are required to reproduce the situation in the region of the damage using high fidelity local models. Thus, the region of the damage must be modeled in full scale, with the model also representing properly the geometry, the mechanical properties, and the real loading in that local region (loading that is coming from the external loads that were applied in the structure or in the component as a whole).
- In many cases, inverse problems may belong to the category of ill-posed problems, which represents an additional challenge in the modeling of the problem at hand. In these cases, the approach for the inverse method may require additional hypothesis to be made (for problem regularization, for example), or that a meta-modeling approach is adopted (such as surrogate models, reduced order models, etc), replacing the original model by the proper meta-model, and then solving this approximate model for the problem.

- The modeling of inverse problems, such as in the case of SHM and/or HUMS, must take into account the proper modeling of the sensor behavior, and also the uncertainties associated to these sensors, as well as the simultaneous use of multiple, independent, techniques for monitoring, with different sensors. The optimal sensor positioning, to maximize the Probability of Detection (PoD), may be seen as a topological optimization problem and/or as a stochastic optimization problem.
- The modeling of inverse problems may involve the detection, localization, and identification of parameters and/or material properties (for example, properties such as elasticity modulus, Poisson's coefficient, etc), which may vary through time (for example, material degradation thought time) and also along the length of the structure or mechanical component (for example, local changes which may occur in the material properties and / or mechanical properties of a composite plate, due to the debonding between the layers of the composite material).

3 Chapter topics in Volume I: Model-Based and Signal-Based Inverse Methods - presentation

Along this Volume I of the Book Series, several topics related to model-based and signal-based methods for inverse problems are presented in the several book chapters, representing the collaborative work from researchers, engineers, scholars, engaged in research, development and applications in the related areas, affiliated to several institutions, universities, industry, and recognized both nationally and internationally.

The book chapters in this Volume I of the Book Series are distributed as follows:

Chapter 1: Introduction to Optimization and Identification Techniques for Model-Based and Signal-Based Inverse Problems

Chapter 2: Overview of Some Optimization and Identification Techniques for Inverse Problems of Detection, Localization and Parameter Estimation

Chapter 3: An overview of Linear and Non-linear Programming methods for Structural Optimization

Chapter 4: Overview of Traditional and Recent Heuristic Optimization Methods

Chapter 5: Application of Machine Learning and Multi-Disciplinary/Multi-Objective Optimization Techniques for Conceptual Aircraft Design

Chapter 6: On a Bio-Inspired Method for Topology Optimization via Map L-Systems and Fractone Modeling

Chapter 7: Fundamentals on the Topological Derivative concept and its classical applications

Chapter 8: Ultrasound Obstacle Identification using the Boundary Element and Topological Derivative Methods

Chapter 9: Fundamental Concepts on Wavelet Transforms

Chapter 10: Application of Wavelet Transforms to Structural Damage Monitoring and Detection

Chapter 11: Inverse Methods using KF, EKF, EIF, PF, and LS Techniques for Detection, Localization, and Parameter Estimation

Chapter 12: Fundamental Concepts for Impedance-based Structural Health Monitoring

Chapter 13: Fundamental Concepts for Guided Lamb Wave-based Structural Health Monitoring

Chapter 14: Machine Learning and Pattern Recognition: Methods and Applications for Integrity Monitoring of Civil Engineering Structures

4 Final remarks and acknowledgements

This chapter presents an overview of the Book Series in Direct Methods, Inverse Methods and Uncertainty Modeling, with focus on its Volume I: Model-Based and Signal-Based Inverse Methods, and includes an introduction to the different topics in Optimization and Identification Techniques comprising the several chapters included in this Volume I of the Book series.

This book series is an initiative of the Graduate Program in Integrity of Engineering Materials (PPG-Integridade) at the University of Brasilia (UnB), Brazil (www.pgintegridade.unb.br).

The editors would like to thank PPG-Integridade and UnB for the initiative, incentive and support for this Book Series project.

The book series is organized as a collaborative work involving researchers, engineers, scholars, engaged in research, development and applications in the related areas, affiliated to several institutions, universities, industry, and recognized both nationally and internationally.

The editors are grateful and would like to show their appreciation to all the co-authors of the book chapters, for their participation, dedication, and support.

The book series is published as a digital version, with ISBN provided by UnB, and DOI for each chapter, provided by the Latin American Journal of Solids and Structures (LAJSS) (www.lajss.org). The scope of the Book series is in the broad areas of interest of LAJSS, and also of the Brazilian Society of Mechanical Sciences and Engineering (ABCM) (www.abcm.org.br) and the Brazilian Association of Computational Methods in Engineering (ABMEC) (www.abmec.org.br). For increased visibility, these three institutions are encouraging the divulgation of the Book Series project in their websites.

The editors would like to express their appreciation to LAJSS, ABCM and ABMEC, for their incentive, encouragement and support for this Book Series project.

References

Jorge, A. B. (2020). *Technological Demonstration Platform for Inverse Methods and Uncertainty Modeling Integrity of Structures and Components (Plataforma demonstradora tecnológica para métodos inversos e modelagem de incertezas em integridade de estruturas e componentes).* University of Brasilia. https://repositorio.unb.br/handle/10482/39570

Chapter 2. Overview of Some Optimization and Identification Techniques for Inverse Problems of Detection, Localization and Parameter Estimation

Chapter details

Chapter DOI:

https://doi.org/10.4322/978-65-86503-71-5.c02

Chapter suggested citation / reference style:

Sousa, Bruno S., et al. (2022). “Overview of Some Optimization and Identification Techniques for Inverse Problems of Detection, Localization and Parameter Estimation.” *In* Jorge, Ariosto B., et al. (Eds.) *Model-Based and Signal-Based Inverse Methods*, Vol. I, UnB, Brasilia, DF, Brazil, pp. 8–64. Book series in Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity.

P.S.: DOI may be included at the end of citation, for completeness.

Book details

Book: Model-based and Signal-Based Inverse Methods

Edited by: Jorge, Ariosto B., Anflor, Carla T. M., Gomes, Guilherme F., & Carneiro, Sergio H. S.

Volume I of Book Series in:

Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity

Published by: UnB City: Brasilia, DF, Brazil Year: 2022

DOI: https://doi.org/10.4322/978-65-86503-71-5

Overview of Some Optimization and Identification Techniques for Inverse Problems of Detection, Localization and Parameter Estimation

Bruno Silva de Sousa[1*], Guilherme Ferreira Gomes[1], Patricia da Silva Lopes Alexandro[1], Sebastião Simões Cunha Jr.[1] and Ariosto Bretanha Jorge[2]

[1]Mechanical Engineering Institute, Federal University of Itajubá, Itajubá, Brazil.

E-mail: bruno_s_sousa@unifei.edu.br; guilhermefergom@unifei.edu.br; patty_lauer@unifei.edu.br; sebas@unifei.edu.br

[2]Post-Graduate Program - Integrity of Engineering Materials, University of Brasilia, Brazil. E-mail: ariosto.b.jorge@gmail.com

*Corresponding author

Abstract

This chapter presents a compilation of the research work being done by the authors and collaborators on the topics of optimization and identification techniques for inverse methods in damage detection and localization.

1 Introduction

In this chapter the work being done in the Research Group in Computational Mechanics (GEMEC) at UNIFEI is presented. In what follows, a short introduction of the different journal articles and conference papers prepared by the authors along the last 15 years is presented, covering different methods and aspects in optimization and identification techniques for inverse methods in damage detection and localization. The subsections below refer to each separated topic being discussed.

1.1 Some concepts and definitions used along this chapter

1.1.1 Structural damage

Many structures, during their useful life, are submitted to several types of static and dynamic loads. These loads and the structural deterioration process can cause different types of structural damages. Damage characterization and the knowledge of the changes in the material properties corresponding to these damages depend on the type of material and on the structural configuration (Lopes *et al.*, 2007). The proper assessment of the damage in a structure can be useful to infer its remaining service life (Suveges *et al.*, 2016).

Ensuring the integrity of a structure is of paramount importance to ensure the safety of workers, the environment and the general public, as many equipment and structures are part of our daily lives. For this, the structure as a whole must be evaluated in an attempt to detect possible damage and carry out the necessary maintenance actions, quickly, effectively and economically viable. Among the various mechanisms that generate damage, there are fatigue, overloads, impacts, corrosion and even natural damage, such as tsunamis, winds, earthquakes, among others (Alves, 2012).

According to (Friswell, 2008) and (Lopes *et al.*, 2010), structural damage can be modeled as changes in the physical and/or geometric properties of the structure. The choice of the damage model will depend on the type of structure under analysis (trusses, beams, plates and others), the type of material, the failure modes and the objectives of the damage assessment. For example, compared to simplified models based on local stiffness reduction, more detailed models that associate a given geometric shape to damage (holes, cavities, inclusions, cracks and others) can provide more information for predicting the remaining service life of the structure. Often times, the detection and identification of structural damage can be difficult, for example, due to the difficulty of accessing the location of the damage (Suveges *et al.*, 2016).

1.1.2 Inverse problem and direct problem

The life time of any structure can be predicted through the correct determination of the damage. To determine the damage must be performed a comparison between measured and simulated data using numerical code. The numerical modeling consists in a direct problem and an inverse problem, according to (Lopes *et al.*, 2008). For the direct problem, a model is required to obtain information on the distribution of the quantity of interest throughout the structure, given the boundary conditions and the presence of the damage. For the inverse problem, a model is required for the procedure of locating the damage in the structure given some (partial) information on the quantity of interest at some particular locations (for example, where some sensors are placed) (Lopes *et al.*, 2010).

The damage detection problem can be ranked as a problem of system identification or an inverse problem. Numerical methods, such as the Boundary Element Method (BEM) or the Finite Element Method (FEM) can be used for modeling the direct problem (Lopes *et al.*, 2010). Parameter identification techniques and optimization techniques can be used to determine the unknown parameters of the damage. Among the parameter identification techniques, one can cite Artificial Neural Networks (ANN's) and Kalman Filter (KF). As for the optimization techniques, one can cite Genetic Algorithms (GA's), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Differential Evolution (DE), Lichtenberg Algorithm (LA), SunFlower Optimization (SFO), belonging to the category of global optimization techniques wherein the global optimum of the system has larger chances of being obtained.

1.2 Overview of the research work in optimization and identification techniques

1.2.1 Damage detection using GA and ANN

In the work of (Lopes *et al.*, 2007) an example of heat flow through a simple conduction at a thin plate is investigated. The BEM is used to simulate the potential values on the external surface of the plate at given points. These potential values represent the distribution of temperatures on the plate. An assumption is made that the conduction of heat through possible internal holes in the plate is considered null (adiabatic holes). The use of thermal techniques shows that the distribution of temperatures on a plate changes due to the variations in the mechanical properties of the plate, what could be related to a determined damage. ANN's and AG's were used for the identification of the number of holes and its locations. Details of this work can be found in Section 2.1.

In the work of (Lopes *et al.*, 2008) the BEM is used as the direct problem, and two independent different techniques GA and ANN were used for the inverse problem, in order to localize and to identify the presence of circular holes in the structure. Details of this work can be found in Section 2.2.

In the work of (Lopes *et al.*, 2010) two BEM formulations were used, for potential and elastostatic problems, respectively. For the potential formulation, the potential values represent the distribution of temperatures on the plate at given points. For the elastostatic formulation, the quantities of interest are the interior point displacements and stresses. The inverse problem was solved using two independent techniques, GA and ANN, thus allowing more reliable information on the damage parameters can be obtained, as a comparison of the results from both approaches can provide a means to verify these results. Details of this work can be found in Section 2.3.

In the work of (Alexandrino *et al.*, 2019) an inverse problem of damage identification and localization in a structure was modelled as a robust optimization problem using a multiobjective GA. In the robust optimization problem, the optimum value and small variations around this optimum value are considered. This variance function was obtained by a Design of Experiment with regression and also through a relation between functional variance and damage parameters found by ANN. As a multiobjective GA obtains multiple solutions, a fuzzy decision making technique finds the better tradeoff solution for the problem. The BEM was used to obtain the distribution of stress to elastostatic problem. Details of this work can be found in Section 2.4.

In the works of (Gomes, Almeida, *et al.*, 2018) and (Gomes, Mendéz, Cunha Jr., *et al.*, 2018) a numerical-experimental inverse problem study of damage and delamination detections in CFRP plates was performed. The direct problem is solved by FEM and then GA was used in order to solve the inverse problem considering experimental modal data from delaminated structures. Details of these works can be found in Section 2.5 for a more general case and in Section 2.6 for an aeronautical structure case.

1.2.2 Damage location, identification and detection using SunFlower Optimization algorithm

In the work of (Gomes, Cunha Jr., *et al.*, 2019a) a new nature-inspired optimization method based on sunflowers' motion was introduced to treat the damage detection problem as an inverse problem with objective function minimization. The proposed SunFlower Optimization algorithm (SFO) technique is a population-based iterative heuristic global optimization algorithm for multi-modal problems. The new method was then applied in an inverse problem of structural damage detection in composite laminated plates. Details of this work can be found in Section 2.7

In the work of (Gomes *et al.*, 2020) an inverse algorithm based on strain fields for damage identification in composite plate structures was presented. The inverse analyses combine experimental tests and digital image correlation (DIC) with numerical models based on finite element update method with great advantage of being a non-contact method. The proposed technique identifies the location and dimension of damages in a CFRP plate using static strains formulated as an objective function to be minimized. The SunFlower Optimization (SFO) was employed to update the unknown model parameters. Details of this work can be found in Section 2.8

1.2.3 Damage identification and detection using Lichtenberg Algorithm

In the work of (J. L. J. Pereira, Francisco, Cunha Jr., *et al.*, 2021) a new metaheuristic Lichtenberg Algorithm (LA) was applied to solve a complex inverse damage identification problem in mechanical structures built by composite material. To verify the performance of the new algorithm, both LA and Finite Element Method (FEM) were used to identify delamination damage, considering particular situations like noisy response and low damage severity. The results were compared to other algorithms such as Genetic Algorithm (GA) and SunFlower Optimization (SFO). Details of this work can be found in Section 2.9.

In the work of (J. L. J. Pereira, Chuman, *et al.,* 2021) the Lichtenberg Algorithm (LA) was implemented to develop a numerical identification and characterization of crack propagation. The damage identification problem was treated as an inverse problem, which combines FEM with LA to identify the propagation direction of cracks in aluminum structures, with emphasis on aeronautical structures when using the 6061-aluminum alloy. Details of this work can be found in Section 2.10.

1.2.4 Damage detection using Ant Colony Optimization and Differential Evolution

In the work of (Suveges *et al.*, 2016) an inverse plate damage detection problem was solved through different global optimization heuristics coupled with BEM. The selected heuristics to solve the inverse problem were GA, ACO, PSO and DE. These heuristics were coupled with the BEM to identify a damage modeled as an elliptical hole in a thin isotropic plate, varying the position, dimension and inclination of the elliptical hole. Details of this work can be found in (Suveges *et al.*, 2016).

1.2.5 Other optimization techniques: CRS Algorithm and Topological Sensitivity Analysis

In the work of (Sousa *et al.*, 2008) a methodology for multiobjective airfoil shape optimization using a global search algorithm was presented, namely, Controlled Random Search Algorithm (CRSA). The multiobjective method implemented was the aggregating approach, in which all the objectives of the problem are transformed into a single one, through the weighting coefficients that representing the relative importance of each objective function of the problem. The airfoil shape is parameterized by two Bezier arcs of high degree representing one the lower surface and other the upper surface. Constraints are incorporated by means of a penalty scheme. As solver was used a modified version of well known viscous-inviscid flow analysis code XFoil. Details of this work can be found in Section 2.11

In the work of (Sousa *et al.*, 2018) two main problems were analyzed, namely the optimal design of multilayered composite laminates and the topological sensitivity analysis in anisotropic elastostatics. Regarding the composite design, minimal weight structures subjected to bending and Hoffmann failure criteria constraints are considered, where the design variables are the shape/topology of each ply and the stacking sequence. The application of topological sensitivity analysis is extended to obtain the optimal topology of composite laminated structures. From the Topological Derivative mapping methodology, considering the total potential energy as an objective function, the optimal topology is obtained by gradual insertion of material in the considered domain. The Topological Derivative defines the shape of the new added plies, and the optimal layup is obtained by using ACO. Details of this work can be found in Section 2.12.

2 Numerical and Experimental Applications

2.1 Damage Detection using Global Optimization and Parameter Identification Techniques

The damage detection is an important branch of engineering where some measurements can be applied to guarantee the structural security. The life time of any structure can be predicted through the correct determination of the damage. In this work, an example of heat transfer through simple conduction at a thin plate is investigated. The Boundary Element Method (BEM) is used to simulate the potential values on the external surface of the plate at given points. These potential values represent the distribution of temperatures on the plate. An assumption is made that the conduction of heat through possible internal holes in the plate is considered null (adiabatic holes). The use of thermal techniques shows that the distribution of temperatures on a plate changes due to the variations in the mechanical properties of the plate, what could be related to a determined damage. The genetic algorithm (GA) is used as the optimization procedure and the artificial neural network (ANN) approach is used as a parameter identification technique to identify the number of holes and their locations. The MATLAB® was used for the development of the damage detection program.

GA is a search method based on the processes of natural evolution. This method works with a set of possible solutions for a given problem (initial population) and the problem variables are represented as genes in a chromosome or an individual. Starting from an initial population, the individuals with better adapted genetic characteristics have higher chances of surviving and reproducing (Lopes *et al.*, 2007). Parameters of the GA influence in the behavior of the method and the most important parameters are: population size, generation number, crossover probability and mutation probability. The choice of the best configuration for the GA parameters is difficult and this choice depends on the realization of a great number of experiments and tests.

To obtain the unknown damage parameters (location and size) through the GA, a functional can be defined as the difference between the ‘measured’ values (‘simulated’ values by BEM) of the potential difference (between undamaged plate and plate with damage) and the ‘calculated’ values obtained from the damage detection program. This functional corresponds to the fitness function of the GA. The minimization of this fitness function allows the damage detection program to find the unknown damage parameters. The potential values are simulated through BEM for the potential in 49 internal points of the plate. The functional formulation is shown in Eq. (1).

$$J_j = \frac{1}{2}\sum_{i=1}^{n}\left(\text{simulated}_i - \text{calculated}_{ji}\right)^2 \tag{1}$$

where n is the number of internal points i (“sensors”in the plate) where the differences are evaluated; simulated$_i$ is the vector of simulated values for the differences obtained using BEM for a given damage, and calculated$_{ji}$ is the vector of differences in potential calculated by the code for each individual j.

To analyze the circular hole detection problem, a plate with the dimensions (0.06×0.06) m was simulated through the BEM, as illustrated in Fig. 1(a) for potential problem.

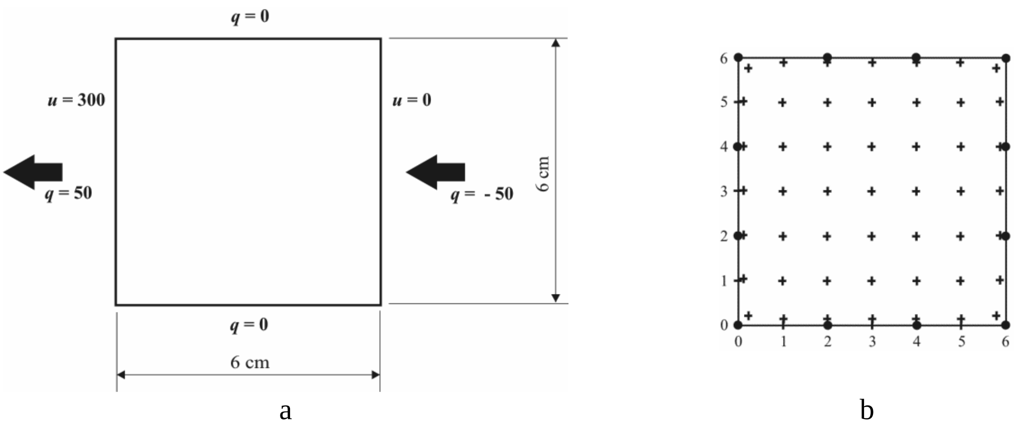


Figure 1: Plate model for potential problem: (a) dimensions, loading and boundary conditions; (b) boundary discretization and sensor locations.

Initially, a plate without damage was simulated through BEM. The plate boundary was discretized into 12 elements and the value of the potential was evaluated at 49 internal points (Fig. 1 (b)). The boundary conditions for the problem were considered the heat flow (q) and the temperature (u) on the external boundary. Then, a plate with a central hole of radius 0.06 cm, with the same dimensions and boundary conditions, was also simulated, and the obtained results for the potential were compared with the plate without damage. For the internal boundary (hole) of the plate, zero heat flow was considered.

The results for the damage detection problem using potential formulation with 330 individuals in the initial GA population are presented in Fig. 2. This population was assembled considering holes of three different radius sizes (0.03, 0.09 and 0.15 cm) in 110 different positions for each radius on the plate, and the values of the potential difference (between undamaged and damage plate) at the 49 internal points. The values of the potential difference were normalized, taking in consideration the largest value of this difference. As the potential values near the right border (temperature equal to zero) of the plate are close to zero, the potential difference is used instead of the direct use of the potential value. The program was run only five (5) times, because there was no significant difference when this value was increased. In Fig. 2, the "real" position of the hole is represented in continuous line and the results found by the GA in non-continuous lines. Insets show the region of hole in detail.

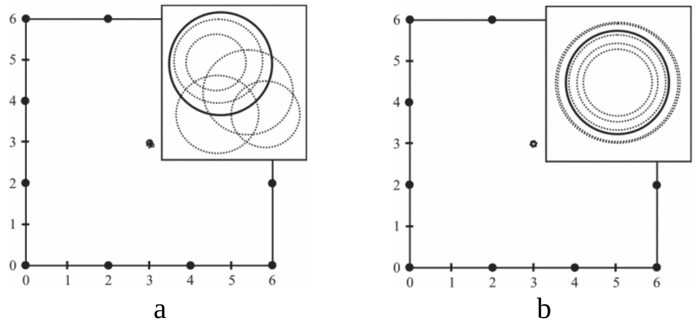


Figure 2: "Real" (continuous line) and simulated hole (non-continuous lines) for potential: (a) for a central hole with elitism equal to 2; (b) for a central hole with elitism equal to 10. Insets show the region of hole in detail.

After several attempts to configure the GA parameters, good results were obtained for the problem. Fig. 2(a) and (b) shows a hole with radius equal to 0.06 cm in the position (3; 3) cm. The difference between the two results is that in Fig. 2(a) the elitism parameter (number of individuals that survive to the next generation) was equal to 2, and in Fig. 2(b) this parameter was equal to 10. Increasing the value of elitism from 2 to 10, the holes were concentric (Fig. 2(b)), and the hole position presented a small uncertainty. Moreover, the radius for every simulation was not much sensitive to the variation of the GA parameters. In simulations, the tolerance of the problem was reached, in other words, there was no improvement in the objective function (fitness function) and the maximum number of generations was not reached, showing a good convergence of the algorithm. Due to the small mutation presence and a crossover function that is different for each run of the algorithm, the results are different for each run of GA approach, in other words, there is an associated occurrence probability. So, the developed program finds an occurrence area of the damage, what can be verified by the presented results.

The previous results were for the program that only detects one hole in the plate. For the program that detects up to two holes, the initial population took into consideration the 330 individuals representing only one hole in the plate and other 330 individuals representing two holes, totaling 660 individuals. How the initial GA population was formed for this case can be seen in (Lopes *et al.*, 2007). The results obtained from this program for the detection of a hole in the plate were similar to the results already presented. However, there was difficulty in detecting two holes in the plate. Perhaps, the values of the radius were very small and, moreover, there was lack of consistent information supplied by the BEM. Finally, the chromosome codification of the GA should be the most random to consider all possible solutions of the problem.

Now considering resolving the inverse problem through parameter identification technique, an ANN is a computational technique that presents a mathematic model to represent the human brain and to try to simulate the learning process of this brain. An ANN is formed by the interconnected neurons whose inputs can be obtained from the outputs of other neurons or from input nodes. Different configurations of the artificial neuron can be made to develop different network topologies that can be set for the layer number, amount of neurons in the layers and the connection type among the neurons (Rao *et al.*, 2006). In this work, a backpropagation neural network (BPN) is used, through a feedforward configuration and the backpropagation learning algorithm. In a feedforward configuration, neurons are interconnected in layers and the data flow only occurs in a direction (CHONG & ZAK, 2004). The backpropagation learning algorithm carries out a supervised training process where the desired outputs are given as part of the training vector. Then, the correct ANN output is found through the weight adjustment among the layers.

The ANN's simulate the non-linear behavior between the measured potential values in the plate and the hole parameters (location and size). In ANN, the potential difference in the plate is supplied in the input of the network, and the parameters (location and radius) of the hole are supplied in the output. After setting network parameters, the created network can be trained and tested for other potential difference data, obtaining as answer, the hole parameters.

Considering the previous problem of heat flow, initially the presence of a single hole in the structure was studied. Then, the influence in the results was verified when the number of sensor at the plate was decreased. The sensors were uniformly distributed on the plate and no positioning study of the sensors was performed. The problem domain is reduced when there is a decrease of the sensor number on the plate. A hole with a radius equal to 0.05 and 0.15 cm in nine different hole positions was considered to assemble the input (potential) and output (hole parameters - location and size) data of ANN. After a few attempts to configure the ANN, the results for a hole of radius 0.10 cm in the positions (3; 3) cm (Fig. 3(a)), (1; 1) cm (Fig. 3(b)), and (5; 5) cm (Fig. 3(c)) for five (5) sensors on the plate could be found. The “real” position of the hole is represented in continuous line and the results found by the ANN in non-continuous lines.

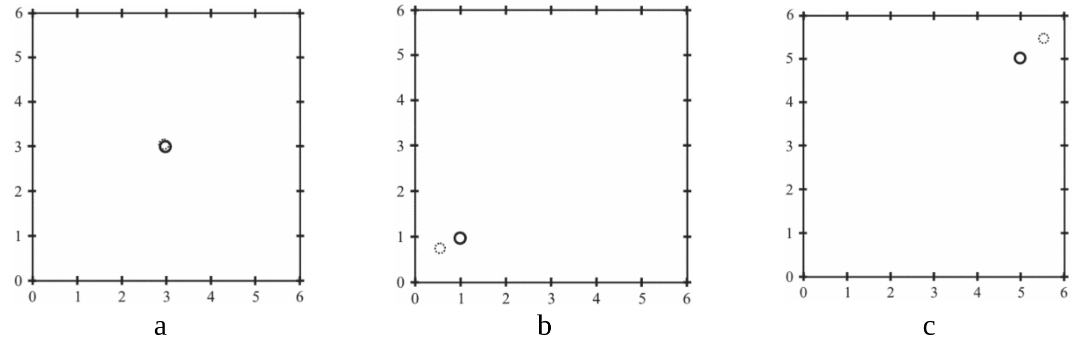


Figure 3: Potential problem: results from the ANN with five sensors for a hole at position: (a) (3; 3) cm; (b) (1; 1) cm and (c) (5; 5) cm.

In order for the ANN to detect more than one hole, the input data in the damage detection program needed to be modified. In this case, 25 sensors were considered on the plate and no reduction in the sensor number was done. The results showed that it is more difficult to detect more than one hole. The results depend on the quality of the input data of the ANN and of the appropriate choice of the configuration parameters of the network. To continue with the detection of more than a hole in the plate through the ANN, the direct problem (data obtained from the BEM) should be gotten better. New loadings on the plate and a new BEM should be considered, allowing to identify circular and elliptic holes, and also cracks, in the structures.

2.2 Detection of Holes in a Plate Using Global Optimization and Parameter Identification Techniques

The life time of any structure can be predicted through the correct determination of the damage. To determine the damage, the numerical modeling consists in a direct problem and an inverse problem. For the direct problem, a model is required to obtain information on the distribution of the quantity of interest throughout the structure, given the boundary conditions and the presence of the damage. For the inverse problem, a model is required for the procedure of locating the damage in the structure given some (partial) information on the quantity of interest at some particular locations (for example, where some sensors are placed). As in Section 2.1, the software MATLAB® was used for the development of the damage detection program.

In this work, the direct problem is modeled by means of the elastostatic formulation of the boundary element method (BEM). In this formulation, the quantities of interest are the interior point displacements and stresses. The problem consisted of a plate with an internal hole, considering some boundary conditions (traction on the external surface of the plate). The stresses at internal holes in the plate are assumed null. The inverse problem of identifying the presence, location and size of damages (circular holes) in a plate structure is modeled using optimization and parameter identification techniques. Again, the genetic algorithm (GA) is used as an optimization technique and the artificial neural network (ANN) is used as a parameter identification technique. GA and ANN are independent techniques to obtain the damage location, thus providing a means to verify the results.

As in Section 2.1, GA is used to find the optimal solution to the problem through a functional. For elastostatic formulation, the same equation (Eq. (1) in Section 2.1) can be used as fitness function, considering the mean stress values instead of the potential values. The minimization of this fitness function allows the damage detection program to find the unknown parameters of the damage.

To analyze a circular hole detection problem, a plate with the dimensions (0.06×0.06) m was simulated through the BEM, as illustrated in Fig. 4(a), for elastostatic problem.

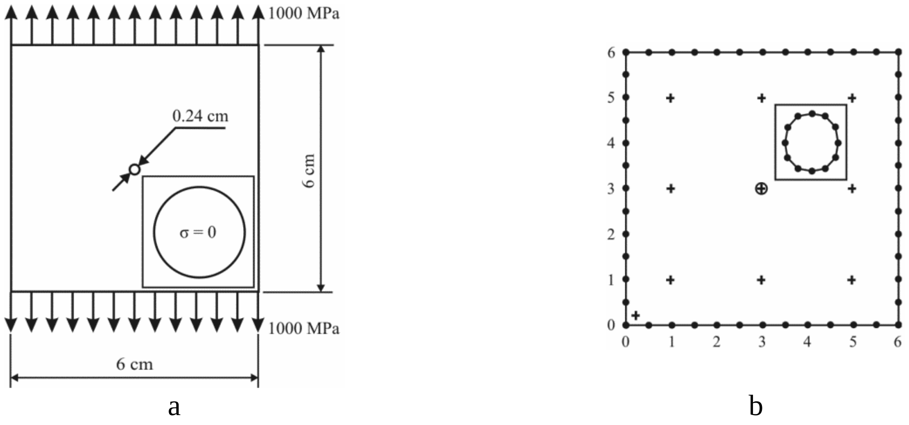


Figure 4: Plate model for elastostatic problem: (a) dimensions, loading and boundary conditions; (b) boundary discretization and sensor locations. The inset shows a stress-free hole and hole discretization for elastostatic problem.

For the elastostatic problem, the boundary conditions (Fig. 4(a)) for the external boundary were considered a pair of equal and opposite tractions (tensile stress equal to 1000 MPa) and for the internal boundary (hole) of the plate were considered zero traction. A study of the influence of numerical errors due to the BEM discretization for the external contour of the plate in the optimization was performed in this problem. Fig. 4(b) shows the discretization for the case of 48 elements in the outer boundary and 12 elements in the hole, as well as the position of the nine sensors uniformly distributed on the plate. The plate was simulated with shear modulus equal to 94,500MPa and a Poisson's ratio for plane strain equal to 0.1.

The results for the problem with 363 individuals in the initial GA population are presented in Fig. 5 for elastostatic formulation. This population was assembled considering holes of three different radius sizes (0.05, 0.10 and 0.15 cm) in 121 different positions for each radius on the plate, and the values of the difference (between undamaged and damage plate) in the mean stress at the 9 internal points. The values of the difference in the mean stress were normalized, taking in consideration the maximum value of this difference. The values of x and y coordinate of the hole center and its radius were also normalized, considering the respective maximum value. The program using GA was run ten (10) times and generated a different optimal solution each time it ran the algorithm due to its own randomness. Nevertheless, the results of the GA approach present a tendency to be concentrated near the "real" hole. The "real" position of the hole is represented in continuous line and the results found by the GA in non-continuous lines.

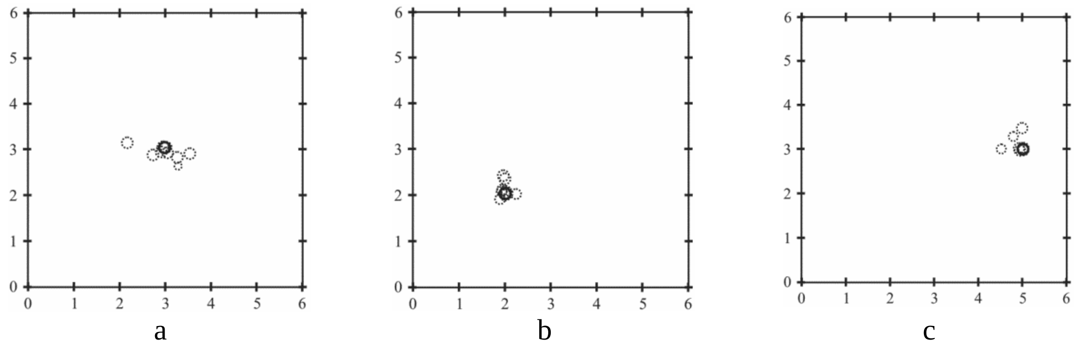


Figure 5: “Real” and simulated hole for mean stress: (a) for a central hole; (b) for a hole at (2; 2) cm and (c) for a hole at (5; 3) cm.

Again, after several attempts to configure the GA parameters, good results were obtained for the problem. Fig. 5(a) shows the results for a central hole; Fig. 5(b) shows a hole located at (2;2) cm; and Fig. 5(c) shows a hole located at (5;3) cm. The radius of each plot was considered equal to 0.12 cm. It is worth noting that for each problem under study, a new configuration of the GA must be performed, which is therefore different from the previous problem. GA also presents a high computational cost due to the several evaluations of the fitness function. The damage detection code using GA can find a region for the probable occurrence of the hole, as this algorithm generates a different optimal solution every time it is run. Thus, a confidence interval, for the different parameters being identified, can be obtained.

Now, considering the elastostatic formulation and the same normalized data from the initial GA population for this formulation, the ANN simulates the non-linear behavior between the values of the local difference in the mean stress (between undamaged and damage plate) and the hole parameters (location and size). Information regarding the difference in the mean stress is supplied in the input of the network, besides the parameters of the hole are supplied in the output of the same network. After creating and training the ANN, this network was tested for a 0.12 cm radius hole in different positions. Fig. 6 shows the results for nine sensors on the plate. The “real” position of the hole is represented in continuous line and the results found by the ANN in non-continuous lines.

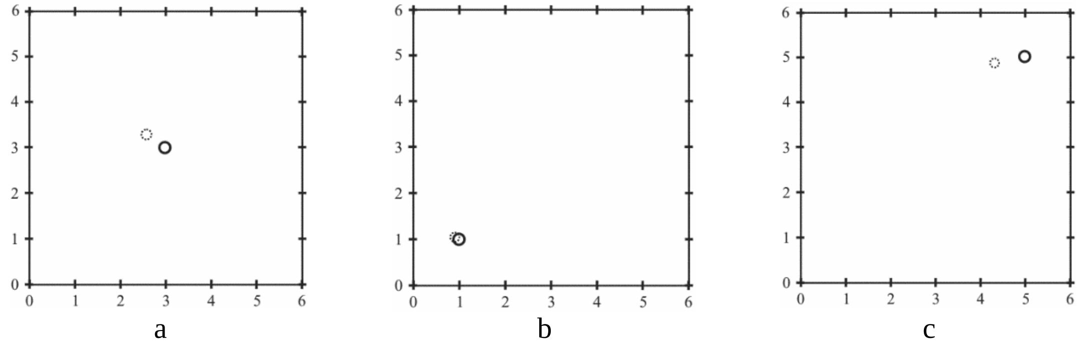


Figure 6: Elastostatic problem: results from the ANN with nine sensors for a hole at position: (a) (3; 3) cm; (b) (1; 1) cm and (c) (5; 5) cm.

In Fig. 6, the results show a small area of uncertainty near the “real” hole and the hole size was obtained with good accuracy. These results were obtained more quickly than in the case of using GA (as a global optimization technique). For this reason, the solution of a damage detection problem through the ANN (as a parameter identification technique) is also known as an online identification. An advantage of the use of ANN in regard to the GA is that, after training the network, holes with different sizes and in different locations can be tested without running the damage detection program again. More information about how ANN was set up and trained can be found in (Lopes *et al.*, 2008).

2.3 Detection of Holes in a Plate Using Global Optimization and Parameter Identification Techniques

Several types of static and dynamic loads and the structural deterioration process can cause different types of structural damage. The knowledge of the change in the material properties corresponding to the damage depends on the type of material and structural configurations. The assessment of the structural damage can be performed through a comparison between measured and simulated data. A measured data represents information about a “real” hole and a simulated data represents information obtained from each run of the inverse problem. Usually, the information on the “real” plate (a plate with a hole with unknown size and location) would be available by means of an experimental device, in which sensors would be put in all selected interior point locations. However, a numerical code is required to obtain both simulated data and measured data, in which a direct model of the problem is consistently used by an inverse problem algorithm.

For the direct problem, two formulations based on Boundary Element Method (BEM) were required to obtain the information on the distribution of the quantity of interest throughout the structure, given the boundary conditions and the presence or absence of the damage. Potential formulation for the heat transfer (conduction) and elastostatic formulation for the distribution of displacements and stresses on the plate at given points. In both cases, a small hole inside the domain is modeled as damage on the plate.

For each run of the direct model, the information about the hole (location and radius), boundary conditions, loading in addition to information on hole and plate discretization are provided. After evaluating the boundary solution, the BEM code evaluates, as a post-processing, some quantities of interest at selected interior points that can be candidates to sensor locations for an experimental setting. Each run of the direct method using the potential formulation provides one piece of information (the potential, i.e. the temperature) at the selected interior points. On the other hand, the elastostatic BEM formulation provides three pieces of information at an interior point (the components of the stress tensor, i.e. two normal stresses and one shear stress). As the goal of the inverse method is to identify and locate the hole, but not to identify any direction-dependent properties, mean stress is used as an independent scalar quantities obtained at the selected interior points.

The inverse problem of identifying size and location of a small hole in a plate structure can be modelled using optimization and parameter identification techniques. The genetic algorithm

(GA) is used as the optimization procedure and the artificial neural network (ANN) approach is used as a parameter identification technique. By solving the inverse problem using two independent techniques (GA and ANN), more reliable information on the damage parameters can be obtained, as a comparison of the results from both approaches can provide a means to verify these results and also allows for the validation of the inverse procedure.

The presence of damage may induce rapid changes in the field variable of the problem, and even discontinuities in the governing equation in the domain. Classical calculus-based optimization methods require evaluation of derivatives of the objective function, which may not be possible to be obtained, or may be numerically obtained, with unacceptable inaccuracy. Besides, these problems can have several local minima (multiple solutions), and thus a global optimization method (such as GA) is a better choice for the numerical solution (Engelhardt *et al.*, 2006; Stavroulakis & Antes, 1998). On the other hand, GA uses multiple points to search for the solution, rather than a single point, and a global minimum has a better chance of being obtained. Also, as GA does not require any evaluation of derivatives, no errors are included in the solution due to the approximation of these derivatives.

Damage detection problem in a thin plate can be formularized as an optimization problem using GA according to the flowchart in Fig. 7. The initial population of GA is a set of possible solutions for a given problem that can be formed by the geometric information of a numerical hole (*x* and *y* coordinates of its center, and also its radius) and also by differences in the quantities of interest (‘difference 1’) calculated at selected interior points. ‘Difference 1’ is the local difference in the potential or the local difference in the mean stress between the undamaged plate and the plate with damage for potential and elastostatic formulations, respectively. ‘Difference 2’ is a set that can be evaluated at the same interior points, representing the ‘measured’ differences for the quantity of interest at these points for the “real” hole (also simulated in this work). To validate the damage detection approach, the value of ‘Difference 2’ was not allowed to be in the initial population of the GA approach. The initial population and also ‘Difference 2’ are employed in the fitness function. The fitness function can be represented as the functional presented in Section 2.1 by Eq. (1) for the potential formulation. For elastostatic formulation, the same equation can be used, considering the mean stress values instead of the potential values.

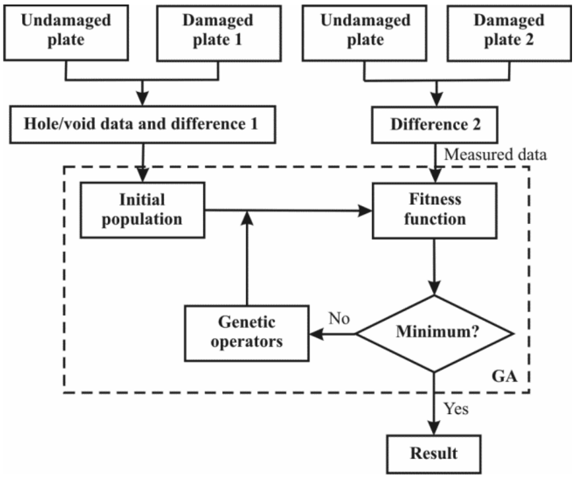


Figure 7: Flowchart for the optimization procedure using GA.

The goal of the GA approach is to look for a minimum value of the fitness function. For that, the algorithm uses genetic operators to modify the population and subsequently reevaluate the fitness function for the new population. As convergence criteria, the maximum number of generations or epochs was assumed, together with a default criterion for the tolerance (more details can be seen in (Lopes *et al.*, 2010)). When the convergence criterion is met, the numerical holes have reached the vicinity of the "real" hole, and thus the information about the location and size of the "real" hole is obtained.

The problem of damage detection in a thin plate also can be formularized as a parameter identification problem (using ANN) according to the flowchart in Fig. 8. In this flowchart a network is created, considering 'Difference 1' (the same 'Difference 1' as in the GA approach) as the input data and the geometric information for the hole (*x* and *y* coordinates of the hole center, and its radius) as the output data. The next step is to train the created network, obtaining, as a result, a NET that contains information about how to proceed for another input data in the problem domain. Finally, the trained network is simulated for 'Difference 2' (same 'Difference 2' as in the GA approach). Similar to the optimization algorithm, convergence criteria (error goal and epochs) was set to this approach. When the convergence criterion is met, the ANN has identified the "real" hole providing the information about its location and size.

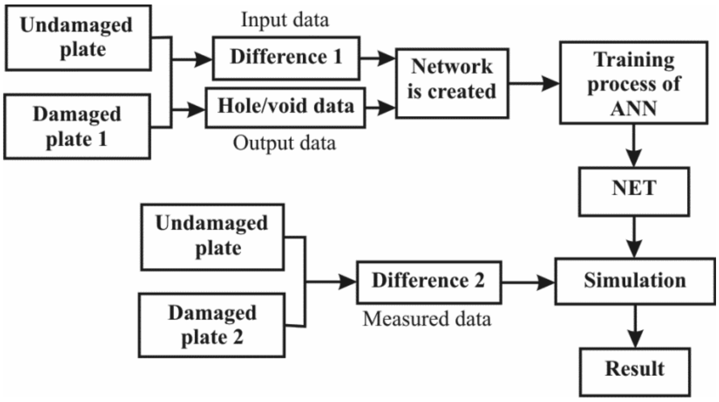


Figure 8: Flowchart for the parameter identification procedure using ANN.

To analyze the circular hole detection problem, a plate with the dimensions (0.06×0.06) m was simulated through the BEM, as illustrated in Section 2.1 by Fig. 1(a) for potential problem and in Section 2.2 by Fig. 4(a) for elastostatic problem. In addition, the plate discretization and sensor location (internal points) were presented in Section 2.1 in Fig. 1(b) for potential problem and in Fig. 4(b) for elastostatic problem. The results for the damage detection problem using GA and ANN, considering the potential formulation, can be seen in Section 2.1. In Section 2.2, the results for the elastostatic formulation are presented for both techniques.

Then, the introduction of random noises into measured data to examine how the inverse method using GA responds to measurement error was investigated. The random noise is a signal formed by a set of random numbers drawn from a normal distribution with zero mean (white noise) and with the coefficient of variation (COV) given as a percentage (5% or 10%) of the measurement value at the sensor location. The flowchart presented in Fig. 9 shows this approach.

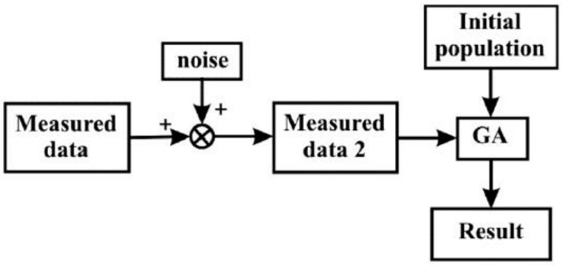


Figure 9: Flowchart for the analysis of the measurement error.

As can be seen in Fig. 9, the noise is added to the measured data to create a set called “Measured data 2”. This new measured data was normalized and then used in the GA approach for the elastostatic problem. The GA approach was run 10 times for each case (5% and 10% noise), always considering the same configuration of parameters as in the case without noise. In each run of the GA, a different noise signal was generated with the proper COV. A hole in (3, 3) cm position with a radius size equal to 0.12 cm was simulated for the elastostatic problem, considering each random noise into measured data. The results show that the GA optimization procedure, for identification and localization of the hole in the structure, presents very small sensitivities to changes in the measured values at the sensors, proving the robustness of the algorithm.

A plate with external dimensions (0.10×0.10) m was simulated for comparison with the literature results (Stavroulakis & Antes, 1998). The results found for the elastostatic problem using GA by both examples are shown in Table 1. In both examples, the loading was applied on the left-hand side of the external boundary of the plate and the right-hand side was fixed, material constants were considered equal to 100 GPa for shear modulus, 0.3 for Poisson’s ratio and the results were obtained after 200 generations of GA. In this work, the results were reached for a static loading of 1000MPa in horizontal and vertical coordinate direction, the GA population was equal to 49 individuals and only a hole with diameter equal to 0.5 was considered in some positions where the test case (“real” hole) was not included in the initial population, validating the results obtained. In (Stavroulakis & Antes, 1998), the plate was subjected to a harmonic dynamic loading in both directions on the left-hand side of the plate, the GA population was equal to 5 individuals and no information was given in that text on how the individuals of the population are placed in the plate.

Table 1: GA approach: comparison with literature results.

Test	“Real” hole	Results presented by Stravoulakis & Antes (1998)			Results in this work		
		Calculated best element	Average for 1000 solutions	Error (%)	Calculated best element	Average for 20 solutions	Error (%)
x	4.0	3.9606	5.59	38.75	3.7336	3.52	12.00
y	4.0	4.0236	4.74	18.50	3.9578	3.95	1.25
Diameter	0.5	0.4968	0.52	4.00	0.5000	0.53	6.00

As shown in Table 1, the GA approach used in this work has presented, for most cases, more accurate results in the identification of the “real” hole dimensions, with respect to the GA approach used in the literature example. In that literature example, an average of 1000 solutions was computed, while in this work, only an average of 20 solutions was performed. Also, for each solution, only a few seconds were needed to run the inverse program using GA on a PC. These features illustrate the accuracy and the low computational cost of the current approach.

In short, the analysis of the results indicates that the damage detection code using GA can only find a region for the probable occurrence of the hole, as this algorithm generates a different optimal solution every time. Moreover, the GA approach in this work was robust in regard to the measurement error, as only a small error was obtained in the results when a noise of 10% was added to the measured data. Also, this GA approach compares well, both in accuracy and in computational cost, with respect to a similar GA approach used in the literature for damage identification. ANN has also generated good results for the several parameters being identified. An important observation is that very small holes are difficult to observe by the damage detection program, mainly when these holes are close to the borders of the plate. The optimization and the identification techniques adopted in this inverse problem can be used concomitantly, as independent procedures to identify the presence of a hole on the plate, thus providing a means to verify the numerical results obtained for the location and size of the damage in the structure, increasing the confidence in the damage identification results.

2.4 A Robust Optimization for Damage Detection Using Multiobjective Genetic Algorithm, Neural Network and Fuzzy Decision Making

Damage can cause changes in the properties of a structure whose effects can be analyzed by inverse damage detection techniques. The inverse problem of damage detection can be modeled through a direct problem, an inverse problem and the presence of uncertainty. In the direct problem, given the boundary conditions and the presence of the damage, the distribution of the quantity of interest throughout the structure is obtained. In the inverse problem, a procedure of locating damage in the structure given some information on the quantity of interest at some particular locations is modeled. Moreover, both direct and inverse problems are stochastic, therefore some kind of treatment of randomness needed to be performed at variables of the problems. Uncertainties are present in modeling of the plate structure under study, at damages in this plate structure and at numerical modeling of the problems.

Considering the direct problem modeling, BEM approach in 2D was used for elastostatics problem. Two BEM model was built for a plate, a model for a circular hole on the plate and a model for a crack that can be represented as an elliptical hole (with semi-minor axis much smaller than the semi-major axis). For the plate model with a circular hole, the parameters of the direct problem are the same as shown in Section 2.3 (hole boundary conditions, plate and hole discretization; mean stress between undamaged and damage plate, etc.). For the plate model with a crack, a plate with the dimensions (1.00×1.00) m was simulated. In this plate, an elliptical hole has as parameters the angle of inclination θ to the horizontal axis, semi-major axis a, semi-minor axis b, and the center of the hole (x,y). Nine internal points were chosen on the plate (uniformly distributed) to provide the desired information. For the inverse problem, after the direct BEM model evaluates the differences in mean stress between the undamaged and damage plate for the interior points, these differences are supplied as input to the multiobjective GA subroutines.

Optimal values to the objective functions and minimum variations of these functions at the optimal point vicinity are the goals of robust optimization. In this case, robust optimization also is a multiobjective problem and the optimal solutions for a problem are robust because these solutions are points in the feasible region where the values of objective function are insensible to small variations around these points. In this work, a robust optimization problem was performed, considering three approaches: (*i*) an approach with a functional function and a functional variance function for normal distribution, considering information of sensors in the population of multiobjective GA; (*ii*) another approach with the same functions, however without information of sensors in the population; and; (*iii*) an approach with functional function and no function for functional variance, but a relation between functional variance and hole parameters (center and radius) found by ANN for each individual in fitness function of multiobjective GA.

Taking the first and the second approaches into account, a function for the variance of the functional (Eq. (1) in Section 2.1) needs to be found. In these cases, the variance function is obtained through a multivariate regression with terms until third order. The independent variables are information about holes (for a circular hole, x and y coordinates of its center, and also its radius r) and the dependent variable is the standard deviation of the functional formulation for each hole. The whole procedure of how the variance function was found is presented in the work (Alexandrino *et al.*, 2019). A natural logarithm of values was used for a change of scale, then, a multivariate regression was performed with regard to x, y and r parameters (considering a 95% confidence level). Since the place of sensors was not considerate at computations of the functional variance function, discontinuities can be avoided at this function. The multivariate regression function found presents a R2 value equal to 83.9% and a p-value equal to 0 for the normal distribution.

In Fig. 10 is presented a flowchart to the robust optimization problem considering information of sensors ("Difference 1") in the population of multiobjective GA (first approach).

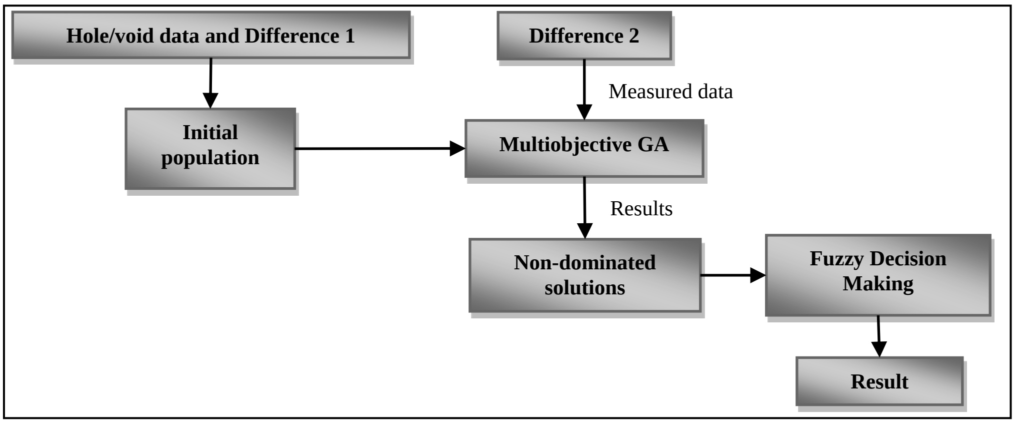


Figure 10: Flowchart for the optimization procedure using multiobjective genetic algorithm e fuzzy decision making.

Considering this Fig. 10, the initial population for the multiobjective GA approach is formed by the geometric information of a numerical hole (*x* and *y* coordinates of its center, and also its radius *r*) and also by differences in the mean stress ("Difference 1") calculated at selected interior points on the plate. This initial population and "Difference 2" ("measured" data for the mean stress in interior points for "real" hole) are employed in the fitness function of multiobjective GA. Bearing in mind that the "Difference 2" values are not in the initial population in order to validate the damage detection approach. The fitness function is formed by two functions (functional formulation and a function for the standard deviation of this functional or square root of the functional variance function). The result obtained from multiobjective GA (considering a variant of NSGA-II algorithm developed by (Deb *et al.*, 2000) in MATLAB®) is a set of Pareto front points (non-dominated solutions) and the best tradeoff solution is found by a decision-making method based on fuzzy set theory.

The initial population for multiobjective GA approach with the presence of sensors information (first approach) and a circular hole was assembled with 168 individuals. The holes of this population had three different radius sizes (0.10 cm, 0.125 cm and 0.15 cm) in different positions for each radius on the plate and the place of sensors was not considerate in the initial population. The other GA parameters can be seen in (Alexandrino *et al.*, 2019). The Pareto front for a hole in (1.0;2.0) cm and radius equal to 0.12 cm is showed in Fig. 11(a). This Pareto front was obtained in GA generation equal to 106. The number of Pareto front points was equal to 126 and these points were represented in non-continuous (dashed) line in Fig. 11(b). In this same figure, the "real" hole is represented in continuous line and the fuzzy decision making results ("Result 1", "Result 2", and "Result 3") in dash-dot line. The "real" hole, Result 1, and some results from multiobjective algorithm are showed with more details at zoom area in this Fig. 11(b).

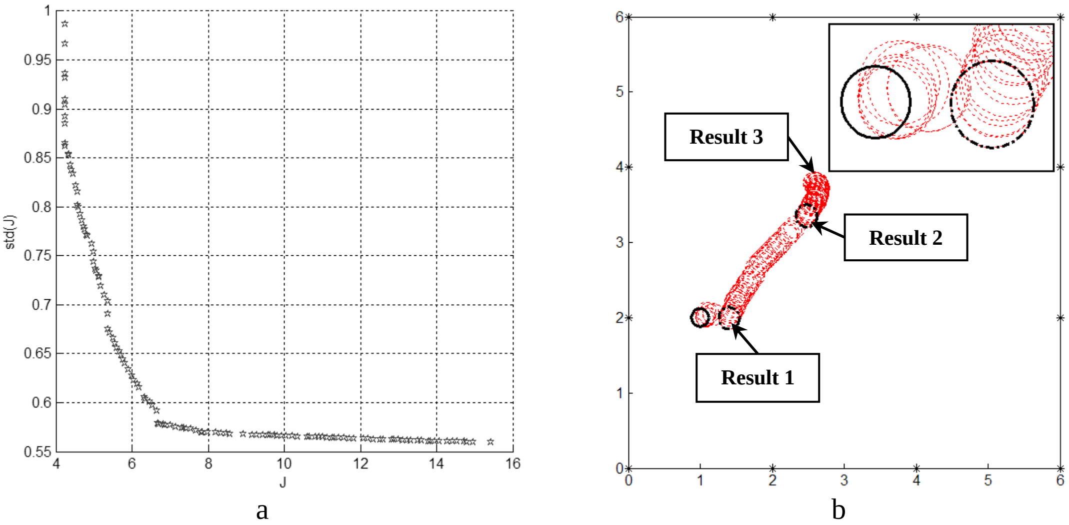


Figure 11: First approach results: a) Pareto front; b) "Real" hole (full line), 126 holes found by the variation of NSGA-II algorithm (dashed line), and fuzzy decision making result (dash-dot line).

These fuzzy decision making results in Fig. 11(b) consider different fuzzy qualifiers that works with imprecise information (Alexandrino *et al.*, 2019). "Result 1" is the result from fuzzy decision making where the functional formulation function is "more important" than standard deviation of this functional. “Result 2” presents the result for no importance (without using the fuzzy qualifier) to the functions (functional formulation and its standard deviation). “Result 3” is the result for the case where only the standard deviation of functional formulation is present. This last result (“Result 3”) corresponds to the hole more distant from “real” hole.

In Fig. 12 is presented a flowchart to the robust optimization problem without information of sensors in the population of multiobjective GA (second approach).

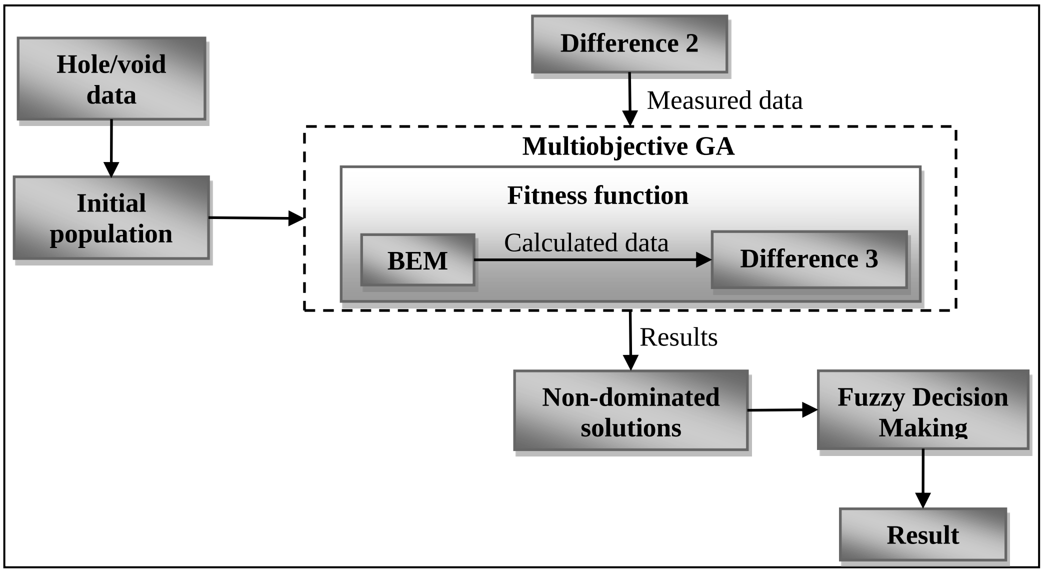


Figure 12: Flowchart for the multiobjective optimization procedure using multiobjective GA, without information of sensors in the population.

Considering this Fig. 12, the initial population for the GA approach is formed by only the geometric information of a numerical circular hole (*x* and *y* coordinates of its center, and also its radius *r*). This initial population and “Difference 2” (“measured” data for the mean stress in interior points for “real” hole) are employed in the fitness function. Again, the fitness function is formed by two functions (functional formulation and a function for the standard deviation of this functional). As “Difference 1” is not in initial population, now the “calculated” vector of the functional formulation is a BEM procedure that finds the differences in mean stress (“Difference 3”) for each individual of population in a generation. The “measured” vector is the “Difference 2” set.

The initial population for multiobjective GA approach without the presence of sensors information (second approach) and a circular hole was assembled with only 6 individuals. The results found by multiobjective GA to a hole at (1.0,2.0) cm and radius equal to 0.12 cm are showed in Fig. 13. The number of points on the Pareto front was equal to 5. The result obtained from multiobjective GA approach using fuzzy decision making shows that the exact location of “measured” hole was found.

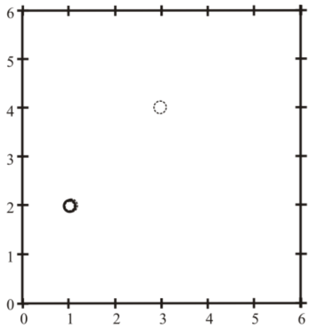


Figure 13: Graphical representation of 5 Pareto front points (dashed line) and "real" hole (full line).

Then, taking the third approach into account, a relationship between the functional variance and the hole parameters using ANN, needs to be found. Figure 14 presents a flowchart to the robust optimization where a relation between functional variance and circular or elliptical hole parameters can be found by ANN. This relation is performed to each individual in fitness function of multiobjective GA so, no function to variance is necessary. The created network is known as "NET" that is used in fitness function to find the variance which mean squared give a standard deviation for each hole information in the fitness function. Again, a set of Pareto front points (non-dominated solutions) is obtained from resolution of multiobjective GA problem and the best tradeoff solution can be found by fuzzy decision making method. In this flowchart, the sensors information ("Difference 1") is considered in the initial GA population.

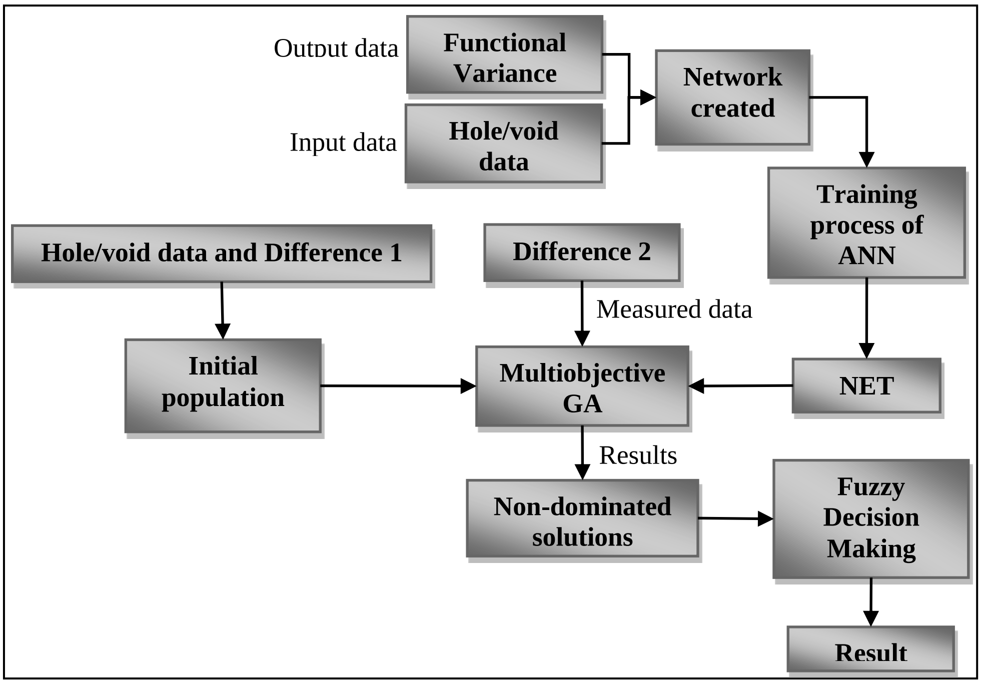
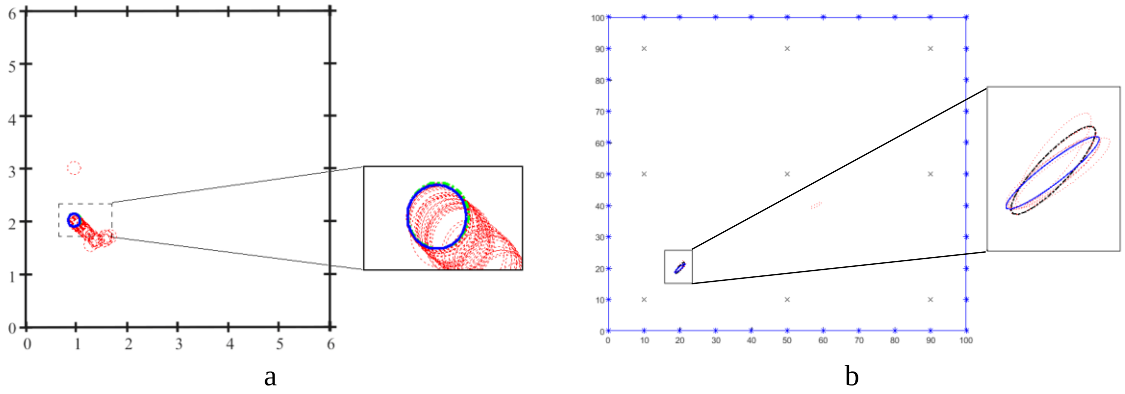


Figure 14: Flowchart for the optimization procedure using multiobjective genetic algorithm, artificial neural network and fuzzy decision making.

Considering a circular hole, the “real” hole is represented in continuous line in position (1.0;2.0) cm with a radius equal to 0.12 cm in Fig. 15(a). The results were obtained in generation equal to 10 for the same initial population as in the first approach. The number of Pareto front points was equal to 77 and these points are represented as circular holes in non-continuous (dashed) line in Fig. 15(a). The fuzzy decision making results are represented in dash-dot line. The “real” hole and some results from multiobjective algorithm are showed with more details at zoom area in this Fig. 15(a).

Now, for elliptical hole representing a crack in hole in (20;20) cm, with semi-major axis equal to 4.0 cm and semi-minor axis equal to 0.75 cm, and angle of inclination equal to 45° is showed in Fig. 15(b) in continuous line. The results for Pareto front points equal to 24 are represented as circular holes in non-continuous (dashed) line. These results were obtained in generation equal to 20 and the initial population of GA consisted of 256 individuals. The initial population and the configuration of the GA approach can be seen in (Alexandrino *et al.*, 2019).



a b

Figure 15: Third approach results: a) for a circular hole; b) for a elliptical hole.

In Fig. 15(a), the result from fuzzy decision making was a hole at location x = 1.006 cm, y = 2.005 cm, and radius r = 0.124 cm. This result considered the functional formulation function “more important” than standard deviation of this functional. An error in x position was found about 0.59%, in y position was found about 0.23%, and an error in radius was found about 3.07%. These error results show that an approach using ANN to find a relation between functional variance and hole parameters (center and radius) is a better choice than an approach where a function of variance functional was found. In Fig. 15(b), the result for functional function considered “much more important” than its standard deviation was a hole in (20.0;20.1) cm, with semi-major axis equal to 4.2 cm and semi-minor axis equal to 1.00 cm, and angle of inclination equal to 54.4°. Due to the inherent randomness of the methodology, if it is performed again, similar results will be obtained. The methodology cannot find the exact values of parameters but can determine the region with the damage with good accuracy.

Still considering an elliptical hole, Fig. 16 shows the convergence behavior of the separately treated J_1 and J_2 (deviation of J_1) functions by addressing the minimum of each function (of the optimal Pareto vector for each one). It is observed the efficient process of convergence of the proposed algorithm.

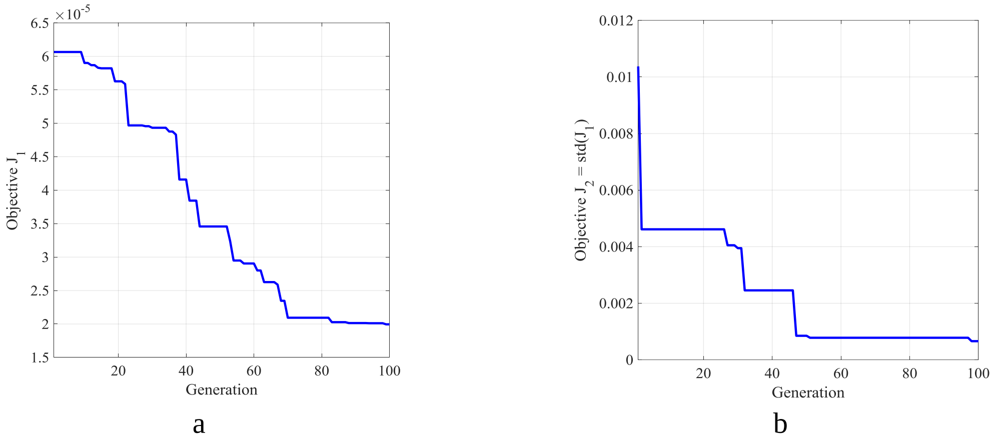


Figure 16: Convergence of the objective functions: a) J_1; b) J_2.

It is known that an elitist GA always favors individuals with better fitness value (rank). A controlled elitist GA also favors individuals that can help increase the diversity of the population even if they have a lower fitness value. It is important to maintain the diversity of population for convergence to an optimal Pareto front. Since a multiobjective optimization was applied to this work, it would be interesting to evaluate the convergence behavior through the Pareto front. Figure 17 shows the Pareto fronts for the inverse problem proposed for some specific generations. This Pareto front is for elliptical hole representing a crack in (20;65) cm, with semi-major axis equal to 2.4 cm and semi-minor axis equal to 0.6 cm, and angle of inclination equal to 0°. The results shown in Fig. 17 show the approximation of the optimal solutions to the axes. In addition, Fig. 18 shows a 3D projection of all 100 Pareto fronts (100 generations) obtained. It can also be seen the convergence of all the non-dominated solutions to the optimal front.

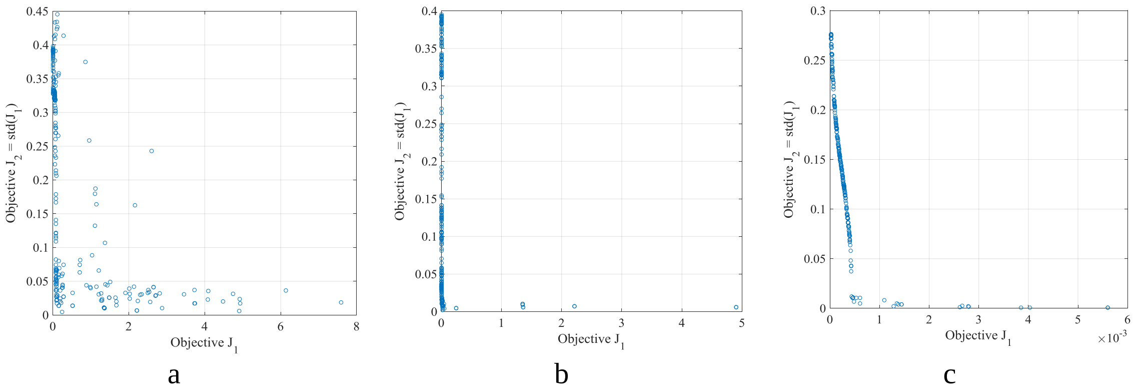


Figure 17: Pareto front convergence for generation: a) 10; b) 40; c) 100.

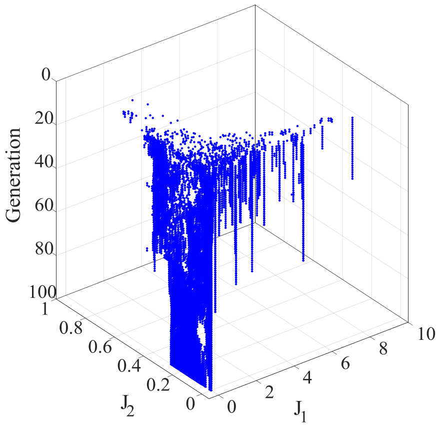


Figure 18: Pareto fronts for the 100 generations evaluated showing the convergence process.

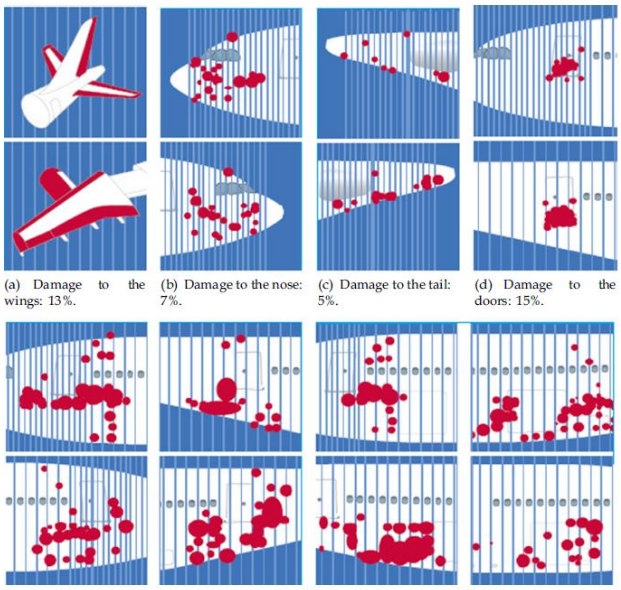
Results for the problem, considering that sensors information was not presented in the GA population, were better than the program where this information was present in the population. However, the inverse problem solution for the first case presented a high computational cost. This difference at computational cost occurred because the BEM routine was executed several times during the run of the damage detection. Finally, better results and small errors were obtained for third approach, mainly considering the error in the radius of hole.

2.5 An estimate of the location of multiple delaminations on aeronautical CFRP plates using modal data inverse problem

Structural health monitoring (SHM) is an interdisciplinary field in engineering that deals with innovative methods of structural monitoring, integrity, and performance without affecting the structure itself or harming its operation. The SHM methodology uses several types of sensors to detect the presence, location, and severity of structural damage. Such technology integrates non-destructive evaluation (NDE) techniques using sensory and intelligent materials to create self-monitoring mechanisms characterized by greater reliability and longer structural life. The method is applied mainly to systems with critical requirements regarding structural performance, where the classical evaluation of localized inspection is costly, difficult, or even impossible in terms of operationality (Stepinski *et al.*, 2013).

SHM is an innovative form of embedded non-destructive testing (NDT) that can be employed to directly assess the integrity of aeronautical structures. The principle of SHM is comparable to that of the human nervous system, where the sensors form a network comparable to the nervous system, detecting and diagnosing structural damage, mechanical loads, or abnormal conditions. The aviation company AIRBUS® interrogates its sensors through a diagnostic system “on-board” or “off-board” and structural condition information is reported to the maintenance team. In contrast to conventional NDT, there is no need for a qualified inspector to access the inspection area and to perform measurements, which in most cases are expensive and time-consuming. Real data is laid out in Fig. 19 showing a significant amount of damage of an A-320 commercial aircraft. The company manages to overcome this problem by employing SHM technology, which has shown great potential for reducing maintenance times and costs in some cases, while also increasing aircraft availability (Wenk & Bockenheimer, 2014).

Currently, there are a number of relevant techniques for identifying and locating structural damage. Although each technique has its advantages and disadvantages, there is no general algorithm that can resolve all types of problems in all types of structures. Every technique tends to be sensitive regarding damage. In other words, a very sensitive technique can produce false positives, while a less sensitive technique can lead to false negatives, the latter being the most problematic. Generally, only damage above a certain size (threshold) can be detected (Montalvão *et al.*, 2006). Therefore, this research deals with aspects related to the detection of delamination in structures of composite material using a vibration measurement approach. Variations in modal behavior strongly indicate structural states, and, when properly analyzed by efficient methods, can indicate the presence and location of a certain types of damage.



(a) Damage to the wings: 13%.

(b) Damage to the nose: 7%.

(c) Damage to the tail: 5%.

(d) Damage to the doors: 15%.

(e) Damage to passenger doors: 31%.

(f) Damage to cargo compartment: 22%.

Figure 19: Mapping of damage in service to the fuselage of an Airbus A-320 aircraft. Locations with damage are marked in red in positions delimited by vertical lines (adapted from Wenk & Bockenheimer, 2014)

The premise for these techniques is that damage causes a change in structural physical properties, especially in stiffness and damping at the damaged locations. These structural property changes in turn alter the dynamic response behavior of the structure respective to its initial state. Therefore, monitoring changes in structural response parameters can be an important tool for assessing structural integrity and identifying damage as early as possible. Based on the technique used for the measured responses in damage identification, methodologies can be classified as either "non-model based" or "model-based" (Bayissa & Haritos, 2007). Model based methods are able to deal with many facets of damage, such as locating and quantifying the severity of the damage. On the other hand, non-model-based methods are often used to identify and locate damage based on two data sets from the undamaged and damaged states.

The global damage identification problem can be summarized in the flowchart in Fig. 20. In a first step, we proceeded to manufacture an undamaged plate. This same structure was modeled via FEM and analyzed according to its modal characteristics in the free vibration test (obtaining the first natural frequencies at this time). The same procedure was performed on the actual plate in the laboratory. To do this, modal assay was performed and the natural frequencies were obtained in the same way. In this step, the real and numerical natural frequencies were compared. As errors can be associated with the material test (signal acquisition, boundary condition, material property), an inverse method was performed using a GA to adjust the properties of the numerical model. This is essential if the numerical and experimental models are to be in perfect harmony.

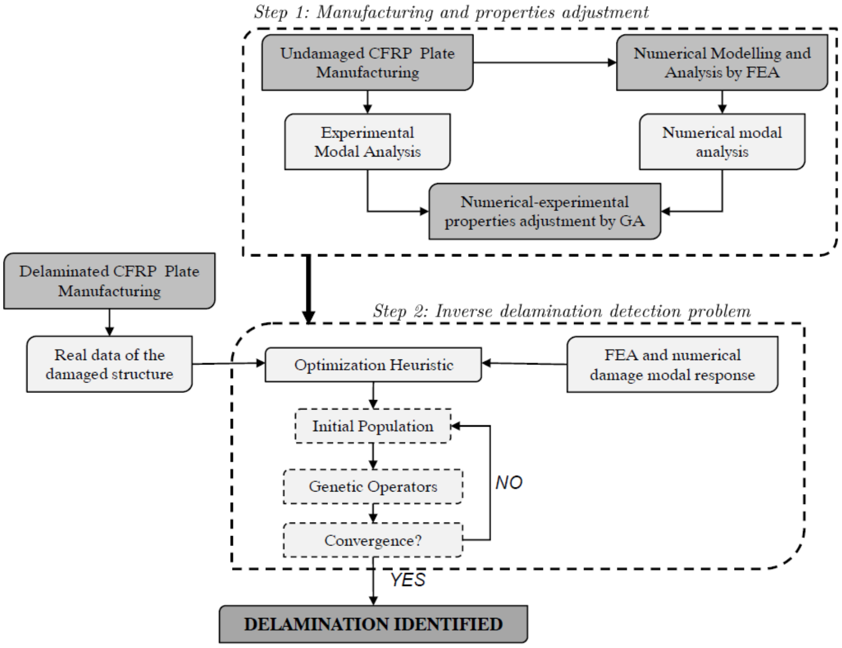
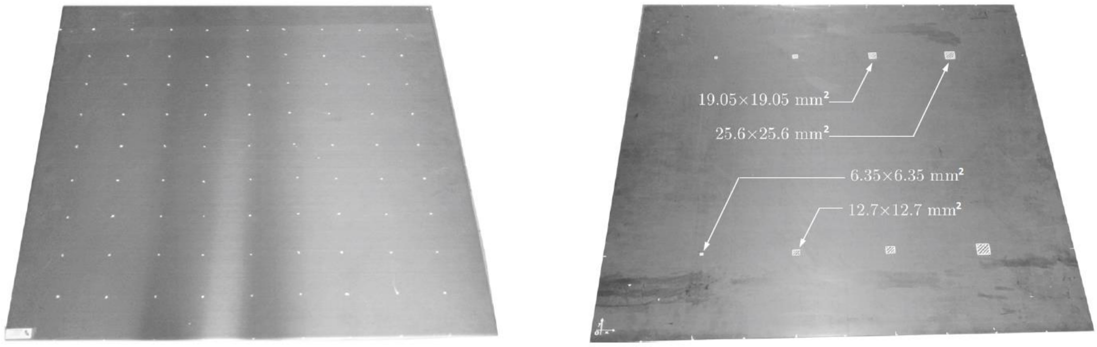

Step 1: Manufacturing and properties adjustment
Undamaged CFRP Plate Manufacturing
Numerical Modelling and Analysis by FEA
Experimental Modal Analysis
Numerical modal analysis
Numerical-experimental properties adjustment by GA
Delaminated CFRP Plate Manufacturing
Step 2: Inverse delamination detection problem
Real data of the damaged structure
Optimization Heuristic
FEA and numerical damage modal response
Initial Population
Genetic Operators
NO
Convergence?
YES
DELAMINATION IDENTIFIED

Figure 20: Flowchart of the delamination identification.

The damaged plate (inserted Teflon modeling delamination) was manufactured after establishing the mechanical properties. Modal experimental analysis was performed on the damaged structure to obtain the natural frequencies. Once the damaged natural frequencies were obtained, the GA was used as an optimization tool for minimizing the objective function that was constructed from the natural frequencies in both the undamaged structure and the delaminated structure.

As delamination alters structural rigidity, the natural frequencies of the delaminated plate are expected to be different from the unaltered plate. Therefore, the algorithm begins to apply random, yet GA controlled damage until the objective function is as low as possible. This occurs when the natural frequencies of both plates are equal. The algorithm proceeds to the convergence criterion, and once finalized, the damage is identified.

The experimental development was carried out in test structures so cordially provided by the Brazilian Aeronautics Company (EMBRAER®). Two plates were analyzed in the experiment. The first one was taken as the reference structure, this being a square plate of dimensions a = 1 m, with 16 layers, and with a stacking sequence [0/45/−45/90]4S and a thickness of t = 0.19 mm in each layer (Fig. 21a). A second plate was then taken, this having the same geometric characteristics as the first, except for the fact that it exhibited delamination damage. The plates were damaged by inserting four different sizes of Teflon in eight different locations (Fig. 21b).



(a) (b)

Figure 21: Test structures used in this work, cordially provided by EMBRAER®: (a) Plate without damage and (b) Plate with damage.

In order to obtain the results, it was necessary to carry out a quality experimental arrangement. The entire experimental apparatus is shown in Fig. 22, which was integrated with an Impact hammer, Laser Vibrometer, data acquisition (LabVIEW programming), and signal analysis.

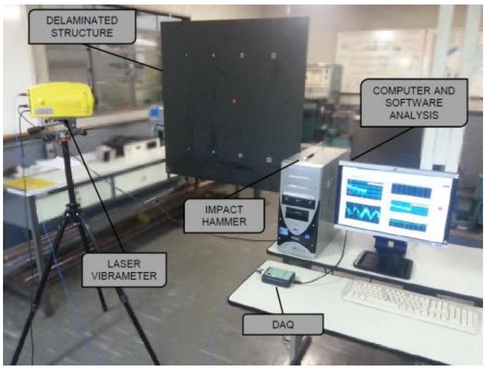


Figure 22: Schematic of the experimental setup.

The method of identifying damage used in this section was developed experimentally. Modal information was taken on the delaminated plate, and the locations of the inserted delamination were known. The problem of identifying damage itself was solved by the inverse problem using genetic algorithms. The approach of solving the actual problem was to minimize the objective function of Eq. (2).

$$J_{\exp} = \sqrt{\sum_{i=1}^{N}\left(1 - \frac{\omega_i^{GA}}{\omega_i^{real}}\right)^2} \tag{2}$$

where ω^{real} the natural frequency of the delaminated plate, ω^{GA} the frequencies that are calculated by the genetic algorithm in function of the design variables, and i the analyzed modes.

It is known that the presence of even small delaminations can lead to changes in the resonant frequencies, and so it is possible to obtain the locations of such damage by minimizing $J_{\exp}$. It is also known that variations in the natural frequencies serve as an excellent overall metric for the structural state. In other words, variations serve as reliable values that indicate whether or not a damage is present. Although single mode variation analysis does not yield a significant amount of information regarding the possible location of damage, by contrast, a set of modes can yield much more information as to the location of structural damage:

The objective function $J_{\exp}$ is composed of the first six nonzero modes. The rigid body modes of the structure with "free" boundary conditions were not taken into account, and the fundamental mode (first) presented an undesirable noise level, justifying the choice of modes i = 2, …, 5. Similar to the numerical identification problems addressed in this section, the same genetic operators were used (crossing of 60%, elitism of 1 individual, mutation of 2%, population of 10 times the number of variables, and maximum number of generations equal to 100). The damage search limits were defined by the maximum number of structural elements ($1 < N_e < 100$) and the total degree of severity of the plate ($0 \leq \alpha < 1$).

Regarding the numerical model in finite elements, and in relation to modeling damage, a level of severity α is associated with a damaged element. However, when multiple elements are considered, only a value of α can be considered. As a damaged plate shows extremely small damage, which in turn has higher performance for large structures, the application of this methodology seeks to detect at least the approximate location (neighborhood) of the largest failures (1-in. dimension square). The minor failures (6.35 × 6.35 mm²) correspond to an area of approximately 0.004% of the total area of the structure. Damage of 1 sq. in. (25.4 × 25.4 mm²) in turn represents approximately 0.0645% of the total area of the structure. After performing the modal test to acquire the modal information, the inputs used in the algorithms were the same as those of the objective function minimization inverse problem $J_{\exp}$ (Eq. 2).

Despite the previous knowledge as to the damage induced on the plate, the algorithm could neither identify the location of the damage, the severity of the damage, nor the extent of the damage present in the structure. In this regard, the solution of the inverse problem was addressed by considering the plate under different failure quantities, i.e., assuming 1, 2, and 8 failures. As such, the idea was to verify the method's capacity of in identifying the location of induced damage. Given the aforementioned, the optimization was performed and results were obtained considering different failure quantities. Figure 23 shows the final results considering one and two failures present on the plate. Considering the only one failure (Fig. 23a), the damage was obtained at $N_e = 9$. Given its proximity to element $N_e = 19$ and considering the damaged element, one can observe that there is an extremely narrow search area. The identified damage is located in the vicinity of the actual damage. Since each element has an area of 0.1 × 0.1 = 0.01 m², the area to be inspected is equivalent to 4 × 0.01 = 0.04 (four elements in the vicinity), that is, 4% of the total area of the plate.

These results in a 96% reduction of the inspected area, thus guaranteeing savings in terms of time, labor, and in costs associated with inspections. Additionally, Fig. 23b shows two failures and that the damage was found in elements $N_e = 10$ and $N_e = 21$. Assuming that the real failures are $N_e = 22$ and $N_e = 19$, the total inspection area is thus equivalent to 8% of the total area of the board, which again translates to large saving in inspection times, as well as other benefits.

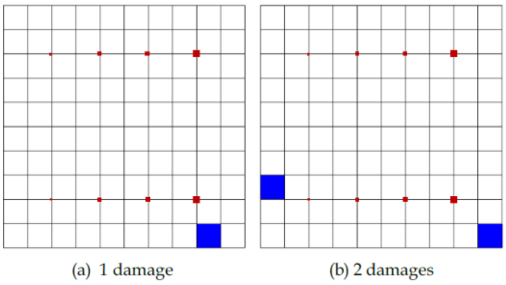


Figure 23: Result of damage identification in the delaminated plate considering 1 and 2 present damages (legend: 🟦 damage detected, 🟥 actual damages)

2.6 Numerical–experimental study for structural damage detection in CFRP plates using remote vibration measurement

Although composite structures are designed to sustain structural damage, reliable structural health monitoring (SHM) systems demand the improvement of structural design and maintenance performance while maintaining safety. Impact damage detection techniques for SHM ring are widely established; however, the associated costs are high because often the damaged area cannot be localized, and hence inspection of the whole component is required. Much research has been conducted to assess the success of non-destructive damage detection techniques, especially on new composite materials used in the aerospace industry (Mujica *et al.*, 2008).

The effect of defect or damage to the structural integrity of composite components is essential for understanding the criticality of the defect. The defects may be grouped into specific categories according to when they arise during the life of composite structure, their relative size, location or origin in the structure of the material. Some examples of damage in composites are shown in Fig. 24. The service components have defects that occur through mechanical action or contact with hostile environments, such as the impact site overload, local heating, chemical attacks, ultraviolet radiation, acoustic vibration, fatigue or inadequate action repair. The size of a defect has a significant influence on its criticality and may be present in isolation from structural features such as slots and bolted joints, or even a random accumulation resulting from the interaction between other defects (Talreja & Singh, 2012).

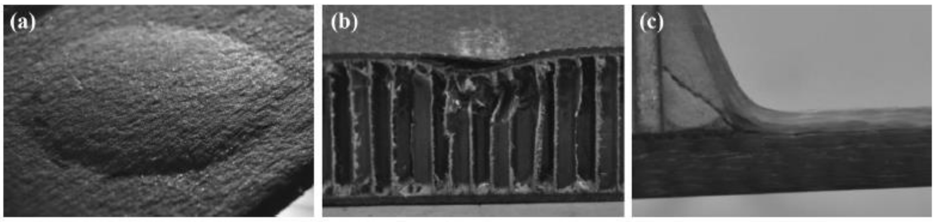


Figure 24: Some characteristic damage in composite materials: surface bubble (a), crush on a sandwich panel (b) and delamination (c) (Adapted from Talreja & Singh, 2012).

Composite structures have excellent performance, although this significantly deteriorates the presence of damage. Unfortunately damage due to impact events, for example, is difficult to visually detect, and, therefore, needs methods for non-destructive testing of these structures. According to (Friswell, 2008), although these materials present other failure modes such as cracks in the matrix, the fiber breakage or delamination damage these mechanisms produce changes in the vibrational response similar to a metal structure when there is a damage. Furthermore, a laminated composite carbon fiber/ epoxy plate was manufactured at NTC/UNIFEI. The carbon fiber is of the type AS4, unidirectional, GA45 and 5052 epoxy resin (Huntsman).

The plate was produced by the VARTM process—transfer molding vacuum-assisted resin symmetrically to 12 layers depending on the orientation of 0° and 90°, i.e., [0/90]3S. Each layer of the laminate in turn has 0.1824 mm, with the final structure being 2.1886 mm thick. The orientation of the fibers of this compound is structured symmetrically about the median plane of the laminate, which means that each layer above the median plane has a layer identical to the similar distance below the average plane. The laminate was made using a 30 cm² edge and subsequently damage was added to the medium, this being a circular hole of 8 mm radius in the central position of the plate (x = 0.15 cm and y = 0.15 cm), as shown in Fig. 25a.

The mechanical properties of the laminate, used in the design of the numerical model, are the result of a study on the estimation of material properties for model adjustment purposes. The numerical results in the following paragraphs will show the process performed for this purpose. For operational and experimental limitations, the laminated plate was simulated with free boundary conditions. The assay was then performed with the aid of a laser vibrometer (Brand: Ometron, Model: VQ-500-D) to avoid contact sensors such as accelerometers and used a portable system for acquisition of data. Figure 25b shows the experimental scheme used in this work. The detection method developed in this work, briefly, will take place in two steps.

The circular hole damage type is parameterized by their Cartesian x and y positions on the plate and the radius r thereof. It is important to note that this damage model is robust and can be interpreted as a hole by itself or by corrosion, erosion, tooth, etc., where there has been localized loss of material and stiffness. Other interpretations can be given to this model, but the goal of the adopted model is to intervene on structural physical characteristics (mass or stiffness) of the composite in question. The main goal of the optimization procedure is to adjust the fractional order α and the damaged element number N_{elem} to obtain the best properties of the damage identification algorithm.

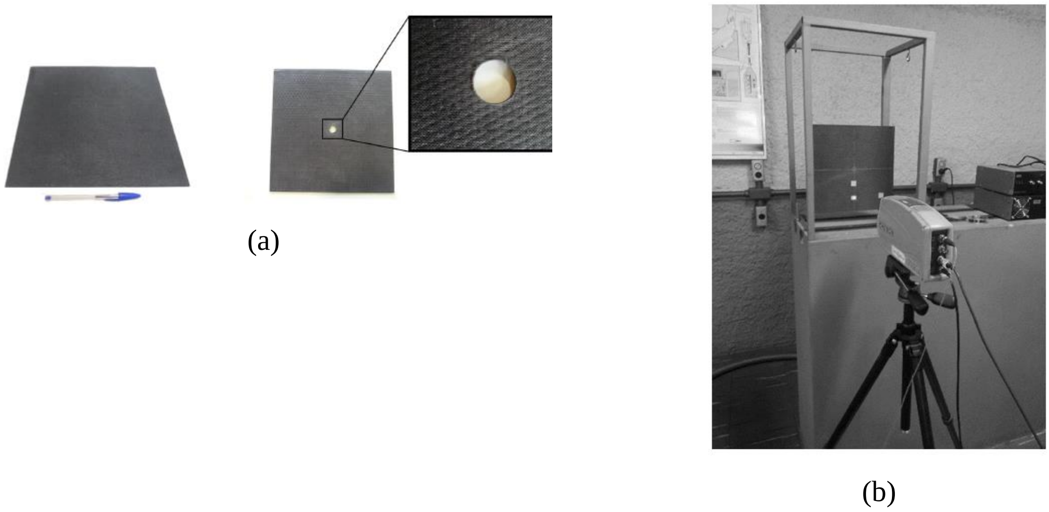
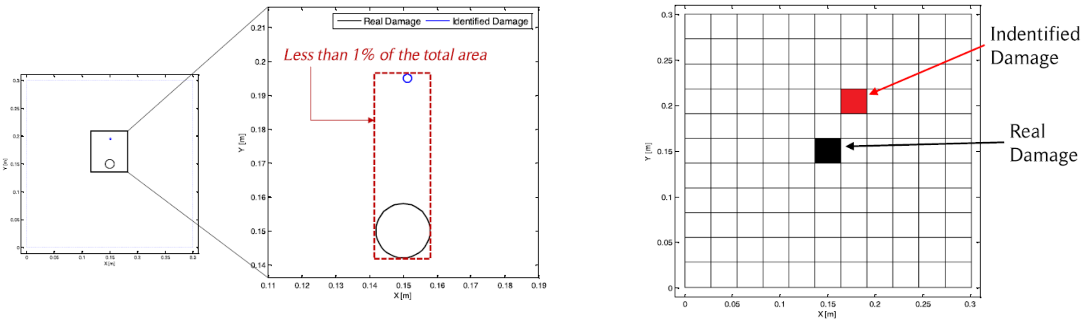


Figure 25: Experimental case: undamaged and damaged composite laminated plate (a) and Experimental setup of damage detection using contactless vibration measurement (b).

Figures 26a and 26b show the result of the search performed by the optimization algorithm for structural damage imposed on the laminate. It is observed that the method was not able to detect with great accuracy the presence of the hole. This mainly happens because the inserted structural damage is not sufficiently great as to cause a significant change in modal properties, in this case the natural frequencies of the laminate. However, the damage is detected at a region that is not so distant from the actual bore which leads for example, in the case of inspection of large structures (fuselage of aircraft, for example), a starting point (region with possible damage) facilitates the identification of the damage.

Following the idea that the plate has a total area of 900 cm², a rectangular imaginary area (red dashed line in figure) may be formed in an area covering both real and average damage that leads to obtaining an area possibly 8.66 cm² damaged, or the method promotes a reduction in an area unknown to be monitored to an area already known with possible damage, and less than the initial, promoting a reduction of about 99% of the region searched in the maintenance process, repair, identification, etc. It was also observed that the method effectively met along the axis of the damage location *x*, with an error of only 0.86%.

According to (Boller, 2000) and (Pawar & Ganguli, 2003) that there is no need to locate damage to within a few millimeters. The cost and efforts involved in predicting damage to a high-level accuracy can be prohibitive. In addition, because of measurement, model and signal processing inaccuracies, systems that claim to predict damage with great accuracy are likely to give false alarms. Hence, a better idea is to roughly locate damage in the structure and then use standard NDT methods such as acoustic emission and ultrasound for closer analysis of damaged area. Modal analysis methods are useful in roughly locating the damage.



(a)

(b)

Figure 26: Minimum area covering the obtained and induced damage: (a) producing a global reduction of area inspection and (b) damage detection showing a calculated damaged element number near real damaged element

2.7 A sunflower optimization (SFO) algorithm applied to damage identification on laminated composite plates

The performance and behavior of composite structures can be significantly affected by degradation caused by exposure to environmental conditions or damage caused by operating conditions such as impacts and structural loads. As a result, corrosion, delamination, cracking and other failures occur once the structure is in service. In the case of composite laminates, such damages are not always visible on the surface, which can lead to catastrophic structural failure. To ensure the performance and integrity of a structure of high structural responsibility, prior recognition of damage is crucial.

Traditionally, visual inspection accompanied by some alternative methods is employed to obtain general information on structural conditions. However, the inspection is limited and time consuming. The development of a comprehensive on-site health monitoring system that can inspect a relatively large area, instantly providing reliable, quantitative structural health data such as type of defect, location, and severity level minimizes and eventually eliminates drawbacks caused by stoppages for monitoring (Zhao *et al.*, 2007).

The advantage of using metaheuristic is because those methods are zero order methods, especially designated for nonlinear and multi-modal problems (Mitchell, 1998). In addiction, when working with optimization in the detection of damages, a functional with multiple local minimums appear (Gomes, Mendéz, Alexandrino, *et al.*, 2018; Gomes, Mendéz, *et al.*, 2019), that justify the use.

The cycle of a sunflower is always the same: every day, they awaken and accompany the sun like the needles of a clock. At night, they travel the opposite direction to wait again for their departure the next morning. (Yang, 2012) proposed a new algorithm based on the flower pollination process of flowering plants considering the biological process of reproduction. In this work, the authors take into account the peculiar behavior of sunflowers in the search for the best orientation towards the sun. The pollination considered here was take randomly along the minimal distance between the flower *i* and the flower *i* + 1. In the real world, each flower patch often release millions of pollen gametes. However, for simplicity, we also assume that each sunflower only produces one pollen gamete and reproduces individually. Another important nature-based optimization here is about the inverse square law radiation. The law says that the intensity of the radiation is inversely proportional to the square of the distance, i.e., the intensity (amount) of radiation reduces in proportion to the square of the increase in distance. If the distance doubles, the intensity reduces by a factor 4, triples, reduces to a factor 9, and so on. In our case, the less the distance from the plant to the sun, the greater the amount of radiation received, and it will tend to stabilize in these vicinity. On the other hand, the more distance a plant is from the sun, the lower the amount of heat received by it, so the same will be followed in this study which will take larger steps to get as close as possible to the global optimum (sun).

The damage detection problem can be formulated as an inverse problem solved via optimization methods. In this approach, it is desired to minimize an objective function that expresses the residues between the predicted and experimental responses. The design variables are the parameters of the parametric model assumed for the damage and once the optimal solution has been found it is assumed that the actual damage was identified, as illustrated by Fig. 27.

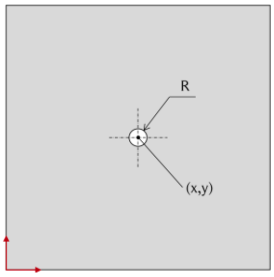


Figure 27: Damage modeling on the plate considering three variables in the inverse problem

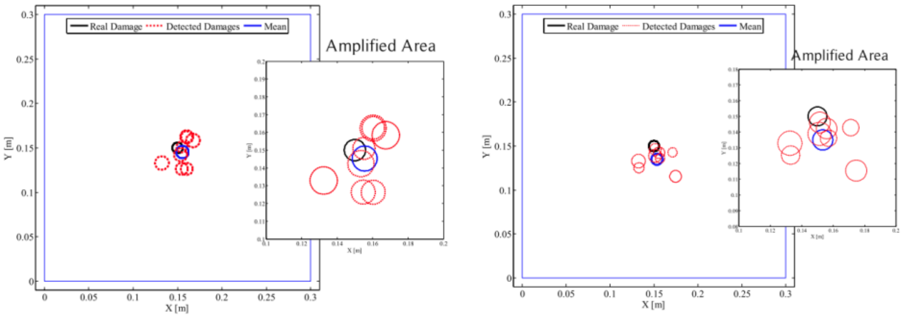
The presence of a hole (damage) affects the dynamic response of the laminate, then, the inverse problem is introduced to find optimal locations where the algorithms best fits the objective function. For this case, the results are obtained using fine mesh considering undamped shell element with eight nodes in each element.

To obtain the unknown parameters of the damage, such as location and size, a functional can be defined as the difference between the known or measured values of the natural frequencies and the calculated values obtained from the optimization algorithm. The minimization of this function, also called in this work as “solar radiation” allows the damage detection algorithm to find the unknown parameters of the damage. The pristine structural values are simulated through FEM. The objective function J based on the change of natural frequencies was defined in Eq. (3).

$$J(\vec{X}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left[1-\frac{\omega_{i}^{real}}{\omega_{i}^{SFO}\left(\vec{X}\right)}\right]^{2}} \tag{3}$$

Where ω^{real} are the natural frequencies obtained from the real damaged structure and ω^{SFO} are the natural frequencies obtained by the optimization procedure. When $J \sim 0$ means that the algorithm found a damage that exactly fits the real values. In addiction, X is the vector containing the project variables defined as the central position of the damage and its extension, i.e., $X = \{x, y, r\}$ and $n = 6$.

As can be seen in Fig. 28, the results of the damage search were satisfactory in the detection of circular holes. In both methods (GA and SFO), the results were very close to known (induced) damage. However, the proposed OS optimization method behaved equally with GA. This is because the proposed method is still in a beta version, programmed in some command lines in MATLAB®, and GA is already a method with a large contribution of several researchers and very well-elaborated programming in the software used in this work in commercial software.



(a) Genetic **(b)** Sunflower

Figure 28: Structural damage (holes) detection in composite plate using GA and SFO algorithm.

The results of the optimization showed that the new optimization method introduced was able to find points of good locations in standard test functions, which proved its good performance. It is intended to improve the version of the SFO algorithm for greater variability in the process of generating new individuals so that there is no stagnation in sub-regions of optimal location in relation to the application of the algorithm in a real non-trivial solution problem. The algorithm was still able to solve the damage identification and obtained a performance very similar to the widely known and used genetic algorithm.

2.8 Inverse structural damage identification problem in CFRP laminated plates using SFO algorithm based on strain fields

The detection of damages is a field of extreme importance in engineering, since through it corrective maintenance can be applied and in this way structural safety can be guaranteed. A prognosis of the structure can be made from the moment that a damage is correctly detected, thus being able to evaluate the integrity of the structure and determine its life time.

Non-destructive inspection/evaluation (NDI/E) techniques such as of X-rays, ultrasonic waves, eddy currents, shearography, and infrared thermography are often employed for the detection, localization, and quantification of flaws and damage in composite materials (Chandarana *et al.*, 2017). However, these methods depend on the skill and experience of an operator. The creation of an effective and autonomous method, approached in this study, enables the SHM methodology and thus avoids the identification of false positives or negatives.

A justification for using digital image correlation (DIC) is due to a non-contact optical technique to measure contour, deformation, vibration and strain on almost any material. The technique can be used with mechanical tests including tensile, torsion, bending and combined loading for both static and dynamic applications. The use of the digital image correlation technique is justified by the possibility of identify damages in composites, from the initial (matrix microcraks) to the final phase (fiber failure). DIC can reveal the elementary mechanisms in composites such as microcracks, debonding and delamination (Hild *et al.*, 2014). It was shown that damage laws can be identified with the help of DIC from mechanical tests imaged at different stages of loading. The complex damage type and failure mechanics theory present during the loading stage in a CFRP laminate are increased due to the presence of a stress concentration factors, causing a wide range of effects, such as stress or strain gradients fields (Caminero *et al.*, 2014). It is therefore more desirable when performing experimental testing on laminated composites structures to obtain extensive full-field strain data, rather than limited strain (by a limited number of sensors) or displacement measurements obtained from traditional electrical strain gauges or extensometers.

In the experiment, the specimen was inserted to the universal test machine as shown in Fig. 29, after which it was subjected to a tensile stress (below the yield). During the experiment, it was decided not to submit the test specimen to compressive stresses due to the possibility of buckling occurrence.

Two cases were evaluated: (*i*) a plate and (*ii*) beam model in the presence of damages. In order to capture the strains generated in the test specimen, a data acquisition system was used consisting of a camera with sensors and a computational apparatus. The resolution of the camera depends on the size of the measurement zone in question, while the maximum size of the measurement zone depends on the monochrome light emitter. In order for the data acquisition system to work correctly, it is necessary that the background color of the test piece is dark, if it is not, the test piece must be painted. As the specimen used in the experiment was already black, there was no need to paint it. Next, paint the test specimen in a spray pattern using white paint. In this experiment, a sponge was used to make this painting; however, there are other methods that can be employed, such as the spray paint itself and even a toothbrush or brush. After the experiment was carried out, all the data generated were collected and processed by Bluehill® software, which is also provided by INSTRON®, manufacturer of the universal testing machine and DIC data acquisition tools.

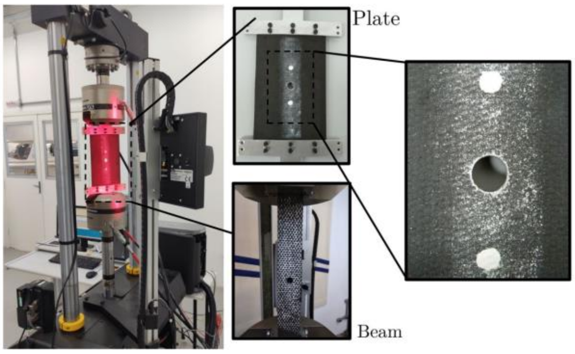


Figure 29: Experimental setup performed for analysis on damaged beam and plate models.

In Fig. 30, a flowchart is introduced to summarize all the methodology that was used in this work, from the initial problem to the solution of this problem, in it we can observe the existence of two main fronts, one focused on the computational solution of the problem and the other solution to the problem

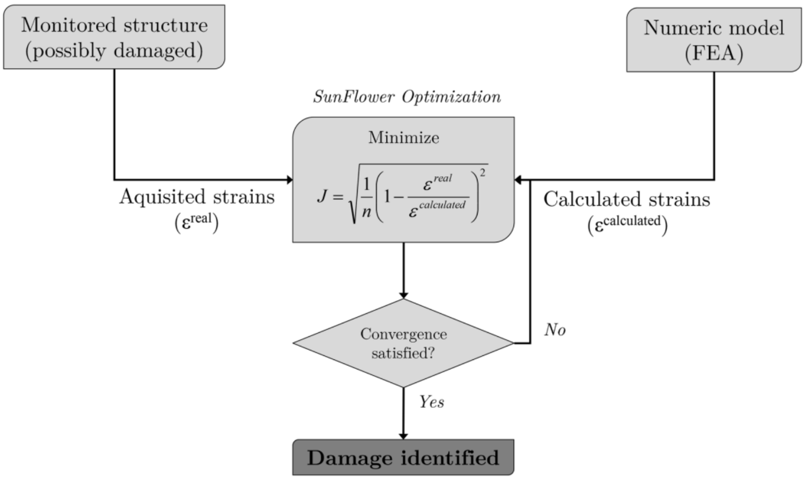
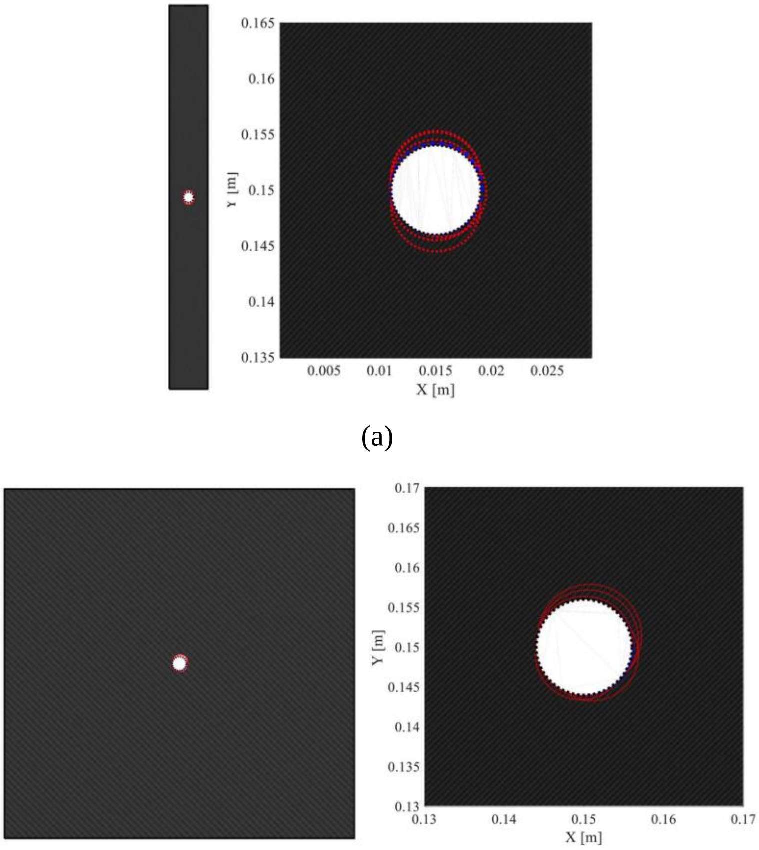

Monitored structure (possibly damaged)
Numeric model (FEA)
SunFlower Optimization
Minimize
$J = \sqrt{\frac{1}{n}\left(1-\frac{\varepsilon^{real}}{\varepsilon^{calculated}}\right)^2}$
Aquisited strains ($\varepsilon^{\text{real}}$)
Calculated strains ($\varepsilon^{\text{calculated}}$)
Convergence satisfied?
No
Yes
Damage identified

Figure 30: Flowchart of the methodology used in this work

As it can be seen in Fig. 31, the results presented were satisfactory, since all the parameters obtained converged to values very close to the actual damage parameters, and the damage was found practically concentric and or tangential when compared to the actual damage. Based on the results obtained, the robustness of the SFO is verified when applied in the detection of damage in both beams and plates.



(a)

(b)

Figure 31: Experimental damage identification results considering: (a) the composite beam and (b) the composite plate.

The present section has shown the potential of DIC for SHM of composite structures. It is revealed as an efficient methodology to identify possible damage in laminates with geometric discontinuities. It is still a challenge though to accurately identify internal damage such as delamination. With this, it can be said that this method has great potential to be applied in several engineering cases: firstly, due to the fact that the method produces relevant results. But mainly because of the practical advantages of the method, since it can be applied in an uninterrupted way by monitoring the structure continuously, it requires little time to carry out the inspection, the results are constant being dependent almost only on the adjustment and the quality of the used instruments and also has low cost with instrumentation and operation. In this way, when compared to conventional methods of damage identification, the method used in this work becomes more practical and efficient in most engineering applications.

2.9 Lichtenberg Optimization Algorithm Applied to Crack Tip Identification in Thin Pate-like Structures

Damage detection in mechanical structures is of great interest, as it is critical to ensure structural safety, prevent accidents and reduce maintenance costs. Structure monitoring allows detecting, locating and even predicting damage to mechanical structures (Gomes, Mendéz, Alexandrino, *et al.*, 2018). Cracks and other damages appear when the structure is in service. To ensure performance and integrity, there is a need for efficient detection, i.e. monitoring that provides fast and reliable results (Gomes, Cunha Jr., *et al.*, 2019b).

Mechanical systems under fatigue cycles can present cracks inside (internal or superficial) and normally these cracks appear in the region of maximum stress and in their direction.

There are many studies in the literature that deal with crack formation and its consequences on the health of a mechanical system. This issue is highlighted in the works of (K. Pereira *et al.*, 2018), (Floros *et al.*, 2019), (Zhu *et al.*, 2019) and (Xu *et al.*, 2018).

The SHM methodology is applied to identification and propagation direction of cracks in aluminum structures, with emphasis on aeronautical structures when using a 6061-aluminum alloy plate. This method allows remote and online monitoring of the SHM. The proposed methodology is based on the use of a new nature-inspired optimization algorithm and the inverse method (by finite element analysis) to detect the location and propagation direction of in-plane cracks. For the inverse problem solution, the metaheuristic Lichtenberg Algorithm (LA) was used. According to (J. L. J. Pereira, Francisco, Diniz, *et al.*, 2021), this powerful method consists of a hybrid algorithm that unites trajectory and population search strategies by exploiting the power of fractals to efficiently explore new solutions in the search space and increase the accuracy of those already found.

The proposed method is a robust one that requires only the information of a number of sensors pre-fixed in the structure. The induced damage could represent a real case, where only a few deformation points can be acquired. The results, based on strain fields, show a good crack detection, including the propagation direction, in plate-like structures using the LA.

The SHM methodology applied here consists of the use of two computational programs: *i)* finite element method (FEM) modeling (direct problem) and *ii)* optimization procedure using LA in order to detect the crack, including the propagation direction (inverse problem).

The plate used is modeled (using FEM) in solid material and has a square shape, with dimensions of 2×2 m² with 1mm thickness. The structure consists of 6061-T6 aluminum alloy, widely used in the aeronautical industry. The mesh has 289 nodes and 356 elements (solid element). As a boundary condition, the shape has all its edges fixed; however, its nodes have freedom of movement. Crack modeling is a force applied to one of the nodes (the yield strength < 255MPa).

Two cases are proposed in this model according (Suveges *et al.*, 2016): *i)* edge crack where the crack propagation occurs only in one end. In this case, there are four variables to determine: x and y positions of the tip and the force components F_x and F_y, *ii)* central crack where the crack propagations occurs in the two ends. Here, there are eight variables to determine: x_1, y_1, Fx_1, Fy_1, x_2, y_2, Fx_2 and Fy_2. The crack propagation direction is given by the direction of the resulting force found in the model. The Fig. 32 shows these two models of crack and the strain after load application.

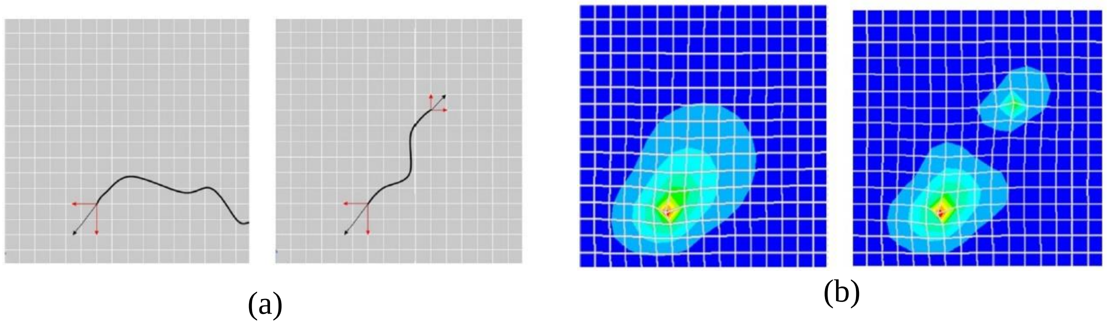


Figure 32: Two models of crack: (a) edge crack and central crack and (b) the strain after load application.

The inverse problem is modeled based on the principle that from the application of a force in the structure, there is a strain in the material that can be detected by properly positioned sensors. Thus, the behavior of the structure subjected to stress is changed. Therefore, monitoring these changes becomes an important strategy to preventively assess the integrity of the structure.

In this way, using an appropriated objective function related to strain of the plate will be possible to determine the position, magnitude and direction of propagating force acting on the crack tips.

The objective functions used are shown in Eq. (4) and Eq. (5):

$$J_k = \sum_{i=1}^{n} \sqrt{\left(\frac{1}{\varepsilon_{i,k}^{calc}} - \frac{1}{\varepsilon_{i,k}^{real}}\right)^2} \tag{4}$$

Where $\varepsilon_{i,k}^{calc}$ is the strain in the k direction on sensor i computed in each iteration by the optimization algorithm and $\varepsilon_{i,k}^{real}$ is the strain computed in the k direction on sensor i by the MEF in the direct method. In this case $k = x$ or y.

$$J = w_1 \times J_x + w_2 \times J_y \tag{5}$$

J_x and J_y are the objective functions related with the strains in x and y direction, w_1 and w_2 are weighting weights (both range from 0 to 1).

In Table 2 is presented the input variables in the edge crack and in the central crack as well as the values of the forces that will be applied to the plate.

Regarding the number of sensors, it was used 01 and 05 sensors in case of edge crack and central crack, as illustrated in Fig. 33. In addition, Fig. 34 summarizes the modeling of the crack identification presented in this study.

Table 2: Input variables for edge and center cracks.

edge crack model	
Parameter	Value
x	0.75 m
y	0.50 m
F_x	-300 N
F_y	-400 N

central crack model	
Parameter	Value
x_1	0.75 m
y_1	0.50 m
F_{x1}	-300 N
F_{y1}	-400 N
x_2	1.25 m
y_2	1.25 m
F_{x2}	200 N
F_{y2}	200 N

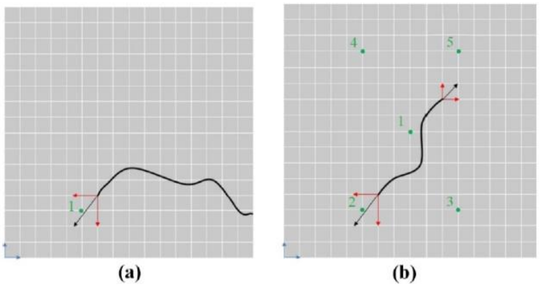


Figure 33: Sensors arrangement.

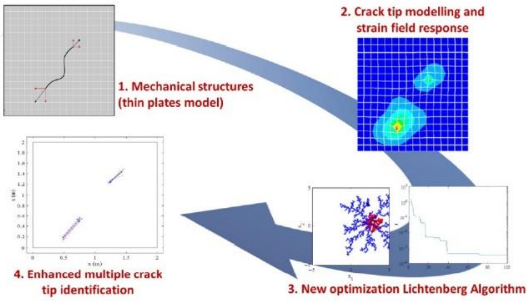

1. Mechanical structures (thin plates model)
2. Crack tip modelling and strain field response
3. New optimization Lichtenberg Algorithm
4. Enhanced multiple crack tip identification

Figure 34: General methodology.

Five simulations were performed for each case (edge and central crack). Table 3 shows the values found by the LA for edge crack detection and Table 4 and Table 5 for the central crack.

The magnitude and direction for crack tips detection using LA for both edge and central crack are illustrated in Fig. 35 and Fig. 36.

Table 3: Values found by the LA for edge crack.

Target	x (m) 0.75	y (m) 0.50	F_x (N) -300	F_y (N) -400	Simulation time
Run #1	0.7737	0.5425	-264.9938	-355.5506	36h 10min
Run #2	0.7845	0.5102	-331.6465	-435.6982	35h 55min
Run #3	0.7323	0.5368	-283.4581	-440.6524	36h 25min
Run #4	0.7866	0.5521	-315.1874	-422.3685	36h 15min
Run #5	0.7658	0.4856	-271.6414	-392.7823	36h 00min
Mean	0.7686	0.5255	-293.3854	-409.4104	-
SD	0.0131	0.0180	4.6772	6.6542	-

Table 4: Values found by the LA for central crack (first end).

Target	x_1 (m) 0.75	y_1 (m) 0.50	F_{x1} (N) -300	F_{y1} (N) -400
Run #1	0.7535	0.5595	-288.5471	-377.4165
Run #2	0.7133	0.5370	-261.3569	-412.6859
Run #3	0.7437	0.4826	-298.6658	-421.5687
Run #4	0.7866	0.5426	-314.2587	-382.1258
Run #5	0.7322	0.5144	-291.3541	-356.3684
Mean	0.7459	0.5272	-290.8365	-390.0331
SD	0.0029	0.0192	6.4795	7.0477

Table 5: Values found by the LA for central crack (second end).

Target	x_2 (m) 1.25	y_2 (m) 1.25	F_{x2} (N) 200	F_{y2} (N) 200
Run #1	1.2357	1.2225	194.5687	221.3644
Run #2	1.2755	1.837	185.2356	94.75145
Run #3	1.2369	1.2578	236.9874	242.3265
Run #4	1.2174	1.2241	278.6984	301.3652
Run #5	1.2512	1.2315	171.3674	257.1459
Mean	1.2433	1.2439	213.3715	223.3907
SD	0.0047	0.0043	9.4551	16.5397

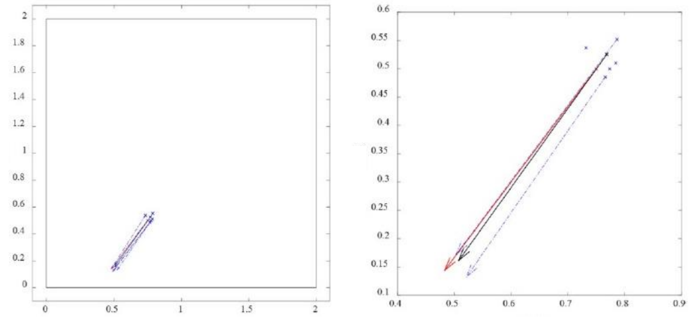


Figure 35: Detection considering only one crack tip: (a) real view and (b) zoomed view.

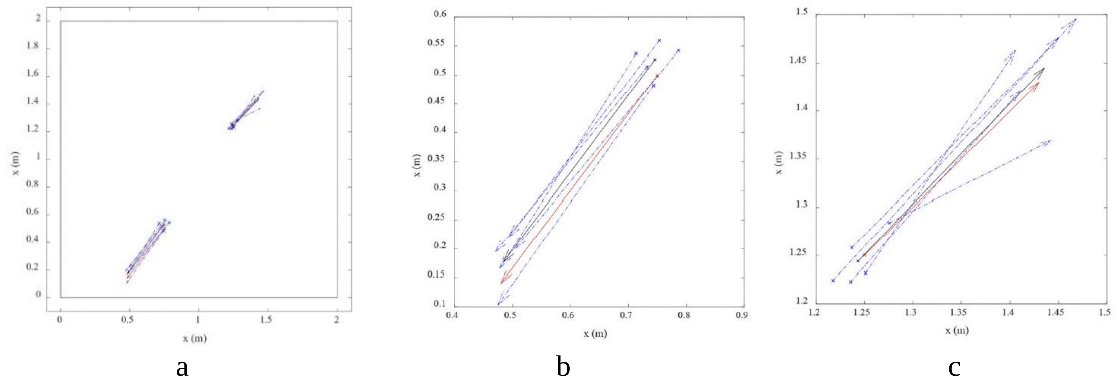


Figure 36: Detection considering two crack tips. (a) real view, (b) and (c) zoomed view.

2.10 A Powerful Lichtenberg Optimization Algorithm: A Damage Identification Case Study

Composite materials have been widely used over the years in the aerospace industry and other engineering applications where structural weight is one of the main reasons for its use. This is due to its excellent advantages, such as high strength and remarkable stiffness related to its specific mass, besides the high capacity to withstand fatigue and corrosion (Kaw, 2005).

However, in service, they may have failure mechanisms, such as fiber breakage, cracks in the matrix, or delamination. Static overload, impact, fatigue, design errors, and overheating are some of the causes of these failures. Delamination is considered the greatest ‘‘weakness’’ of laminated composite materials, as it can spread throughout the laminate of a composite structure and lead to catastrophic failures if not detected (Chakraborty, 2005).

Structural Health Monitoring (SHM) inspections that explore vibration measures are methods based on the principle that degradation due to damage in a structure changes the vibration parameters such as natural frequencies, mode shapes, and structural damping. Then, by analyzing the output vibration parameters of a system, it is possible to identify the presence of damage using techniques such as inverse modeling and computational intelligence (Gomes, Cunha Jr., *et al.*, 2019b).

This study is dedicated to identifying structural damages in composite laminated structures with focus in the detection of delamination using a new metaheuristic based on the Lichtenberg figures phenomena called Lichtenberg Algorithm (LA).

According to (Garg, 1988), delamination is an important form of failure in composite materials, which may not be visible on the structural surface and can affect strength and stiffness (local loss of stiffness) of the material. Figure 37 shows a case of delamination in Carbon Fiber Reinforced Polymer (CFRP).

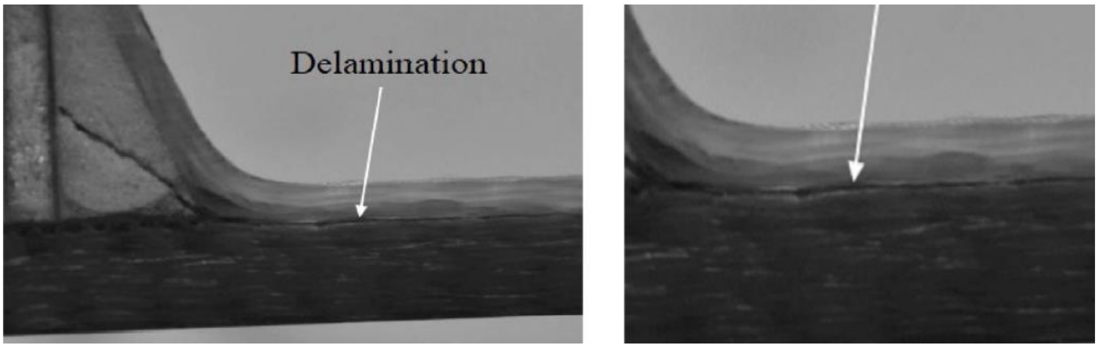


Figure 37: Composite laminated structure with delamination. Adapted from Heslehurst (2014).

The Lichtenberg Algorithm (LA) was first introduced by (J. L. J. Pereira, Francisco, Diniz, *et al.*, 2021) and has been used in engineering problems such as detection and characterization of crack propagation in thin plates of composite material (J. L. J. Pereira, Chuman, *et al.*, 2021).

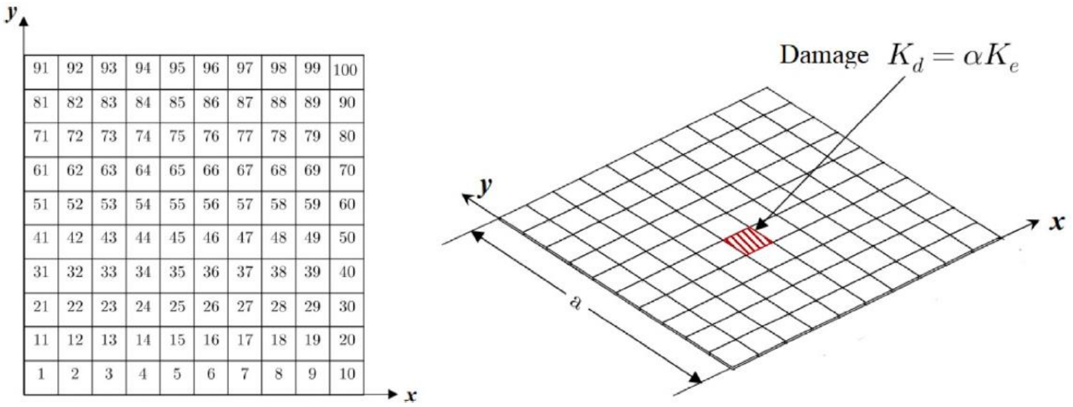
Here, it will be assesses the potential of LA applied in a damage detection using incomplete and noisy modal data in SHM systems.

The SHM methodology consists of using two computational routines: *i)* finite element method (FEM) modeling (direct problem) and *ii)* optimization procedure using LA in order to detect the delamination.

The modeled geometry is a square plate with uniform composite thickness laminated with or without delamination in a linear elastic regime. The plate was discretized according to 10 × 10 elements through a uniform and mapped mesh. It was used shell elements with 8 nodes and 6-DOF by node. The boundary condition is free (FFFF - Free- Free- Free- Free) on the four boundaries sides of the plate.

The square plate has 30 cm of side and is a symmetrical laminate of composite material consisting of 12 layers with different orientations arranged in the form [0/90]3S. Damping is not considered in numerical modeling.

Stiffness reduction, due to delamination, is represented by a non-dimensional parameter that changes a local stiffness but conserves the mass of the system (Santos *et al.*, 2000). The parameter of local stiffness reduction in percentage terms is given by β = (1-α)×100. In Fig. 38 outlines the finite element model and indicates a possible damage position.



91	92	93	94	95	96	97	98	99	100
81	82	83	84	85	86	87	88	89	90
71	72	73	74	75	76	77	78	79	80
61	62	63	64	65	66	67	68	69	70
51	52	53	54	55	56	57	58	59	60
41	42	43	44	45	46	47	48	49	50
31	32	33	34	35	36	37	38	39	40
21	22	23	24	25	26	27	28	29	30
11	12	13	14	15	16	17	18	19	20
1	2	3	4	5	6	7	8	9	10

Figure 38: FEM model of the discretized system with damage position.

For delamination detection, it was used the objective function proposed by (Gomes *et al.*, 2019) represented in Eq. (6).

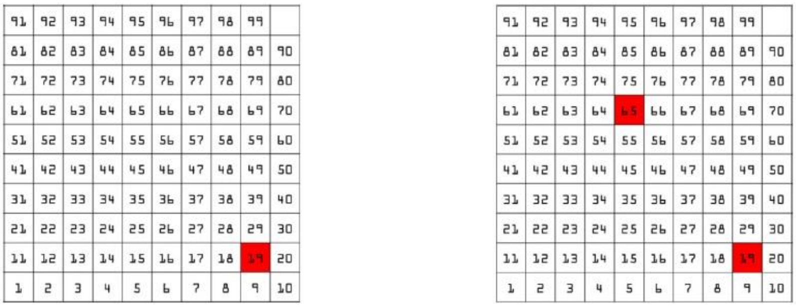
$$J_{\Phi} = \sum_{i=1}^{n} \sqrt{\left(1 - \frac{\Phi_{i,s}^{calculated}}{\Phi_{i,s}^{real}}\right)^2} \tag{6}$$

Where $\Phi_{i,s}^{calculated}$ are the nodal displacements obtained by the FEM numerically analyzing each random point of each iteration generated by the optimization algorithm referring to the mode shape *i*. $\Phi_{i,s}^{real}$ are the known displacements of the structure that has structural damage.

Here, three case will be analyzed: *i)* single delamination and *ii)* multiple delamination and *iii)* single delamination with noise in the measures.

i) Single Delamination: damage in a single element, element (N_e) 19, Fig. 39 (a), with a damage rate (α) of 0.2, 0.5, and 0.9.

Table 6 shows the results for the damage located in element number 19 with different severity rates:



(a) Single delamination (b) Multiple delamination

Figure 39: Induced delamination location: (a) Single delamination and (b) Multiple delamination.

Table 6: Results for the damage located in element number 19 with different severity rates.

Target			Results		
N_e	α		N_e^*	α^*	J_{min}
19	0.2	Mean SD	19 0	0.2000 0.0002	0.0001 0.2372
19	0.5	Mean SD	19 0	0.5000 0.0001	0.0000 0.0017
19	0.9	Mean SD	19 0	0.9000 0.0041	0.0000 0.0005

ii) Multiple Delamination: the system has two elements with local stiffness reduction. It was used the same stiffness reduction rate for both elements. The delamination is in the elements 19 and 65, Fig. 39 (b), with a damage rate of 0.2.

iii) Single Delamination: damage in a single element, element (N_e) 19 with a damage rate (α) of 0.2 and noise in the measures.

Table 7 shows the results for the damage located in elements 19 and 65:

The noisy signals have intensities of 1, 5 and 10% and Table 8 shows the results for this important applications:

Table 7: Results for the damage located in elements 19 and 65.

Target				Results			
N_{e1}	N_{e2}	α		N_{e1}^*	N_{e2}^*	α^*	J_{min}
19	65	0.2	Mean SD	19 0	65 0.6325	0.2732 0.0745	0.3890 0.3238

Table 8: Results for the damage located in elements 19 and 65.

Noise Level (%)		N_e^*	α^*	J_{min}
1	Mean SD	19 0	0.1967 0.0117	0.2284 0.0413
5	Mean SD	19 0	0.2278 0.0417	0.8391 0.1189
10	Mean SD	19 0	0.1776 0.0402	1.6197 0.1949

2.11 Multiobjective Optimization Using a Controlled Random Search Algorithm (CRSA)

A direct multiobjective optimization methodology is presented in (Sousa *et al.*, 2008), based on CRSA version proposed by (Manzanares Filho *et al.*, 2005), in which two objectives are treated using as aggregating approach the well-known weighting method technique. To represent the airfoil geometry is used a Bezier curve parameterization scheme, based on two higher degree curves that define extrados and intrados, in the same guide lines to those one implemented by (Pehlivanoglu & Hacioglu, 2006). The evaluation of aerodynamic coefficients used as objective functions are performed by airfoil flow analysis code XFoil, developed by (Drela & Giles, 1987), based in a panel method with viscous effects incorporated.

In treatment of multiobjective optimization problems, the CRSA is one option of population-set based algorithm considered duo to low computational cost associated to each interaction and facility in implementation, when compared with others evolutionary optimization algorithms like GA and DE as showed by (Ali *et al.*, 1997) and (Ali & Törn, 2004).

Initially proposed by (Price, 1977), and improved and modified by (Ali & Törn, 2004) and (Manzanares Filho *et al.*, 2005), the CRSA have been shown as a good alternative optimization algorithm to apply in aerodynamic shape optimization design problems.

All CRSA versions start with a random population generation with P individuals, this number of individuals is kept during optimization process. Each individual has N design variables, defined within upper limit U and lower limit L, thus creating a design space. The version used in this work makes selective use of quadratic interpolations in the trial point search, considering function objective variability around the best point of current population. In the way to execute quadratic interpolations are selected three points in the current population, best point l, namely r_1, and others two randomly choose, r_2 and r_3, respectively. Objective functions values are assumed as $f_1 = f(r_1)$, $f_2 = f(r_2)$ and $f_3 = f(r_3)$. Varying design variables $j = 1, \ldots, N$ are constructed quadratic interpolations for each one of sets r_{1j} , r_{2j} and r_{3j}, where trial point design variables p_j are defined as minimum of the parabola, as described by Eq. (7) and illustrated in Fig. 40.

$$p_j = \frac{1}{2}\frac{\left(r_{2j}^{\ 2} - r_{3j}^{\ 2}\right) f_1 + \left(r_{3j}^{\ 2} - r_{1j}^{\ 2}\right) f_2 + \left(r_{1j}^{\ 2} - r_{2j}^{\ 2}\right) f_3}{\left(r_{2j} - r_{3j}\right) f_1 + \left(r_{3j} - r_{1j}\right) f_2 + \left(r_{1j} - r_{2j}\right) f_3}, \qquad j = 1,\ldots,N \tag{7}$$

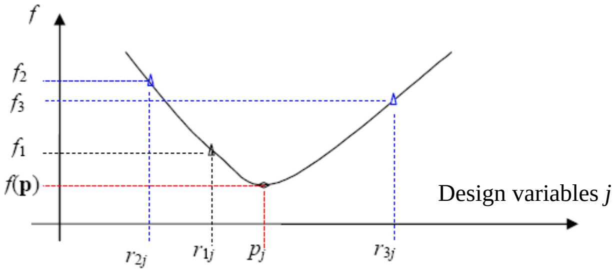


Figure 40: Graphical representation of quadratic interpolation.

To control the use of quadratic interpolations, in the way to avoid it to become ill-conditioned or present trial point design variables as maximum of the parabola, are used a mean objective function value, f_g , and a local variability measure around the best point, α, which are calculated as follow in Eq. (8) and Eq. (9).

$$f_g = \frac{1}{2}(f_2 + f_3) \tag{8}$$

$$\alpha = \frac{f_g - f_l}{f_h - f_l} \tag{9}$$

These equations are used when quadratic interpolation is well-conditioned and best point is not contained between others two points. In this case are defined a set of centroidal design variables g_j, Eq. (10), and through them trial point design variables p_j, Eq. (11), are defined by variability based reflection around best point.

$$g_j = \frac{(f_2 - f_1)r_{2j} + (f_3 - f_1)r_{3j}}{(f_2 - f_1) + (f_3 - f_1)} \tag{10}$$

$$p_j = (2-\alpha)r_{1j} - (1-\alpha)g_j \tag{11}$$

If quadratic interpolation is well-conditioned and best point is contained between others two points, the trial point design variables are defined normally according to quadratic interpolation. Finally, if quadratic interpolation is considered ill-conditioned, trial point design variables are defined randomly within design space *S*.

Constraints can be introduced in all CRSA versions by means of a penalty scheme, which more detailed in (Sousa *et al.*, 2008). This choice is problem dependent a too small factor can accelerate the algorithm, but may not be effective in promoting constraint satisfaction. On the other hand, a too large factor may lead to a loss of information about the original objective function and a hampering of the algorithm convergence.

To treat multiobjectives within CRSA was implemented the weighting method technique, which converts several objectives into a single one as described in Eq. (12). Each of the k objective functions has a w_i weights associated, and the sum of the weights must equal the unit.

$$Min.\quad \sum_{i=1}^{k} w_i f_i \tag{12}$$

Varying weights are determined Pareto optimum set, and consequently, is constructed the Pareto front of the multiobjective optimization problem. However, this technique is not able to represent concave parts of Pareto front, according to (Coello Coello *et al.*, 2007). The main advantages of this technique are ease in implementation and low computational cost.

Figure 41 presents a comparison of objective functions between numerical results obtained through multiobjective optimization and airfoil base NACA 65_1-412.

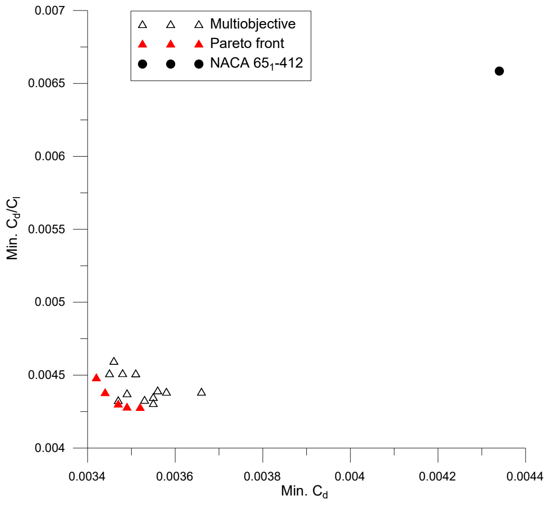


Figure 41: Comparison between multiobjective optimization results with NACA 651-412 airfoil.

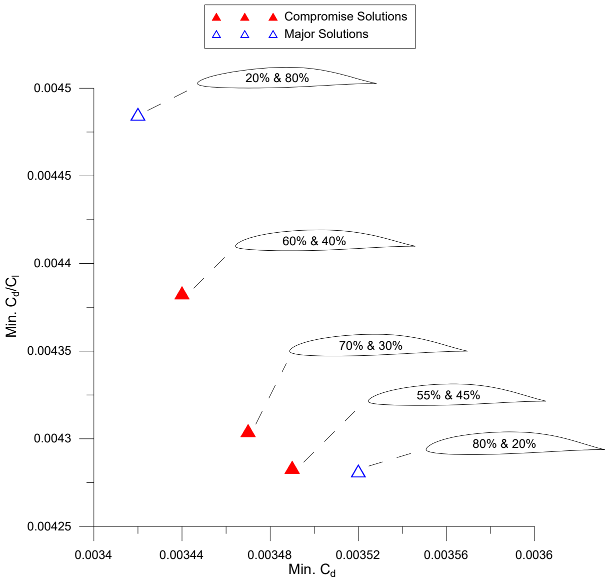


Figure 42: Detail of Pareto front.

Observing Fig. 41, can be noted that results obtained were sensitively improved in relation to airfoil base NACA 651-412. In addition, the spread of results was caused duo to C_l/C_d relations behavior, which depend directly of the airfoil shapes modified when weights are varied, in the way to minimize the objective function formed by two real objectives. Thus, in this optimization example, the weights do not reflect proportionally the relative importance of objective functions.

Observing the shapes of airfoils that form Pareto front in Fig. 42, where first percentage value correspond to C_l/C_d relation and second to C_d minimization, can be noted from major solutions that, as hopped, reduction on maximum camber associated with it position beyond 50% of chord, favor drag minimization. In the same way that increasing on maximum thickness and camber, positioned close to 50% of chord, favor C_l/C_d maximization. Compromise solutions are given by combinations of these modifications on airfoil geometric parameters. In addition, must be noted too, between airfoils that form Pareto front, modifications on maximum thickness and camber values were smaller than promoted on respective maximum positions.

2.12 Topological Sensitivity Analysis Applied to Composite Structural Design

A new approach for Topological Sensitivity Analysis is presented in (Sousa *et al.*, 2018) applied to composite structural design. Topological Sensitivity Analysis allows for the assessment of the sensitivity of both the objective function and the constraints when the problem definition domain changes shape and/or topology. According to (Novotny *et al.*, 2003), Topological Sensitivity Analysis results in a scalar function, called the Topological Derivative, which provides the sensitivity of the objective function for each point in the problem definition domain when a change is created at this point. The calculation is based on a mathematical proof that establishes a relationship between the Analysis of Sensitivity to the Change of Form and the Topological Derivative, thus leading to a modified, simpler, and general formulation.

The original formulation of the calculation of the Topological Derivative which was developed in the works of (Eschenauer *et al.*, 1994), (Schumacher, 1995) and (Céa *et al.*, 2000), in a way, limits the field of application of the Topological Sensitivity Analysis, due to the mathematical difficulty of obtaining the Topological Derivative, and also due to the fact that several simplifying hypotheses were adopted, mainly with regards to the boundary conditions in the border of the holes. Engineering optimization applications were explored in the works of (C. E. L. Pereira & Bittencourt, 2008), (Bojczuk & Mróz, 2009) and (Bojczuk & Mróz, 2012).

In the Topological Derivative original formulation, the original domain of the problem, denoted by Ω, after the creation of a small hole B_ε of radius ε becomes Ω_ε, just as the initial boundary Γ becomes Γ_ε, after domain perturbation. Establishing the performance function in the domains Ω and Ω_ε, $\psi(\Omega)$ and $\psi(\Omega_\varepsilon)$ are obtained, as graphically exemplified in Fig. 43. Thus, the Topological Derivative is defined as shown in Eq. (13).

$$D_T^*\left(\hat{x}\right)=\lim_{\varepsilon\to 0}\frac{\psi\left(\Omega_\varepsilon\right)-\psi\left(\Omega\right)}{f\left(\varepsilon\right)} \tag{13}$$

where $f(\varepsilon)$ is a regularizing function, such that $f(\varepsilon) \to 0$ when $\varepsilon \to 0$, and so that $0 \leq |D_T^*\left(\hat{x}\right)| \leq \infty$. The choice of function will depend on the problem being analyzed. According to Cordeiro (2007), the regularizing function used in elastic problems corresponding to the hole area B_ε, that is, the difference between the values of the performance function for the initial topology and the disturbed topology is weighted by the size of the perturbation in the hole area created in the domain. The great difficulty in working with Eq. (13) lies in the fact that when a hole is created in the domain, it is no longer possible to establish an inverse mapping between Ω and Ω_ε, leading to mathematical difficulties in calculating the Topological Derivative.

The modification in the calculation of the Topological Derivative, proposed by (Novotny *et al.*, 2003) in her doctoral thesis, would start from a domain with a pre-existing perturbation B_ε, Ω_ε being the initial domain with contour Γ_ε. When a small variation δ_ε is caused in the perturbation B_ε, it is denoted by $B_\varepsilon + \delta_\varepsilon$, and a new domain $\Omega_\varepsilon + \delta_\varepsilon$ and new contour $\Gamma_\varepsilon + \delta_\varepsilon$ are defined, as graphically exemplified by Fig. 44. In this way, the Topological Derivative can be redefined as shown in Eq. (14).

$$D_T\left(\hat{x}\right) = \lim_{\substack{\varepsilon\to 0\\ \delta\varepsilon\to 0}} \frac{\psi\left(\Omega_{\varepsilon+\delta\varepsilon}\right)-\psi\left(\Omega_\varepsilon\right)}{f\left(\varepsilon+\delta\varepsilon\right)-f\left(\varepsilon\right)} \tag{14}$$

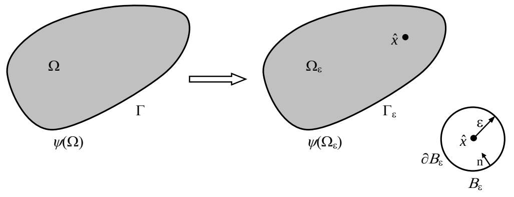


Figure 43: Concept of Topological Derivative in its original form.

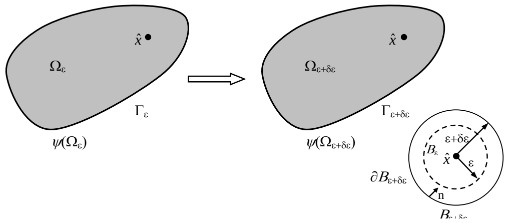


Figure 44: Concept of Topological Derivative in its modified form.

The innovation brought about by the definition of the Topological Derivative given by Eq. (14) is that it is now possible to establish the inverse mapping between the domains Ω_ε and $\Omega_\varepsilon + \delta_\varepsilon$, and also to allow for the use of the concepts of Analysis of Sensitivity to Shape Change to obtain the Topological Derivative. The understanding that expanding a hole of radius ε, when $\varepsilon \to 0$, would be nothing more than creating it, leads to the thought that it would be possible to use the Topological Derivative to map regions of the domain where it would be necessary to insert material instead of removing it, thus creating optimal topology.

The motivation for developing this work by (Sousa *et al.*, 2018) started from the idea that, as predicted in the theory described by (Novotny *et al.*, 2003), the optimal topology could also be obtained by progressively inserting material in the domain, since this procedure had not yet been explored for this purpose. The insertion of material into the domain is a widely used procedure for the eventual corrections and smoothing over of the topology if any criteria have been exceeded, as illustrated by Fig. 45.

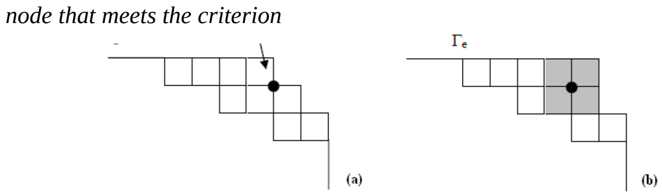


Figure 45: Addition of material in the domain; (a) region of the contour to be corrected; (b) corrected region. (Cordeiro, 2007).

Because the topology of a structural component made from laminated composite material is generated by layer superposition, it would be intuitive to start from an initial layer and then to add the other layers until the design goals, constraints, and performance criteria specified in the design have been met. Therefore, the premise of this methodology is to start from an undersized structure and to add material in regions of the domain defined by the Topological Derivative as being more sensitive, until the optimum structure topology has been obtained.

By mapping the Topological Derivative, it is possible to determine the format of the new layer to be created, obviously while respecting the criteria related to the manufacturing process, such as minimum size of a layer to be added and the format of the layer to be added.

Total Potential Energy was used as the performance function. This choice was due to the fact that the Total Potential Energy gives information as to the sum of the effects of the Deformation Energy, and the Potential of the External Forces, which leads to a more precise identification of the points in the domain that need to be added and shows how the load applied to the domain can influence the change in the disturbance. The smoothing function is given by the area of the layer being created. Thus, the expression for the calculation of the Topological Derivative for elasticity problems involving laminated composite material is given by Eq. (15).

$$D_T\left(\hat{x}\right) = \lim_{\substack{\varepsilon\to 0\\ \delta\varepsilon\to 0}} \frac{\pi_p\left(\Omega_{\varepsilon+\delta\varepsilon}\right) - \pi_p\left(\Omega_\varepsilon\right)}{f\left(\varepsilon+\delta\varepsilon\right) - f\left(\varepsilon\right)} \tag{15}$$

where π_p is the Total Potential Energy defined as the sum of the deformation energy (U) and the potential of the external forces (Ω), i.e., $\pi_p = \mathrm{U} + \Omega$ and Ω_ε is the initial domain. That is, the first layer, $\Omega_\varepsilon + \delta_\varepsilon$ is the domain disturbed by the addition of a new layer, ε the area of the initial layer and $\varepsilon + \delta_\varepsilon$ the area of the new layer added to the starting area.

However, in the design of structures made of laminated composite, the orientation of each layer, and the stacking sequence in which they are arranged, strongly influence the stiffness and resistance characteristics of the final topology. The orientation of each layer and the stacking sequence of the laminated follow the manufacturing constraints and were defined by ACO algorithm for each new layer insertion.

In the example proposed for the application test, the determination of the optimum number of layers, and the optimal stacking sequence were sought by minimizing the variability of the Topological Derivative, and consequently, by homogenizing its values throughout the domain. The minimizing of the variability in the topological derivative has a physical response of stiffness increasing. For the thickness of the layers, the constant value of 0.25 mm was adopted so as to consider laminate manufacturing issues.

The domain is defined as a square plate, $L_x = L_y = 0.2$ m, simply supported on the four edges, subjected to a transverse load $P = 50$ kPa, evenly distributed over the entire surface, as shown in Fig. 46. The maximum allowable displacement at the center of the plate (y_{max}), and the value of the maximum failure criterion (H_{max}), for any layer of the laminate were defined as feasibility criteria. It can be ensured that the laminate configuration is a point belonging to the viable design region when these criteria are met. The formulation of the optimization problem is given by Eq. (16).

$$\text{minimize}\left[\text{variability } D_T\left(\hat{x}\right)\right]$$
$$\text{subject to:}\begin{cases} H_{\max} \leq 0.80 \\ y_{\max} \leq 1.0\,mm \\ \text{Manufacturing} \end{cases} \tag{16}$$

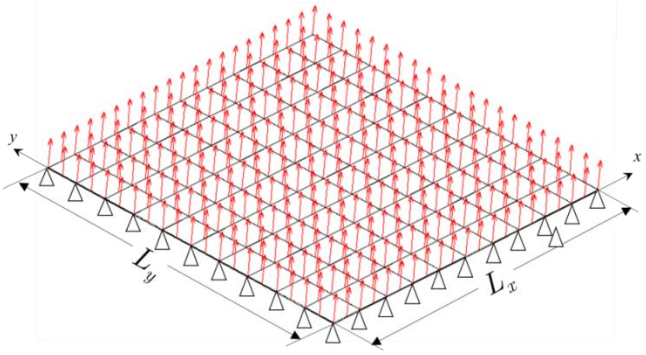
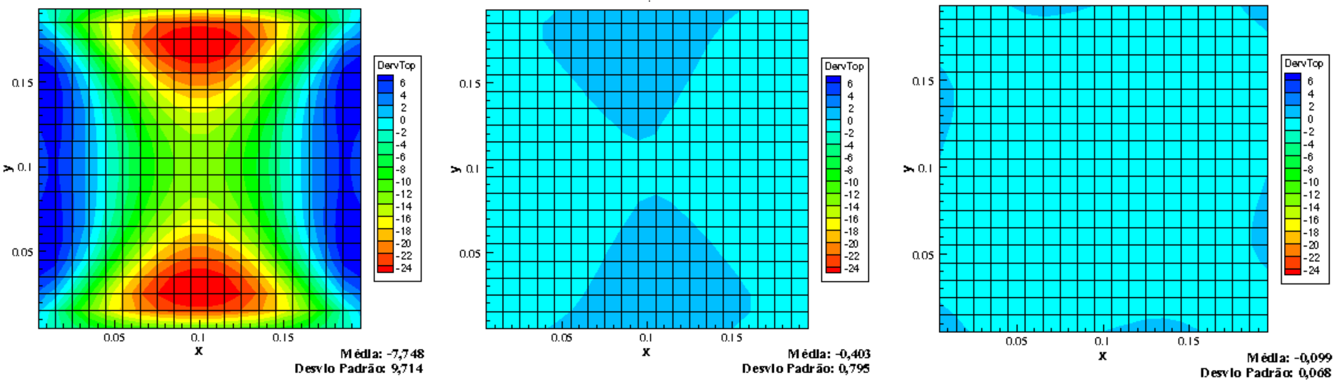


Figure 46: Square plate simply supported on the edges, with uniformly distributed loading along the entire surface, represented in a simplified way by the vectors in red.

In Fig. 47 the Topological Derivative mappings are shown for some iterations for the first execution of the algorithm.



(a) Iteration 1 (b) Iteration 4 (c) Iteration 15

Figure 47: Topological Derivative Mapping of some iterations for the first execution of the algorithm.

In order to verify the behavior of the mean value of the Total Potential Energy, $(\pi p)_{avg}$, with respect to the increased thickness of the laminate, h, throughout the iterations, until the viable configuration of the laminate is obtained for each execution of the algorithm, a graph of $(\pi_p)_{avg}$ x h is shown in Fig. 48, in which all the viable configurations of obtained laminates are grouped.

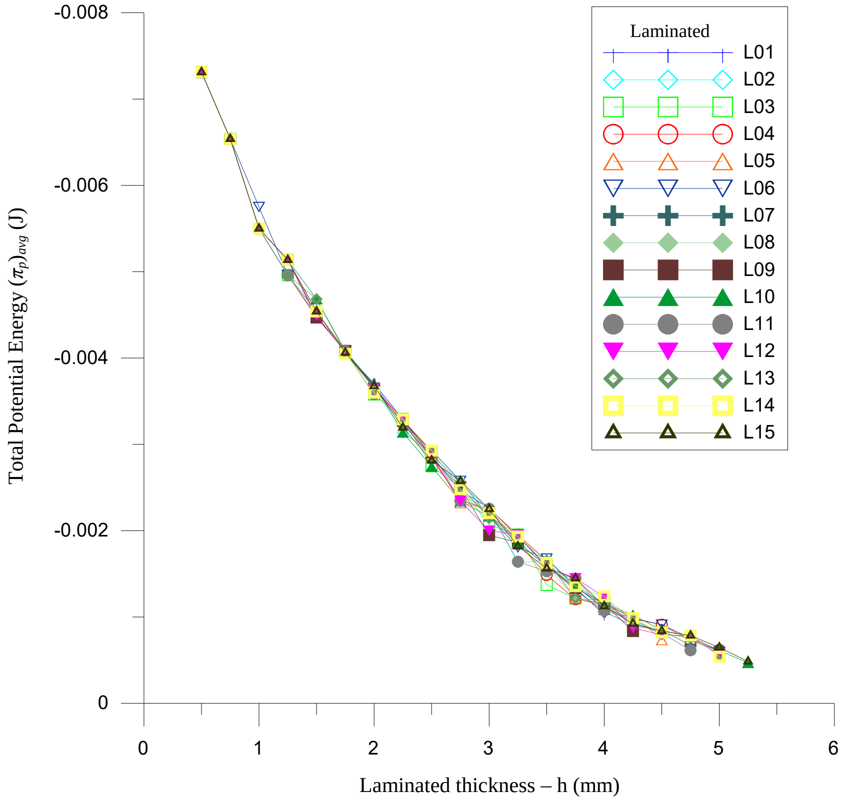


Figure 48: Behavior of *$(\pi_p)_{avg}$* x *h* over the iterations for all laminates obtained.

By analyzing the curves, it can be seen that the mean (π_p) values are constantly decreasing as the laminate thickness increases due to the addition of new layers. Note that there is a greater dispersion of $(\pi_p)_{avg}$ between the thickness range of 2 and 4.5 mm. Precisely in this range, the largest variations occurred in the stacking sequences between the laminate configurations. But even so, the downward trend of $(\pi_p)_{avg}$ remains. Thus, it is demonstrated that the Deformation Energy of the laminate is more sensitive to the variation of thickness than to variations in the stacking sequence.

Based on the analysis of the results obtained from the application example, it can be concluded that the Topological Sensitivity Analysis methodology may be applied in the design of laminated composite structures, showing that the calculation of the Topological Derivative satisfactorily indicates the region of the domain where a new layer should be added.

3 Concluding Remarks

This chapter presented previous work performed in the Research Group in Computational Mechanics (GEMEC) at UNIFEI. Details of the different journal articles and conference papers prepared by the authors along the last 15 years were presented, covering different methods and aspects in optimization and identification techniques for inverse methods in damage detection and localization. Current work in the research group includes a follow-up of this work with new and modern optimization and identification techniques and approaches.

References

Alexandrino, P. da S. L., Gomes, G. F., & Cunha Jr., S. S. (2019). A robust optimization for damage detection using multiobjective genetic algorithm, neural network and fuzzy decision making. *Inverse Problems in Science and Engineering*, *28*(1), 26. https://doi.org/https://doi.org/10.1080/17415977.2019.1583225

Ali, M. M., & Törn, A. (2004). Population set-based global optimization algorithms: some modifications and numerical studies. *Computers & Operations Research*, *31*(10), 1703–1725. https://doi.org/https://doi.org/10.1016/S0305-0548(03)00116-3

Ali, M. M., Törn, A., & Viitanen, S. (1997). A Numerical Comparison of Some Modified Controlled Random Search Algorithms. *Journal of Global Optimization*, *11*(1), 8. https://doi.org/https://doi.org/10.1023/A:1008236920512

Alves, V. N. (2012). *Study of new strategies for identifying structural damage to from vibrational data (Estudo de novas estratégias para identificação de danos estruturais a partir de dados vibracionais)* [Federal University of Ouro Preto]. http://www.repositorio.ufop.br/jspui/handle/123456789/3285

Bayissa, W. L., & Haritos, N. (2007). Structural damage identification in plates using spectral strain energy analysis. *Journal of Sound and Vibration*, *307*(1–2), 226–249. https://doi.org/https://doi.org/10.1016/j.jsv.2007.06.062

Bojczuk, D., & Mróz, Z. (2009). Topological sensitivity derivative and finite topology modifications: application to optimization of plates in bending. *Structural and Multidisplinary Optimization*, *39*(1), 1–15. https://doi.org/https://doi.org/10.1007/s00158-008-0333-5

Bojczuk, D., & Mróz, Z. (2012). Topological sensitivity derivative with respect to area, shape and orientation of an elliptic hole in a plate. *Structural and Multidisplinary Optimization*, *45*(1), 153–169. https://doi.org/https://doi.org/10.1007/s00158-011-0710-3

Boller, C. (2000). Next generation Structural Health Monitoring and its integration into aircraft design. *International Journal of Systems Science*, *31*(11), 1333–1349. https://doi.org/https://doi.org/10.1080/00207720050197730

Caminero, M. Á., Lopez-Pedrosa, M., Pinna, C., & Soutis, C. (2014). Damage Assessment of Composite Structures Using Digital Image Correlation. *Applied Composite Materials*, *21*, 91–106. https://doi.org/https://doi.org/10.1007/s10443-013-9352-5

Céa, J., Garreau, S., Guillaume, P., & Masmoudi, M. (2000). The shape and topological optimizations connection. *Computer Methods in Applied Mechanics and Engineering*, *188*(4), 713–726. https://doi.org/https://doi.org/10.1016/S0045-7825(99)00357-6

Chakraborty, D. K. (2005). Artificial neural network based delamination prediction in laminated composites. *Materials & Design*, *26*(1), 1–7. https://doi.org/https://doi.org/10.1016/j.matdes.2004.04.008

Chandarana, N., Sanchez, D. M., Soutis, C., & Gresil, M. (2017). Early Damage Detection in Composites during Fabrication and Mechanical Testing. *Materials*, *10*(7), 685. https://doi.org/https://doi.org/10.3390/ma10070685

CHONG, E. K. P., & ZAK, S. H. (2004). *An Introduction to Optimization*.

Coello Coello, C. A., Lamont, G. B., & Van Veldhuizen, D. A. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems* (2nd ed.). Springer.

Cordeiro, M. F. A. (2007). *Structural Optimization Technique Using the Topology Sensitivity Analysis*. Universidade Federal do Rio de Janeiro.

Deb, K., Agrawal, S., Amrit, P., & Meyarivan, T. (2000). A Fast Elitist Non-dominated Sorting Genetic Algorithm for Multi-objective Optimization: NSGA-II. In *Lecture Notes in Computer Science* (pp. 849–858).

Drela, M., & Giles, M. B. (1987). Viscous-Inviscid Analysis of Transonic and Low Reynolds Number Airfoils. *AIAA Journal*, *25*(10), 9. https://doi.org/https://doi.org/10.2514/3.9789

Engelhardt, M., Stavroulakis, G. E., & Antes, H. (2006). Crack and flaw identification in elastodynamics using Kalman filter techniques. *Comput Mech*, *37*(1), 249–265. https://doi.org/https://doi.org/10.1007/s00466-005-0709-y

Eschenauer, H. A., Kobelev, V. V., & Schumacher, A. (1994). Bubble method for topology and shape optimization of structures. *Structural Optimization*, *8*(1), 42–51. https://doi.org/https://doi.org/10.1007/BF01742933

Floros, D., Ekberg, A., & Larsson, F. (2019). Evaluation of crack growth direction criteria on mixed-mode fatigue crack growth experiments. *International Journal of Fatigue*, *129*. https://doi.org/https://doi.org/10.1016/j.ijfatigue.2019.04.013

Friswell, M. I. (2008). Damage Identification using Inverse Methods. In A. Morassi & F. Vestroni (Eds.), *Dynamic Methods for Damage Detection in Structures* (CISM Cours, p. 230). SpringerWienNewYork.

Garg, A. C. (1988). Delamination—a damage mode in composite structures. *Engineering Fracture Mechanics*, *29*(5), 557–584. https://doi.org/https://doi.org/10.1016/0013-7944(88)90181-6

Gomes, G. F., Almeida, F. A. de, Alexandrino, P. da S. L., Cunha Jr., S. S., Sousa, B. S. de, & Ancelotti Jr., A. C. (2019). A multiobjective sensor placement optimization for SHM systems considering Fisher information matrix and mode shape interpolation. *Engineering with Computers*, *35*, 519–535. https://doi.org/https://doi.org/10.1007/s00366-018-0613-7

Gomes, G. F., Almeida, F. A. de, Ancelotti Jr., A. C., & Cunha Jr., S. S. (2020). Inverse structural damage identification problem in CFRP laminated plates using SFO algorithm based on strain fields. *Engineering with Computers*, *37*(1), 21. https://doi.org/https://doi.org/10.1007/s00366-020-01027-6

Gomes, G. F., Almeida, F. A. de, Cunha Jr., S. S., & Ancelotti Jr., A. C. (2018). An estimate of the location of multiple delaminations on aeronautical CFRP plates using modal data inverse problem. *The International Journal of Advanced Manufacturing Technology*, *99*(1), 20. https://doi.org/https://doi.org/10.1007/s00170-018-2502-z

Gomes, G. F., Cunha Jr., S. S., & Ancelotti Jr., A. C. (2019a). A sunflower optimization (SFO) algorithm applied to damage identification on laminated composite plates. *Engineering with Computers*, *35*(1), 8. https://doi.org/https://doi.org/10.1007/s00366-018-0620-8

Gomes, G. F., Cunha Jr., S. S., & Ancelotti Jr., A. C. (2019b). A sunflower optimization (SFO) algorithm applied to damage identification on laminated composite plates. *Engineering with Computers*, *35*(1), 8. https://doi.org/https://doi.org/10.1007/s00366-018-0620-8

Gomes, G. F., Mendéz, Y. A. D., Alexandrino, P. da S. L., Cunha Jr., S. S., & Ancelotti Jr., A. C. (2018). The use of intelligent computational tools for damage detection and identification with an emphasis on composites – A review. *Composite Structures*, *196*, 44–54. https://doi.org/https://doi.org/10.1016/j.compstruct.2018.05.002

Gomes, G. F., Mendéz, Y. A. D., Alexandrino, P. da S. L., Cunha Jr., S. S., & Ancelotti Jr., A. C. (2019). A Review of Vibration Based Inverse Methods for Damage Detection and Identification in Mechanical Structures Using Optimization Algorithms and ANN. *Archives of Computational Methods in Engineering*, *26*, 883–897. https://doi.org/https://doi.org/10.1007/s11831-018-9273-4

Gomes, G. F., Mendéz, Y. A. D., Cunha Jr., S. S., & Ancelotti Jr., A. C. (2018). A numerical–experimental study for structural damage detection in CFRP plates using remote vibration measurements. *Journal of Civil Structural Health Monitoring*, *8*(1), 15. https://doi.org/https://doi.org/10.1007/s13349-017-0254-3

Hild, F., Périé, J.-N., & Roux, S. (2014). Evaluating Damage with Digital Image Correlation: C. Applications to Composite Materials. In G. Voyiadjis (Ed.), *Handbook of Damage Mechanics* (pp. 1–21). Springer. https://doi.org/https://doi.org/10.1007/978-1-4614-8968-9_26-1

Kaw, A. K. (2005). *Mechanics of Composite Materials* (2nd ed.). CRC Press. https://doi.org/https://doi.org/10.1201/9781420058291

Lopes, P. da S., Cunha Jr., S. S., & Jorge, A. B. (2007). Damage Detection Using Global Optimization and Parameter Identification Techniques. *19th International Congress of Mechanical Engineering*.

Lopes, P. da S., Jorge, A. B., & Cunha Jr., S. S. (2010). Detection of holes in a plate using global optimization and parameter identification techniques. *Inverse Problems in Science and Engineering*, *18*(4), 24. https://doi.org/https://doi.org/10.1080/17415971003624306

Lopes, P. da S., Jorge, A. B., & Cunha Jr., S. S. (2008). Detection of holes in a plate using global optimization and parameter identification techniques. *EngOpt 2008 - International Conference on Engineering Optimization*, 6.

Manzanares Filho, N., Moino, C. A. A., & Jorge, A. B. (2005). An Improved Controlled Random Search Algorithm for Inverse Airfoil Cascade Design. *6th World Congresses of Structural and Multidisciplinary Optimization*, 10.

Mitchell, M. (1998). *An Introduction to Genetic Algorithms*. The MIT Press.

Montalvão, D., Maia, N. M. M., & Ribeiro, A. M. R. (2006). A review of vibration-based structural health monitoring with special emphasis on composite materials. *The Shock and Vibration Digest*, *38*(4), 295–324. https://doi.org/10.1177/0583102406065898

Mujica, L. E., Vehi, J., Staszewski, W., & Worden, K. (2008). Impact damage detection in aircraft composites using knowledge-based reasoning. *Structural Health Monitoring*, *7*(3), 215–230. https://doi.org/https://doi.org/10.1177/1475921708090560

Novotny, A. A., Feijóo, R. A., Taroco, E., & Padra, C. (2003). Topological Sensitivity Analysis. *Computer Methods in Applied Mechanics and Engineering*, *192*(7–8), 803–829. https://doi.org/https://doi.org/10.1016/S0045-7825(02)00599-6

Pawar, P. M., & Ganguli, R. (2003). Genetic fuzzy system for damage detection in beams and helicopter rotor blades. *Computer Methods in Applied Mechanics and Engineering*, *192*(16), 2031–2057. https://doi.org/10.1016/S0045-7825(03)00237-8

Pehlivanoglu, Y. V., & Hacioglu, A. (2006). Inverse Design of 2-D Airfoil Via Vibrational Genetic Algorithm. *JOURNAL OF AERONAUTICS AND SPACE TECHNOLOGIES*, *2*(4), 8. http://jast.hezarfen.msu.edu.tr/index.php/JAST/article/view/142

Pereira, C. E. L., & Bittencourt, M. L. (2008). Topological sensitivity analysis in large deformation problems. *Structural and Multidisplinary Optimization*, *37*(1), 149–163. https://doi.org/https://doi.org/10.1007/s00158-007-0223-2

Pereira, J. L. J., Chuman, M., Cunha Jr., S. S., & Gomes, G. F. (2021). Lichtenberg optimization algorithm applied to crack tip identification in thin plate-like structures. *Engineering Computations*, *38*(1), 15. https://doi.org/https://doi.org/10.1108/EC-12-2019-0564

Pereira, J. L. J., Francisco, M. B., Cunha Jr., S. S., & Gomes, G. F. (2021). A powerful Lichtenberg Optimization Algorithm: A damage identification case study. *Engineering Applications of Artificial Intelligence*, *97*(1), 17. https://doi.org/https://doi.org/10.1016/j.engappai.2020.104055

Pereira, J. L. J., Francisco, M. B., Diniz, C. A., Oliver, G. A., Cunha Jr., S. S., & Gomes, G. F. (2021). Lichtenberg algorithm: A novel hybrid physics-based meta-heuristic for global optimization. *Expert Systems With Applications*, *170*(1), 12. https://doi.org/https://doi.org/10.1016/j.eswa.2020.114522

Pereira, K., Bhatti, N., & Wahab, M. A. (2018). Prediction of fretting fatigue crack initiation location and direction using cohesive zone model. *Tribology International*, *127*, 245–254. https://doi.org/https://doi.org/10.1016/j.triboint.2018.05.038

Price, W. L. (1977). A controlled random search procedure for global optimisation. *The Computer Journal*, *20*(4), 367–370. https://doi.org/https://doi.org/10.1093/comjnl/20.4.367

Rao, H. S., Ghorpade, V. G., & Mukherjee, A. (2006). A genetic algorithm based back propagation network for simulation of stress–strain response of ceramic-matrix-composites. *Computers & Structures*, *84*(5–6), 330–339. https://doi.org/https://doi.org/10.1016/j.compstruc.2005.09.022

Santos, J. V. A. dos, Soares, C. M. M., Soares, C. A. M., & Pina, H. L. G. (2000). Development of a numerical model for the damage identification on composite plate structures. *Composite Structures*, *48*(1–3), 59–65. https://doi.org/https://doi.org/10.1016/S0263-8223(99)00073-2

Schumacher, A. (1995). *Topologieoptimierung von bauteilstrukturen unter verwendung von lopchpositionierungkrieterien*. Universitat-Gesamthochschule-Siegen.

Sousa, B. S. de, Gomes, G. F., Jorge, A. B., Cunha Jr., S. S., & Ancelotti Jr., A. C. (2018). A modified topological sensitivity analysis extended to the design of composite multidirectional laminates structures. *Composite Structures*, *200*(1), 18. https://doi.org/https://doi.org/10.1016/j.compstruct.2018.05.145

Sousa, B. S. de, Manzanares Filho, N., & Jorge, A. B. (2008). Multiobjective Laminar-flow Airfoil Shape Optimization Using a Controlled Random Search Algorithm. *EngOpt 2008 - International Conference on Engineering Optimization*, 8.

Stavroulakis, G. E., & Antes, H. (1998). Flaw identification in elastomechanics: BEM simulation with local and genetic optimization. *Structural Optimization*, *16*(1), 162–175. https://doi.org/https://doi.org/10.1007/BF01202827

Stepinski, T., Uhl, T., & Staszewski, W. (2013). *Advanced Structural Damage Detection: From Theory to Engineering Applications*. John Wiley & Sons.

Suveges, J. M. C., Gomes, G. F., Souza, A. M., Cunha Jr., S. S., & Lopes, P. da S. (2016). Comparative Study of Optimization Techniques Applied to Structural Damage Detection Problems (Estudo Comparativo de Técnicas de Otimização Aplicadas em Problemas de Detecção de Danos Estruturais). *9th National Congress of Mechanical Engineering (Congresso Nacional de Engenharia Mecânica) - CONEM 2016*, 11.

Talreja, R., & Singh, C. V. (2012). *Damage and failure of composite materials*. Cambridge University Press. https://doi.org/https://doi.org/10.1017/CBO9781139016063

Wenk, L., & Bockenheimer, C. (2014). Structural Health Monitoring: A real-time on-board ‘stethoscope for ConditionBased Maintenance. *Airbus Technical Magazine*, *54*(1), 22–29.

Xu, Y., Bao, Y., Chen, J., Zuo, W., & Li, H. (2018). Surface fatigue crack identification in steel box girder of bridges by a deep fusion convolutional neural network based on consumer-grade camera images. *Structural Health Monitoring*, *18*(3), 653–674. https://doi.org/https://doi.org/10.1177/1475921718764873

Yang, X.-S. (2012). Flower Pollination Algorithm for Global Optimization. In J. Durand-Lose & N. Jonoska (Eds.), *Unconventional Computation and Natural Computation. UCNC 2012. Lecture Notes in Computer Science* (pp. 240–249). Springer. https://doi.org/https://doi.org/10.1007/978-3-642-32894-7_27

Zhao, X., Gao, H., Zhang, G., Ayhan, B., Yan, F., Kwan, C., & Rose, J. L. (2007). Active health monitoring of an aircraft wing with embedded piezoelectric sensor/actuator network: I. Defect detection, localization and growth monitoring. *Smart Materials and Structures*, *16*(4), 1208. https://doi.org/10.1088/0964-1726/16/4/032

Zhu, L.-F., Ke, L.-L., Zhu, X.-Q., Xiang, Y., & Wang, Y.-S. (2019). Crack identification of functionally graded beams using continuous wavelet transform. *Composite Structures*, *210*, 473–485. https://doi.org/https://doi.org/10.1016/j.compstruct.2018.11.042

Chapter 3: An overview of Linear and Non-linear Programming methods for Structural Optimization

Chapter details

Chapter DOI:

https://doi.org/10.4322/978-65-86503-71-5.c03

Chapter suggested citation / reference style:

Choze, Sergio B., et al. (2022). "An overview of Linear and Non-linear Programming methods for Structural Optimization". *In* Jorge, Ariosto B., et al. (Eds.) *Model-Based and Signal-Based Inverse Methods*, Vol. I, UnB, Brasilia, DF, Brazil, pp. 65–106. Book series in Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity.

P.S.: DOI may be included at the end of citation, for completeness.

Book details

Book: Model-based and Signal-Based Inverse Methods

Edited by: Jorge, Ariosto B., Anflor, Carla T. M., Gomes, Guilherme F., & Carneiro, Sergio H. S.

Volume I of Book Series in:

Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity

Published by: UnB City: Brasilia, DF, Brazil Year: 2022

DOI: https://doi.org/10.4322/978-65-86503-71-5

Overview of Linear and Non-linear Programming Methods for Structural Optimization

Sergio B. Choze[1*] Rogerio R. Santos[2]
Ariosto B. Jorge[3] Guilherme F. Gomes[4]

[1]Consultant. E-mail: sergio.butkewitsch@gmail.com
[2]Division of Mechanical Engineering, Technological Institute of Aeronautics, Brazil. E-mail: rsantos9@gmail.com
[3]Post-Graduate Program - Integrity of Engineering Materials, University of Brasilia, Brazil. E-mail: ariosto.b.jorge@gmail.com
[4]Mechanical Engineering Institute, Federal University of Itajuba - UNIFEI, Brazil. E-mail: guilhermefergom@unifei.edu.br

*Corresponding author

Abstract

This chapter describes the theoretical and practical aspects of deterministic optimization methods. The introductory section describes the basic equations commonly found in numerical optimization and lists some standard approaches in structural optimization. Next, the concepts of Multi-criteria Optimization and Multidisciplinary Design Optimization are described. The rest of the text is devoted to clarifying specific variations of the methods in the context of Linear and Non-linear Programming. Unconstrained and constrained cases are handled. Algorithms and source code fragments are presented to enrich the discussion about the methodologies. Some practical considerations for applied optimization are described throughout the text.

1 Introduction

As a decision-making problem crafted with mathematical formality, an optimization procedure aims at finding the extreme point that minimize or maximize the value of a function (Vanderplaats [1998], Butenko and Pardalos [2014]) as defined in Equation (1)

$$\min, \max[F(\{X\})] \tag{1}$$

expressing the objective function F, dependent upon an n-dimensional vector of input variables $\{X\}$, which contains the parameters to be determined/modified in order to achieve the desired optimization/decision goals. As such, these parameters are usually called decision or design variables, and relate directly to the resources available to supply a certain demand Haldar and Mahadevan [1999]. In the particular context of structural optimization, these decision variables will determine the load bearing capabilities (supply) to resist the operating conditions (demand).

In a realistic decision-making context, however, changes applied in the content of $\{X\}$ will likely affect not only the objective F, but also produce side effects in other quantities. These other quantities are most plausibly not allowed to vary freely and thus represent constraints that need to be factored into the determination of $\{X\}$ in the pursuit of improving F. Moreover, it is customary to classify these constraints into 2 categories, defined in Equations (2) and (3) as inequality and equality constraints, respectively:

$$G_j(\{X\}) \leq 0 \tag{2}$$

$$H_k(\{X\}) = 0 \tag{3}$$

Inequality constraints represent more flexible considerations corresponding to a threshold, as for example when the stress in a structural member should not exceed an allowable, but a margin of safety ($\leq$) would not be, barring any other considerations, harmful. On the other hand, if a structural failure mode is deliberately designed to undertake some pre-determined course or sequence, then the load bearing capabilities of its weakest link (and all others thereafter, for that matter) must fulfill a much more stringent target, giving rise to equality constraints. Please also note that the nominal values for the failure limits in these 2 scenarios may differ significantly, and it is standard practice to get them normalized by moving their values into the left-hand side of inequality (2) and equality (3) relationships, both then uniformly defaulted to 0, which becomes the reference for all equality and inequality constraints alike (they are considered violated once they surpass zero).

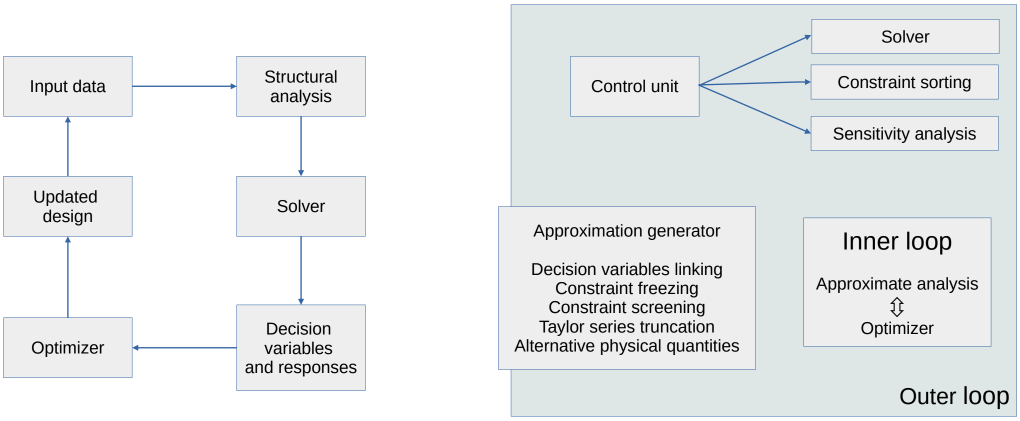
In addition to the constraints that arise in the form of functions other than the objective, a special kind of limitation acknowledges the fact that the decision variables are themselves restricted, as are the resources associated with structural sizing and material properties. These are the so-called side constraints, which complete the statement of an optimization problem by specifying limits, both lower (LB) and upper (UB) bounds, onto the decision variables, as in Equation (4):

$$\{X\}^{LB} \leq \{X\} \leq \{X\}^{UB} \tag{4}$$

Mass, strength, stiffness and natural frequencies and associated mode shapes of vibration are examples of objectives/constraints relevant to structural design, and geometry/materials/manufacturing processes are examples of the collectively shared decision variables.

2 Structural Optimization

The set of $j + k + 1$ functions defined in Equations (1)-(3) all share the same status, being responses (or outcomes) that depend on the same set of decision variables (or inputs) $\{X\}$. As such, they are in principle interchangeable in terms of which response is the objective and which ones are to be constrained. The caveat is that



(a) Direct link

(b) Approximate link

Figure 1: Comparison of the direct and approximate link between the Finite Element solver and the optimizer.

some of these combinations (alternative formulations for the optimization problem) are more amenable for solution by existing methods than others. Accordingly, the practice of structural optimization since its early days has relied upon specific formulations intended to streamline the solution processes. More recently, even with the emergence of powerful computational resources, these applied optimization frameworks continue to be improved with focus on formulation, most often dedicated to mediating the transfer of data between a Finite Element solver and optimization methods to be detailed in the remainder of the current chapter.

In this context, the direct coupling between analysis and optimization modules, as indicated in Figure (1a), results in the evaluation of the entire numerical model, by means of the analysis module, at all iterations performed by the optimizer along its search for an optimal solution. Since the CPU effort associated to a single analysis is multiplied by the (usually high) number of function evaluations during optimization, this scheme is often associated with a computational overhead that discourages its use.

Based on the fact that not all the parts of the model contribute all the time, and with the same intensity/relevance to the progress of the optimization procedure, a set of algorithms known as structural synthesis techniques (Schmidt, 1960) has been proposed to rationalize the use of computer resources when coupling optimizers to numerical analysis software. Figure (1b) presents the framework of the synthesis approach.

It is important to highlight the role of the approximate problem generator in the framework presented in Figure (1b). This set of algorithms is the ultimate responsible for the feasibility of coupling numerical optimizers to analysis software, since its use results in significant drop of the need for computer resources in comparison to the optimization scheme presented in Figure (1a). The methods listed therein operate described in the next 4 topics, for the duration of this section.

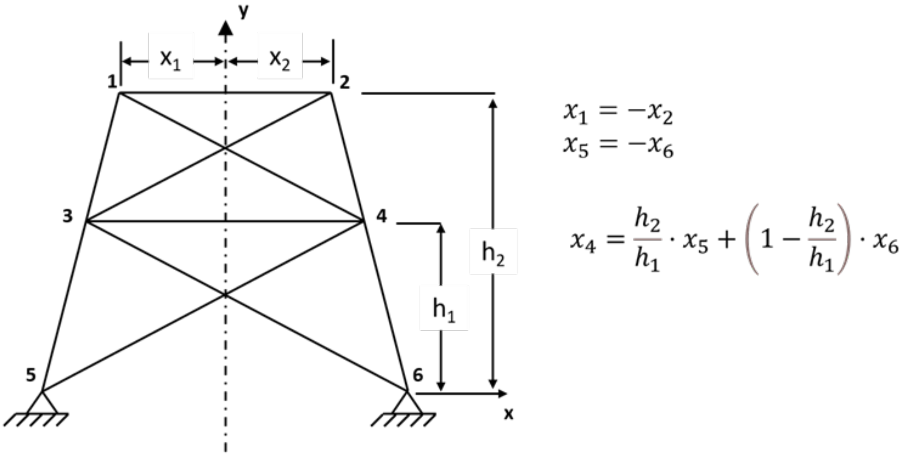

y
x_1
x_2
1
2
3
4
5
6
h_2
h_1
x
$x_1 = -x_2$
$x_5 = -x_6$
$x_4 = \frac{h_2}{h_1} \cdot x_5 + \left(1 - \frac{h_2}{h_1}\right) \cdot x_6$

Figure 2: Schematic depiction of variable linking in a truss type of structure to generate approximations capable of reducing the computational cost of structural optimization, while simultaneously enforcing symmetry.

2.1 Decision Variable Linking

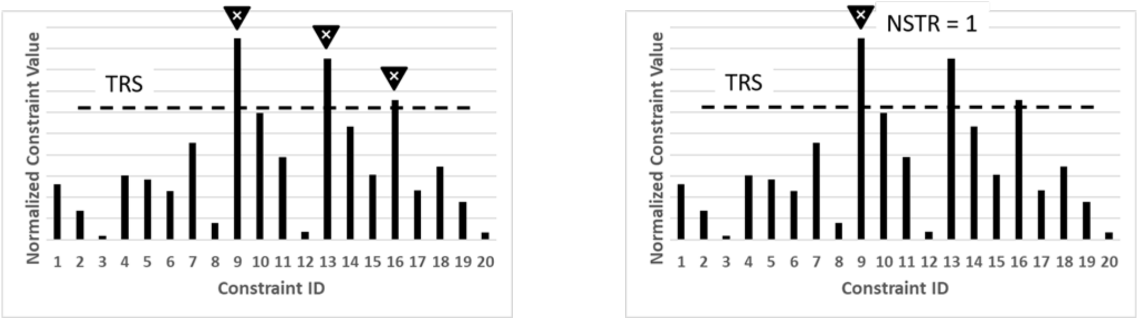
By establishing linear relations among design variables, one can reduce the number of independent variables to be evaluated. Economy of CPU resources is noticeable due to the drop in the number of partial derivatives of the responses, to be calculated with respect to the independent decision variables. Additional advantages introduced by this technique are: 1) The relations among decision variables are directly controlled by the user, which helps to keep physical insight; 2) The laws of dependence can be useful to enforce desirable design features, such as symmetry and parallelism, as shown in Figure (2).

2.2 Constraint Freezing and Screening

If a given subset of all the prescribed constraints has no risk of violation, it is useless to waste CPU time with their evaluation and the calculation of their derivatives with respect to the design variables. Hence, for the sake of feasibility, the constraints far from the violation threshold (TRS) can be neglected until their importance grows (i.e., risk of violation arises) at a different stage of the automated design process. This concept is illustrated graphically in Figure (3a).

For the purpose of further CPU economy, it should be considered that numerical models are discrete, which leads to the unfolding of the constraints prescribed in the statement of the optimization problem into a much larger number of components. For instance, consider a group of thousands of shell finite elements employed to model an aeronautic airfoil. If one desires to minimize the structure's weight keeping track of stress levels, constraints must be prescribed over all the elements of the airplane's wing. Just a few (NSTR) elements, however, can be considered at each optimization iteration, due to their superior representativity in comparison with the others. The effect of NSTR on constraint screening is illustrated in Figure (3b).

It should be noted that the procedures of constraint freezing and constraint



(a) (b)

Figure 3: Schematic illustration of the combined constraint freezing/constraint freezing procedure. Constraints are normalized.

screening are overlapped (i.e., used in conjunction) in order to avoid the heavy calculations involved in the evaluation of unnecessary constraint functions: the number of such functions is thus reduced to the least possible.

2.3 Taylor Series Truncation

Up to this point, the model reduction techniques presented acted on quantitative basis, that is, alternatives to reduce the number of independent design variables and constraints were indicated. Although these solutions are effective, more CPU power can be saved by addressing the qualitative aspects related to the functions evaluated during the optimization procedure. If several simple functions are eliminated by means of constraint deletion and screening but a few very complex, highly non-linear functions remain, still too much computer effort will result. For this reason, simplification by linearization may be beneficial, and this can be performed by expanding the functions in a Taylor Series to be truncated at a lower order/first term, as indicated in the Equations (5) and (6) below:

$$f(x^0 + \Delta x) = f(x^0) + \left.\frac{df}{dx}\right|_{x^0} \cdot \Delta x + \left.\frac{d^2 f}{dx^2}\right|_{x^0} \cdot \frac{\Delta x^2}{2!} + \left.\frac{d^3 f}{dx^3}\right|_{x^0} \cdot \frac{\Delta x^3}{3!} + \cdots \tag{5}$$

$$\hat{f}(x^0 + \Delta x) = f(x^0) + \left.\frac{df}{dx}\right|_{x^0} \cdot \Delta x \tag{6}$$

where the approximation of function f within an interval Δx around the reference point x^0 can the represented by a Taylor series expansion, with as many terms as are deemed necessary and sufficient to balance accuracy and expediency including, at the limit, just the initial linear term.

2.4 Alternative Physical Quantities

Still in the effort of simplifying the functions involved in the numerical non-linear optimization process, a special group of algorithms was developed, having in mind engineering situations often present in design optimization tasks.

As far as specific approaches go, it is useful to recall that a very common structural optimization task aims to obtain the minimum possible mass without violating stress constraints. Since the material (and consequently its density) is very seldom considered as design variables, the optimizer has to impose changes to the geometric parameters and the area is chosen, most of the times, as the design variable.

In such a formulation, the objective function displays a linear, explicit relation with respect to the design variables. The same, however, does not hold true for the constraints, because stresses and areas relate with each other by means of a reciprocal mathematical function. This situation poses a special difficulty for the optimizer because the optimization problem is usually strongly driven by the constraints, which get directly represented as nonlinear entities with respect to the decision variables in this particular kind of formulation. Indeed, one would prefer, for computational efficiency reasons, the objective function to became nonlinear, and the constraints linear with respect to the design variables. This switch can be done if the design variables became the reciprocal of the areas, which is equivalent to integrate the stresses with respect to the areas, obtaining the internal forces (Vanderplaats [1998]). Hence, it is a usual procedure to replace stresses by internal forces in structural synthesis problems.

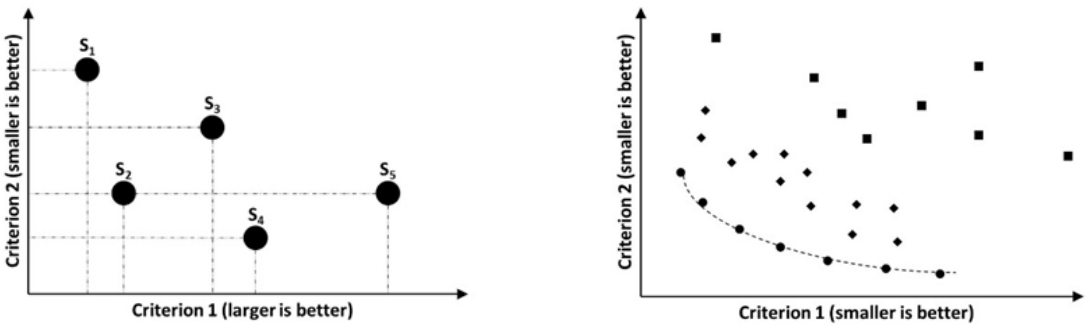
Conversely, in the case of structural dynamics problems, the corresponding computer cost mitigation approach consists of replacing the natural frequencies by the equivalent Rayleigh Coefficient (Canfield [1993]), with Equation (7) representing it as a dimensionless scalar that also relates the kinetic and elastic energies of the vibrating structure:

$$\lambda = \omega^2 = \frac{\phi^T \cdot k\phi}{\phi^T \cdot M\phi} = \frac{U}{T} \tag{7}$$

where K, M and ϕ are the stiffness, mass, and modal matrices, respectively. The scalars ω, λ, U and T stand for the natural frequency, its associated eigenvalue and the potential and kinetic energies.

3 Multi-criteria Optimization

Another noteworthy scenario with respect to possible combinations of the of $j+k+1$ functions defined in Equations (1)-(3) is that in which more than one response can participate within a set of objectives, configuring a multi-criteria optimization scenario (Gandibleux [2006], Odu and Charles-Owaba [2013]). The solutions for multi-criteria optimization problems are presented in terms of sets with multiple possible combinations of decision variables, each of them representing not strictly an optimal solution, in the absolute sense, but an optimal trade-off among the participating criteria. The solutions belonging to this set inherently group in a locus of the responses space, generally alluded to as the Pareto frontier (Wilson et al. [2001]), in a manifestation of a property called dominance or Pareto-optimality. By definition, a solution S1 to a multicriteria optimization problem dominates another solution S2 if S1 is no worse than S2 in all criteria and, furthermore, if S1 is better than S2 in at least one criterion. The concept of dominance in multicriteria decision-making can be graphically illustrated as in Figure 4. In panel (a), a two-dimensional response or output space (for 2 criteria), the best trade-offs would reside on the lower



(a) 2 criteria - larger is better AND smaller is better.

(b) 2 criteria - both smaller is better.

Figure 4: Schematic illustration of Pareto-dominance in the solution space for optimization problems with 2 criteria.

right extreme of the graph. However, driven by other constraints (explicit or not), there are cases such as S3 relative to S2 and S5 relative to S4 where Criterion 1 improves only at the expense of Criterion 2. In terms of dominance, both S1 and S5 dominate S2, whereas neither S2 nor S3 dominate any of the other solutions shown explicitly. Conversely, panel (b), shows the case of a two-dimensional output space where the 2 involved responses have to be minimized, and compete with each other. Even though the illustrations cover only two responses, for simplicity and clarity, the concept is valid for whatever arbitrary number of multiple criteria.

In the intersection between structural and multi-criteria optimization, a common situation arises when a certain region is discretized through some number of finite elements and, each of them having their own individual stress level, a single stress constraint is applied over the entire region. More generally, this situation is representative of the case in which a vector of responses that are connected/correlated to each other are subjected to the same practical constraint. In this scenario, a simple yet ineffective workaround would be to impose the constraint over the maximum value in the vector, that is, the maximum stress in this particular example, in the vein of being conservative. Because max(·) is a highly discontinuous function, numerical difficulties are to be expected, therefore the alternative presented in Equations (8) and (9) is recommended (Butkewitsch and Steffen, Jr. [1999]):

$$\min(\beta),\ 0 \leq \beta \leq 1 \tag{8}$$

subject to

$$G(T,\beta) = \beta - \frac{T}{T'} \geq 0 \tag{9}$$

For numerical conditioning, the objective defined by the auxiliary variable β is normalized in the unit interval, and its value is driven down subjected to the constraint that it still remains positive while the values in the true target T (the components of the vector constraint), normalized by T' are subtracted from it. These conditions are only satisfied as the values of T itself are reduced, as initially intended.

4 Multidisciplinary Design Optimization

As described in Gandomi et al. [2013], multidisciplinary design optimization (MDO) has been an established field for as long as optimization migrated from a purely mathematical to a fully applied science. It gained significant momentum from the mid-1990s, as asserted by references Ragon et al. [2003], Giunta et al. [1997], Alexandrov and Kodiyalam [1998], Venter and Haftka [1999], mainly impelled by the challenge and opportunity of performing optimal decision making, in the mathematical sense, as in the present material, however targeting applications in aircraft design, that could leverage ever increasing computer processing power. Given that aircraft are highly complex and integrated systems themselves, a natural derivation merged MDO and Systems Engineering into MSDO (Multidisciplinary Systems Engineering), in which the mathematical framework to solve MDO problems has been applied to systems (References Neufville et al. [2004], Gu et al. [2000], Agarwal and Renaud [2004]).

While settling itself as MDO and evolving into MSDO, this entire knowledge field has consolidated a number of approaches to handle problems with competing domains, as described in the sequence.

4.1 All-at-Once (AAO)

For being the most straightforward, All-at-once is also the most usual approach for optimizing complex systems. It should be noted, however, that the clear exchange here is to gain simplicity potentially giving up some performance, since a single optimizer is in charge of determining all of the decision variables, driving all of the existing objectives and constraints alike. It is assumed, of course, that some representation of the system can supply the output values (either exact or approximate), given a set of inputs. Many options of simulation software exist for the increasingly preferred computational representation of such input to output relationships (References Altiok and Melamed [2010], Sturrock and Pegden [2011], Waller [2012], Ucar et al. [2017]). The AAO construct is depicted in Figure 5.

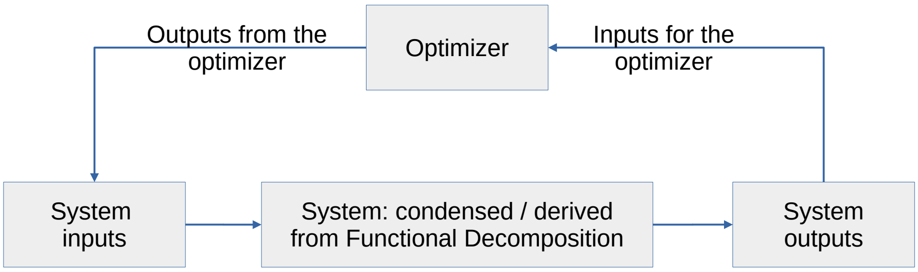

Outputs from the optimizer
Optimizer
Inputs for the optimizer
System inputs
System: condensed / derived from Functional Decomposition
System outputs

Figure 5: All-at-Once (AAO) approach for Multidisciplinary Systems Design Optimization (MSDO).

4.2 Individual Discipline Feasibility (IDF)

Contrary to All-at-Once, there is no overarching system representation invoked every time the optimizer iterates. Each sub-system is optimized in isolation (although the local mechanism is the same as that applied to the entire system within the AAO approach) and, at the sub-system level, each optimum is, by definition, feasible (or locally feasible, to be strict). The system-level optimizer has then to tune the decision variables if coupling constraints across the multiple-subsystems are violated in any form. These adjustments will penalize the local optima at the sub-systems in exchange of overarching feasibility. The notion of giving-up some optimality is intrinsic to partitioning the system in levels so that, in contrast with the All-at-Once approach, optimality is sacrificed for performance, since sub-systems can be optimized independently, and certainly in parallel. A potential drawback is the possibly unlimited complexity in how to define boundaries between sub-systems. Figure 6 captures the whole IDF idea.

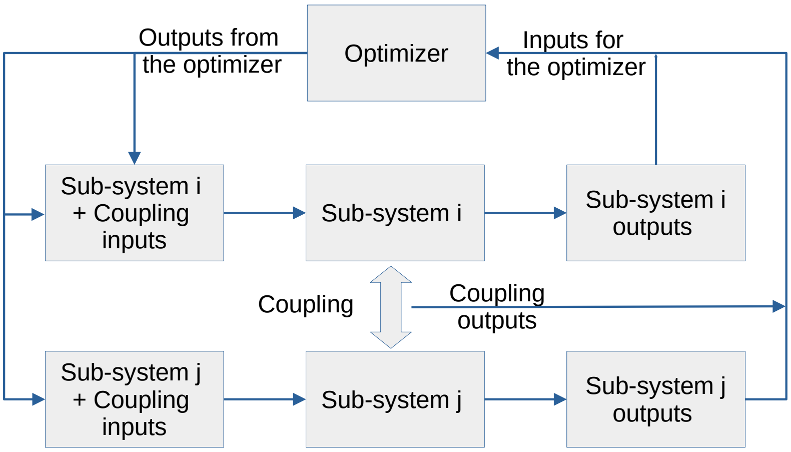


Figure 6: Individual Discipline Feasibility (IDF) approach for Multidisciplinary Systems Design Optimization (MSDO).

4.3 Collaborative Optimization (CO)

With Collaborative Optimization, the notion of a hierarchical approach is again leveraged, with decision variables at 3 levels: the system level, the coupling (inter) level and the sub-system (intra) level. It is hence similar to the IDF approach, but with the additional sophistication of decision variables preserved at the system level. The larger this system-level subset is, the stronger is the effort to preserve optimality despite the system partitions necessary to address local versus global aspects of the optimization process, as indicated in Figure 7.

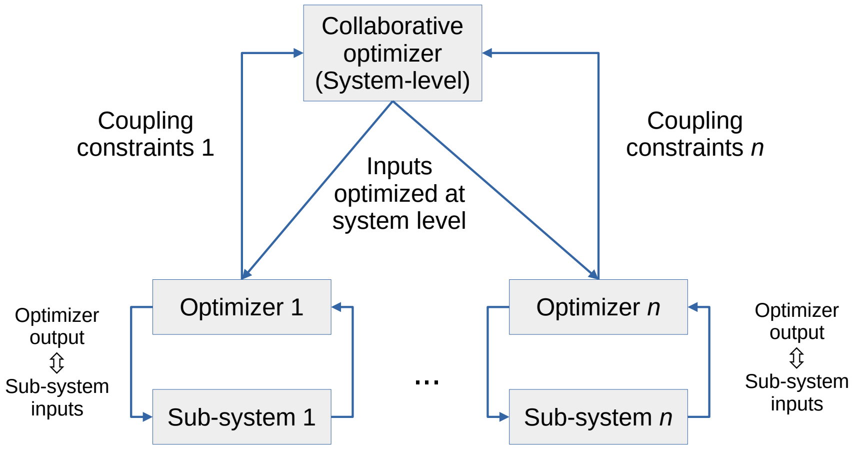

Collaborative optimizer (System-level)
Coupling constraints 1
Coupling constraints *n*
Inputs optimized at system level
Optimizer 1
Optimizer *n*
Sub-system 1
Sub-system *n*
...
Optimizer output ⇳ Sub-system inputs
Optimizer output ⇳ Sub-system inputs

Figure 7: Collaborative Optimization (CO) approach for Multidisciplinary Systems Design Optimization (MSDO).

4.4 Bi-level Integrated System Synthesis (BLISS)

Due to leveraging features/strengths of the many the previous approaches, the BLISS method is somewhat popular and works as follows:

1. Define a starting configuration
2. Perform an entire system analysis, as in each loop of the AAO approach, to check both optimality and feasibility
3. If overall feasibility exists (no violated constraints at any sub-system) and optimality is satisfactory, exit
4. If step 3 above is not satisfied, perform sensitivity analysis at each sub-system in preparation to optimize them individually, recalling that sensitivities are objective measurements of rates of change caused in outputs, given variations on the inputs, in a small enough range or neighborhood (i.e., in the local sense, as opposed to global sensitivity analysis such as discussed in Rashedi et al. [2018])
5. Repeat step 4, but at the system level (i.e., consider sensitivity of the couplings relative to inputs)
6. Optimization at the sub-system level (leveraging step 4)
7. Optimization at the system level (leveraging step 5)
8. Returns to step 3 to either terminate the process or launch the next iteration, until convergence criteria are met.

It should be noted that in very complex systems, where it is justified to unfold a hierarchical tree into many levels, nested BLISS configurations give rise to multi-level system optimization. One could devise something along these lines to encompass from supply chain elements (at the higher end of the system spectrum), all the way down to structural design of individual parts. Contemporary computer tools and system simulation software make it realistic to address such a use case at the time of writing (References Altiok and Melamed [2010], Sturrock and Pegden [2011], Waller [2012], Nardin et al. [2009]). Figure 8 depicts BLISS graphically.

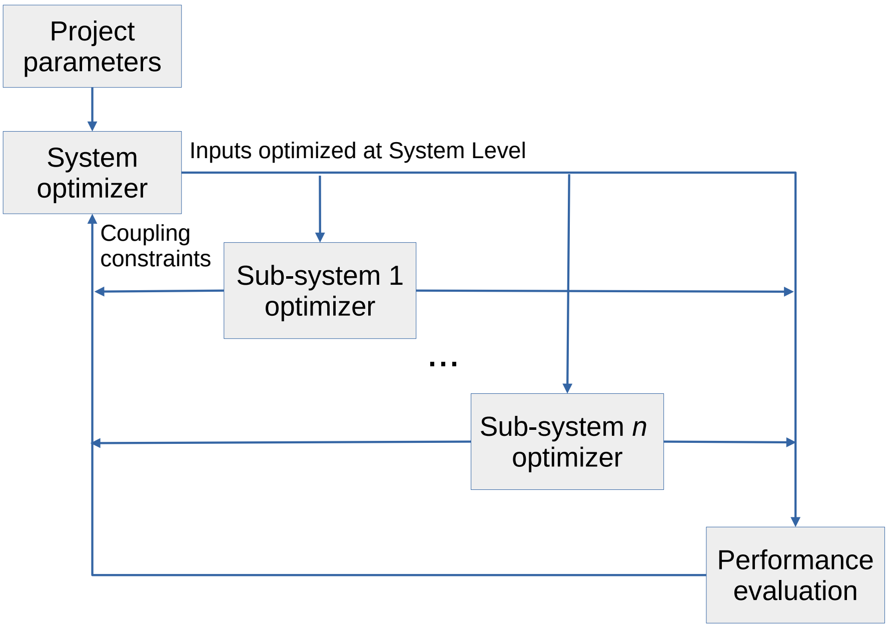


Figure 8: Bi-level Integrated Systems Synthesis (BLISS) approach for Multidisciplinary Systems Design Optimization (MSDO).

5 Solution Techniques

This section describes and discusses approaches to solve the optimization problem stated in its various forms, as per the preceding topics within this chapter. A number of computational implementations is offered to illustrate the main solution techniques, all of them presented as “Code” crafted in the R statistical computing platform R Core Team [2020].

For this purpose, the command `install.packages` is the main tool to add new packages to the R software. It takes a vector of names and a destination library, downloads the packages from the repositories (from a local folder or remote web server) and installs them in the R environment. To install the packages needed to run the codes in this chapter, for example, enter the following `install.packages('Rglpk', 'nloptr')`

Help is available at the command prompt via `?install.packages` command.

5.1 Classical Optimization Techniques

Classical, calculus-based or gradient-based optimization techniques will employ a collection of methods which, for the most part, operate by the use of derivatives to map a decision space between starting point(s) and the optimum, navigating from the former to the latter in an iterative fashion. Regardless of how specifically these iterations are undertaken by each specific method, the process is deemed convergent into an optimal configuration $\{X^*\}$ upon satisfaction of the Karush-Kuhn-Tucker conditions, as in Equations (10)-(12):

$$G_j(\{X^*\}) \leq 0 \tag{10}$$

$$\lambda_j \cdot G_j(\{X^*\}) = 0 \tag{11}$$

$$\nabla F(\{X^*\}) + \sum_j \lambda_j \cdot \nabla G_j(\{X^*\}) + \sum_k \lambda_k \cdot \nabla H_k(\{X^*\}) = 0 \tag{12}$$

The first of the three conditions states that a solution must be feasible (i.e., satisfies all the constraints) to be considered optimal. Second and third conditions introduce the Lagrange Multipliers λ, which scale the gradient vectors (∇) for both equality and inequality constraints such that, when they are added to the gradient vector of the objective function itself, the resultant is zero at the optimum point $\{X^*\}$. This reflects an equilibrium condition, mathematically expressed by the solution of a homogeneous system of linear equations in $\{X\}$, tackled by the methods described in the remainder of this chapter.

6 Linear Programming

Linear programming consists in a class of problems from mathematical programming which the objective index and constraints are described by linear equations. Hence, they are particularly kin to structural optimization problems that can be linearized and, more specifically, when the decision space is discrete, such as when selecting structural members from a pre-defined catalog. Despite the great variety of problems and mathematical formulations encompassing the category, the linear program can be written in the standard form (Noble and Daniel [1988])

$$\max c_1x_1 + c_2x_2 + \cdots + c_nx_n \tag{13}$$

subject to

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \tag{14}$$

$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \tag{15}$$

$$\cdots \tag{16}$$

$$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m \tag{17}$$

$$x_i \geq 0,\ \forall i = 1, \cdots, n \tag{18}$$

where $c \in \mathbb{R}^{n\times 1}, a \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^{m\times 1}$ are fixed real values and $x \in \mathbb{R}^{n\times 1}$ are real numbers. The variables $x_i,\ i = 1, \cdots, n$ are the design variables.

In a compact vector notation the standard problem becomes:

$$\max\{C\}^T\{X\} \tag{19}$$

subject to

$$A\{X\} = \{B\} \tag{20}$$

$$\{X\} \geq 0 \tag{21}$$

With Equations (19) and (20) reinforcing the linear nature of the entire formulation through the dot product (linear combination) and system of simultaneous linear equations, respectively. Given $B \in \mathbb{R}^{m\times m}, m < n$ a set of m linearly independent columns of matrix A in Equation (20), the matrix is non-singular and therefore admits a unique solution for the expression:

$$B\{X_B\} = \{B\}. \tag{22}$$

If all $n - m$ components of $\{X\}$ not associated with columns of a matrix B are zero, the corresponding solution is a basic solution of Equation (22) with respect to B, which is mathematically equivalent to the second condition in Equation (11). The components of the design variable $\{X\}$ associated with columns of B are the basic variables. A vector $\{X\}$ is feasible if it satisfies equations (20) and (21). If a feasible solution is also a basic solution, it is said to be a basic feasible solution.

A fundamental theoretical result states that if there is a feasible solution to the linear program then there is a basic feasible solution. Furthermore, if there is an optimal feasible solution, there is an optimal basic feasible solution. This result is important because it converts the task of solving a linear program to a task of computing basic feasible solutions. For a problem represented by n variables and m constraints there are

$$\frac{n!}{m!(n-m)!} \tag{23}$$

or less basic solutions. This is an upper limit on the number of solution candidates for the linear program. There are several algorithms in the literature dealing with the task of finding an optimal design. The main idea behind a popular method “The Simplex Method” is outlined in the following.

6.1 The SIMPLEX method

Given the general linear programming problem expressed in Equations (19)-(21), where $\{X\} \in \mathbb{R}^{n\times 1}, \{C\} \in \mathbb{R}^{n\times 1}, A \in \mathbb{R}^{m\times n}$ and $\{B\} \in \mathbb{R}^{m\times 1}$, the optimal solution is also a basic feasible solution. Solving such problem requires the addition of slack variables $x_{n+1}, \cdots, x_{n+m}$ to convert the inequality constraint given by Equation (20) in equality constraint. As a result, the extended version of the general linear programming problem is given by

$$\max M = \{C_e\}^T\{X_e\} + u \tag{24}$$

subject to

$$A_e\{X_e\} = \{B\} \tag{25}$$

$$\{X_e\} \geq 0 \tag{26}$$

where $u \in \mathbb{R}, \{X_e\} \in \mathbb{R}^{q\times 1}, \{C_e\} \in \mathbb{R}^{q\times 1}, A_e \in \mathbb{R}^{m\times q}$ and $\{B\} \in \mathbb{R}^{m\times 1}$. Adopting this representation as a general linear programming formulation, it follows that $A_e = [A, I_m], \{C_e\}^T = \left[\{C\}^T, \{0\}^T\right], q = m + n$ and $u = 0$ express the equivalence between Equations (19)-(21) and Equations (24)-(26).

The first step of the Simplex Method consists in the definition of the matrix

$$T = \begin{bmatrix} A_e & \{B\} \\ -\{C_e\}^T & u \end{bmatrix} \tag{27}$$

and then a number of elementary operations are performed over the lines of the matrix. The only allowed operation over the last line of T is the addition of linear combination of the lines above. As a result, this new matrix is obtained

$$T' = \begin{bmatrix} A'_e & \{B'\} \\ -\{C'_e\}^T & u' \end{bmatrix} \tag{28}$$

and the following linear programming problem is equivalent to the problem given by Equations (24)-(26). Along the Simplex Method a sequence of matrix operations is performed, according to the matrix representation of T, Equation (28), which elements where updated by transformations over the lines. For each T, a basic vector of viable solution exists. Suppose that at the step r, a feasible basic solution is found and the basic variables are $x_1, x_2, \cdots, x_m$. Furthermore, if the columns of T_r corresponding to the basic variables are unitary vectors, then T_r is expressed as

$$T_r = \begin{bmatrix} 1 & \cdots & 0 & a_{1,m+1} & \cdots & a_{1,q} & b_1 \\ 0 & 1 & 0 & a_{2,m+1} & \cdots & a_{2,q} & b_2 \\ 0 & \cdots & 1 & a_{m,m+1} & \cdots & a_{m,q} & b_m \\ 0 & \cdots & 0 & -c_{m+1} & \cdots & -c_q & u_r \end{bmatrix}. \tag{29}$$

In addition, the objective now is written as

$$M = u_r + c_{m+1}x_{m+1} + \cdots + c_q x_q \tag{30}$$

Equation (30) holds regardless of the values of $x_1, x_2, \cdots, x_q$. The basic solution available at this stage is found by using $x_{m+1} = x_{m+2} = \cdots = x_q = 0$ and the maximum value is $M = u_r$. At this stage, if all $c_j < 0$, for all $j = m+1, \cdots, q$, any increase in the values of $x_{m+1}, \cdots, x_q$ will lead to a decrease in the objective index M. This test is an indicative of the optimality of the solution.

On the other hand, if there is at least one $c_j > 0$ for $j = s$, the value of the objective index M will be increased by updating the value of $x_s \geq 0$. In this case, a new step is performed and the matrix T is updated. As a result, the numerical procedure is summarized as follows:

1. Find the column index $s \in \{m+1, m+2, \cdots, q\}$ at the last line of the matrix T whose element $-c_s$ has the lowest value.

2. Define $k = \frac{b_i}{a_{i,s}}$ with the smallest value of $\frac{b_i}{a_{i,s}}$ considering cases in which $a_{i,s} > 0$ holds.

3. Make a matrix pivoting at elements $a_{i,s}$ using the k value calculated above.

The succession of steps described above will lead to a sequence of feasible results that will maximize the objective index M. Since the number of basic solutions is finite, this procedure may lead to the optimal design.

Degeneracy occurs when a basic feasible solution contains a smaller number of non-zero variables than the number of independent constraints. In this case the values of some basic variables are zero and the replacement ratio is the same. Cycling in the algorithm will occur when the next basic solution to be investigated is one that was already investigated before. More advanced schemes are proposed to deal with the effects of degeneracy and cycling in linear programming. A comprehensive discussion about the Simplex Method and typical variants can be found in Luenberger and Ye [2008] and Vanderplaats [1998].

6.2 Duality and Sensitivity

Once again alluding to Equations (19)-(21) for a linear program, and considering Y^* to be its optimal solution, the maximum objective value of this formulation is also the minimal objective value of the following, rewritten linear program:

$$\min\{B\}^T\{Y\} \tag{31}$$

$$A^T\{Y\} \geq \{C\} \tag{32}$$

$$\{Y\} \geq 0 \tag{33}$$

and Y^* is also a solution for this problem. The linear program (13)-(18) is the so called primal problem and the new linear program (31)-(33) is the dual problem. There are a number of theoretical and computational properties shared by the formulations:

- The dual problem of a given dual problem is the primal.
- If $\{X\}$ satisfies the primal constraints (20)-(22) and {Y} satisfies the dual constraints (30)-(31) then $\{C\}^T\{X\} \leq \{B\}^T\{Y\}$.
- If $\{X\}$ satisfies the primal constraints (20)-(21), $\{Y\}$ satisfies the dual constraints (32)-(33) and $\{C\}^T\{X\} = \{B\}^T\{Y\}$ then $\{X\}$ is an optimal solution of the primal problem and $\{Y\}$ is an optimal solution of the dual problem.
- If exists non degenerated primal and dual feasible vectors then both primal and dual programs have optimal design solutions and the optimal objective values are identical.
- If the optimal program or the dual program has no feasible vector then both programs have no optimal solution.

These results allow a direct comparison of primal and dual feasible vectors and can be useful in evaluating the optimality of a solution. The results can also be used to convert a formulation with a large number of constraints into variables, that is, if $\{A \in \mathbb{R}^{m\times n} | m \gg n\}$ then the dual problem will be computed through the constraint matrix $\{A^T \in \mathbb{R}^{n\times m} | m \ll n\}$ and the number of matrix operations over the constraints is largely reduced. This scheme is sometimes required due to the concern with computational efficiency in problems with large number of constraints.

Code 1: Linear Programming method.

```
# maximize: 4 x_1 + 3 x_2 + 2 x_3
# subject to: 3 x_1 + x_2 + 2 x_3 <= 80
# 2 x_1 + 4 x_2 + 2 x_3 <= 70
# x_1 + 3 x_2 + 2 x_3 <= 70
# x_1, x_2, x_3 >= 0

library(Rglpk)

obj = c(4, 3, 2)
mat = matrix(c(3,2,1,1,4,3,2,2,2), nrow=3, ncol=3)
sig = c("<=", "<=", "<=")
rhs = c(80, 70, 70)
Rglpk_solve_LP(obj, mat, sig, rhs, max = TRUE)

$optimum
[1] 115

$solution
[1] 25 5 0
```

Another important concept refers to the sensitivity analysis of an optimal solution. Given a basic optimal solution $(\{X_B\}, 0)$ of the linear program (24)-(26) associated to an optimal basis B, where $\{X_B\} = B^{-1}B$, the solution of the corresponding dual formulation is $\{\lambda\}^T = \{C_B\}^T B^{-1}$.

If there is no degeneracy in the solution, the basis B is also optimal when small changes in the bounds $\{B\}$ of the constraint (Equation (25)) occur. Thus, for $\{B\} + \{\Delta B\}$ the optimal solution is

$$\{X\} = (\{X_B\} + \{\Delta X_B\}, 0) \tag{34}$$

where $\{\Delta X_B\} = B^{-1}\{DeltaB\}$. The corresponding change in the objective function value (Equation (30)) is

$$\Delta M = \{C_B\}^T \{\Delta X_B\} \tag{35}$$

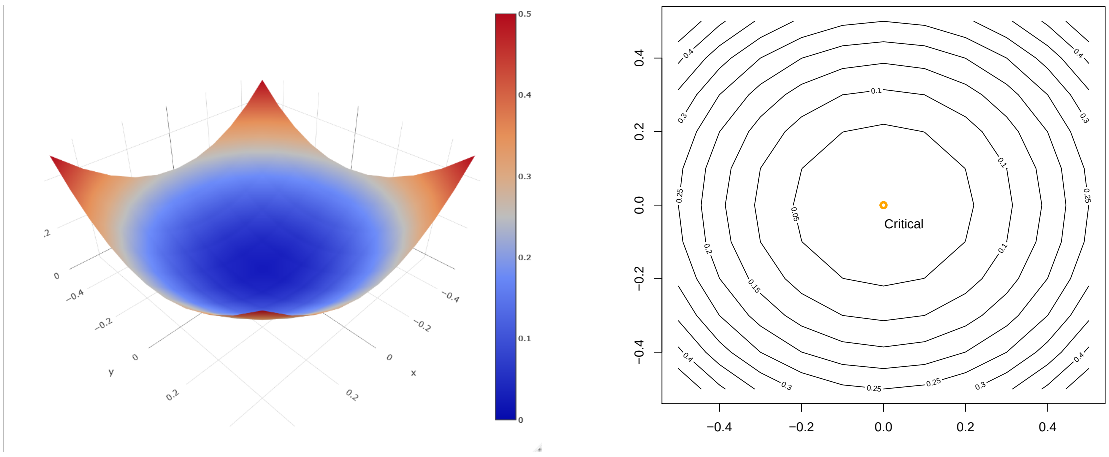
This expression addresses the sensitivity of the optimal value when small changes in $\{B\}$ are found. It can be understood as a marginal price of the component $\{B\}$, since when b_j changes to $b_j + \Delta b_j$, the value of the objective changes accordingly.

To summarize the Linear Programming approach, Code 1 implements it in the R statistical computing platform (Theussl and Hornik [2019]).

7 Non-Linear Programming

In the absence of linearization (either in the objective function and/or the constraints) or convexity within the decision space, the optimality conditions defined in Equations (10)-(12) should be resolved via non-linear optimization methods, to be detailed under each topic within this section.

The plot of $f(x, y) = x^2 + y^2$ is shown in Figure 9, along with a contour plot of the search space. The constrained case will be discussed later in section 8.



(a) Image of $f(x, y)$

(b) Contour plot of $f(x, y)$

Figure 9: Plot of a nonlinear function.

7.1 One-dimensional minimization methods

One-dimensional or line search is the mechanism taking place at each iteration of a non-linear optimization procedure, whereby the values of the decision variables in the current iteration (i) are updated from their values in the previous one $(i - 1)$ according to a step of magnitude α taken along the direction delimited by $\{S\}$, a line or one-dimensional search along this vector's direction, as in Equation (36):

$$\{X\}^{(i)} = \{X\}^{(i-1)} - \alpha\{S\}^{(i)} \tag{36}$$

Since the previous status of $\{X\}$ is known, the line search boils down to determining both the search direction and the magnitude of the step along it. Many methods exist, and the broad categories with the most important ones are the object of the forthcoming bullet points. It should be noted that the choice of search direction should influence not only the one-dimensional line to be pursued for the current iteration but also, for the higher-order methods, how the successive lines of search are concatenated and, ultimately, how the overall landscape of the decision space is mapped. In regards to the step size, it plays the role of a learning rate, for which it is necessary to balance not too large a number that the procedure will diverge/overshoot, neither an excessively small one that will unnecessarily penalize the performance.

The determination of the search direction $\{S\}$ is discussed next.

7.2 Zero-order methods

Methods in this category require function values only, but none of their derivatives, which could still be taxing depending upon the size and complexity of the model used for analysis of the objective function (for now, in an unconstrained scenario).

Usually at the expense of a very high number of function evaluations, these methods tend to be robust enough to circumvent issues related with non-convex and discontinuous functions, extensible to the case of discrete decision variables.

The primary representatives of this class are Random Search and Powell's method (described in greater detail in the sequence), while other methods are also documented in the literature (Vanderplaats [1998]).

7.2.1 Random Search

Practical implementations of Random Search depart from limiting the overall scope (and computational burden) by stipulating bounds as in Equation (4), and then modify Equation (36) accordingly:

$$\{X\}^{(i)} = \{X\}^{(i-1)} + r\left[\{X\}^{UB} - \{X\}^{LB}\right] \tag{37}$$

where r is randomly drawn from the $[0, 1]$ interval.

The important balance to be established relates to the width of the interval between the side constraints $\{X\}^{LB}$ and $\{X\}^{UB}$: make it too wide, and the convergence becomes too expensive, whereas excessive narrowing may lead to missing the global optimum for more complex decision spaces. Ultimately, these factors may be balanced by way of the heuristic search methods detailed in the next chapter.

It should be noted that in Equation (37), the search direction $\{S\}$ is replaced by a randomly chosen point within the side constraints, obtained by the subtraction of the bounds directly, which gets multiplied by the aleatory scale factor r. Hence, the direction itself is fixed, as it arises from $\{X\}^{LB}$ and $\{X\}^{UB}$ that remain constant. A more efficient variation affects each component of $\{S\}$ separately and, for an n-dimensional search space, works as:

$$\{S\}^{(i)} = 2 \cdot [r_i - 0.5],\ i = 1, \cdots, n. \tag{38}$$

7.2.2 Powell's method

This approach relies theoretically on the concept of conjugate directions, which underpins most of the more powerful classical optimization methods, and can yield significant success in practice if the problem to be solved is or can be reasonably approximated as a quadratic one, which is a plausible scenario for structural optimization (via Taylor series, for example).

Code 2: The BOBYQA algorithm.

```
library(nloptr)

fobj <- function(x) {
  return( 100 * (x[2] - x[1] * x[1])^2 + (1 - x[1])^2 )
}

x0 = c(-2, 2); xl = c(-10, -10); xu = c(10, 10)
opt1 = nloptr(x0=x0, eval_f = fobj, lb = xl, ub = xu,
  opts=list(algorithm=’NLOPT_LN_BOBYQA’, ’maxeval’ = 500))
cat(’Optimal␣value:␣’, opt1$objective,
  ’\nOptimal␣design:␣’, opt1$solution)

Optimal value: 4.334152e-05
Optimal design: 1.001892 1.004418
```

Denoting two directions (p) and (q), not necessarily in sequence but indexed as such for notational simplicity, are conjugate if

$$\left(\{S\}^{(p)}\right)^T \cdot [H] \cdot \{S\}^{(q)} = 0. \tag{39}$$

The first direction is transposed to ensure dimensional compatibility with respect to the Hessian square matrix of order n, denoted as H, and holding the second order derivatives of the objective function.

According to Powell's method, a first set of searches following Equation (36) is performed across the n orthogonal directions that constitute the coordinates of the search space, finding each minimum associated with a corresponding α. While neither of these directions are necessarily conjugate, they provide the starting point for building subsequent directions that satisfy this condition. Having completed all n uni-dimensional searches, each of them becomes a column of the following approximation to the Hessian matrix:

$$[\hat{H}] = \left[\alpha_1 \cdot S^{(1)}, \alpha_2 \cdot S^{(2)}, \cdots, \alpha_n \cdot S^{(n)}\right] \tag{40}$$

which is then used as a generator for conjugate directions by summation of the columns of $[H]$, as follows:

$$S^{(i+1)} = \sum_{n} \alpha_i \cdot S^{(i)} \tag{41}$$

The method is repeated thereafter updating the iteration index, disposing of the leftmost column of $[H]$, shifting all remaining columns leftwards and appending the most recent conjugate direction into the (now empty) rightmost column. The flowchart in Figure (10) displays the entire method.

An advanced method using this concept is the BOBYQA (Powell [2009]), which usage is shown in Code 2 (Johnson [2020]).

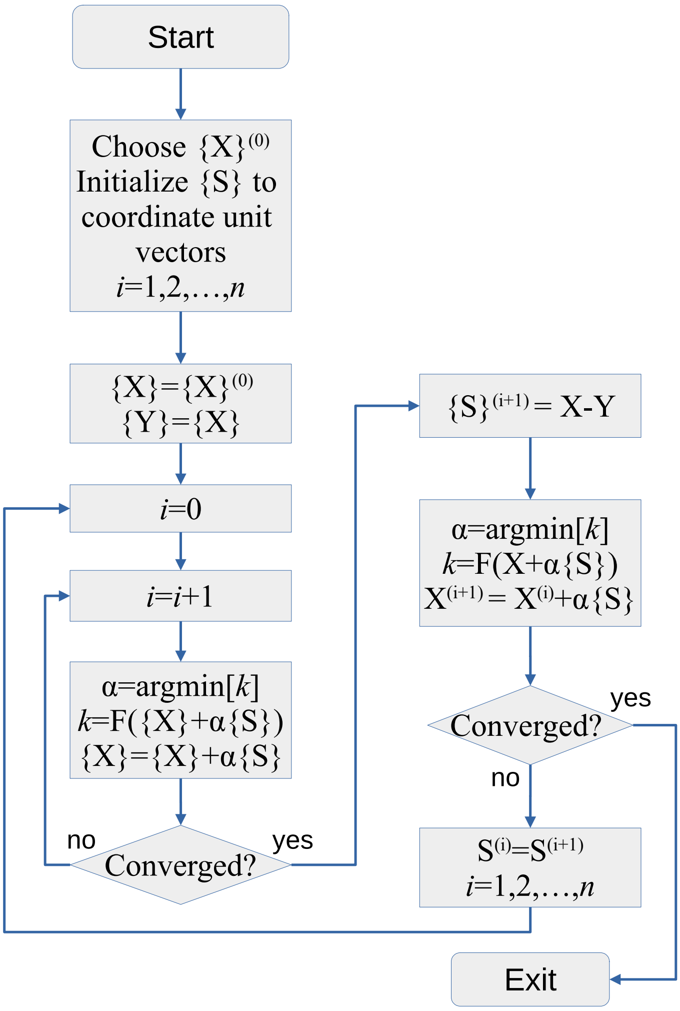


Figure 10: Flowchart with step-by-step description of Powell's zero-order optimization method.

7.3 First-order methods

The foremost first-order method is the Steepest Descent algorithm, which leverages the property of directional derivatives whereby the path of most intense variation of a function $F\{X\}$ at point $\{X^0\}$ is that of the gradient of the function at the same point. In essence, it is as simple as replacing $\{S\}$ by $\nabla F(\{X\})$ at each iteration, until Equations (10)-(12) are reasonably satisfied.

It should be noted however that while the instantaneous variation is the steepest by following the direction of the gradient to the function at each point, the overall performance of the method is severely penalized by the fact that the gradient is always perpendicular to the function surface. Along the successive iterations, this results in a series of search directions in which the current one is perpendicular to its prior. In effect, this means that each search direction retains no information whatsoever about the previous one (by orthogonality), and hence the decision landscape is poorly mapped.

To counter this loss of information that reduces the efficiency of the search, the

Conjugate Direction method (also known as the Fletcher-Reeves method) requires only a simple modification of the Steepest Descent approach, yet significantly improves the convergence rate. In effect, its initial search direction is the gradient of $F\{X\}$, to be updated according to Equations (42) and (43):

$$S^{(i)} = \nabla F(\{X\})^{(i-1)} + \beta^{(i)} \cdot S^{(i-1)} \tag{42}$$

$$\beta^{(i)} = \frac{\left|\nabla F(\{X\})^{(i-1)}\right|^2}{\left|\nabla F(\{X\})^{(i-2)}\right|^2} \tag{43}$$

This approach is conceptually similar to Powell's method, in the sense that the initial search directions are meant only to initialize the procedure, but different from the perspective that all of the search directions are conjugate from the beginning. The flowchart is in Figure 11.

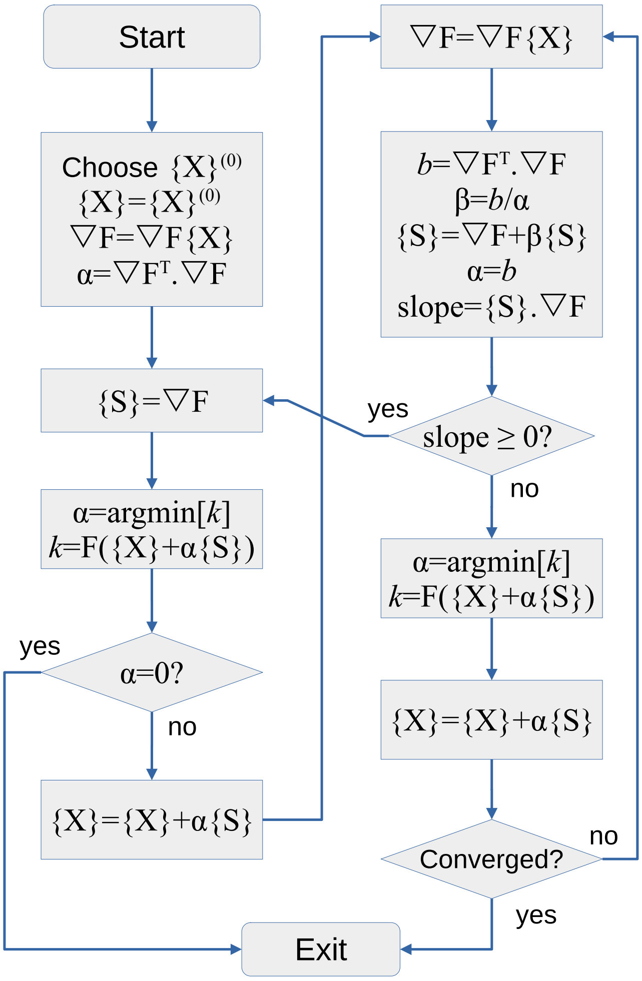


Figure 11: Flowchart with step-by-step description of Fletcher-Reeves Conjugate Direction method.

7.4 Second-order methods and approximations thereof

At this point, it becomes noticeable that adequate mapping of the decision landscape (i.e., topology of the objective function $\{F\}$ and how it varies in n-dimensional space) bears beneficial influence in terms of how efficient the optimization process occurs. On this vein, mapping the second derivatives will not only entail the information about the instantaneous reward (as in each isolated iteration of the Steepest Descent method), but also provide some anticipation on how the decision space varies thereafter, connecting a more plausible succession of search directions other than the sequentially perpendicular ones. As seen in Equation (41), the Hessian matrix $[H]$ is the container to store all information relative to the second-order derivatives, as in Equation (44):

$$[H(F(\{X\}))] = \begin{bmatrix} \frac{\partial^2 F(\{X\})}{(\partial X_1)^2} & \frac{\partial^2 F(\{X\})}{(\partial X_1)\cdot(\partial X_2)} & \cdots & \frac{\partial^2 F(\{X\})}{(\partial X_1)\cdot(\partial X_n)} \\ \frac{\partial^2 F(\{X\})}{(\partial X_2)\cdot(\partial X_1)} & \frac{\partial^2 F(\{X\})}{(\partial X_2)^2} & \cdots & \frac{\partial^2 F(\{X\})}{(\partial X_2)\cdot(\partial X_n)} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 F(\{X\})}{(\partial X_n)\cdot(\partial X_1)} & \frac{\partial^2 F(\{X\})}{(\partial X_n)\cdot(\partial X_2)} & \cdots & \frac{\partial^2 F(\{X\})}{(\partial X_n)^2} \end{bmatrix} \quad (44)$$

which is square of order n and symmetric with respect to the main diagonal, where the second-order derivatives are with respect to each individual decision variable in $\{X\}$, with cross terms in the off-diagonal cells.

The paramount approach for utilizing $[H]$ as part of the optimization procedure is Newton's method, which starts with the Taylor series expansion in Equation (45), whose notation differs from that in Equation (5) to embed the notion of search iterations indexed by (i):

$$\begin{aligned} F(\{X\}) \;=\;& F(\{X\}^{(i)}) + \nabla F(\{X\}^{(i)})^T \cdot \delta(\{X\}) \\ & + \frac{1}{2}\delta(\{X\})^T \cdot \left[H(F(\{X\})^{(i)})\right] \cdot \delta(\{X\}) \end{aligned} \quad (45)$$

where

$$\delta(\{X\}) = \{X\}^{(i+1)} - \{X\}^{(i)}. \quad (46)$$

Solving Equation (45) for the stationary condition gives:

$$\delta(\{X\}) = -\left[H(F(\{X\})^{(i)})\right]^{-1} \cdot \nabla F(\{X\}^{(i)}). \quad (47)$$

Next, Equation (48) can be obtained by combining a re-arranged form of Equation (46) with Equation (47), and further simplified into Equation (45)

$$\{X\}^{(i+1)} = \{X\}^{(i)} + \delta(\{X\}) = \{X\}^{(i)} - \left[H(F(\{X\})^{(i)})\right]^{-1} \cdot \nabla F(\{X\}^{(i)}) \quad (48)$$

$$\{S\}^{(i)} = -\left[H(F(\{X\})^{(i)})\right]^{-1} \cdot \nabla F(\{X\}^{(i)}) \quad (49)$$

Complementary, Equation (49) can be derived from Newton's (also known as Newton-Raphson) method for determining roots of equations. In the present case, the goal

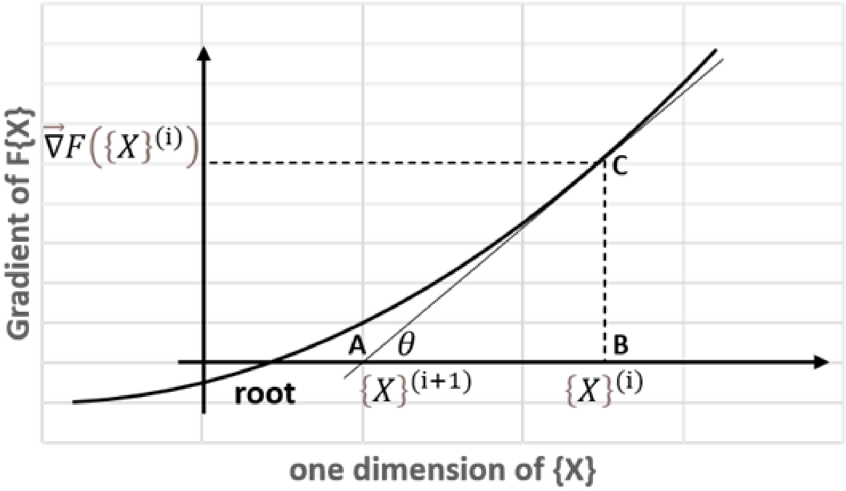


Figure 12: Newton-Raphson root finding approach to derive Newton's second-order search method.

is determining the roots of the equation corresponding to $\nabla F(\{X\}) = 0$, which indicates the stationary point of a function of n variables contained in $\{X\}$. Figure 12 describes the root finding procedure, projected to a single dimension, and how it circles back to Equations (47)-(49).

Starting at point $\{X\}^{(}i)$, a tangent (derivative or gradient component) to the function is drawn until it intercepts the horizontal axis, much closer to the root at $\{X\}^{(i+1)}$. It is possible to see that iterating a few more times from this new point (corresponding to the vertex A of the triangle ABC), convergence to the root will eventually occur. However, freezing into the current view and back to the triangle ABC, it is possible to see that the motion described in Equation (50) is:

$$\tan(\theta) = \left.\frac{d(\nabla F)}{d\{X\}}\right|_{\{X\}^{(}i)} = \frac{\nabla F(\{X\}^{(i)})}{\{X\}^{(i+1)} - \{X\}^{(i)}} \tag{50}$$

and given that

$$\{X\}^{(i+1)} - \{X\}^{(i)} = \frac{\nabla F(\{X\}^{(i)})}{\left.\frac{d(\nabla F)}{d\{X\}}\right|_{\{X\}^{(i)}}} \tag{51}$$

and

$$\left.\frac{d(\nabla F)}{d\{X\}}\right|_{\{X\}^{(i)}} = \frac{d^2F(\{X\})}{dx^2} \tag{52}$$

Equations (51) and (52) can be combined as in Equation (53) and, when generalizing the second derivative back into its matrix form, confirm Equation (47):

$$\{X\}^{(i+1)} - \{X\}^{(i)} = \frac{\nabla F(\{X\}^{(i)})}{\frac{d^2F(\{X\})}{dx^2}} \tag{53}$$

The computational counterpart of Equations (44)-(53) in R format is provided within Code 3 (Dembo and Steihaug [1983], Johnson [2020]):

Code 3: The Newton algorithm.

```
library(nloptr)

fobj <- function(x) {
  return( 100 * (x[2] - x[1] * x[1])^2 +
  (1 - x[1])^2 )
}
gfobj <- function(x) {
  return( c( -400 * x[1] * (x[2] - x[1] * x[1]) - 2 *
  (1 - x[1]), 200 * (x[2] - x[1] * x[1])) )
}

x0 = c(2, 2); xl = c(-10, -10); xu = c(10, 10)
opt1 = nloptr(x0=x0, eval_f = fobj, eval_grad_f = gfobj,
  lb = xl, ub = xu,
  opts=list(’algorithm’=’NLOPT_LD_TNEWTON’,
  ’xtol_rel’ = 1.0e-6))
cat(’Optimal␣value:␣’, opt1$objective,
  ’\nOptimal␣design:␣’, opt1$solution)

Optimal value: 6.252339e-28
Optimal design: 1 1
```

Let’s now make some considerations regarding the overhead for computing the second-order derivatives to fill the n-dimensional, symmetrical square matrix $[H]$. A total of $\frac{n(n-1)}{2}$ derivatives is to be calculated, resulting in computational complexity $O(N^2)$. Therefore, while the detailed mapping of the decision space to be navigated is useful, it has to be pondered against the calculation burden, and the trade-off is approximating $[H]$ instead of calculating it explicitly. Davidon-Fletcher-Power (DFP) and Broyden-Fletcher-Goldfarb-Shanno (BFGS), the paramount methods to implement this philosophy, are detailed in the sequence. Both of them share Fletcher and Reeves philosophy about storing information pertinent to previous iterations, but do that using a vector instead of the single scalar β within Equations (42) and (43), as well as Figure (11).

Along these lines, to approximate the Hessian matrix, these methods start by making it equal to the Identity Matrix, which is equivalent to start the search through the direction of Steepest descent (constant gradient across the main diagonal). Then, interactively, this approximate Hessian is updated as follows:

$$[H]^{(i+1)} = [H]^{(i)} + [D]^{(i)} \tag{54}$$

where

$$\begin{aligned} [D]^{(i)} &= \frac{\sigma+\theta}{\sigma^2}\cdot\delta(\{X\})\cdot\delta(\{X\})^T + \frac{\theta-1}{\tau}\cdot[H]^{(i)}\cdot y\cdot([H]^{(i)}\cdot y)^T \\ & -\frac{\theta}{\sigma}\cdot\left[[H]^{(i)}\cdot y\cdot\delta(\{X\})^T + \delta(\{X\})\cdot([H]^{(i)}\cdot y)^T\right] \end{aligned} \tag{55}$$

Code 4: The BFGS method.

```
library(nloptr)

fobj <- function(x) {
  return( 100 * (x[2] - x[1] * x[1])^2 + (1 - x[1])^2 )
}
gfobj <- function(x) {
  return( c( -400 * x[1] * (x[2] - x[1] * x[1]) - 2 *
  (1 - x[1]), 200 * (x[2] - x[1] * x[1])) )
}

x0 = c(-2, 2); xl = c(-10, -10); xu = c(10, 10)
opt1 = nloptr(x0=x0, eval_f = fobj, eval_grad_f = gfobj,
  lb = xl, ub = xu,
  opts=list(’algorithm’=’NLOPT_LD_LBFGS’,
  ’xtol_rel’ = 1.0e-6))
cat(’Optimal␣value:␣’, opt1$objective,
’\nOptimal␣design:␣’, opt1$solution)

Optimal value: 1.243396e-18
Optimal design: 1 1
```

with the vector y being the difference between the gradients at two successive iterations, as in Equation (55), and the scalars σ and τ defined as in Equations (57) and (58), respectively:

$$y = \nabla F(\{X\}^{(i)}) + \nabla F(\{X\}^{\{}(i-1)\}) \tag{56}$$

$$\sigma = \delta(\{X\})^T \cdot y \tag{57}$$

$$\tau = y^T \cdot [H]^{(i)} \cdot y \tag{58}$$

DFP assumes $\theta = 0$ and BFGS uses $\theta = 1$, while other methods within this family, also known as quasi-Newton or variable metric (because of the nature of the function expressed in Equation (55)) have their own operating strategies. Code 4 provide an illustrative R implementation of the BFGS method (Liu and Nocedal [1989], Johnson [2020]).

While the emergence of high performance computers reduces the performance related difficulties of tackling full-fledged Hessian matrices, methods such as DFP and BFGS retain their relevance with respect to numeric stability, since they avoid inversion of matrices that often grow to very large sizes.

The determination of the step size/learning rate α is discussed next.

Once a search direction $\{S\}$ is determined, as discussed in the preceding item, it is then time to decide upon the magnitude of the motion along $\{S\}$, for which the following methods are mainstream.

7.4.1 Golden-section method

Consider the following logic to create a Fibonacci series which, apart from its two initial terms at 0 and 1, consists of progressively adding up the two latest terms to determine the next one:

$$\begin{aligned} F_0 &= 0 \\ F_1 &= 1 \\ F_i &= F_{i-1} + F_{i-2},\ i > 1 \end{aligned} \tag{59}$$

After stabilizing at its recursive portion for $i > 1$, the ratio between any two consecutive terms is equal to the Golden Section ratio ϕ, a constant calculated as in Equation (54):

$$\varphi = \frac{1+\sqrt{5}}{2} = 1.61803 \tag{60}$$

The constant defined as such is useful for finding the optimum of a one-dimensional function (which is the essence of one-dimensional or line searches) when, similarly to the bisection method for root finding, one could interactively determine the minimum of a function bounded within a known interval. Specifically, if in the neighborhood of the minimum we can find three points $x_1 < x_2 < x_3$ with functional values $f(x_1) > f(x_2) > f(x_3)$, then the minimum is located between x_1 and x_2 (or exactly at x_2, at the limit).

Finding this minimum starts by picking a brand-new point x_4 located between x_2 and x_3, such that either the minimum is located between x_4 and x_3 (upwards) or between x_1 and x_4 (downwards). Regardless, a new interval containing 3 points, similar to but narrower than the previous one, is delimited. For efficiency, we prefer that the remaining interval to be interactively shrunk from the previous one is as small as possible, which is obtained by intervals of same width irrespective to which side the search pivots to. Equality of interval lengths using the above notation is enforced as in Equation (55):

$$(x_2 - x_1) + (x_4 - x_2) = (x_3 - x_2) \tag{61}$$

which gets rearranged into

$$(x_2 - x1)^2 + (x_2 - x_1) \cdot (x_3 - x_2) - (x_3 - x_2)^2 = 0 \tag{62}$$

because $(x_3 - x_2) = (x_3 - x_4) + (x_4 - x_2)$. Solving Equation (61) and choosing the positive of the two roots yields a ratio between $(x_2 - x_1)$ and $(x_3 - x_2)$ which is exactly the golden section ϕ of Equation (60). This leads to the following update law for the iterative process, to be deemed converged upon reaching a suitable, typically problem dependent tolerance:

$$x_1^{(i+1)} = x_2^{(i)} \tag{63}$$

$$x_2^{(i+1)} = x_3^{(i)} \tag{64}$$

$$x_3^{(i+1)} = x_3^{(i)} + \varphi \cdot \left(x_3^{(i)} - x_2^{(i)} \right) \tag{65}$$

7.4.2 Root finding/Interpolation methods (quadratic and cubic cases)

A unique parabola may be fitted through 3 points x_1, x_2 and x_3 discussed in the golden section method above, resulting in the functional form and optimum expressed in Equations (66) and (67):

$$a \cdot \alpha^2 + b \cdot \alpha + c \tag{66}$$

$$\alpha_{\text{optimum}} = \frac{-b}{2 \cdot a} \tag{67}$$

Given the functional values at x_1, x_2 and x_3 chosen for the golden search approach, it is guaranteed that $a > 0$ in Equation (66) and, therefore, the optimum α does correspond to a minimum. Moreover, computational efficiencies can be gained if one of the three points within the working interval is made to be 0 and/or they are evenly spaced.

In some cases, even narrowly bounded intervals will contain functional behavior that is highly non-linear, requiring higher order interpolation to be performed. Equation (68) shows the cubic case. Optimality is guaranteed by finding the roots of Equation (69), which is the quadratic form of obtained by deriving the cubic interpolate. Moreover, minimization is obtained when the second derivative, which is linear, is strictly positive as shown in Equation (70). Calculation of the coefficients a and b that satisfy this condition is possible by having four gradient/function evaluations or, alternatively, three gradient evaluations.

$$a \cdot \alpha^3 + b \cdot \alpha 2 + c \cdot \alpha + d \tag{68}$$

$$3 \cdot a \cdot \alpha^2 + 2 \cdot b \cdot \alpha + c = 0 \tag{69}$$

$$6 \cdot a \cdot \alpha + 2 \cdot b > 0 \tag{70}$$

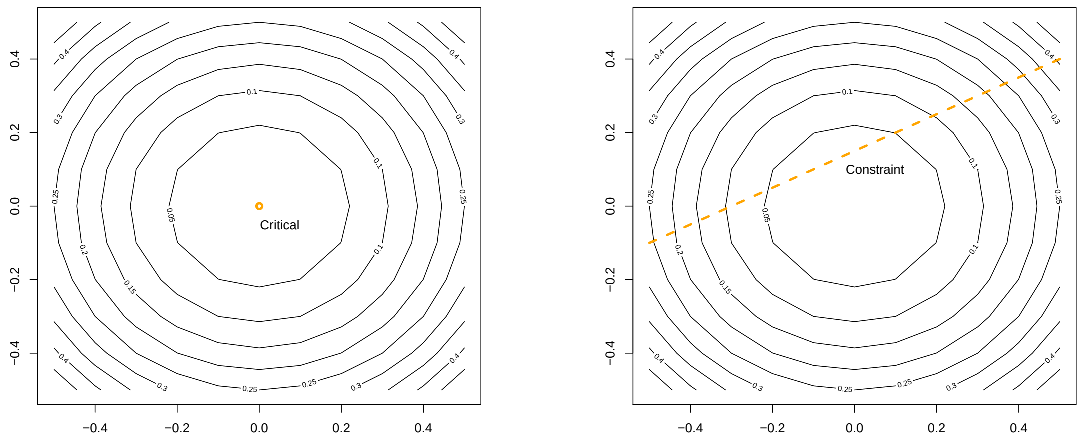
8 Constrained optimization techniques

Insofar, motion along the search direction $\{S\}$ at steps with length α did not account for encountering any constraint. While this is an adequate simplification for methodological purposes, not incorporating constraints prevents actual applicability of these methods, which will then be amended to represent the full extent of the optimization problem statement introduced in Equations (1)-(4). Hence, constrained optimization problems are solved by extensions of the well-established methods for the unconstrained scenarios.

A geometric representation of a constraint is shown in Figure 13. In the case of an equality constraint, the search space is reduced to the dashed line. In the case of an inequality constraint, the search space is a subset above or below the dashed line.

8.1 Sequential Methods

One of the strategies for leveraging unconstrained optimization methods into problems that actually do pose constraints is to penalize the search outcome to limit constraint violation.



(a) Unconstrained case

(b) Constrained case

Figure 13: Contour plot of a search space.

While simple, this idea has side effects as it results in numerical ill-conditioning, due to discontinuous nature of the penalties, which behave like "jumps" during the search process. This obstacle may be circumvented by modulating the penalties with a weight factor r_p as in Equation (71), which is moderate at the earlier search stages and increases as the solution approaches the optimum, operating in a sequentially increasing penalization mode, yielding the name for this class of methods:

$$\phi(X, r_p) = F(\{X\}) + r_p \cdot P(\{X\}) \tag{71}$$

The pseudo-objective $\phi(X, r_p)$ results from penalizing the original $F(\{X\})$ with the variable penalty $P(\{X\})$. Once again, the explicit dependency of the penalty on the current value of the decision variables stresses its adaptive nature.

8.1.1 Exterior Penalty Method

A penalty function (Equation (72)) is built in a fashion that it is innocuous if no constraint is violated:

$$P(X) = \sum_{i=1}^{m} [\max(0, G_j(\{X\}))]^2 + \sum_{i=1}^{k} [H_k(\{X\})]^2 \tag{72}$$

accounting for both inequality and equality constraints in $G_j(\{X\})$ and $H_k(\{X\})$, respectively. Squaring the constraints ensures a slope of zero for the constraint function at its violation boundary, improving numerical conditioning of the pseudo-objective defined in Equation (56).

Remaining ill-conditioning issues can be resolved by balancing the choice of r_p. When it is too small, the search process is very stable but lax for allowing constraints violation. Conversely, large r_p values lead to strict observance of the constraints, but induce the "jumps" alluded to before. It is customary to modulate this process by sequentially increasing the value of r_p by a constant factor γ, which is then fine-tuned for each problem. Figure 14 represents the Exterior Penalty method in graphical form.

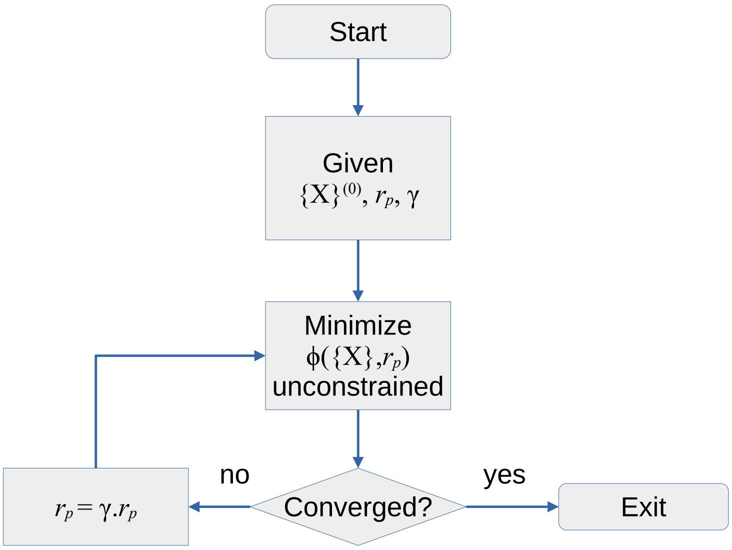


Figure 14: Flowchart for the Exterior Penalty method for constrained optimization.

8.1.2 Interior Penalty Method and Extension

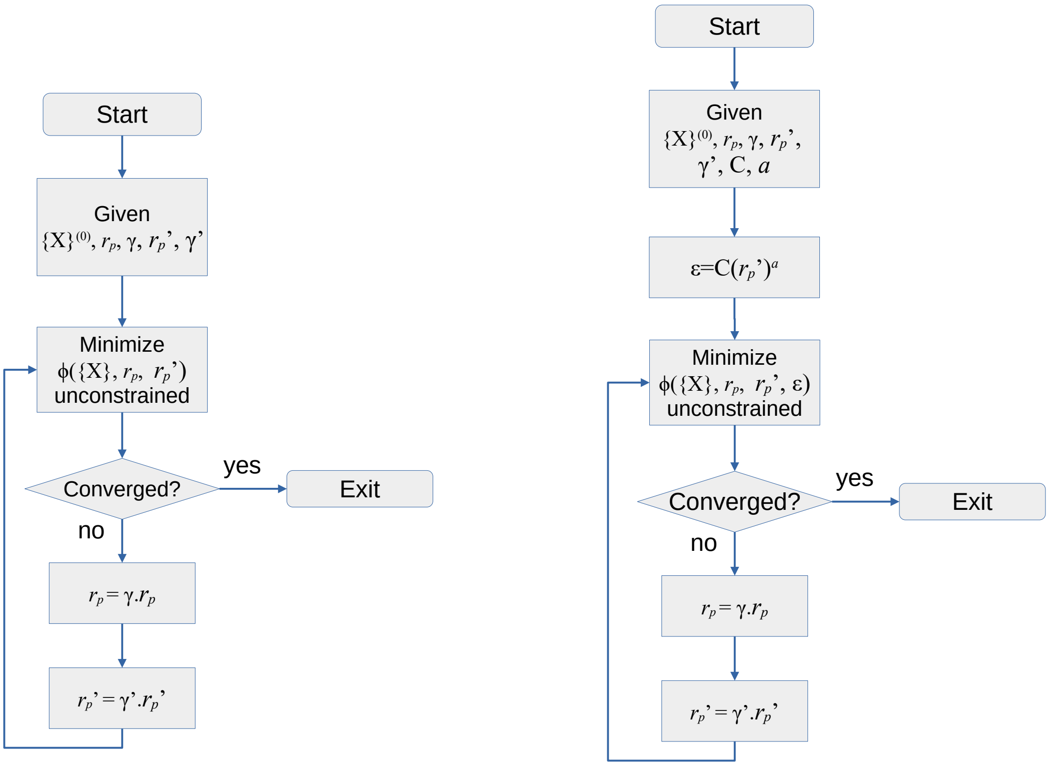
By providing a special treatment for inequality constraints, whereby the penalties are actually reduced instead of decreased over the search process (Equations (73)-(74)), this method aims at simplifying the unconstrained search portion of the solution:

$$\phi(X, r_p', r_p) = r_p' \cdot \sum_{i=1}^{m} -\frac{1}{G_j(\{X\})} + r_p \cdot \sum_{i=1}^{k} [H_k(\{X\})]^2 \tag{73}$$

$$\phi(X, r_p', r_p) = r_p' \cdot \sum_{i=1}^{m} -\log(-G_j(\{X\})) + r_p \cdot \sum_{i=1}^{k} [H_k(\{X\})]^2 \tag{74}$$

with both Equations (73) and (74) exchanging the discontinuous max(·) function present in Equation (72) for continuous ones, with variation "b" being reputed for slightly better numerical conditioning. Also, an additional weight factor r_p' is introduced to handle the inequality constraints separately from the equality ones. Over the course of the search, r_p' will decrease at a rate modulated by the constant parameter γ'.

The extended version of the method accounts for the specific degree of constraint violation relative to a tolerance level ε, a small negative quantity dependent upon constants C and a, that marks the transition between the interior penalty method to its extended version. Detailed formulation is described in Equations (75)-(78), and a flowchart of both interior penalty approaches is presented in Figure 15.



(a) Original method

(b) Extended method

Figure 15: Interior penalty method for constrained optimization.

$$P(\{X\}) = \sum_{i=1}^{m} \bar{G}_j(\{X\}) \tag{75}$$

where

$$\bar{G}_j(\{X\}) = -\frac{1}{G_j(\{X\})} \quad \text{if } G_j(\{X\}) \leq \varepsilon \tag{76}$$

$$\bar{G}_j(\{X\}) = -\frac{2\varepsilon - G_j(\{X\})}{\varepsilon^2} \quad \text{if } G_j(\{X\}) > \varepsilon \tag{77}$$

and

$$\varepsilon = -C \cdot (r_p')^a, \ \frac{1}{3} < a < \frac{1}{2}. \tag{78}$$

Closer examination of Equations (75)-(78) reveals that the penalty functions have continuous first derivatives at $G_j(\{X\}) = \varepsilon$, but the continuity of the second derivative is not assured unless the order of the penalty function is increased as in its quadratic form displayed at Equation (79):

$$\bar{G}_j(\{X\}) = -\frac{1}{G_j(\{X\})'} \quad \text{if } G_j(\{X\}) \leq \varepsilon \tag{79}$$

$$\bar{G}_j(\{X\}) = -\frac{1}{\varepsilon} \cdot \left[\left(\frac{G_j(\{X\})}{\varepsilon} \right)^2 - 3 \cdot \frac{G_j(\{X\})}{\varepsilon} + 3 \right] \quad \text{if } G_j(\{X\}) > \varepsilon \tag{80}$$

8.1.3 Additional Penalty Methods for improved numerical conditioning

Relatively recent developments are meant to ensure better balance between feasibility (the optimal solution respects the constraints) and numerical stability. Along these lines, a family of variable penalty functions is proposed as in Equations (81) to (89), and the logarithmic approach introduced in Equation (74) is improved as per Equations (90) to (91).

$$\bar{G_j}(\{X\}) = -\frac{-[G_j(\{X\})]^{1-s}}{s-1} \quad \text{if } G_j(\{X\}) \leq \varepsilon \tag{81}$$

and

$$\begin{aligned}\bar{G_j}(\{X\}) \;&=\; (-\varepsilon)^{1-s} \cdot \left[A \cdot \left(\frac{G_j(\{X\})}{\varepsilon} - 1\right)^3 + \frac{1}{2} \cdot \left(\frac{G_j(\{X\})}{\varepsilon} - 1\right)^2\right. \\ &\left. - \left(\frac{G_j(\{X\})}{\varepsilon} - 1\right) + \frac{1}{s-1}\right] \quad \text{if } G_j(\{X\}) > \varepsilon \end{aligned}\tag{82}$$

which undergo the following modifications when $s = 1$:

$$\bar{G_j}(\{X\}) = -\log\left(-G_j(\{X\})\right) \quad \text{if } G_j(\{X\}) \leq \varepsilon \tag{83}$$

and

$$\begin{aligned}\bar{G_j}(\{X\}) \;&=\; A \cdot \left(\frac{G_j(\{X\})}{\varepsilon} - 1\right)^3 + \frac{1}{2} \cdot \left(\frac{G_j(\{X\})}{\varepsilon} - 1\right)^2 \\ &- \left(\frac{G_j(\{X\})}{\varepsilon} - 1\right) - \log(-\varepsilon) \quad \text{if } G_j(\{X\}) > \varepsilon. \end{aligned}\tag{84}$$

Both Equations (82) and (84) depend on parameters A and s, whose values and fine tuning contemplate literature recommendations as expressed in Equations (85) to (90), connecting with the notion of the constraint violation tolerance ε and the sequential penalty approaches, such as those contained in Equations (71) to (74):

$$\varepsilon = -\beta \cdot (r_p')^q \tag{85}$$

where β is a positive constant chosen such that ε remains near zero at the start of the search, while q is recommended to be within the range in Equation (86)

$$\frac{1}{2+s} \leq q \leq \frac{1}{s}. \tag{86}$$

On its turn, A should be defined as in Equation (87)

$$A = \frac{1+s}{3} \quad \text{if } G_j(\{X\}) \leq 0 \tag{87}$$

and

$$A = \left[(1-s) \cdot \left(\frac{C^*}{\varepsilon - 1}\right)\right] \left[\frac{1}{3}\left(\frac{C^*}{\varepsilon - 1}\right)^{-2}\right] \quad \text{if } G_j(\{X\}) > 0 \tag{88}$$

indicating violated constraints, at a maximum C^*, and falling back into Equation (89) when all constraints are violated.

$$A = \frac{s}{6 \cdot \left(1 - \frac{C^*}{\varepsilon}\right)} \tag{89}$$

Regarding the logarithmic penalty function, also known as log-barrier function, its enhancement towards better numerical conditioning involves the modification in Equation (90). While it only explicitly addresses inequality constraints, the equality ones can be either handled by the standard exterior penalty method, or treated as two equal but opposite inequality constraints.

$$P(x, r'_p, \lambda^i) = r'_p \cdot \sum_{j=1}^{m} \lambda^i_j \cdot \log\left[1 - \frac{G_j(\{X\})}{r'_p}\right]. \tag{90}$$

Lastly, the Lagrange multipliers λ, recalled from the KKT convergence conditions in Equations (10)-(12), are updated along the i iterations as described in Equations (91)-(92).

$$\lambda^{i+1}_j = \frac{\lambda^i_j}{\left[1 - \frac{G_j(\{X\})}{r'_p}\right]} \quad \text{if } G_j(\{X\}) \leq k \cdot r'_p, \text{ with } k < 1 \tag{91}$$

$$\lambda^{i+1}_j = \frac{\lambda^i_j}{2r'_p(1-k)} \quad \text{if } G_j(\{X\}) > k \cdot r'_p, \text{ with } k < 1. \tag{92}$$

The next topic is dedicated to cover the role of Lagrange Multipliers in greater detail, including additional methods in which they have a more prominent role in solving constrained optimization problems.

8.1.4 Augmented Lagrange Multiplier Method

Adding a penalty term to the original objective function to account for the constraints continues to be the governing idea, but with direct influence from the KKT convergence condition of Equations (10)-(12). Historically, this approach was first derived for equality constraints (Equation (93)) and then generalized to include the inequality ones (Equation (99)). Both approaches are similar, as highlighted by the flowcharts in Figure 16.

$$A(X, \lambda, r_p) = F(\{X\}) + \sum_k 2\lambda_k H_k(\{X\}) + r_p[H_k(\{X\})]^2 \tag{93}$$

Observe that Equation (93) reduces to the exterior penalty method of Equation (72) if all λ_k are equal to 0. More interestingly, if the values of the Lagrange multipliers are chosen to be optimal, it is possible to prove that the true optimum of $F(\{X\})$ can be determining (for positive, but finite values of r_p), skipping the constrained portion of the method altogether and requiring a single unconstrained search. Since these optimum values of the Lagrange multipliers are actually unlikely to be available, it is possible to update them from an initial guess by using Equation (94):

$$\lambda^{i+1}_k = \lambda^i_k + 2r_p H_k(\{X\}^i) \tag{94}$$

Now, Equation (96) will provide the inequality constraints version of Equation (93), after the inequalities from Equation (2) are evened out into equalities, by adding the term Z^2 as in Equation (95). The value Z is squared due to the numerical conditioning ease of parabolic functions.

$$G_j(\{X\}) + Z^2 = 0 \tag{95}$$

$$A(X, \lambda, Z, r_p) = F(\{X\}) + \sum_j \left[\lambda_j \cdot \left(G_j(\{X\}) + Z^2\right) + r_p \cdot \left(G_j(\{X\}) + Z^2\right)^2\right] \tag{96}$$

By way of mathematical equivalence, Equation (79) brings an advantage over Equation (96) for it achieves the same conversion of inequality into equality constraints, but dispensing with the potentially many slack variables Z^2. Connecting with the optimization theory presented in the linear programming section, it should be noted that there will be as many Z^2 as there are inequality constraints, and then their number can be also contained by using the duality principle.

$$A(X, \lambda, r_p) = F(\{X\}) + \sum_j \left[\lambda_j \cdot \psi_j + r_p \cdot \psi_j^2\right] \tag{97}$$

$$\psi_j = \max\left(G_j(\{X\}), \frac{-\lambda_j}{2r_p}\right). \tag{98}$$

Finally, the all-encompassing case, with both equality and inequality constraints, can be represented in Equation (99) by combining Equations (93) and (97):

$$\begin{aligned} A(X, \lambda, r_p) &= F(\{X\}) + \sum_j \left[\lambda_j \cdot \psi_j + r_p \cdot \psi_j^2\right] \\ &\quad + \sum_k \lambda_{k+m} H_k(\{X\}) + r_p[H_k(\{X\})]^2. \end{aligned} \tag{99}$$

Among the many advantages of this formulation, the following should be highlighted:

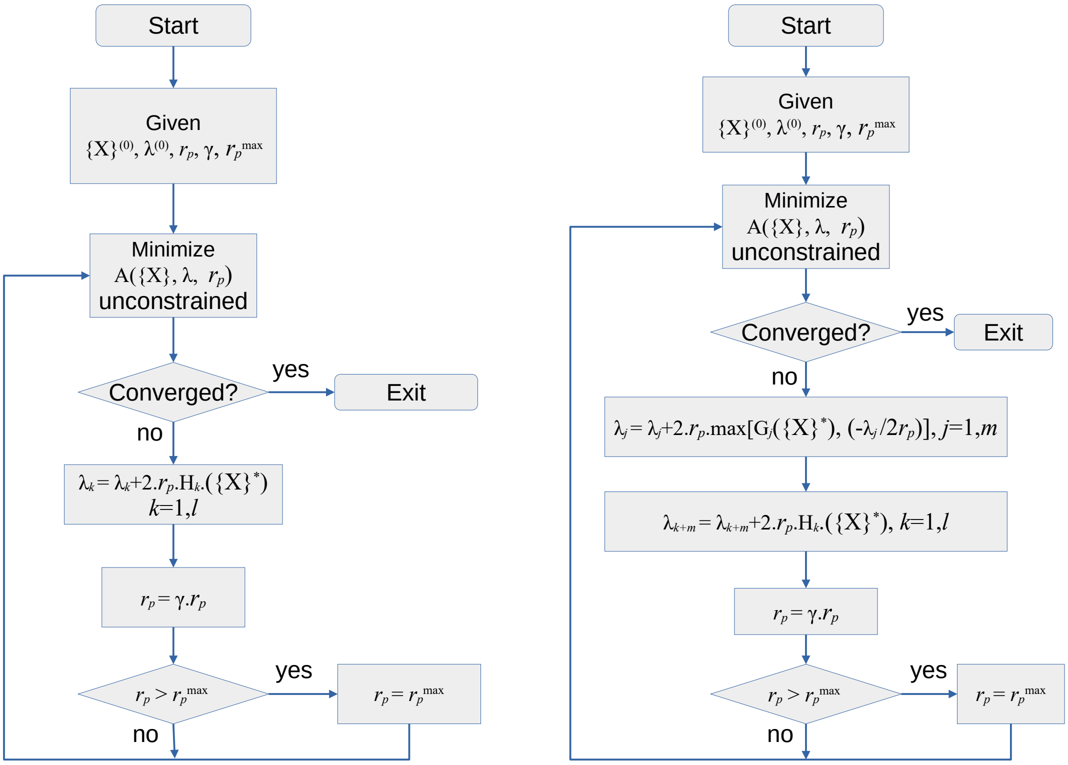
- Relative insensitivity with respect to the value of r_p;
- Precise zero values of both equality and inequality constraints are handled in a numerically stable way;
- Acceleration can be achieved by updating the values of the Lagrange multipliers, as in Equations (100) and (101):

$$\lambda_j^{(i+1)} = \lambda_j^{(i)} + 2r_p \cdot \max\left(G_j(\{X\}), \frac{-\lambda_j^{(i)}}{2r_p}\right), \ j = 1, \ldots, m \tag{100}$$

$$\lambda_{k+m}^{(i+1)} = \lambda_{k+m}^{(i)} + 2r_p \cdot H_k(\{X\}^{(i)}) \tag{101}$$

where k is the number of equality constraints.

Code 5 provides an illustrative implementation of the Augmented Lagrangian Multiplier method (Birgin and Martínez [2008]) in the R statistical computing platform (Johnson [2020]).


Start
Given
$\{X\}^{(0)}, \lambda^{(0)}, r_p, \gamma, r_p^{\max}$
Minimize
$A(\{X\}, \lambda, r_p)$
unconstrained
Converged?
yes
Exit
no
$\lambda_k = \lambda_k + 2.r_p.H_k.(\{X\}^*)$
$k=1,l$
$r_p = \gamma.r_p$
$r_p > r_p^{\max}$
yes
$r_p = r_p^{\max}$
no
Start
Given
$\{X\}^{(0)}, \lambda^{(0)}, r_p, \gamma, r_p^{\max}$
Minimize
$A(\{X\}, \lambda, r_p)$
unconstrained
Converged?
yes
Exit
no
$\lambda_j = \lambda_j + 2.r_p.\max[G_j(\{X\}^*), (-\lambda_j/2r_p)], j=1,m$
$\lambda_{k+m} = \lambda_{k+m} + 2.r_p.H_k.(\{X\}^*), k=1,l$
$r_p = \gamma.r_p$
$r_p > r_p^{\max}$
yes
$r_p = r_p^{\max}$
no

(a) Only equality constraints (b) Equality and inequality constraints

Figure 16: Augmented Lagrangian method for constrained optimization.

9 Direct Methods

The naming for this class of methods reflects their strategy of dealing with the constraints directly, instead of using modified versions of methods initially intended to solve unconstrained problems to tackle constrained ones.

9.1 Sequential Linear Programming (SLP)

This method consists of using a truncated Taylor series around a point X^0 as a linearized version of the general non-linear optimization problem, to be updated at every iteration. Equations (102) to (105) reflect the resulting problem statement, as it stems from the original Equations (1) to (4):

$$\min, \max F(\{X\}) \approx F(\{X^0\}) + \{\nabla F(\{X^0\})\}^T \cdot \delta\{X\} \tag{102}$$

subject to

$$G_j(\{X\}) \approx G_j(\{X^0\}) + \{\nabla G_j(\{X^0\})\}^T \cdot \delta\{X\} \leq 0 \tag{103}$$

$$H_k(\{X\}) \approx H_k(\{X^0\}) + \{\nabla H_k(\{X^0\})\}^T \cdot \delta\{X\} = 0 \tag{104}$$

and side constraints

$$\{X\}^{LB} \leq \{X\} + \delta\{X\} \leq \{X\}^{UB}. \tag{105}$$

Code 5: Augmented Lagrange Multiplier Method.

```
library(nloptr)

fobj <- function(x) {
  return( 100 * (x[2] - x[1] * x[1])^2 + (1 - x[1])^2 )
}
gcon <- function(x) {
  return(x[1] + x[2] - 2.5) # x1 + x2 >= 2.5
}

x0 = c(-2, 2); xl = c(-10, -10); xu = c(10, 10)
opt1 = auglag(x0=x0, fn = fobj, hin = gcon,
  lower = xl, upper = xu)
cat('Optimal␣value:␣', opt1$value,
  '\nOptimal␣design:␣', opt1$par)

Optimal value:  0.02504044
Optimal design:  1.158152 1.341849
```

It should be noted that, although this theoretical construct is such that any linear programming method as described in section 6 could in principle be applied to its solution, practitioners have devised more specific approaches, which rely on several tunning parameters to enhance the overall performance. As a result, SLP is usually a very robust alternative when solving complex engineering problems, even if the attained optimality would vary in a problem dependent fashion.

An example of the SLP method (Nelder and Mead [1965]) implemented in R is provided by Code 6 (Johnson [2020]).

9.2 Method of Feasible Directions and Modified Method of Feasible Directions (MFD and MMFD)

In trying to deal with the actual non-linearity of the optimization problem in an explicit manner, MFD attempts to follow the constraint boundaries, but not in a precise tangential fashion. It does so by leveraging the concepts of usable and feasible search directions $\{S\}$, as defined in Equations (106) and (107), respectively:

$$\{\nabla F(\{X^0\})\}^T \cdot \{S\} \leq 0 \tag{106}$$

$$\{\nabla G(\{X^0\})\}^T \cdot \{S\} \leq 0 \tag{107}$$

Equation (106) means that a small move from $\{X^0\}$ along $\{S\}$ will improve the value of the objective function $F(\{X\})$. Equation (107) means that a small move from $\{X^0\}$ along $\{S\}$ will not violate any constraint $G(\{X\})$, represented here without the sub-indices for brevity and generality.

As in Equation (108), a positive parameter θ is added to Equation (107) to ensure feasibility, that is, the constraint will not be violated despite movements along $\{S\}$

Code 6: Sequential Linear Programming Method.

```
library(nloptr)

fobj <- function(x) {
  return( 100 * (x[2] - x[1] * x[1])^2 + (1 - x[1])^2 )
}
x0 = c(-2, 2); xl = c(-10, -10); xu = c(10, 10)
opt1 = nloptr(x0=x0, eval_f = fobj, lb = xl, ub = xu,
  opts=list('algorithm'='NLOPT_LN_NELDERMEAD',
  'xtol_rel' = 1.0e-6))
cat('Optimal␣value:␣', opt1$objective,
  '\nOptimal␣design:␣', opt1$solution)

Optimal value:  0.003021801
Optimal design:  0.9569033 0.9122515
```

while trying to remain strictly tangent to a function with curvature:

$$\{\nabla G(\{X^0\})\}^T \cdot \{S\} + \theta \leq 0 \tag{108}$$

From vector dot product theory, the geometric interpretation of Equation (108) is that the cosine of the angle between the search direction and the gradient of the constraint function has a strictly negative value. Hence, the search direction is separated from the constraint boundary by an angle always greater than 90 deg, meaning that they are being moved away from each other, fulfilling the feasibility goal (no constraint violation). On the other hand, we can balance this separation motion (between the search direction and the constraint boundary) by prescribing the usability from Equation (106):

$$\{\nabla G(\{X^0\})\}^T \cdot \{S\} - \left(\{\nabla F(\{X^0\})\}^T \cdot \{S\}\right) \cdot \theta \leq 0 \tag{109}$$

Furthermore, the usability condition itself can be simplified, rendering Equation (96), in which the maximization of β will minimize the dot product involving the gradient of the objective function and the search direction, making the search direction further usable, that is, capable of improving the objective (note that the search direction no longer improves the objective when the dot product in Equation (106) crosses into positive territory, and therefore optimality is enhanced by having it minimized further into the negative one).

Combining Equations (109) and (110),

$$\{\nabla F(\{X^0\})\}^T \cdot \{S\} + \beta \leq 0 \tag{110}$$

the feasibility requirement becomes:

$$\{\nabla G(\{X^0\})\}^T \cdot \{S\} + \theta\beta \leq 0. \tag{111}$$

Besides rewriting the usability and feasibility conditions initially expressed in Equations (106) and (107), Equations (110) and (111) also hold as constraints applicable to the following transformation of the optimization problem statement, with Equation (112) subject to (113):

$$\max \beta \tag{112}$$

subject to

$$\{S\}^T \cdot \{S\} \leq 1. \tag{113}$$

The ensemble of Equations (110) to (113) is the method of feasible directions, based on maximizing a parameter β as long as the solution remains feasible, usable, the search direction does not degenerate into a vector whose magnitude explodes into infinity (Equation (113)).

The MMFD method, on its hand, once more rewrites the statement of the optimization problem as follows:

$$\max \left(\{-\nabla F(\{X^0\})\}^T \cdot \{S\}\right) \tag{114}$$

which is still equivalent to minimize Equation (106) in the spirit of usability, subject to Equation (113) and also

$$\{\nabla G_j(\{X^0\})\}^T \cdot \{S\} + \theta \leq 0 \tag{115}$$

which slightly but importantly rewrites Equation (107) to highlight individual constraints, allowing for a special treatment for those that are initially infeasible. Moreover, the equality constraints are also addressed in a specific manner, resulting in the following two additional constraints appended to the objective stated in Equation (114) and the original constraint in Equation (113) to complete the formulation:

$$[A] \cdot \{S\} \leq 0 \tag{116}$$

$$[B] \cdot \{S\} = 0 \tag{117}$$

where $[A]$ contains the gradients of inequality constraints only if they are active (with the safety margin term θ from Equation (109) dropped), and $[B]$ contains the gradients of the equality constraints.

9.3 Sequential Quadratic Programming (SQP)

As an extension of the linear approximation introduced via Equations (102) to (105), the SQP method will define a search direction creating a quadratic approximation for the augmented objective function and a linear approximation to the problem constraints, such that:

$$\min \left(F(\{X\}) + \{\nabla F(\{X\})\}^T \cdot \{S\} + \frac{1}{2} \cdot \{S\}^T \cdot [H_L] \cdot \{S\}\right) \tag{118}$$

subject to

$$\{\nabla G_j(\{X\})\}^T \cdot \{S\} + \delta_j \cdot G_j(\{X\}) \leq 0 \tag{119}$$

$$\{\nabla H_k(\{X\})\}^T \cdot \{S\} + \tilde{\delta} \cdot H_k(\{X\}) = 0 \tag{120}$$

Code 7: Sequential Quadratic Programming Method.

```
library(nloptr)

fobj <- function(x) {
  return( 100 * (x[2] - x[1] * x[1])^2 + (1 - x[1])^2 )
}
gfobj <- function(x) {
  return( c( -400 * x[1] * (x[2] - x[1] * x[1])
    - 2 * (1 - x[1]), 200 * (x[2] - x[1] * x[1])) )
}

x0 = c(-2, 2); xl = c(-10, -10); xu = c(10, 10)
opt1 = nloptr(x0=x0, eval_f = fobj, eval_grad_f = gfobj,
  lb = xl, ub = xu,
  opts=list('algorithm'='NLOPT_LD_SLSQP',
  'xtol_rel' = 1.0e-6))
cat('Optimal␣value:␣', opt1$objective,
  '\nOptimal␣design:␣', opt1$solution)

Optimal value: 1.174518e-19
Optimal design: 1 1
```

where H_L starts as an identity matrix and gradually converges to the Hessian of the Lagrangian, which is directly derived from the KKT conditions of Equations (10)-(12) (expressed below in Equation (121)), with parameters δ_j and $\tilde{\delta}$ providing the other angle of the update process throughout the method iterations, as per the rules in Equations (122)-(124):

$$\nabla F(\{X\}) + \sum_j \lambda_j \cdot \nabla G_j(\{X\}) + \sum_k \lambda_k \cdot \nabla H_k(\{X\}) \tag{121}$$

$$\delta_j = 1, \quad \text{if } G_j(\{X\}) < 0 \tag{122}$$

$$\delta_j = \tilde{\delta}, \quad \text{if } G_j(\{X\}) \geq 0 \tag{123}$$

$$0 \leq \tilde{\delta} \leq 1, \quad \text{(typically, towards the upper bound).} \tag{124}$$

An example of the SQP method (Kraft [1988]) implemented in R is provided by Code 7 (Johnson [2020]).

This method will provide a compromise about efficiency in numerical computation (typical of linear methods) and precision in evaluation (typical of nonlinear methods). A number of nonlinear programming methods will rely on SQP as internal engine to solve nonlinear problems with reduced computational cost.

References

H. Agarwal and J. Renaud. Reliability based design optimization using response surfaces in application to multidisciplinary systems. *Engineering Optimization*, 36(3):291–311, 2004.

N. Alexandrov and S. Kodiyalam. Initial results of an MDO method evaluation study. In *Proceedings of the 7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, September 1998.

T. Altiok and B. Melamed. *Simulation modeling and analysis with Arena.* Elsevier, 2010.

E. G. Birgin and J. M. Martínez. Improving ultimate convergence of an augmented lagrangian method. *Optimization Methods and Software*, 23(2):177–195, 2008.

S. Butenko and P. M. Pardalos. *Numerical Methods and Optimization: An Introduction.* Chapman & Hall/CRC Numerical Analysis and Scientific Computing Series, 2014.

S. Butkewitsch and V. Steffen, Jr. A case study in frequency response optimization. In *DINAME 99 - Applied Mechanics in the Americas*, volume 8, 1999.

R. A. Canfield. Design of frames against buckling using a rayleigh quotient approximation. *AIAA journal*, 31(6):1143–1149, 1993.

R. S. Dembo and T. Steihaug. Truncated newton algorithms for large-scale optimization. *Math. Programming*, 26:190–212, 1983. doi: http://doi.org/10.1007/BF02592055.

X. Gandibleux. Multiple criteria optimization: state of the art annotated bibliographic surveys. 2006.

A. H. Gandomi, X.-S. Yang, S. Talatahari, and A. H. Alavi. *Metaheuristic applications in structures and infrastructures.* Elsevier, 2013.

A. A. Giunta, V. Balabanov, D. Haim, B. Grossman, W. H. Mason, L. T. Watson, and R. T. Haftka. Aircraft multidisciplinary design optimisation using design of experiments theory and response surface modelling. *Aeronautical Journal*, 101 (1008):347–356, 1997.

X. Gu, J. E. Renaud, S. M. Batill, R. M. Brach, and A. S. Budhiraja. Worst case propagated uncertainty of multidisciplinary systems in robust design optimization. *Structural and Multidisciplinary Optimization*, 20(3):190–213, 2000.

A. Haldar and S. Mahadevan. *Probability, Reliability, and Statistical Methods in Engineering Design.* Wiley, 1999.

S. G. Johnson. The NLopt nonlinear-optimization package. 2020. URL `https://cran.r-project.org/package=nloptr`.

D. Kraft. A software package for sequential quadratic programming. Technical report, Technical Report DFVLR-FB 88-28, Institut für Dynamik der Flugsysteme, Oberpfaffenhofen, July 1988.

D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Math. Programming*, 45:503–528, 1989.

D. G. Luenberger and Y. Ye. *Linear and Nonlinear Programming.* Springer, 3rd edition, 2008.

L. Nardin, K. Sørensen, S. Hitzel, and U. Tremel. modeFRONTIER, a framework for the optimization of military aircraft configurations. In *MEGADESIGN and MegaOpt-German Initiatives for Aerodynamic Simulation and Optimization in Aircraft Design*, pages 191–205. Springer, 2009.

J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7:308–313, 1965.

R. D. Neufville, O. D. Weck, D. Frey, D. Hastings, R. Larson, D. Simchi-Levi, K. Oye, A. Weigel, and R. Welsch. Uncertainty management for engineering systems planning and design. In *Proceedings of the Engineering Systems Symposium, MIT*, March 2004.

B. Noble and J. W. Daniel. *Applied Linear Algebra.* Prentice-Hall, 1988.

G. O. Odu and O. E. Charles-Owaba. Review of multi-criteria optimization methods–theory and applications. *IOSR Journal of Engineering (IOSRJEN)*, 3 (10):1–14, 2013.

M. J. Powell. The BOBYQA algorithm for bound constrained optimization without derivatives. *Cambridge NA Report NA2009/06, University of Cambridge, Cambridge*, pages 26–46, 2009.

R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2020. URL `https://www.R-project.org/`.

S. A. Ragon, Z. Gürdal, R. T. Haftka, and T. J. Tzong. Bilevel design of a wing structure using response surfaces. *Journal of Aircraft*, 40(5):985–992, 2003.

E. Rashedi, E. Rashedi, and H. Nezamabadi-pour. A comprehensive survey on gravitational search algorithm. *Swarm and evolutionary computation*, 41:141–158, 2018.

D. T. Sturrock and C. D. Pegden. Recent innovations in Simio. In *Proceedings of the 2011 Winter Simulation Conference (WSC)*, pages 52–62. IEEE, 2011.

S. Theussl and K. Hornik. *Rglpk: R/GNU Linear Programming Kit Interface*, 2019. URL `https://CRAN.R-project.org/package=Rglpk`. R package version 0.6-4.

I. Ucar, B. Smeets, and A. Azcorra. Simmer: Discrete-event simulation for R. *arXiv preprint arXiv:1705.09746*, 2017.

G. N. Vanderplaats. *Numerical Optimization Techniques for Engineering Design*. Vanderplaats Research and Development, 1998.

G. Venter and R. T. Haftka. Using response surface approximations in fuzzy set based design optimization. *Structural Optimization*, 18(4):218–227, 1999.

A. Waller. Witness simulation software. In *Proceedings of the Winter Simulation Conference*, pages 1–12, 2012.

B. Wilson, D. Cappelleri, T. W. Simpson, and M. Frecker. Efficient pareto frontier exploration using surrogate approximations. *Optimization and Engineering*, 2(1): 31–50, 2001.

Chapter 4: Overview of Traditional and Recent Heuristic Optimization Methods

Chapter details

Chapter DOI:

https://doi.org/10.4322/978-65-86503-71-5.c04

Chapter suggested citation / reference style:

Choze, Sergio B., et al. (2022). “Overview of Traditional and Recent Heuristic Optimization Methods”. *In* Jorge, Ariosto B., et al. (Eds.) *Model-Based and Signal-Based Inverse Methods*, Vol. I, UnB, Brasilia, DF, Brazil, pp. 107–142. Book series in Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity.

P.S.: DOI may be included at the end of citation, for completeness.

Book details

Book: Model-based and Signal-Based Inverse Methods

Edited by: Jorge, Ariosto B., Anflor, Carla T. M., Gomes, Guilherme F., & Carneiro, Sergio H. S.

Volume I of Book Series in:

Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity

Published by: UnB City: Brasilia, DF, Brazil Year: 2022

DOI: https://doi.org/10.4322/978-65-86503-71-5

Overview of Traditional and Recent Heuristic Optimization Methods

Sergio B. Choze[1*]   Rogerio R. Santos[2]   Guilherme F. Gomes[3]

[1]Consultant. E-mail: sergio.butkewitsch@gmail.com
[2]Division of Mechanical Engineering, Technological Institute of Aeronautics - ITA, Brazil. E-mail: rsantos9@gmail.com
[3]Mechanical Engineering Institute, Federal University of Itajuba - UNIFEI, Brazil. E-mail: guilhermefergom@unifei.edu.br

[*]Corresponding author

Abstract

This chapter describes theoretical and practical aspects of Heuristic Optimization Methods. The introductory section describes the basic equations commonly found in numerical optimization and enumerates some open questions in the field. Next, general principles are outlined, and an overview of some heuristic optimization methods is presented, followed by considerations about the main aspects of numerical computations. The rest of the text is devoted to the clarification of method-specific variations and novel methods. Algorithms and fragments of source code are showed to enrich the discussion about the methodologies. Some practical considerations for applied optimization are the concluding remarks.

1 Introduction

Heuristic optimization methods are meant to handle special conditions of the same fundamental mathematical optimization problem introduced in the Chapter "Overview of Linear and Non-linear Programming Methods for Structural Optimization", as defined in Equations (1) - (4) below. This means that while the problem statement or formulation remains unchanged, the solution methods no longer rely on derivatives, either because 1) they are not available at all, or 2) because they are not effective to determine a search path leading to a satisfactory solution.

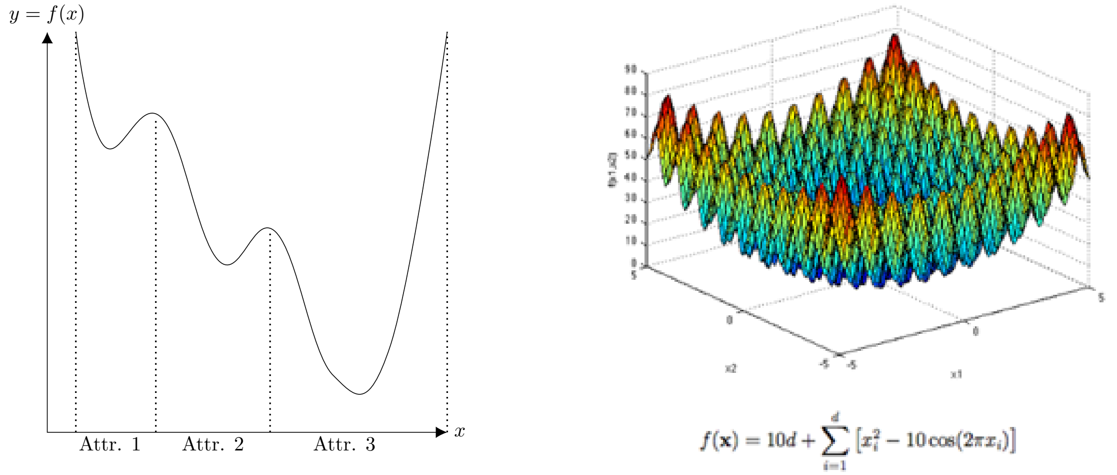
$$\min, \max F(\mathbf{x}) \tag{1}$$

subject to:

$$G_j(\mathbf{x}) \leq 0 \tag{2}$$

$$H_k(\mathbf{x}) = 0 \tag{3}$$

$$\mathbf{x}_{LB} \leq \mathbf{x} \leq \mathbf{x}_{UB} \tag{4}$$



(a) Three basins of attraction seen across one dimension.

(b) 2-dimensional case of the Rastrigin function, showing multiple basins of attraction.

Figure 1: Function with multiple basins of attraction.

Situation 1 is prone to arise in various circumstances, even when the optimization process is driven by actual physical experimentations that cannot be undertaken, but in the context of computational mechanics are usually associated with non-linear and discontinuous analyses in which convergence of a certain simulation scenario is not attainable, and hence information for calculation of derivatives is missing.

More often though, still in the realm of highly complex and non-linear search spaces, situation 2 happens when derivative based methods may be uncapable of navigating throughout the decision space at large, being confined to a narrow search domain which does not necessarily contains the best possible decision, which at this point is unknown or undetermined. The narrow sections of the overall decision space are called basins of attraction, which may contain local or global optima, as represented in Figure 1. Panel (a) sketches a simplified (for illustration purposes) one-dimensional function (or a one-dimensional cut of a multivariate function) whereby 3 adjacent basins of attraction are identified. Panel (b) shows the 2-dimensional case of the Rastrigin function [1], a common test function for the performance of the methods to be covered in this chapter. The functions have multiple basins of attraction: (a) schematic decision space with basins of attraction 1, 2, and 3, with the latter one containing a global optimum (function minimum) concerning the other two and (b) 2-dimensional Rastrigin function, with multiple basins of attraction and known analytical global optimum at (0, 0).

Many problems in engineering can be formulated as optimization problems, subject to complex nonlinear objective functions and constraints. As already stated by Yang (2020), the solutions of highly nonlinear problems usually require sophisticated optimization algorithms, and traditional algorithms may struggle to deal with such problems.

Many real-world engineering applications involve the optimization of certain objectives such as the minimization of costs, energy consumption, environment and the maximization of performance, efficiency and sustainability (Yang [2020]). In many cases, the optimization problems that can be formulated are highly nonlinear with multimodal objective landscapes, subject to a set of complex, nonlinear constraints. Such problems are challenging to solve. Even with the ever-increasing power of modern computers, it is still impractical and not desirable to use simple brute force approaches. Thus, whenever possible, efficient algorithms are crucially important to such applications. However, efficient algorithms may not exist for most of the optimization problems in applications. Though there are a wide range of optimization algorithms such as gradient-based algorithms, the interior-point method and trust-region method, most of such algorithms are gradient-based and local search algorithms, which means that the final solutions may depend on the initial starting points. In addition, the computation of derivatives can be computationally expensive, and some problems such as the objective with discontinuities may not have derivatives in certain regions.

Despite the effectiveness of nature-inspired algorithms and their popularity, there are still many challenging issues concerning such algorithms, especially from theoretical perspectives.

The paper Yang [2020] highlights five main challenging issues that justify the use of advanced metaheuristic techniques. However, much remains to be explored in this field of research. The following stand out:

- Open Problem 1: How to build a unified framework for analyzing all nature-inspired algorithms mathematically, so as to obtain in-depth information about their convergence, rate of convergence, stability, and robustness?
- Open Problem 2: How to best tune the parameters of a given algorithm so that it can achieve its best performance for a given set of problems? How to vary or control these parameters so as to maximize the performance of an algorithm?
- Open Problem 3: What types of benchmarking are useful? Do free lunches exist, under what conditions?
- Open Problem 4: What are the most suitable performance metrics for fairly comparing all algorithms? Is it possible to design a unified framework to compare all algorithms fairly and rigorously?
- Open Problem 5: How to best scale up the algorithms that work well for small-scale problems to solve truly large-scale, real-world problems efficiently?

The authors of Yang [2020] concludes that there are other open problems concerning nature-inspired algorithms, including how to achieve the optimal balance of exploitation and exploration, how to deal with nonlinear constraints effectively, and how to use these algorithms for machine learning and deep learning. Nature-inspired computation is an active area of research. It is hoped that the above five open problems we have just highlighted can inspire more research in this area in the near future.

As indicated by the nomenclature, basins of attraction will draw the search method towards themselves, which is neither harmful nor advantageous per se, depending primarily on: 1) how sooner or later the attraction happens in the search process and 2) how much decision making elements are available to ascertain that the optimum contained in a basin of attraction is the best decision, given a search cost requirement and 3) what are the resources available to afford a larger or smaller search effort, without explicit guarantee that a better solution is available at a different basin of attraction.

Despite of these considerations, a common feature of all basins of attraction is that they tend to contain derivative values of zero for functions that represent the decision spaces where the search is being performed, which means the stalling of classical, derivative driven optimization methods. In optimization jargon, this phenomenon is known as “being trapped at a local optimum” , and the goal of this chapter is to discuss the so-called heuristic optimization methods, which are meant to avoid this phenomenon by being more flexible than strict derivatives to balance exploration (mapping of a broader decision space, identifying as many as possible basins of attraction) with exploitation (deciding upon the best solution to be exploited, conditioned on the knowledge of the decision space accrued thus far by exploration and the budget to further the search). This flexibility is achieved by replacing the point-to-point search strategies characteristic of derivative driven methods by population-based approaches, which signify an additional cost to perform the search.

2 General principles

Heuristic optimization methods depart from multiple starting points, on the vein of balancing the odds across multiple potential (and often unknown, at least a priori) basins of attraction. Hence, they are from the onset population based, and balance should be achieved between how representative the population is (typically an artifact of its size, barring any biases) and how feasible it is to handle the computational cost of evaluating larger populations as they grow.

Once the multiple starting points are determined, instead of proceeding with derivative based searches for each of every of them (the multi-start concept in classical optimization (De Jong [2016])), which would ensue a totally deterministic operation of the methods thereafter, aleatory components are actually introduced to further balance the odds with respect to finding or avoiding the possible basins of attraction. On this vein, the values of the decision variables are randomly perturbed according to an approach particular to each method, which also rank the resulting perturbed decision points (candidate optima) based on their own particular criteria. Rather than continuing to randomly perturb the values of the decision variables, the design candidates arising from these perturbations are somehow combined or mixed (again, according to each method?s particularities) so that the effects of randomness apply population-wise, rather than just at each individual setup. The combined candidate designs are then appraised and, for the most part, those with superior performance will comprise a larger fraction of the candidates pool for the next iteration, with a minority of relatively underperforming individuals still kept for diversity of the overall population and feasibility of reaching to alternative basins of attraction. After multiple repetitions of this sample → perturb → combine → appraise → resample sequence, it is expected that deterministic and aleatory search components are ideally balanced and, between the extremes of classical and totally random searches, the best tradeoff has been met and the most suitable design, within the best relative basin of attraction, has been achieved at the minimum computational cost.

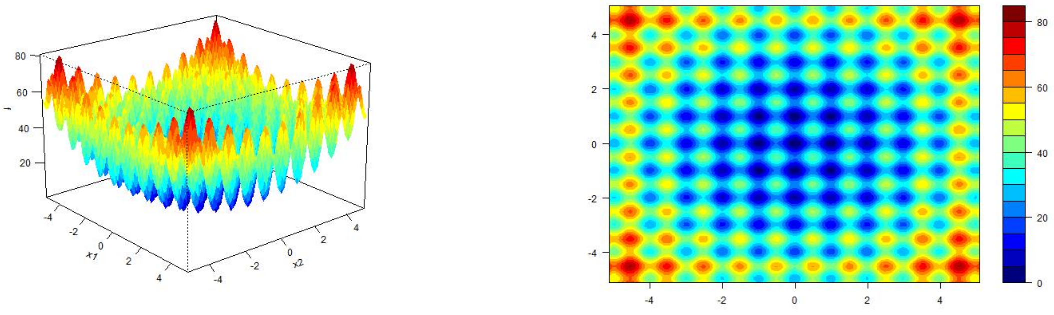
It should be noted that, along these lines, no formal convergence criteria similar to the Karush-Kuhn-Tucker (KKT) conditions apply, and the most adequate nomenclature is actually search termination, either because the allotted budget has been exhausted and/or the balance between random and deterministic search elements is such that no plausible improvement can be achieved. In order to strike this best balance even in the absence of formal criteria, intense research is devoted to the understanding and characterization of heuristic search methods as random processes, that may be modeled according to one of the canonical forms or combinations thereof available in this field of knowledge.

Lastly, since the aleatory-deterministic search balance effects one given function (typically the optimization objective expressed in Equation (1)), it is common not to account explicitly for constraint functions (equality or inequality, as in Equations (2) and (3)) for the execution of heuristic optimization methods. Because the decision making is not practical without any form of consideration of the constraints, they are either verified offline (simplistic approach) or embedded into the objective by way of a penalty (similarly to the constrained optimization methods detailed in Chapter "Overview of Linear and Non-linear Programming Methods for Structural Optimization", section 8) or, in the most sophisticated approaches, the problem statement is completely reconfigured as a multicriteria (Chapter "Overview of Linear and Non-linear Programming Methods for Structural Optimization", section 3) or multidisciplinary (Chapter "Overview of Linear and Non-linear Programming Methods for Structural Optimization", section 4) procedure.

Table 1 summarizes how the general principles outlined above are handled by each of the established search methods. Since these common principles are shared, the methods are inevitably similar to each other in many aspects, with the different approaches and respective nomenclatures arising more as an artifact of successful practice rather than formal mathematical features. The forthcoming sections are dedicated to explore each of the methods listed in Table 1 in greater detail.

Table 1: Overview of heuristic optimization methods according to the elements they use to balance aleatory and deterministic search components in pursuit of (potentially) global optima.

Method	Individual Unit	Population Unit	Iteration Unit	Operators	Comments
Genetic Algorithm	chromosomes, as a collection of genes (codified decision variables)	Population comprised of a determined number of individuals, each one with its chromosome	Generations, which get updated by successive application of the genetic operators	Selection, Crossover, Mutation	Basis (both historically and algorithmically) for most of the other heuristic methods. Multiple variations exist, but the core consists of applying the genetic operators in the order they appear in the previous column
Evolutionary Program	chromosomes, as a collection of genes (codified decision variables)	Population comprised of a determined number of individuals, each one with its chromosome	Generations, which get updated by successive application of the genetic operators	Mostly random mutations, at various categories and intensities	Similar to GAs, emphasizing the randomized element of the search through various modalities of mutations
Simulated Annealing	Candidate designs, as cohesive sets of decision variables	Sets of candidate designs	Annealing cycle	Heating, Cooling	
Ant Colony	Path traveled by individual ants	Path traveled by all ants	Excursions from and back to the ant colony, through a random path	Pheromone deposition and evaporation	
Particle Swarm	Particles	Swarm of many particles	Particle's Position and Velocity variations	Inertia, individual best and social best	
Differential Evolution	chromosomes, as a collection of genes (codified decision variables)	Population comprised of a determined number of individuals, each one with its chromosome	Generations, which get updated by successive application of the genetic operators	Mutation, Selection, Crossover	Same basic operators as GAs, but with mutation taking place earlier at each perturbation cycle



(a)

(b)

Figure 2: The (a) Rastrigin test function and (b) the contour plot.

3 Numerical Computation

As outlined in Equation (1), the maximization and minimization of functions are similar from the computational perspective. Several algorithms are implemented aiming minimization. However, it is worth to note the difference in computational implementation that will switch between max and min. This feature will be used on snippet of computational codes along the text, implemented within the R statistical computation platform (R Core Team [2020]).

For this purpose, the reader may please refer to the command `install.packages` as the main tool to add new packages to the R software. It takes a vector of names and a destination library, downloads the packages from the repositories (from a local folder or remote web server) and installs them in the R environment. To install the packages needed to run the codes in this chapter, for example, enter the following

```
install.packages('GA', 'ecr', 'DEoptim', 'GenSA', 'pso',
  'metaheuristicOpt')
```

Help is available at the command prompt via `?install.packages` command.

As an example, consider the Rastrigin function, which is provided as test function aiming minimization. The global minimum value is 0 at the design $x = (0, .., 0)$. It is shown in Figure 2.

A two-dimensional implementation for minimization algorithms is given in Code 1.

Code 1: R implementation defining the 2D version of the Rastrigin function as the unconstrained objective to be minimized by various heuristic optimization methods.

```
obj2min <- function(x)
{
  obj=20+x[1]^2+x[2]^2-10*(cos(2*pi*x[1])+cos(2*pi*x[2]))
  return (obj)
}
```

On the other hand, a two-dimensional implementation aiming *maximization* algorithms is shown in Code 2.

Code 2: R implementation defining the 2D version of the Rastrigin function as the unconstrained objective to be maximized by various heuristic optimization methods.

```
obj2max <- function(x)
{
  obj=20+x[1]^2+x[2]^2-10*(cos(2*pi*x[1])+cos(2*pi*x[2]))
  return (-obj)
}
```

That is, the objective `obj2max` evaluated by a maximization algorithm will lead to the same result as the objective `obj2min` evaluated by the minimization algorithm. As a result, the multiplication of the objective function by -1 is sufficient to translate an unconstrained optimization problem between the minimization and maximization perspectives.

Following this template, the corresponding code snippets are offered the methods described throughout sections 4.1 to 4.6 below, as well as sections 5.3 and 5.4. These exemplary codes are offered whenever actual R packages (basic R topic modules) are available at the time of writing and able to offer implementations for each of the methods alluded within the aforementioned sections (group 4 for the more established techniques, and group 5 for the more recent developments).

The reader should notice minor variations in formatting and structure, due to particularities of each package implementing the applicable functions. At the same time, in an effort to create a multipurpose heuristic methods engine within the R statistical computing platform, Riza et al. [2019] proposed the `metaheuristicOpt` package, in which any of the specific algorithms listed (in alphabetical order) within Code 1 can be instantiated as determined by the parameter "`algorithm`" (Table 2) illustrated at the associated code fragment. The length of the list indicates the level of research activity in the field, as well as the commonality of nature inspired analogies underpinning the proposition of different heuristic methods that share several common principles. While the entire field is evolving, several novel methods introduce additional tuning parameters that may require a significant level of experimentation for fine tuning, and hence the most complex option is not necessarily the one yielding superior performance.

4 Method specific variations

4.1 Genetic Algorithms

As the very foundational of heuristic optimization methods, Genetic Algorithms balance random and deterministic decision-making efforts by way of the following sequence of operations:

- An encoding method, or schema, is adopted to codify combinations of decision variables in a string (per the biological analogy, a chromosome) that lead to

Code 3: Code fragment for minimization of the Rastrigin function through the metaheuristicOpt R package.

```
# Input #
obj2min <- function(x)
{
  return( 20 + x[1]^2 + x[2]^2 - 10 * (cos(2*pi*x[1]) +
    cos(2*pi*x[2]) ) )
}
lower_bound = c(-7, -7); upper_bound = c(7, 7)

library('metaheuristicOpt')
BA_ans = metaOpt(obj2min, optimType='MIN', algorithm='BA',
  numVar = 2, rangeVar = rbind(lower_bound, upper_bound),
  control = list(numPopulation = 20, maxIter = 700) )
cat('\nOptimal␣value␣', BA_ans$optimumValue,
  '\nOptimal␣design␣', BA_ans$result)

# Output #
Optimal value  0
Optimal design  3.238586e-292 1.289825e-29
```

Table 2: Collection of heuristic optimization methods available through the metaheuristicOpt R package.

Algorithm Name/Option	Description of the `algorithm` parameter
ABC	Artificial Bee Colony Algorithm
ALO	Ant Lion Optimizer
BA	Bat Algorithm
BHO	Black Hole Based Optimization Algorithm
CLONALG	Clonal Selection Algorithm
CS	Cuckoo Search Algorithm
CSO	Cat Swarm Optimization Algorithm
DA	Dragonfly Algorithm
DE	Differential Evolution Algorithm
FFA	Firefly Algorithm
GA	Genetic Algorithm
GBS	Gravitation Based Search Algorithm
GOA	Grasshopper Optimization Algorithm
GWO	Grey Wolf Optimizer
HS	Harmony Search Algorithm
KH	Krill Herd Algorithm
MFO	Moth Flame Optimizer
PSO	Particle Swarm Optimization
SCA	Sine Cosine Algorithm
SFL	Shuffled Frog Leaping Algorithm
WOA	Whale Optimization Algorithm

the corresponding values of responses and ensuing fitness. In other words, a standard indexed stream of information, like a vector or list, becomes the genetic identity of each decision. More often, this encoding is binary, but representation via real numbers is also possible in GA and mainstream in DE;

- A random sample of individuals (population) is created for initialization. Therefore, sample size in itself is a configuration parameter important to balance random diversity and computational effort;
- Individuals within the initial population, and also thereafter, are ranked according to a scaling function intended to systematically gauge their relative fitness;
- A fitness-based selection operation will sub-sample from the original population to propagate features belonging to the fittest individuals throughout the subsequent steps;
- Crossover of the selected individuals, generating new ones (subsequent generation) that combine features associated with the superior fitness driving the selection step;
- Mutation (aleatory changes into randomly chosen features of random individuals) to increase the stochastic element of the overall decision-making process;
- Check of termination criteria, which are not formal convergence metrics but rather a heuristic verification of improvement vis-Ã -vis the ability to afford additional computational effort. At this point, the cycle is either finalized or a new generation starts back into the selection step and onwards.

Algorithm 1 summarizes the outline above in a computer centered manner, with Figure 3 being its flowchart counterpart.

Algorithm 1: Genetic Algorithm.

```
Generate initial population
while termination criteria not satisfied do
    Calculate the fitness of each element in population
    for n steps do
        Use fitness to select a pair of elements for the new generation
        Create new elements $e_1$ and $e_2$
    end
    Randomly mutate some elements
    Evaluate fitness of new elements
    New population ← elements with best fitness among new elements and
      current population
end
```

Code 4 implements a genetic algorithm function call in R (Scrucca [2013]) to maximize the negative of the 2-dimensional Rastrigin function available through

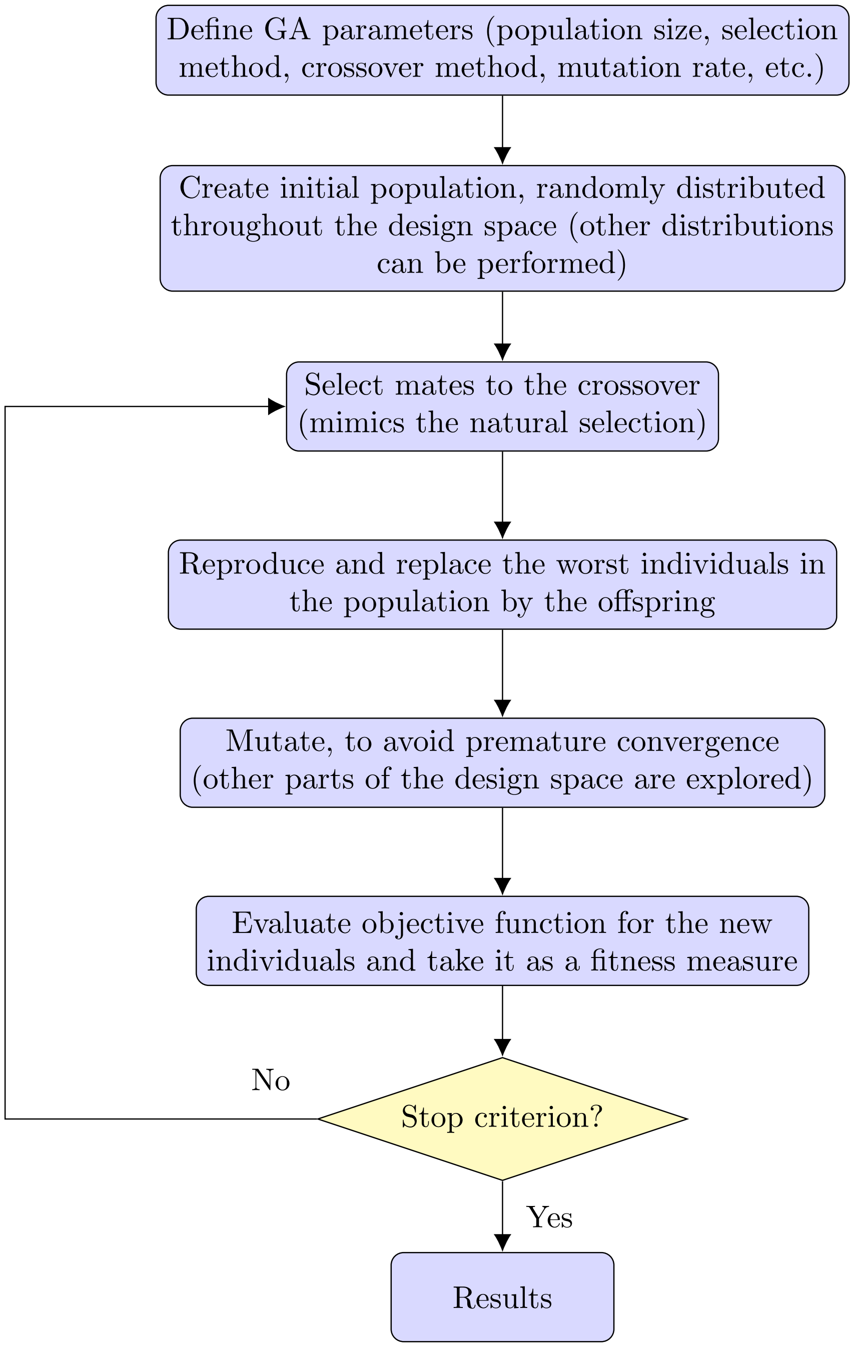

Define GA parameters (population size, selection method, crossover method, mutation rate, etc.)
Create initial population, randomly distributed throughout the design space (other distributions can be performed)
Select mates to the crossover (mimics the natural selection)
Reproduce and replace the worst individuals in the population by the offspring
Mutate, to avoid premature convergence (other parts of the design space are explored)
Evaluate objective function for the new individuals and take it as a fitness measure
No
Stop criterion?
Yes
Results

Figure 3: Genetic Algorithm flowchart.

Code 4: R implementation calling Genetic Algorithms to maximize the 2D version of the Rastrigin function as an unconstrained objective.

```
# Input #
obj2max <- function(x)
{
  obj = 20 + x[1]^2 + x[2]^2 - 10 *
  ( cos(2*pi * x[1]) + cos( 2*pi * x[2] ) )
  return (-obj)
}
lower_bound = c(-7, -7); upper_bound = c(7, 7)

library('GA')
GA_ans <- ga(type = "real-valued", fitness = obj2max,
  lower = lower_bound, upper = upper_bound, popSize = 20,
  maxiter = 300)
cat('\nOptimal␣value␣', GA_ans@fitnessValue,
  '\nOptimal␣design␣', GA_ans@solution)

# Output #
Optimal value  -2.1734e-06
Optimal design  8.811057e-05 5.64903e-05
```

Code 2. The results of one function call are summarized in Figure 4, and are expected to vary from run to run due to the underlying stochastic nature of heuristic optimization methods (if affordable, a reasonable amount of repetitions should establish a stable enough trend for each application in context).

Figure 4 further illustrates the populational nature of the method, with a distribution of results at each iteration (generation) of the search. Within these distributions, some individuals are fitter than others, and statistical summaries such as the mean and the median are useful to characterize how the fitness of the entire population converges towards the intended maximum.

4.2 Evolutionary Program

Another heuristic optimization method that operates similarly to Gas, Evolutionary Program (EP) relies on the behavioral linkage between parents and their offspring, as outlined in these 3 fundamental steps (De Jong [2016]).

- Randomly initialize the initial population, whose size strongly influences the speed of optimization. The caveat is the absence of definite rules, which means that practical applications shall require some level of experimentation for fine tuning the nest tradeoff between speed and maximum coverage of loci which might contain the global optimum;
- A second generation clones the initial one, such that a spectrum of various mutation intensities is applied over the offspring solutions, according to a

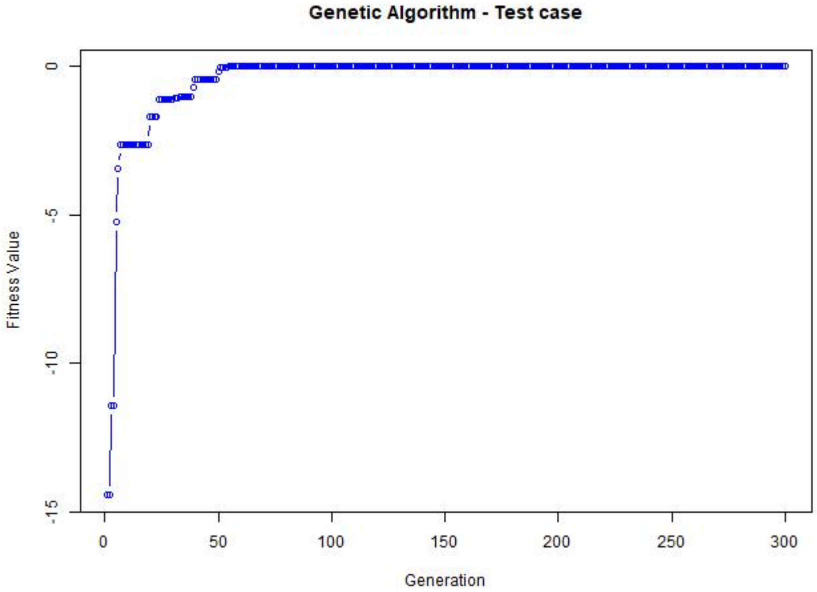


Figure 4: Graphic Output highlighting maximization of the objective along iterations.

distribution of mutation types, according to how strongly they affect the fitness of the parents (initial population);

- An arbitrarily chosen number of solutions (keeping or varying the population size determined in the initialization) within a fitness rank is for the next generation.

Algorithm 2 formalizes these steps, while Code 5 contains both the input and output to the corresponding routine in R (Bossek [2017]), in text format.

Algorithm 2: Evolutionary Programming.

```
Given the parameters $a$ and $b$
Generate initial population
while termination criteria not satisfied do
    Calculate the fitness of each element in population
    for each individual do
        Generate a random vector $n_i \in$ Gaussian$(0, 1)$
        New $x_i \leftarrow x_i + n_i * \sqrt{a * f(x_i) + b}$
    end
    Evaluate fitness of new elements
    New population $\leftarrow$ elements with best fitness among new elements and
      current population
end
```

A striking feature of the `ecr` library (Bossek [2017]) is the ability to easily incorporate user-defined functions for fitness evaluation, mutation strategies, and stop-

Code 5: R implementation calling Evolutionary Programming Optimization to minimize the 2D version of the Rastrigin function as an unconstrained objective.

```
# Input #
obj2min <- function(x)
{
  return( 20 + x[1]^2 + x[2]^2 - 10 *
    ( cos(2*pi*x[1]) + cos(2* pi * x[2]) ) )
}
lower_bound = c(-7, -7); upper_bound = c(7, 7)

library(’ecr’)
EC_ans = ecr(fitness.fun = obj2min,
  representation = ’float’,
  n.dim = 2, n.objectives = 1, survival.strategy = ’plus’,
  lower = lower_bound, upper = upper_bound, mu = 20,
  lambda = 10, mutator = setup(mutGauss, sdev = 2,
  lower = lower_bound, upper = upper_bound),
  terminators = list(stopOnIters(300)))
cat(’\nOptimal␣value␣’, EC_ans$best.y,
  ’\nOptimal␣design␣’, EC_ans$best.x[[1]][1:2])

# Output #
Optimal value  1.059243
Optimal design  0.01795939 0.9936067
```

ping criteria, among others. Therefore, this library would be an adequate platform for development and evaluation of new evolutionary strategies.

4.3 Simulated Annealing

Annealing is a term from metallurgy used to describe a process in which a metal is heated to a high temperature, inducing strong perturbations to its atom?s positions. Providing that the temperature drop is slow enough, the metal will eventually stabilize into an orderly structure and, otherwise, an unstable atom structure arises.

Simulated annealing can be performed in design optimization by randomly perturbing the decision variables and keeping track of the best resulting objective value. After many tries, the most successful design is set to be the center about which a new set of perturbations will take place. In an analogy to the metallurgical annealing process, let each atomic state (design variable configurations) result in an energy level (objective function value) E. In each step of the algorithm, the atoms positions are given small random displacements due to the effect of a prescribed temperature T (standard deviation of the random number generator). As an effect, the energy level undergoes a change ΔE (variation of the objective function value). If $\Delta E \leq 0$, the objective stays the same or is minimized, thus the displacement is accepted, and the resulting configuration is adopted as the starting point of the next step. If $\Delta E > 0$, on the other hand, the probability that the new configuration is accepted is given by Equation (5):

$$P(\Delta E) = e^{\Delta E/k_b T} \tag{5}$$

where k_b is the Boltzman constant, set equal to 1. Since the probability distribution in Equation (5) is chosen, the system evolves into a Boltzman distribution. The random numbers r are obtained according to a uniform probability density function in the interval $(0, 1)$. If $r < P(\Delta E)$ the new configuration is retained. Else, the original configuration is used to start the next step.

The temperature T is simply a control parameter in the same units as the objective function. The initial value of T is related to the standard deviation of the random number generator, whilst its final value indicates the order of magnitude of the desired accuracy in the location of the optimum point. Thus, the annealing schedule starts at a high temperature which is discretely lowered (using a factor $0 < rt < 1$) until the system is “frozen”, hopefully at the optimum, even if the design space is multimodal.

Similarly to previous sections, Algorithm 3 reflects the computational aspects of this procedure (Xiang et al. [2013]), actually implemented as in Code 6 to create the numeric and graphic outputs shown in panels (a) and (b) of Figure 5, respectively.

4.4 Ant Colony Optimization

As directly alluded to by its name, the natural analogy underpinning this method mirrors how ants move between their colonies and sources of food (Blum [2005]). Major components of this process are a) ants should minimize the total path travelled to gather food into their colony; b) ants may not be able to carry all the food

Algorithm 3: Simulated Annealing.

```
Set initial temperature $T > 0$
Cooling function: $C(T) \in [0, T]$
Generate initial population
$x_0 \leftarrow$ optimal design
**while** *termination criteria not satisfied* **do**
    Generate a new candidate design $x_1$
    **if** $f(x_1) > f(x_0)$ **then**
        $u \leftarrow$ Uniform$(0, 1)$
        **if** $u < e^{([f(x) - f(x_1)]/T)}$ **then**
            New $x_0 \leftarrow x_1$
        **end**
    **else**
        New $x_0 \leftarrow x_1$
    **end**
    $T = C(T)$
**end**
```

Code 6: R implementation calling Simulated Annealing to minimize the 2D version of the Rastrigin function as an unconstrained objective.

```
# Input #
obj2min <- function(x)
{
  return( 20 + x[1]^2 + x[2]^2 - 10 *
    (cos(2*pi*x[1]) + cos(2*pi*x[2]) ) )
}
lower_bound = c(-7, -7); upper_bound = c(7, 7)
library('GenSA')
SA_ans = GenSA(fn=obj2min, lower=lower_bound,
  upper=upper_bound, control = list(maxit = 300))
cat('\nOptimal␣value␣', SA_ans$value,
  '\nOptimal␣design␣', SA_ans$par)

# Output #
Optimal value 0
Optimal design -5.591672e-12 3.819151e-14
```

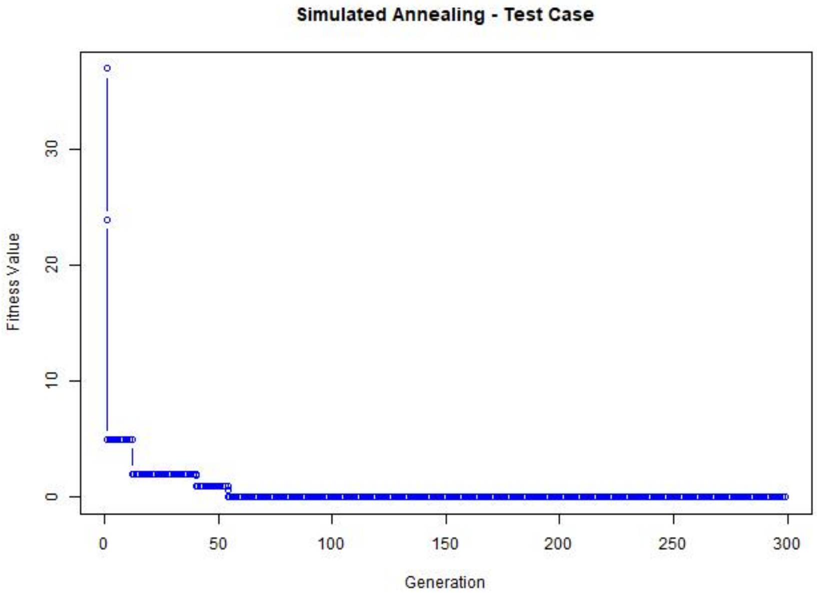


Figure 5: Graphic Output highlighting minimization of the objective along iterations.

available to the colony at once, and should need a pheromone trail (memory) to find their way back to the remaining food at the source and c) in subsequent excursions, ants will follow the shorter paths, since they result more dense on pheromones. Mathematically, this key step "c" corresponds to increasing the probability associated with a certain solution path, given its relative success as measured by earlier experimentations (when the method is exploring the decision space). In some variations, the pheromone trail is updated in each movement of an ant from one location to another, while others opt to update after all ants completed their tour. Simplifying this search into a 2-dimensional map for clarity, Equation (6) represents the amount of pheromone deposited T at a location (x, y) as a function of itself in the previous iteration $(i - 1)$, discounting an evaporation rate ρ (loss of memory in the process) and adding the reinforcement of each ant (counted by k) that travels to the same location to gather food.

$$\tau_{x,y}^{(i)} = (1-\rho)\tau_{x,y}^{(i-1)} + \sum_k \Delta\tau_{x,y}^k \tag{6}$$

Furthermore, the additional pheromone deposited in Equation (6) is parameterized as follows:

$$\Delta\tau_{x,y}^k = \begin{cases} Q/L_k, & \text{if the ant reinforces the same path to } x, y \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

where Q is a calibration constant and L_k is the cost of ant k traveling that arc into location (x, y). The incentive/penalty is then balanced and measured along the iterations, as the commensurate amount of pheromone is deposited. Algorithm 4 describes this mechanism from the computational perspective:

Algorithm 4: Ant Colony Optimization.

```
Given the population of ants, the problem dimension and the discretized
  interval
Evaluate fitness of initial population
while termination criteria not satisfied do
    for each ant do
        for each dimension do
            for each discretized interval do
                Set probability $p$
            end
            Update pheromone index with probability $p$
        end
    end
    $C_i \leftarrow$ cost of solution found by ant $i$
    for each dimension do
        for each discretized interval do
            for each ant do
                Update pheromone quantity according to pheromone index
            end
        end
    end
end
The optimal design is represented by the interval with biggest pheromone
  concentration
```

4.5 Particle Swarm Optimization

The natural analogy underpinning this method is that of the behavior and motion of collectives of animals, such as birds and fish (Mercangöz [2021]). Individual animals within the swarm are like alternative solutions known as “particles”, without knowing which one is the best, but being able to measure each particle?s fitness. Per Equation (8), from each iteration i to the next $(i+1)$, the position P of each particle k is updated by a velocity term V, itself defined in Equation (9):

$$P_k^{(i+1)} = P_k^{(i)} + V_k^{(i+1)} \tag{8}$$

$$V_k^{(i+1)} = \omega \cdot V_k^{(i)} + c_1 \cdot r_1 \cdot \left(P_{\text{BEST}}^{(i)} - P_k^{(i)}\right) + c_2 \cdot r_2 \cdot \left(P_{\text{GLOBAL\_BEST}}^{(i)} - P_k^{(i)}\right) \tag{9}$$

The update of the velocity term occurs by adding up 3 terms: 1) the inertia (pondered by ω), which factors the preference for a current solution/basin of attraction; 2) the individual best (pondered by c_1 and r_1), connected to the best solution/basin of attraction ever attained by an individual particle k and 3) the collective or social best (pondered by c_2 and r_2), connected to the best solution/basin of attraction ever attained by any individual. As summarized in Equation (10), both r_1 and r_2 vary along a small range (typically the unit interval $[0, 1]$) and are unique search hyperparameters for each particle and each iteration.

Parameters c_1 and c_2, on the other hand, may take any positive real value and work as the elitism factor in Genetic Algorithms (the larger they are, the higher is the weight attributed to the best solution found up to that point of the search, which reduces the exploration into alternative basins of attraction).

$$r_{1,2} \in [0, 1],\ \ c_{1,2} \in \mathbb{R}^+ \tag{10}$$

The method is described in computational terms by Algorithm 5 and implemented following Bendtsen [2012] in Code 7, with both numeric and visual outputs displayed in Figure 6.

Algorithm 5: Particle Swarm Optimization.

```
Define neighborhood, maximum influence and maximum velocity
Generate initial population
Set each element velocity vector
$x_0 \leftarrow$ optimal design
while termination criteria not satisfied do
    for each individual do
        Compute the velocity of neighbors
        Update the velocity of the element
        $x_1 \leftarrow$ new position of the element
    end
end
$x_0 = \min(f(x_0), f(x_1))$
```

Code 7: R implementation calling Particle Swarm Optimization to minimize the 2D version of the Rastrigin function as an unconstrained objective.

```
# Input #
obj2min <- function(x)
{
  return(20 + x[1]^2 + x[2]^2 - 10 *
    (cos(2*pi*x[1]) + cos(2* pi * x[2])))
}
lower_bound = c(-7, -7); upper_bound = c(7, 7)
library('pso')
PSO_ans = psoptim( par = rep(NA,2), fn = obj2min,
  lower = lower_bound, upper = upper_bound,
  control = list(trace=1, maxit=300, trace.stats=TRUE))
cat('\nOptimal␣value␣', PSO_ans$value,
  '\nOptimal␣design␣', PSO_ans$par)

# Output #
Optimal value 0
Optimal design 6.107478e-11 -3.494791e-10
```

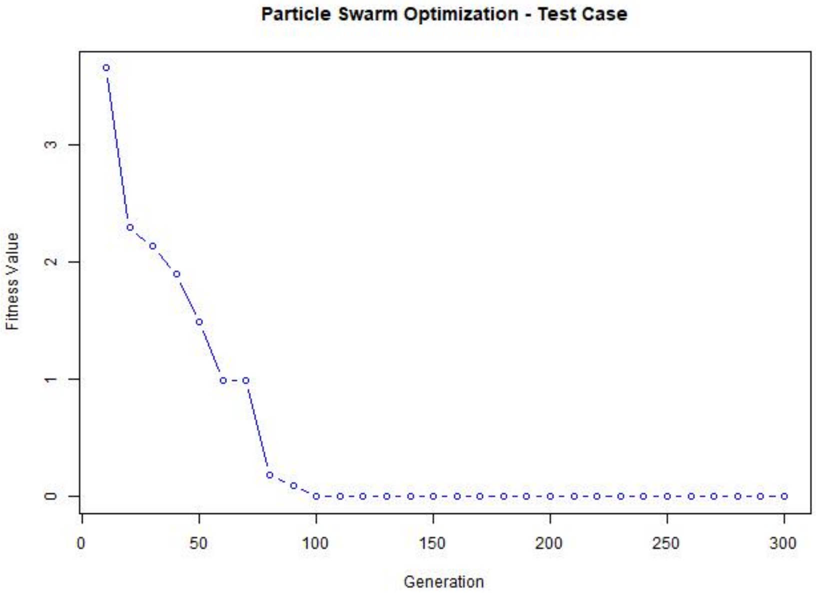


Figure 6: Graphic Output highlighting minimization of the objective along iterations.

4.6 Differential Evolution

Expanding along the lines of GA, DE is also population based stochastic technique, that is used to optimize non-linear (or even discontinuous), non-differentiable problems which are otherwise difficult to solve using classical optimization techniques. Among the GA/DE similarities, there is the reliance on the same list of core operators (selection, crossover, mutation). Also, the fitness evaluation is done for the current generation, serving as guidance for the creation of the subsequent ones. However, unlike GA, where crossover is performed initially and later followed by mutation, DE mutates the individuals prior to a recombination step which is, in essence, crossover. This particular sequence is formally captured in Algorithm 6 and may be better visualized in Figure 7. Code 8 contains its implementation (Mullen et al. [2011]) within the R statistical platform to minimize the 2-dimensional Rastrigin function, with results of a representative run summarized in Figure 8.

Algorithm 6: Differential Evolution.

```
Generate initial population
while termination criteria not satisfied do
    for each element $x_i$ in population do
        Set random parameters
        Compute mutation of current design
        for each dimension do
            Apply mutation with a given probability
        end
        Define new design $n_i$
        if $f(n_i) < f(x_i)$ then
            $x_i \leftarrow n_i$
        end
    end
end
```

5 Miscellaneous novel methods operating upon nature inspired analogies

Over the past decades, many different meta-heuristic algorithms have been proposed for complex optimization problems such as Genetic Algorithms (GA), Ant Colony Optimizaton (ACO) and Particle Swarm Optimization (PSO). Others new nature-inspired algorithms have emerged in recent years in order to optimize complex multimodal objective functions such as Firefly algorithm (FA), Sunflower Optimization (SFO), Bat algorithm (BA), Grey Wolf Optimization (GWO), Lichtenberg Algorithm (LA) and many others.

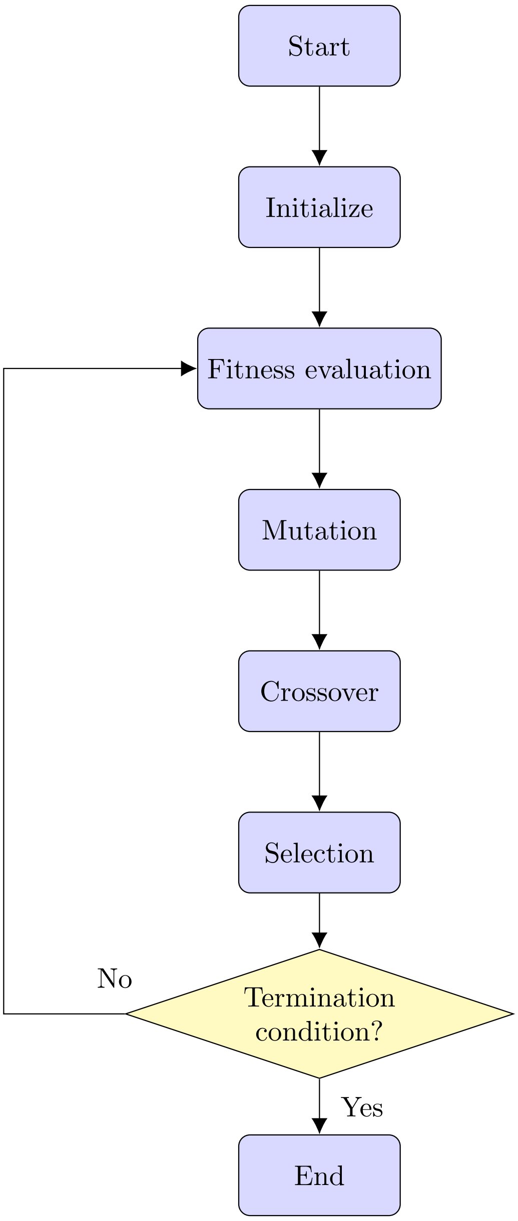


Figure 7: Schematic representation of the DE workflow.

Code 8: R implementation calling Differential Evolution to minimize the 2D version of the Rastrigin function as an unconstrained objective.

```
# Input #
obj2min <- function(x)
{
  return( 20 + x[1]^2 + x[2]^2 - 10 *
    (cos(2*pi*x[1]) + cos(2*pi*x[2])) )
}
lower_bound = c(-7, -7); upper_bound = c(7, 7)
library(’DEoptim’)
DE_ans = DEoptim( fn = obj2min, lower = lower_bound,
  upper = upper_bound,
  control = list( NP = 20, itermax = 300, trace = T))
cat(’\nOptimal␣value␣’, DE_ans$optim$bestval,
  ’\nOptimal␣design␣’, DE_ans$optim$bestmem)

# Output #
Optimal value 1.445954e-12
Optimal design -3.099187e-08 -7.951384e-08
```

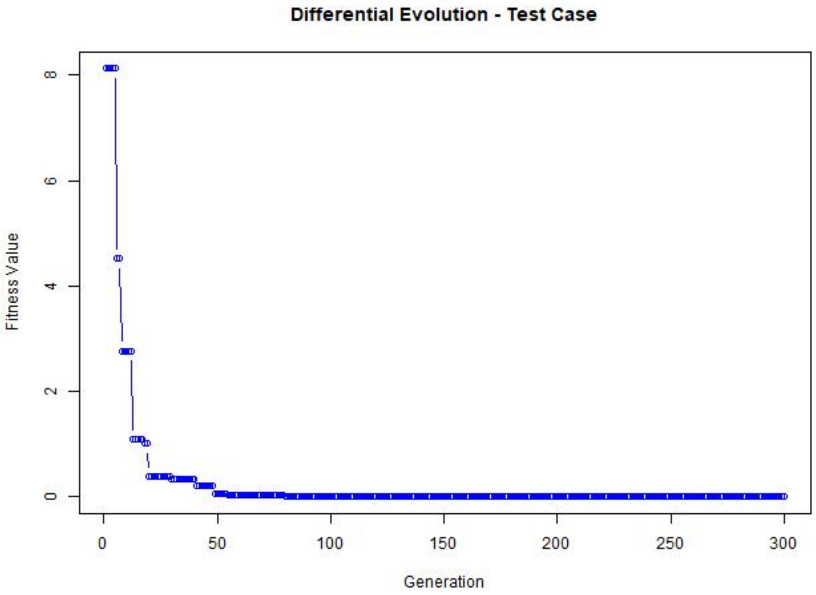


Figure 8: Graphic output highlighting minimization of the objective along iterations.

5.1 Sunflower Optimization Method

The Sunflower Optimization (SFO) method, proposed by Gomes et al. [2019], is a metaheuristic optimization method that is based on the flower pollination process proposed by Yang (2010) with the addition of a movement of plants towards the sun, which increases the convergence speed of the method, in relation to its predecessor. A population of N_{pop} plants containing parameters between a specified lower and upper limit is randomly created. The sun is assumed to be the best plant among those generated. At the beginning of each iteration $m(\%)$ of the plants will die and give way to new plants generated randomly, $p(\%)$ of the plants will pollinate each other and give rise to the new plants according to Equation 11. In addition, Algorithm 7 shows the pseudocode of the SFO methodology (Gomes and de Almeida [2020]).

$$\mathbf{x}_{i,j+1} = \mathbf{x}_{i+1,j} + rand(\mathbf{x}_{i,j} - \mathbf{x}_{i+1,j}) \tag{11}$$

The random function multiplies the vector, term-to-term by a random value between 0 and 1 generated evenly. The rest of the plants will take steps towards the sun. The direction of plants in the sun will be the second displayed in Equation 12.

$$\mathbf{s}_i = \frac{x - x_i}{||x - x_i||}, \ i = 1, 2, ..., N_{pop} \tag{12}$$

The step of each plant will be calculated according to Equation 13 and the maximum step is calculated in Equation 14.

$$d_i = \lambda \cdot P_i(||x_i + x_{i-1}||) \cdot ||x_i + x_{i-1}|| \tag{13}$$

$$d_{max} = \frac{||x_{max} - x_{min}||}{2 \cdot N_{pop}} \tag{14}$$

Finally, the new plantation will be calculated by Equation 15.

$$\mathbf{x}_{i,j+1} = \mathbf{x}_{i,j} + d_i \cdot \mathbf{s}_i \tag{15}$$

The sunflower optimization source code in MATLAB language can be found in Gomes [2021].

5.2 Lichtenberg Spectrum Algorithm

The Lichtenberg algorithm (LA) was recently developed by Pereira et al. [2021]. It is inspired in the physical phenomenon of radial propagation of an intra-cloud lightning. The author used the Lichtenberg figures (LF) as a model for the construction of the optimizer. Figure 9 shows an example of LF.

A theory that is called Diffusion-Limited Aggregation (DLA) based on cluster growth can be used to describe an FL. This model is based on delimiting a region and fixing a particle in the center. A second particle is added at random and moves towards the fixed point, adhering and becoming part of the cluster. This happens with n particles until the figure is completely formed.

Algorithm 7: Sunflower Optimization Method.

```
A random population begins with $n$ plants
Find the sun (Individual aiming closer to zero)
while (k < Maximum number of iterations) do
    $p(\%)$ of plants pollinate each other
    $m(\%)$ of the plants are removed and new random plants will be
      generated
    Or remaining plants will pollinate around the sun
    Evaluates new individuals
    if (New individual is better than its predecessor) then
        The new individual is stored in place of the old
    end
    if (New individual is a great overall) then
        Updates the sun
    end
end
Best solution found
```

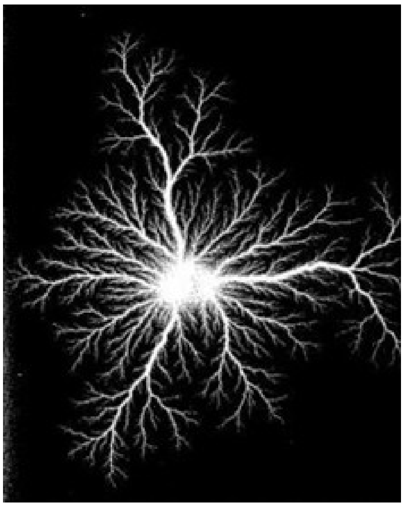
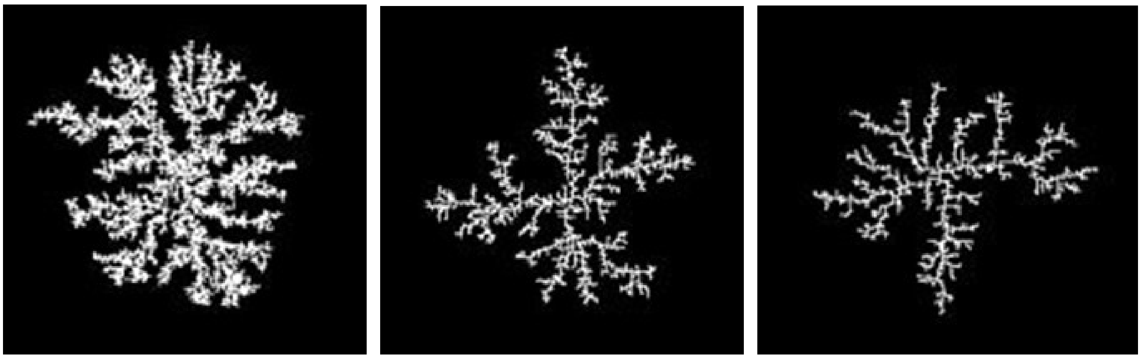


Figure 9: FL in tetra-fluoride gas under $30kV$ voltage and $30.3atm$ pressure (Pereira et al. [2021]).



(a) $S = 0.1$ (b) $S = 0.5$ (c) $S = 1$

Figure 10: Influence of the Adhesion Coefficient S on the density of the cluster.

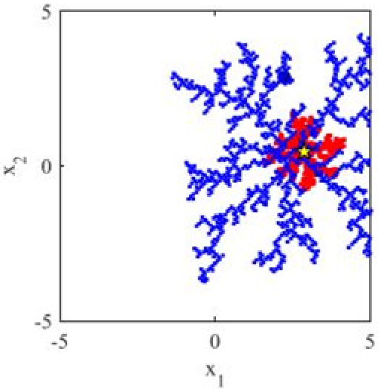


Figure 11: Local Figure (red) with 30% of the global size (blue).

Every time a particle meets the cluster, a random value is generated that is compared with an adhesion coefficient S. If the generated value is less than S, the particle is fixed, otherwise, it escapes. The lower the S, the less likely the particles will join the cluster and the greater the density. Figure 10 shows the LF formed with different values of S.

Some points about the LA must be highlighted: the maximum number of points used to build the LF is one of the input variables. If the figure exceeds the dimension of the search space, the algorithm is finalized before using all points; A random variable between zero and one is used to determine the scale factor of the figure, i.e., the size of the LF is different in each interaction, being delimited by the search space; A random factor is used to rotate the figure, avoiding repeated points; A refinement factor is used to create a second LF with the same trigger point, same rotation and smaller scale factor than the primary figure, helping to improve the local search. Figure 11 shows an example of global LF (blue) and local LF (red).

In this way, LA generates several ramifications for each iteration, always looking for the optimum point. LA has been tested in some cases with excellent results. Thus, this algorithm will be used to evaluate its performance and the already established PSO will be used too. Figure 12, adapted from Pereira et al. [2021], shows a flowchart summarizing the functioning of the Lichtenberg algorithm. In addition, A summary of the algorithm code can be seen through the pseudocode in Algorithms 8 and 9.

In addition to Algorithm 8, the Lichtenberg Algorithm optimization source code in MATLAB language can be found in Pereira et al. [2021].

5.3 Firefly Optimization Algorithm

Based on the behavior of fireflies attracted by light sources, Yang [2009] proposed the Firefly Algorithm that has its pseudo-code shown in Algorithm 10.

In this algorithm, there are two central questions: the variation of the brightness intensity and the attractiveness of each individual. In a simple way, one can assume that the attractiveness of a firefly is determined by its brightness, which is directly related to the value of the objective function. As in the real case, it will be considered that the medium absorbs the intensity of the light emitted by the fireflies, based on the distance between each individual. It is called the light absorption coefficient. The expression for intensity I is given by Equation 16.

$$I(r) = I_0 e^{-\lambda r^2} \tag{16}$$

where I_0 is the original intensity of the light and r is the distance between the fireflies. The attractiveness β, which is proportional to the intensity of light and can be defined and written as shown in Equation 17.

$$\beta(r) = \beta_0 e^{-\lambda r^2} \tag{17}$$

where β_0 is the attractiveness at $r = 0$.

The distance between two fireflies i and j in x_i and x_j, respectively, is the Cartesian distance given by Equation 18.

$$r_{ij} = ||x_i - x_j|| = \sqrt{\sum_{k=1}^{d}(x_{i,k} - x_{j,k})^2} \tag{18}$$

where $x_{i,k}$ is the $k-th$ component of the space coordinate x_i of the $i-th$ firefly. The motion of a firefly i is determined by a more attractive (bright) firefly j and is defined by Equation 19.

$$x_i = x_i + \beta_0 e^{-\gamma r_{ij}^2}(x_j - x_i) + \alpha(\text{rand} - 1/2) \tag{19}$$

where the second term represents attractiveness while the third term determines randomness, α being a parameter of randomness. The expression *rand* is a random number generator evenly distributed in $[0, 1]$. In addition, some studies indicate that the FA is particularly suited for parallel implementation and may outperform existing algorithms, such as PSO, GA, SA, and Differential Evolution, in terms of efficiency and success rates.

Leveraging implementations available at the R statistical computing platform, Code 9 allows application of the Firefly method for the minimization of the Rastrigin

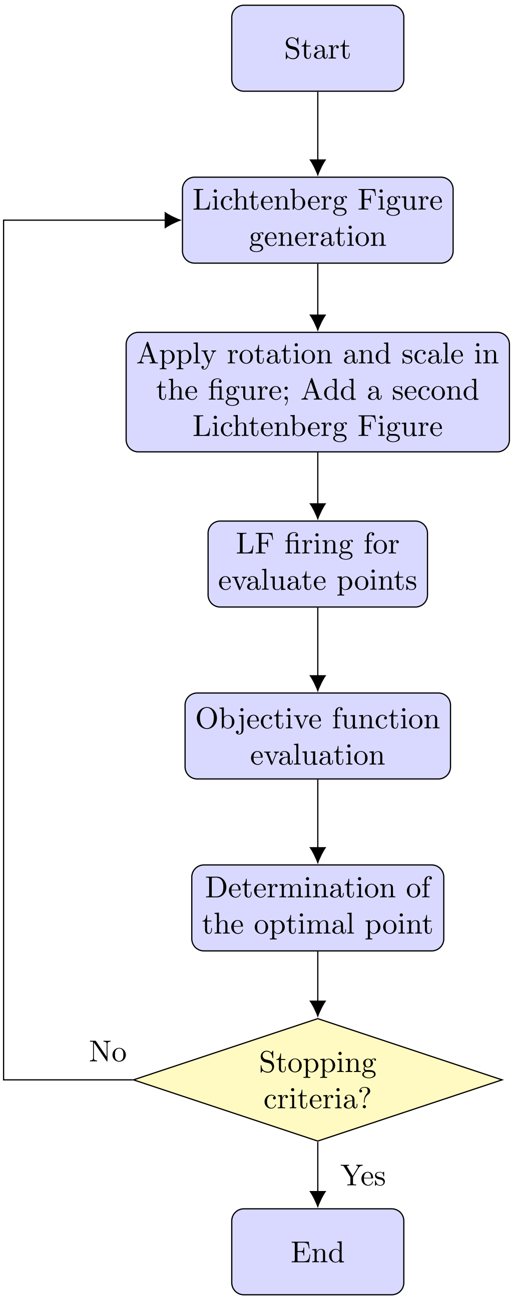

Start
Lichtenberg Figure generation
Apply rotation and scale in the figure; Add a second Lichtenberg Figure
LF firing for evaluate points
Objective function evaluation
Determination of the optimal point
Stopping criteria?
No
Yes
End

Figure 12: Lichtenberg Algorithm Flowchart.

Algorithm 8: Lichtenberg Spectrum Optimization Method - Main code.

```
Main:
Set objective function and search space $J$, upper and lower bounds
Set number of iterations and population $N_{iter}$, $Pop$ (common to all optimizers)
Set Refinement and Parameter for changing the LF: $Ref, M$ (LA routine parameters)
Set LF Parameters $R_c, N_p, S$
if $M$ = 2 then
    Load LF
end
if $M$ = 0 then
    Create a LF
end
while (iter ¡ Niter) do
    if $M$ = 1 then
        Create a LF
    end
    $X_{trigger} \leftarrow$ search space center (trigger point of the first LF)
    if ref = 0 then
        Apply random scale and rotation
        Initialize random population through LF, $X_i$ $(i = 1, 2, ..., Pop)$
    else
        copy LF to create a second LF of size $ref * LF(Local)$
        Apply the same random scale and rotation to both
        Initialize global random population through LF, $Xglobal_i$ $(i = 1, 2, ..., 0.4 * Pop)$
        Initialize local random population through LF, $Xlocal_j$ $(j = 1, 2, ..., 0.6 * Pop)$
        $X_i = Xglobal_i + Xlocal_j$
    end
    Calculate the fitness of each $X_i$
    $Xbest \leftarrow$ the lowest $X_i$ value found
    $Xtrigger \leftarrow Xbest$
    $iter = iter + 1$
end
return $Xbest$
```

Algorithm 9: Lichtenberg Spectrum Optimization Method - Sub-routine creation of LF.

```
Sub-routine: creation of LF
Create an matrix of $R_c$ sized zeros
Place a unitary particle in its center
while ($i < Np$) do
    Randomly place a unitary particle in the matrix
    if the plotted unitary particle t is next to another unitary particle then
        if $rand < S$ then
            This new unitary particle is placed in the matrix
            $i = i + 1$
        else
            The plotted unitary particle is eliminated
        end
    end
    if the cluster of unitary particles reaches $R_c$ then
        The simulation is finished
    end
end
$X \leftarrow$ coordinates of all unitary particles for Cartesian space in the size of
  the search space.
```

Algorithm 10: Firefly Optimization Algorithm.

```
Objective function $f(x),\ x = (x_1, x_2, ..., x_n)^T$
Generate an initial population of $n$ fireflies $x_i,\ (i = 1, 2, ..., n)$
Light intensity $I_i$ at $x_i$ is determined by $f(x_i)$
Define light absorption coefficient $\gamma$
while ($t < MaxGenerations$) do
    for $i = 1 : n$ (all $n$ fireflies) do
        for $j = 1 : n$ (inner loop: all $n$ fireflies) do
            if ($I_i < I_j$) then
                Move firefly $i$ towards $j$
            end
            Vary attractiveness with distance $r$ via $e^{-\gamma r^2}$
            Evaluate new solutions and update light intensity
        end
    end
    Rank the fireflies and find the current global best $g^*$
end
Postprocess results and visualization
```

Code 9: R implementation calling the Firefly optimization method to minimize the 2D version of the Rastrigin function as an unconstrained objective.

```
# Input #
obj2min <- function(x)
{
  return( 20 + x[1]^2 + x[2]^2 - 10 *
    (cos(2*pi*x[1]) + cos(2*pi*x[2])) )
}
lower_bound = c(-7, -7); upper_bound = c(7, 7)
library('metaheuristicOpt')
FFA_ans = metaOpt(obj2min, optimType = 'MIN',
  algorithm = 'FFA', numVar = 2,
  rangeVar = rbind(lower_bound, upper_bound),
  control = list(numPopulation=20, maxIter=700))
cat('\nOptimal␣value␣', FFA_ans$optimumValue,
  '\nOptimal␣design␣', FFA_ans$result)

# Output #
Optimal value 4.510703
Optimal design 0.9207517 1.081387
```

Function, in the present case presenting both input and output (a dedicated post-processing module is still to be implemented to enable the generation of graphical output, such as the value of the objective function over iterations, for example).

5.4 Bat Optimization Algorithm

The Bat Optimization Algorithm (BA) is inspired in the micro-bats behavior and its ability of echolocation. This algorithm uses as parameters the loudness and pulse emission, it was also the first to take and account the frequency tuning.

Bats utilize a type of sonar, called echolocation, to recognize prey, stay away from obstacles, and locate their roosting crevices in the obscurity. These bats emanate a loud sound pulse tune in for the reverberation that bounces back from the surrounding objects. Their pulses fluctuate in properties which can be corresponded with their chasing methodologies and strategies, depending on the species. Most bats utilize short, frequency-modulated signals to sweep through about an octave; others all the more frequently utilize consistent recurrence signals for echolocation. Their sign data transfer capacity changes with species and frequently increments by utilizing more harmonics (Yang [2010]).

The BA uses the diversity of pulse emission rate r and loudness A to control exploration and exploitation characteristics. In the main BA, Equations 20, 21 and 22 present the main equations for the evolutionary process (Yang [2010]).

$$f_i = f_{min} + (f_{max} - f_{min})\beta \tag{20}$$

$$v_i^t = v_i^{t-1} + (x_i^{t-1} - x^*)f_i \tag{21}$$

$$x_i^t = x_i^{t-1} + v_i^t \tag{22}$$

where $\beta in[0,1]$ is a random vector drawn from a uniform distribution so that the frequency can vary from f_{min} to f_{max}. Equally important, these updating equations are also associated with the pulse r and loudness A via a uniformly distributed random number ϵ. Selection is done by the current best solution x^* found so far by all the bats.

It can be seen that both above equations are linear in terms of x_i and v_i. However, the control of exploration and exploitation of the BA is defined by the changes of loudness A from a high value to a lower value and the emission rate r from a lower to a higher value (Equation 23).

$$A_i^{t+1} = \mu A_i^t,\ r_i^{t+1} = r_i^0(1 - e^{-\gamma t}) \tag{23}$$

where $0 < \mu < 1$ and $\gamma > 0$ are two random parameters. As a result, the actual algorithm can have a weak nonlinearity. Consequently, BA can have a faster convergence rate in comparison with PSO. The basic steps of BA can be summarized as the schematic pseudo code shown Algorithm 11.

Algorithm 11: Bat Optimization Algorithm.

```
Initialize the bat population $x_i$ and $v_i$
Initialize frequencies $f_i$, pulse rates $r_i$ and the loudness $A_i$
while termination criteria not satisfied do
    Generate new solution by adjusting frequency
    Update velocities and locations/solutions
    if $rand > r_i$ then
        Select a new solution
        Generate local solution around the selected
    end
    Generate a new solution by flying randomly
    if $(rand < A_i)$ and $(f(x_i) < f(x^*))$ then
        Accept the new solutions
        Increase $r_i$ and reduce $A_i$
    end
    Rank the bats and find the current best $x^*$
end
```

Similarly, to the previous section, the existence of an implementations available at the R statistical computing platform allows the construction of Code 10, which uses the Bat Optimization method for the minimization of the Rastrigin Function. (Once again both input and numeric output are presented, with the availability of visual results pending a dedicated post-processing module).

Code 10: R implementation calling the Bat optimization method to minimize the 2D version of the Rastrigin function as an unconstrained objective.

```
# Input #
obj2min <- function(x)
{
  return( 20 + x[1]^2 + x[2]^2 - 10 *
    (cos(2*pi*x[1]) + cos(2*pi*x[2]) ) )
}
lower_bound = c(-7, -7); upper_bound = c(7, 7)
library('metaheuristicOpt')
BA_ans = metaOpt(obj2min, optimType = 'MIN',
  algorithm = 'BA', numVar = 2,
  rangeVar = rbind(lower_bound, upper_bound),
  control = list(numPopulation = 20, maxIter = 700))
cat('\nOptimal␣value␣', BA_ans$optimumValue,
  '\nOptimal␣design␣', BA_ans$result)

# Output #
Optimal value 0
Optimal design 3.238586e-292 1.289825e-292
```

6 Practical considerations for applied optimization in general and structural optimization in particular

Noteworthy of population-based methods is the increase in computational effort resulting from simultaneous consideration of multiple designs per iteration instead of a singular one, as in the case of classical (derivative based) optimization methods.

For this reason, it is often impractical to couple heuristic optimizers directly into high-fidelity analysis codes (Finite Element and others), and it is standard practice to evaluate the functions pertaining to the optimization problem by way of approximations, such as the statistical surrogates (or metamodels) described in another Chapter of the present volume. While a faster to calculate surrogate will enable heuristic optimization methods to work properly and improve the decision making process even with the complexity added by multiple basins of attraction, attention is required to manage the propagation of approximation errors, which takes place a) preemptively, by building accurate enough approximations and b) correctively, by updating/enhancing initial versions of surrogates as needed and, in certain cases, making calls to the actual analysis software at a lower frequency (not at every iteration or step of the search process) so that error propagation can be contained. The remaining challenge is to ensure that the search is not prone to invoke a design candidate whose analysis and/or derivative calculation are not feasible, and addressing it is often dependent upon sound judgment about physical regimes and their changes.

On a different but related note, it is not unusual that a singular heuristic search method is not the best alternative to solve a given optimization problem, regardless of how powerful (and intrinsically computationally sophisticated/expensive it might be). Hence, practitioners have developed the so-called lifecycle approaches (Viana et al. [2007]), so that combinations of optimization algorithms (calculus based and/or heuristic) are employed in tandem, subject to several configuration and transition criteria that evolve with the progress of the search itself and rely on more than one method over the duration of the optimization procedure (also known as “lifecycle”, hence the approach name).

References

C. Bendtsen. pso: Particle swarm optimization. *R package version 1.0.3*, 2012. URL `https://CRAN.R-project.org/package=pso`.

C. Blum. Ant colony optimization: Introduction and recent trends. *Physics of Life reviews*, 2(4):353–373, 2005.

J. Bossek. Ecr 2.0: A modular framework for evolutionary computation in r. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 1187–1193, 2017.

K. De Jong. Evolutionary computation: a unified approach. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion*, pages 185–199, 2016.

G. F. Gomes. Sunflower optimization (sfo) algorithm. Technical report, MATLAB Central File Exchange, 2021. URL `https://www.mathworks.com/matlabcentral/fileexchange/69076-sunflower-optimization-sfo-algorithm`.

G. F. Gomes and F. A. de Almeida. Tuning metaheuristic algorithms using mixture design: Application of sunflower optimization for structural damage identification. *Advances in Engineering Software*, 2020. doi: https://doi.org/10.1016/j.advengsoft.2020.102877.

G. F. Gomes, S. S. da Cunha, and A. C. Ancelotti. A sunflower optimization (sfo) algorithm applied to damage identification on laminated composite plates. *Engineering with Computers*, 35(2):619–626, 2019. doi: https://doi.org/10.1007/s00366-018-0620-8.

B. A. Mercangöz. *Applying Particle Swarm Optimization*. Springer, 2021.

K. Mullen, D. Ardia, D. L. Gil, D. Windover, and J. Cline. Deoptim: An r package for global optimization by differential evolution. *Journal of Statistical Software*, 40(6):1–26, 2011.

J. L. J. Pereira, M. B. Francisco, C. A. Diniz, G. A. Oliver, S. S. Cunha Jr, and G. F. Gomes. Lichtenberg algorithm: A novel hybrid physics-based meta-heuristic for global optimization. *Expert Systems with Applications*, 170, 2021. doi: https://doi.org/10.1016/j.eswa.2020.114522.

R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2020. URL `https://www.R-project.org/`.

L. S. Riza et al. metaheuristicopt: Metaheuristic for optimization. *R package version 2.0.0*, 2019. URL `https://CRAN.R-project.org/package=metaheuristicOpt`.

L. Scrucca. Ga: a package for genetic algorithms in r. *Journal of Statistical Software*, 53(4), 2013.

F. A. C. Viana, V. Steffen Jr, S. Butkewitsch, and M. de Freitas Leal. About the optimum design of an aircraft pressure bulkhead by using multi-fidelity and lifecycle algorithm. In *IPDO-Inverse Problems, Design and Optimization Symposium*, 2007.

Y. Xiang, S. Gubian, B. Suomela, and J. Hoeng. Generalized simulated annealing for global optimization: the gensa package. *R Journal*, 5(1):13, 2013.

X.-S. Yang. Firefly algorithms for multimodal optimization. In *International symposium on stochastic algorithms*, pages 169–178. Springer, 2009.

X.-S. Yang. A new metaheuristic bat-inspired algorithm. In *Nature inspired cooperative strategies for optimization (NICSO 2010)*, pages 65–74. Springer, 2010.

X.-S. Yang. Nature-inspired optimization algorithms: Challenges and open problems. *Journal of Computational Science*, 46, 2020.

Chapter 5: Application of Machine Learning and Multi-Disciplinary/Multi-Objective Optimization Techniques for Conceptual Aircraft Design

Chapter details

Chapter DOI:

https://doi.org/10.4322/978-65-86503-71-5.c05

Chapter suggested citation / reference style:

Mattos, Bento S., et al. (2022). "Application of Machine Learning and Multi-Disciplinary/Multi-Objective Optimization Techniques for Conceptual Aircraft Design". *In* Jorge, Ariosto B., et al. (Eds.) *Model-Based and Signal-Based Inverse Methods*, Vol. I, UnB, Brasilia, DF, Brazil, pp. 143–236. Book series in Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity.

P.S.: DOI may be included at the end of citation, for completeness.

Book details

Book: Model-based and Signal-Based Inverse Methods

Edited by: Jorge, Ariosto B., Anflor, Carla T. M., Gomes, Guilherme F., & Carneiro, Sergio H. S.

Volume I of Book Series in:

Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity

Published by: UnB City: Brasilia, DF, Brazil Year: 2022

DOI: https://doi.org/10.4322/978-65-86503-71-5

Application of Machine Learning and Multi-Disciplinary/Multi-Objective Optimization Techniques for Conceptual Aircraft Design

Bento S. de Mattos[1 *], Felipe A. Bortolete[2], José A. T. G. Fregnani[3],
Ariosto B. Jorge[4], William M. Alves[5], Ronaldo V. Cruz[6]

[1]Associate Professor, Instituto Tecnológico de Aeronáutica, bmattos@ita.br

[2]Ph.D. Candidate, Instituto Tecnológico de Aeronáutica, felipebortolete@gmail.com

[3]Visiting Professor, Instituto Tecnológico de Aeronáutica, fregnani@ita.br

[4]Post-Graduate Program - Integrity of Engineering Materials, University of Brasilia, Brazil. E-mail: ariosto.b.jorge@gmail.com

[5]Ph.D. Candidate, Instituto Tecnológico de Aeronáutica, wma.mecatronica@gmail.com

[6]Adjunct Professor, Instituto Tecnológico de Aeronáutica, ronaldoc@ita.br

*Corresponding author

Abstract

The development of fuel-efficiency airliners is crucial for the survival of aircraft manufacturers and airlines. In this context, multi-disciplinary optimization encompassing multiple objectives has provided great contributions to the aerospace industry. Thanks to the increase of computational power and efficient optimization algorithms, many tasks usually performed in the preliminary design phase are now carried out into the conceptual phase. Aeroelastic analysis and complex finite element models are already part of the conceptual phase. In addition, accurate and efficient surrogate models made faster and more stable computations possible. The content of this Chapter discusses the quest for the incorporation of high aspect wings into aircraft configurations, provides an overview of machine learning algorithms, and presents three engineering applications: two multi-objective optimizations of transport aircraft and a fluid-structure calculation of KC-135. The first optimization task is concerned with an aerodynamic optimization of an airliner wing by using a surrogate model to calculate aerodynamic coefficients based on artificial neural networks. The second optimization application encompasses the incorporation of flutter speed constraint for airliners of metallic construction into a multi-disciplinary optimization platform tailored for transport aircraft design. For this purpose, a package for structural sizing and aeroelastic analysis was integrated into the optimization framework developed at the Institute Technological of Aeronautics. This package is known as the Next-Generation Conceptual Aero-Structural Sizing (NeoCASS) from the University of Milano. NeoCASS is used to estimate the flutter speeds of individuals that arise during an optimization process. Finally, the detailed aircraft wing structure model of the KC-135 aerial refueller was built by using preliminary sizing procedures based on estimated main aerodynamic loadings, flight enveloped, V-n diagram, and detailed structural layout reports. Then a fluid-structure interaction (FSI) computer simulation procedure and compared to available experimental data.

Greek symbols

α	Angle of attack
Γ_{ϕ}	Diffusion coefficient
ϕ	Potential of velocities
Υ	Neuron activation function
κ	Thermal conductivity
μ	Dynamic viscosity
ρ	Fluid density
σ_{ij}	Stress tensor
θ	Neuron bias

Other Symbols and abbreviations

ADS-B	ADS-B stands for Automatic Dependent Surveillance Broadcast. ADS-B is a tracking system based merely on data transmission - in Mode-S (S stands for Select) - by the transponder at the frequency of 1090 MHz Broadcasted are several aircraft parameters
AF2	Approximate factorization (Scheme 2) introduced by Ballhaus and Steger [1]
ANN	Artificial neural network
AR	Aspect ratio
AR_w	Wing aspect ratio
BOW	Basic operating weight
BPR	Engine by-pass ratio
C_{DI}	Induced drag coefficient
CFD	Computational Fluid Dynamics
C_L	Lift coefficient
$C_{L.max}$	Maximum lift coefficient
C_p	Pressure coefficient
CSD	Computational structural dynamics
D	Drag force
DOC	Direct operating cost
DOE	Design of experiment
e	Oswald’s factor (Related to induced drag)
e_{lweb}	The thickness of the front spar
e_{tweb}	The thickness of the rear spar
e_{upanel}	The thickness of upper skin
e_{lpanel}	The thickness of lower skin
$S_{Boom,i}$	Area of idealized boom
B_i	Area of real boom
EI and EI_{Liner}	Efficiency index of an aircraft configuration
FAR	Federal aviation regulation
F_d	Fuselage equivalent diameter
FEM	Finite element method

F_{hw}	Fuselage height-to-width ratio
FSI	Fluid-structure interaction
h_{lweb}	Height of front spar
h_{tweb}	Height of rear spar
L	Lift force
L/D	Lift-to-frag ratio
L_f	Fuselage length
LCO	Limit cycle oscillation
LRC	Long-range cruise
M	Number of Mach
M_d	Dive Mach number
M_∞	Freestream Mach number
MAC	Mean aerodynamic chord
MMO	Maximum operating Mach number
MTOW	Maximum takeoff weight
MZFW	Maximum zero-fuel weight
NeoCASS	Next-Generation Conceptual Aero-Structural Sizing
OPR	Engine overall pressure ratio
p	Pressure
PAX	Passenger/Passengers
RANS	Reynolds averaged Navier Stokes
Re	Reynolds number (The ratio between the viscous and inertia forces in a fluid)
R_{nm}	Range in nautical miles
RoC	Rate of climb
SMARTCAD	Simplified Models for Aeroelasticity in Conceptual Aircraft Design
S_w	Wing reference area
tc_root	The maximum relative thickness of the root wing station
TFL	Takeoff field length
TOGW	Take-off gross weight
TW	Thrust-to-weight ratio
V	Speed
VT	Vertical tail
VMO	Maximum operating calibrated speed
wbi	Width of the torsion box

1. The quest for high aspect-ratio wings and new aircraft configurations

1.1 Induced drag

Induced drag is drag linked to the generation of lift, which is a phenomenon that involves the production of vorticity at the trailing edge and wingtip of trapezoidal wings of transport aircraft (Delta wings presents other sources of vorticity). The vortex intensity generated at the trailing edge is directly proportional to the loading distribution (product of the chord in each section by the lift coefficient of the section) along the span. **Eq. 1** shows the basic formula for the calculation of induced drag coefficient:

$$C_{DI} = \frac{C_L^2}{\pi \cdot AR_w \cdot e} \tag{1}$$

Oswald's factor shown in **Eq. 1** is an efficiency parameter that measures how the loading distribution along the wingspan departs from a hypothetical elliptical one. Thus, shock waves, the lift itself and any other things that impact aerodynamics will also exert an influence on Oswald's factor. Niță and Scholz elaborated an empirical and low-fidelity model to calculate Oswald's factor with a satisfactory degree of accuracy [2]. The considerable increase of wing aspect ratio typically represents the pragmatic approach to reduce induced drag. However, this will bring some great disadvantages to the airplane configuration. Flutter is a primary cause of concern for design teams considering the adoption of high aspect-ratio wings. Usually, the design and optimization of an aeroelastic wing structure may be undertaken to minimize a structural mass that satisfies many different types of constraints.

The flutter boundary of an aeroelastic system is defined as the lowest airspeed at which a small perturbation to a static equilibrium leads to self-powered oscillatory motion that does not decay. Thus, it is defined as the lowest airspeed at which the system has a complex-conjugate pair of eigenvalues with zero real part. For linear systems, the dynamics predicted by flutter analysis apply globally and thus any perturbation at flutter speed leads to a divergent oscillation. The structural system is then dynamically unstable, and disturbances may catastrophically grow at an extremely low time [3].

Indeed, flutter ordinarily occurs due to a speed and lift coefficient combination. In this specific issue, there is an analogy with flight dynamics. The stability derivatives may be indicators of how a given aircraft will behave dynamically in a flight condition. However, after merely applying some commands from an initial condition it is then possible to analyze whether the trajectory will be stable or not. Similarly, the modal frequencies of an aircraft structure alone do not indicate dynamic behavior. The aircraft structure must be typically experiencing some loading to proceed cautiously with the comprehensive analysis of its flutter characteristics.

Nonlinear mechanisms within the aeroelastic system may attenuate the disturbance propagation and produce a self-sustained limit cycle oscillation [4]. For nonlinear structures, like that, merely represented by the Timoshenko beam model, the dynamics of the perturbation beyond the local region are unknown. Timoshenko's theory of beams constitutes an improvement over the Euler-Bernoulli theory, in that it incorporates shear and rotational inertia effects

As the deformation in the system becomes significant, changes in structural properties mean the modal interactions necessary for a flutter to inevitably take place may be naturally affected. In this manner, the dynamics are no longer topologically equivalent to that of a linear system [5].

At the flutter boundary itself, there are two distinct possibilities for a nonlinear system. The first comprises a flawless transition to a stable LCO when the critical airspeed is exceeded, the amplitude of which increases from zero for speeds higher than the flutter speed. In this case, the nonlinear flutter point coincides with a so-called supercritical Hopf bifurcation (**Figure 1**). This region is beneficial when compared to the linear case (**Figure 1**), as the unbounded oscillation is replaced by a reversible response via airspeed reduction (see arrows in **Figure 1**). Unfortunately, a harmful case is also part of the space of states: a subcritical Hopf bifurcation may occur, characterized by an unstable LCO region for speeds lower than the flutter one and a path turn featuring periodic oscillations, suddenly leading to a dangerous situation with large stresses and structural deflections [6]. Even decreasing the speed below the flutter point the system dynamics will not return immediately to the origin, resulting in a hysteresis loop [6].

The existence of subcritical behavior as described before leads to new considerations in aircraft optimization concerning posing flutter speed as a constraint for non-linear systems. Even considering the harmful situation that may occur in the subcritical region, below flutter speed, a gust, for example, can bring the system dynamics onto the stable large-amplitude limit cycle branch [6]. Naturally increasing wing aspect ratio leads to increased deflections when loading is present. However, the slenderness and reduced overall stiffness of wings optimized for minimal weight may lead to more complex deflections of more considerable magnitudes, resulting in nonlinear behavior. Such nonlinear effects can undoubtedly significantly influence the aeroelastic behavior of civil aircraft fitted properly with high aspect-ratio wings.

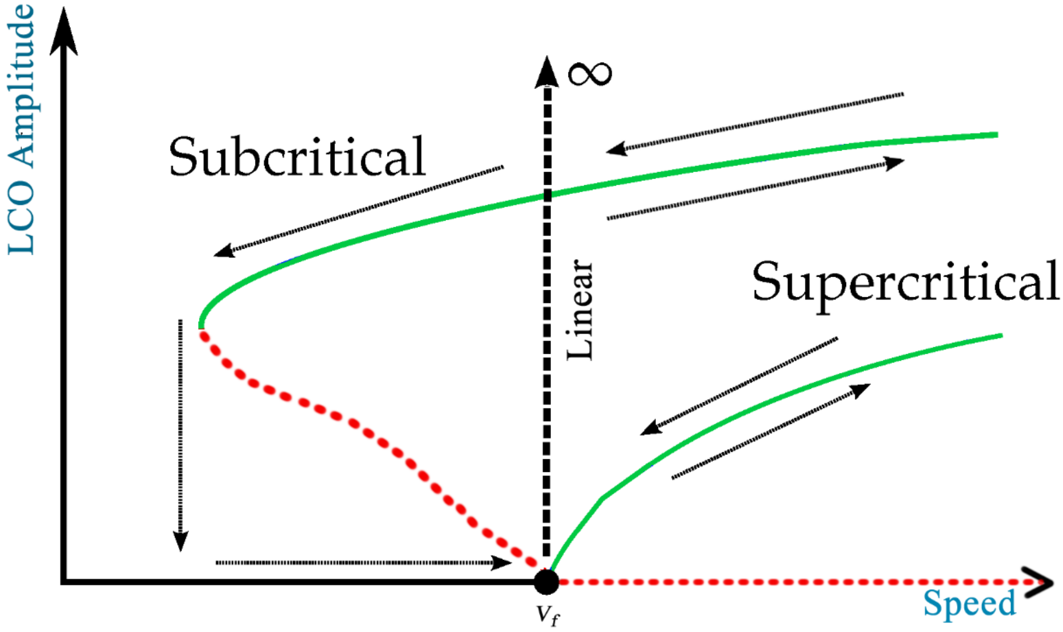


Figure 1: Generic flutter possibilities: linear, supercritical Hopf bifurcation, and subcritical Hopf situation; green color means stable behavior; red, unstable [5] [6]

The aerospace community has intensively been researching the efficient utilization of high aspect-ratio wings for commercial airplanes. Boeing developed its new 787 long-range airliner featuring extremely flexible wings in 2011, with All Nippon Airways as the launch customer [7]. Such highly flexible wings benefit from the stable LCO characteristics of nonlinear systems. Airbus chose a composite wing for the A350XWB airliner after a deep redesign of the initial proposition of an improved A330 [8]. The European Consortium adopted a figure of 9.49 for the A350-900 version [8]. The A350's wing features a quarter-chord sweep of 31.9 degrees, advanced shaped winglets, and spoilers that droop with flap extension to fully seal the gap between the Fowler flaps and wing to reduce drag [9]. Embraer developed a new variant of its E195 airliner and designated it E195E2, which features a considerable increase in wing aspect ratio when compared to previous products of that company. Both E195 and E195E2 feature wings of conventional wing aluminum structure. **Table 1** shows some compared characteristics of the original E195 to its variant E195E2 [10] [11] [12] [13] [14]. According to **Table 1**, a considerable BOW increase for the E2 version can be observed.

Table 1: Comparison of some characteristics of E195E1 and E2

	E-195E1 AR	E-195E2
Wing reference area [m^2]	92.50	103
Wing quarter-chord sweepback angle	23.5°	27.5°
Wing aspect ratio	8.1	9.4
Engine by-pass ratio	5.4:1	12:1
Single-engine dry weight [kg]	1,700	2,177
Max. passenger accommodation	124 @ 31" | 30" | 29" pitch	146 @ 28" pitch
Range (Full PAX, LRC)	2,300 nm (typical reserves)	2,655 nm (100 nm alternate)
MTOW/BOW [kg]	52,290/28,700	61,500/35,750
BOW/MTOW	0.55	0.58
Maximum usable fuel [kg]	13,100	13,500

The reduction of induced drag is the main reason behind the incorporation of high aspect ratio wings into transport aircraft configurations. Induced drag is dependent on the square of lift coefficient, aspect ratio, and the loading distribution along the wingspan. Despite its benefit due to the way it affects the induced drag, high-aspect-ratio wings bring with them several drawbacks:

- Flutter speed is reduced.
- Structural weight increases sharply [15].
- In combination with high-sweepback angles, they may lead to pitch-up behavior.
- The average chord is usually smaller, therefore, increasing the parasite drag.
- Potentialize the adverse yaw effect.

Transport airplanes with higher aspect ratio wings of metallic construction may require flutter ballast in front of the wing elastic line. Increasing the stiffness of the outer wing structure may not be enough to increase flutter speed. A concentrated mass of lead is typically used in such cases. The addition of winglets may require additional ballast due to pitch inertia increase at wingtips aggravating critical flutter modes [16]. For the Boeing 737-800 airliner, a reduction in the low altitude operating speed was avoided by adding 41 kg of ballast per wing in the outboard leading edge [16].

The necessity of incorporating a wing with a considerable aspect ratio into an aircraft will depend on the mission and other requirements of the project. Jet airplanes conceived for oceanic crossings will perform the cruising mission at optimal $M \times {}^{L}/_{D}$ value. This translates into a cruise Mach number lower than the MMO, resulting in a lift coefficient where the induced drag will be a considerable part of the overall drag. This may not be the case for regional airplanes, which perform the cruise phase is shorter when compared with long-haul airplanes. This type of airplane should be optimized for the airline network they will operate [17]. **Table 2** shows the average stage length flown in the United States of some major airlines [18].

Table 2: Average stage length in the U.S. flown by small narrow-body fleet [18]

	1995	**1997**	**2000**	**2010**	**2016**
American	852	898	850	860	762
Continental	731	822	903	1,003	-
Delta	552	569	609	745	665
Northwest	612	614	639	-	-
United	692	721	767	995	977
US Airways	514	521	554	779	-
America West	620	738	854	-	-
--sub Network	**664**	**699**	**727**	**835**	**766**

Due to the structural and aeroelastic reasons pointed out before, the strength and fatigue advantages of carbon fiber encouraged aircraft manufacturers to incorporate higher-aspect-ratio wings into airliner configurations. Besides the utilization of carbon fiber, unusual airplane configurations have been also proposed to mitigate the adverse effects of high-aspect-ratio wings (**Figure 2**). The unusual aircraft configurations being considered to address the induced-drag issue are summarized below:

- Twin-fuselage configuration
- truss-braced and strut-braced airliner
- highly flexible wings
- joined wings

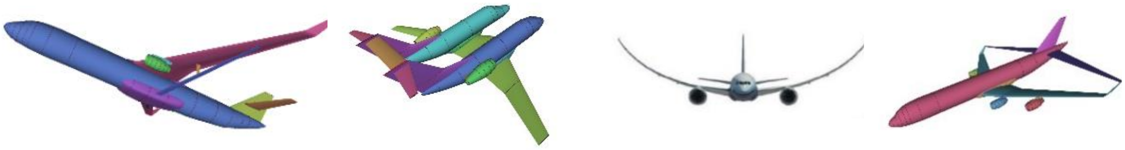


Figure 2: Some configurations usually proposed to mitigate drawbacks related to the adoption of high-aspect-ratio wings

1.2 Bracing

In aircraft design, bracing comprises a peculiar kind of structural component that stiffens the functional airframe to accrescent nominal rigidity and strength under loading. Structural reinforcements are applied both internally and externally and may take the form of the strut, which withstands compression or tension. Exposed wires, which resist tension, only, were commonly used to provide structural strength to vintage airplanes.

Strut-braced wing (SBW) and truss-braced wing (TBW) configurations provide a didactical example of drag reduction benefits that a radical change in wing configuration could bring. Such configurations are commonly seen on high-wing general aviation airplanes like the Cessna Skylane and Skyhawk and almost universal on parasol-winged airplanes such as the Consolidated PBY Catalina (**Figure 3**). Less commonly, some low-winged monoplanes like the Piper Pawnee agricultural airplane are characterized by lift struts mounted on the wing upper surface, acting in compression in flight and tension on the ground (**Figure 4**). For aircraft of moderate engine power and speed, lift struts represent a compromise between the higher drag of a fully cross-braced structure and the higher weight of a cantilevered wing.

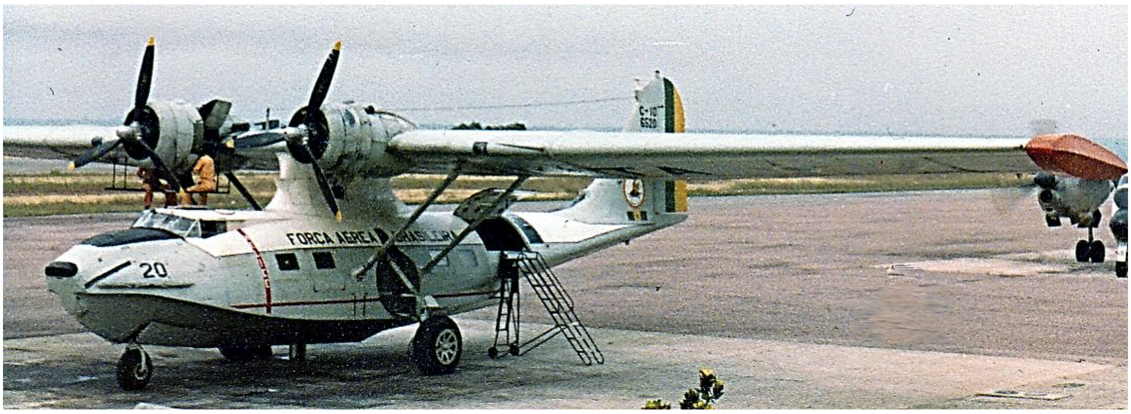


Figure 3: PBY Catalina (Photo: B. Mattos)

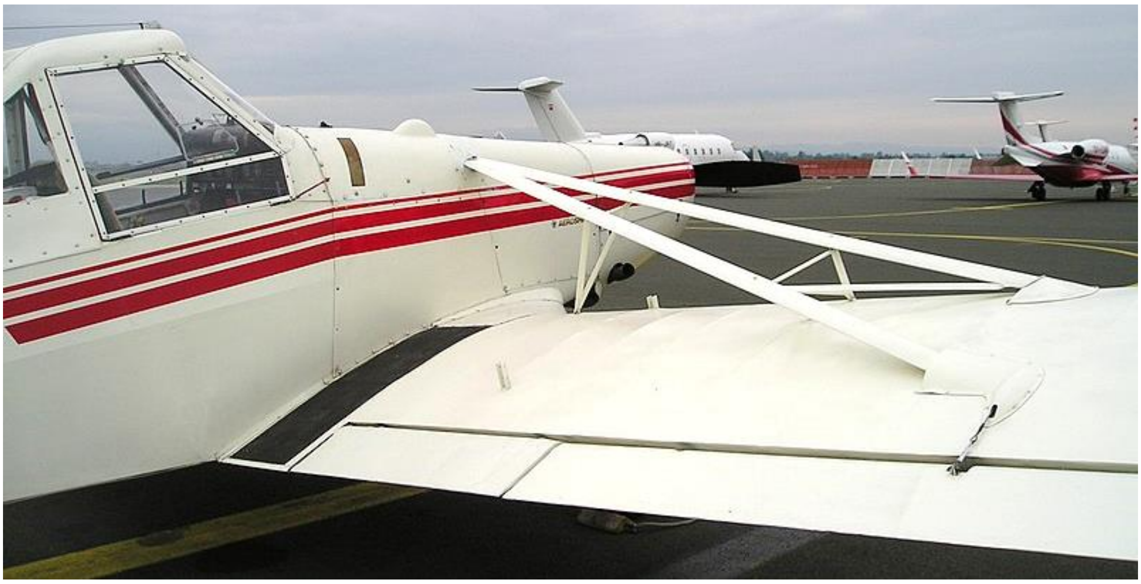


Figure 4: Piper PA-25 Pawnee (Photo released to the public domain)

On a high-wing aircraft, a strut connects an outboard point on the wing with a point lower on the fuselage to form a rigid triangular structure. While in flight the strut acts in tension to carry wing lift to the fuselage and hold the wing level, while when back on the ground it acts in compression to hold the wing up. This kind of airplane has been studied by universities and aerospace manufacturers as a viable concept for medium-sized airliners. TBW offers potential for performance improvements in terms of fuel efficiency thanks to the higher aspect ratio wings it allows. Gern et al. claim that a strut-braced wing enables lower airfoil thickness, which will lower the transonic wave drag and hence require a lower wing sweepback angle [19]. In turn, they continue affirming that the lower wing sweep and high-aspect ratios will produce natural laminar flow thanks to low Reynolds numbers. Consequently, a significant increase in the overall aircraft performance is achieved. However, a lower airfoil thickness-to-chord ratio will inevitably lead to a heavier wingbox. The trade-off among several factors, lower wing sweepback angle, and the strut itself may bring a lighter airplane. In addition, there are also challenges for structural designers as the mitigation buckling of the strut or truss [20].

In 2000, Grossman et al. perform a comparative study for a 325-passenger class airliner fitted with two GE-90 engines [21]. The truss-braced airplane presents fuselage-mounted engines. Using contemporary multi-disciplinary design optimization techniques available at that time integration of the aerodynamic and structural design requirements was carried out. They considered a hexagonal wingbox and optimized area/thickness ratios for spar webs and caps, stringers, and skins. The results for the truss-braced configuration indicated that the take-off gross weight was reduced by more than 10-percent when compared to an equivalent cantilever wing airplane. According to Ref. [21], significantly larger weight reductions (19% TOGW) are obtained for the wing-mounted engine case.

To illustrate some characteristics of a strut-braced beam and compare it with a cantilever beam, the derivation of shear force and bending moments for beams was carried out and shown in the last part of this Section.

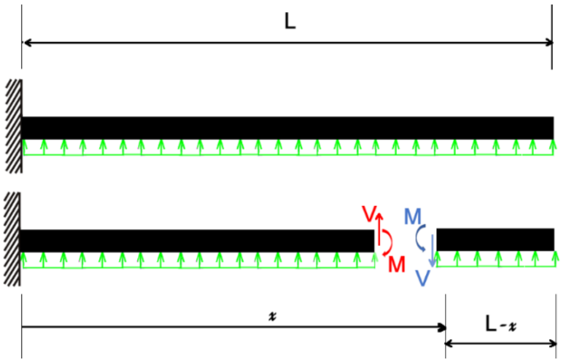


Figure 5: Internal reactions of a cantilever beam

Figure 5 shows a cantilever beam and the internal forces and moments at the boundary of two slices of it. *V* is the shear force; *M* is the bending moment and *w* is a uniform loading expressed in terms of force per unit length. The equilibrium of forces and moments promptly allows the derivation of two equations:

$$\sum F_y = 0 \tag{2}$$

$$\sum M_{x=L} = 0 \tag{3}$$

Inserting the right parameters in **Eqs. 2** and **3**, the following expressions for shear force and bending moment distributions can be derived:

$$w(L-x) - V = 0 \Longrightarrow V = w(L-x) \tag{4}$$

$$M + V * (L-x) - w(L-x)\frac{(L-x)}{2} = 0 \Longrightarrow M = -w\frac{(L-x)^2}{2} \tag{5}$$

Figure 6 shows the shear force and bending moment distributions from **Eqs. 4** and **5** for a 20-m-length beam and *w* = 5000 N/m. The shear force presents a linear variation with the horizontal distance from the wall and the bending moment a quadratic one.

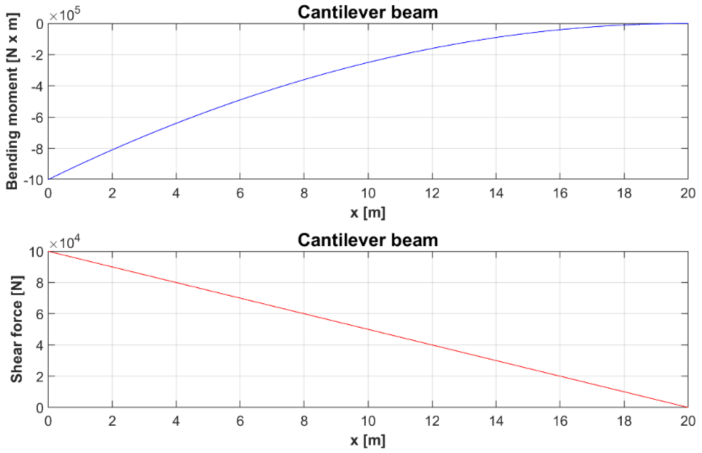


Figure 6: Shear force and bending moment along a cantilever beam

The same calculation procedure is now applied to obtain the distributions regarding a beam with a single strut, also subjected to a uniform lifting load. Assuming that the strut-wing junction is located at one-third of the overall beam length from the wall, the schematics of forces and moments for the whole beam are depicted in **Figure 7**.

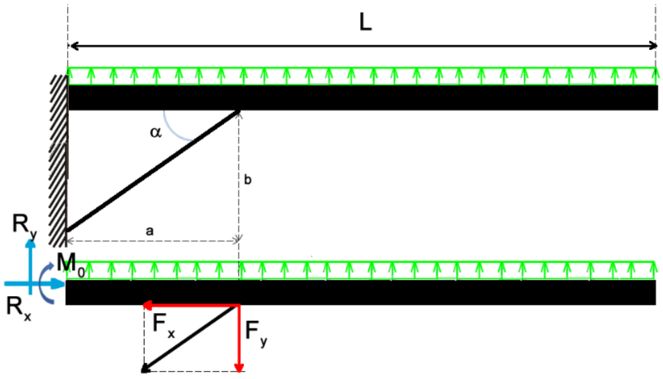


Figure 7: Schematics of reactions of a beam with a single strut

The balance equations for the whole beam for this case are as follows:

$$\sum F_x = 0 \Rightarrow R_x - F_x = 0 \Rightarrow R_x = F_x \tag{6}$$

$$\sum F_y = 0 \Rightarrow R_y - F_y + wL = 0 \tag{7}$$

$$\sum M_{x=0} = 0 \Rightarrow wL\frac{L}{2} - F_y a - M_0 = 0 \tag{8}$$

Considering that the reaction moment *M₀* at the wall is zero, we obtain the vertical force acting on strut:

$$F_y = \frac{wL^2}{2a} \tag{9}$$

Now, F_x can be obtained:

$$F_x = \frac{F_y}{tg\alpha} = \frac{a}{b}F_y = \frac{wL^2}{2b} \Rightarrow R_x = \frac{wL^2}{2b} \tag{10}$$

For the obtention of shear force and bending moment distributions, it is necessary to make the force diagram of generic beam elements of the inner and outer beam (**Figure 8**).

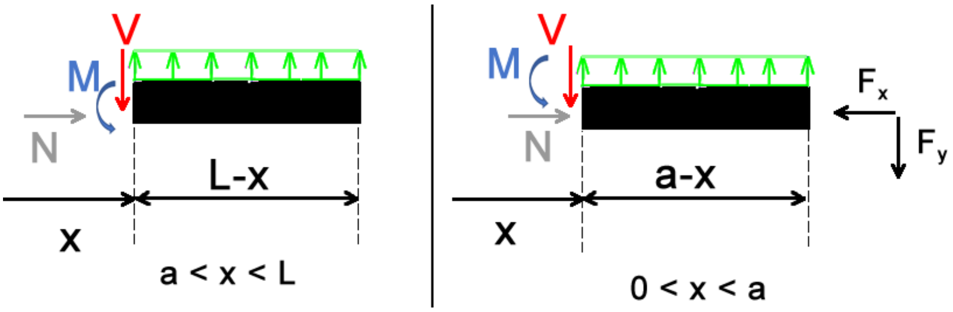


Figure 8: Force and moment schematics for two slices of the beam with strut

Considering the three balance equations for the outer beam, the shear force and bending moment distributions for this part can be easily obtained:

$$\sum F_x = 0 \Rightarrow N = 0 \tag{11}$$

$$\sum F_y = 0 \Rightarrow w(L-x) - V = 0 \Rightarrow V = w(L-x) \tag{12}$$

$$\sum M_{x=a} = 0 \Rightarrow M + w(L-x)\frac{(L-x)}{2} = 0 \Rightarrow M = -\frac{w}{2}(L-x)^2 \tag{13}$$

A straightforward calculation procedure is then carried out for the inner beam:

$$\sum N - F_x = 0 \Rightarrow N = F_x \text{ (compression)} \tag{14}$$

$$\sum F_y = 0 \Rightarrow w(L-x) - V - F_y = 0 \Rightarrow V = w(L-x) - F_y$$
$$\Rightarrow \qquad V = w(L - \frac{L^2}{2a} - x) \tag{15}$$

$$\sum M_{x=0} = 0 \Rightarrow M + w(L-x)\frac{(L-x)}{2} - F_y(a-x) = 0$$
$$\Rightarrow \qquad M = \frac{w}{a}\frac{L^2}{2}(a-x) - \frac{w}{2}(L-x)^2 \tag{16}$$

Figure 9 shows the shear force and bending moment distributions

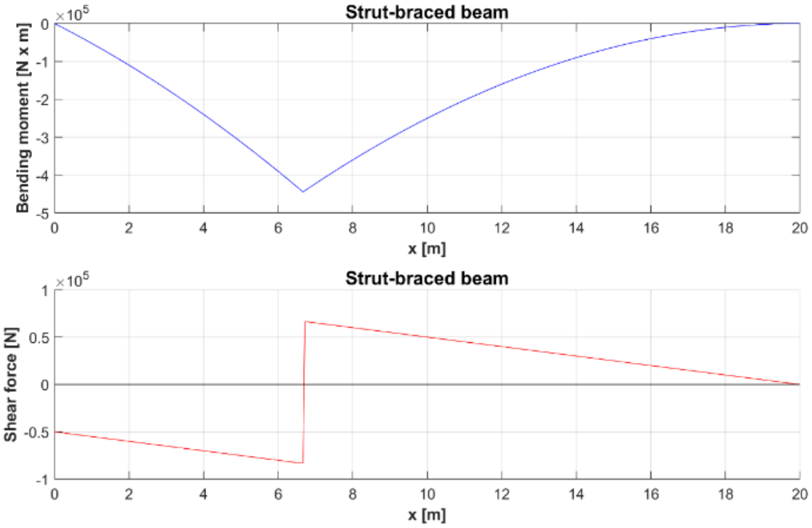


Figure 9: Shear force and bending moment along a strut-braced beam

An advantage of the beam with a strut is a reduction of the bending moment at the root station. However, the inner part of the beam becomes subject to compressive forces and therefore may suffer from buckling more prematurely.

1.3 Dual-fuselage aircraft

The main reason to consider a double-fuselage concept for airliners is the utilization of high aspect-ratio wings without aeroelastic penalties of an equivalent cantilever wing attached to a single fuselage. The stiffness of the resulting set is enough to provide higher flutter speeds.

Figure 10 shows Mach contours and **Figure 11** is a compound image of streamlines and contour surface that resulted from an Euler Simulation of a dual-fuselage configuration able to accommodate 34 passengers. Despite a relatively low freestream Mach number, many supersonic regions can be seen in the calculated flow over the configuration. A strong shock wave affects the wing between the fuselages, probably leading to flow separation at the Mach number of this simulation and higher. Indeed, the flow experiences a high acceleration in the region in between the two fuselages, behaving similarly as flow in a Venturi tube. Thus, the central wing must be carefully designed with additional considerations for a sound and aerodynamically clean design.

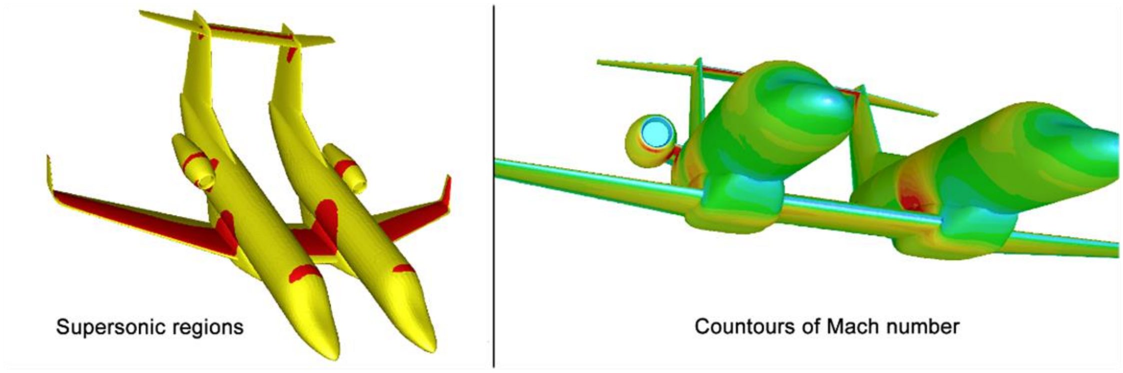


Figure 10: Euler calculation for a double-fuselage configuration (Mach = 0.76, α = 0º). At left: red regions indicate supersonic flow. At right: red color and yellow colors are associated with lower speeds; green color indicates that the local speed is close to the freestream Mach number

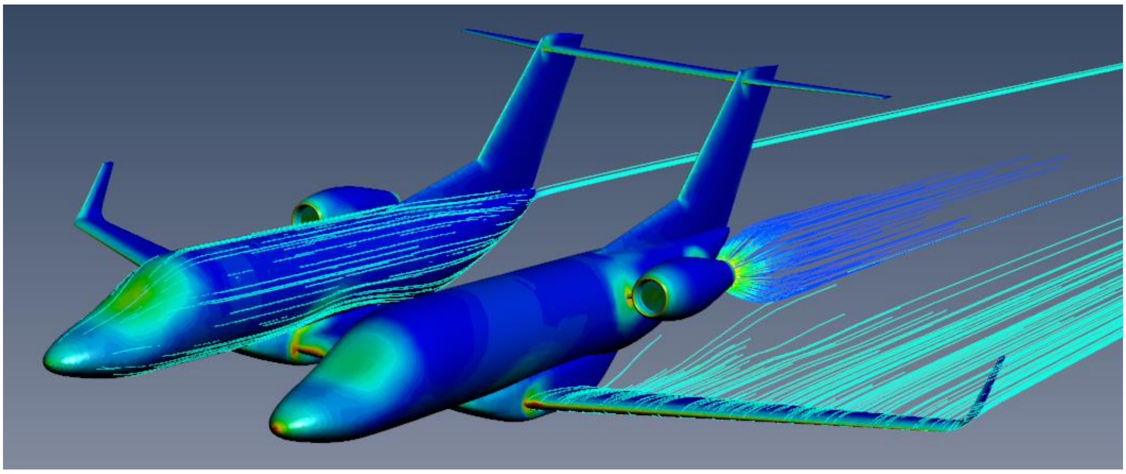


Figure 11: Streamlines over the dual-fuselage configuration (Mach = 0.76, α = 0^o). Surface geometry is colored according to the calculated Mach number. Notice the rapid expansion of engine exhaust gases

To illustrate how important, it is in aircraft design to raise relevant topics that influence the manufacture, performance, and operation of the aircraft, here are some of the advantages of a dual fuselage aircraft and its disadvantages.

On the bright side:

- *Wing with a high aspect ratio is feasible*. The central wing will not have any wingtip (i.e., the wing starts and ends in the fuselage spanwise). The absence of a wingtip reduces the induced drag generated by the central wing. The overall wing could be of a higher aspect ratio contributing to the induced drag reduction. However, as shown before, the central wing must be carefully designed to prevent flow separation and high wave drag coefficients.
- Two smaller vertical tails can eventually save weight, depending on the design requirements and the mission of the aircraft. In addition, in a catastrophic event where one VT is damaged, the remained one may continue to be operated.
- A smaller fuselage will not require bigger production facilities and lighter support equipment around. Fuselage skin thickness is strongly dependent on the fuselage diameter; therefore, the fuselage weight grows exponentially to its diameter. However, are the loads in the flight envelope that will determine the structural sizing of the aircraft.
- *Cargo and passenger airplane*. Each fuselage could separately accommodate cargo and passengers facilitating handling and operation for this type of aircraft.
- *A dedicated fuselage for the business class passenger*. One of the fuselages could be configurated to accommodate business and/or first-class passengers only, allowing this way an exclusive service for them.
- It provides some advantages to specialized airplanes such as Scaled Composites VMS Eve (**Figure 12**). This aircraft is an example of a mother ship that carries a parasite aircraft between the two fuselages, releasing it later to perform a high-altitude flight or a sub-orbital spaceflight.

Some disadvantages of dual-fuselage configs are given as follows:

- Disturbances in airflow over the wing caused by the two fuselages may lead to higher interference drag and negatively impact the load distribution, which will contribute to an increase in the induced drag.
- *Larger rolling moments*: A dual-fuselage airplane will need larger rolling moments and hence, more effective ailerons.
- The central wing must keep the two fuselages together. So, the wing must take a bit of tensile/compressive loads. The forces separating the two fuselages are not the dominant forces, but still, something to keep in mind while designing. Also, the middle wing must maintain the two fuselages at the same pitch, so add a bit of twisting moments to the wing design, apart from aerodynamic twisting forces.
- *Roll stability*: In addition to the traditional parameters affecting the roll stability, i.e., the outer wing sweep and the outer wing dihedral, now there is a new parameter in the picture, the position of the two fuselages. I don't know if the twin-fuselage configuration has reduced or better roll stability, but either way, it complicates the things for achieving reasonable roll stability margins.
- A higher number of control surfaces is a challenge to the control system design. The by-wire system is a must for modern double-fuselage airliners.
- Ground operations will become more complex and difficult such as passenger boarding and deplaning, waste servicing, and catering.

In the past, some dual-fuselage configurations were developed **(Figure 13**). The Heinkel He 111Z was a combination of two existing He 111 tactical bombers. It was produced in a few units just to tug the huge transport glider Messerschmitt Me 321. To take advantage of the war surplus of P-51 Mustangs, a dual fuselage variant was developed for aerial reconnaissance missions (**Figure 13**).

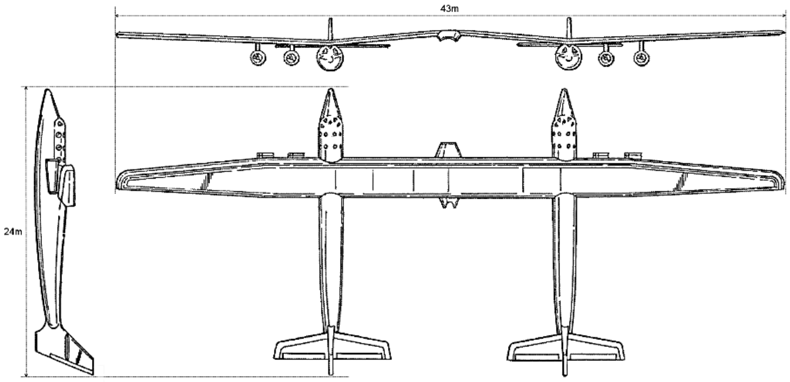


Figure 12: Scaled Composites VMS Eve (Drawing released to the public domain)

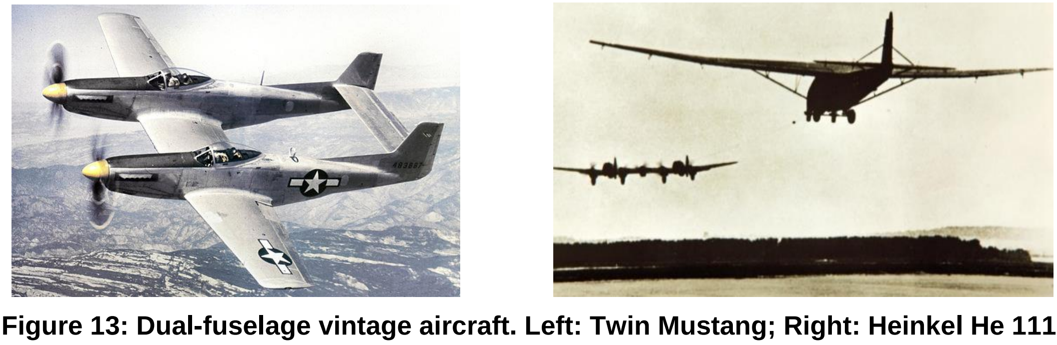


Figure 13: Dual-fuselage vintage aircraft. Left: Twin Mustang; Right: Heinkel He 111Z (Public domain photos)

2. A simple method to calculate wing structural weight

The primary structure of wings is represented by the torsion box or wingbox. An example of the structural layout of the torsion box of a wing is given in **Figure 14**. As can be seen in 1, the primary wing structure is something complex, and it is not possible to scale in a simple and fast way, the main objective of this work. For this, a simplified torsion box model is considered. Megson [22] elaborated a simplified wingbox model for its sizing, which was adopted for the calculation of the wing structural weight of some airliners (**Figure 15**). Stringers and spar flanges are replaced by concentrations of area, known as booms, over which the direct stress is constant, and which are located along the lines representing the skin. The content of this Section is an extended work of Videiro [23].

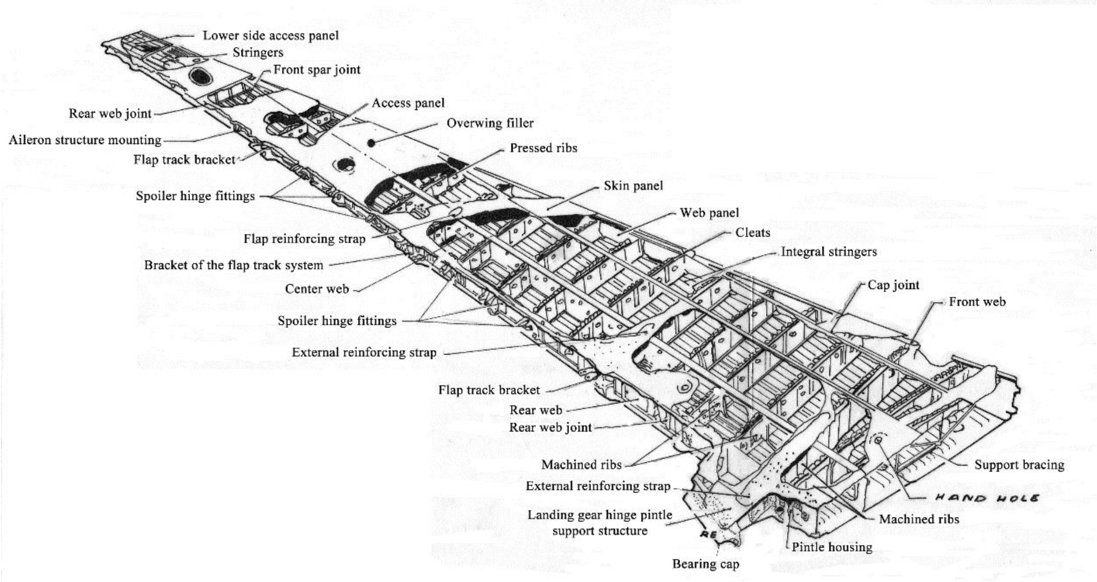


Figure 14: Illustration of a wingbox of a cantilever wing (Drawing from ITA's Aircraft Design Department)

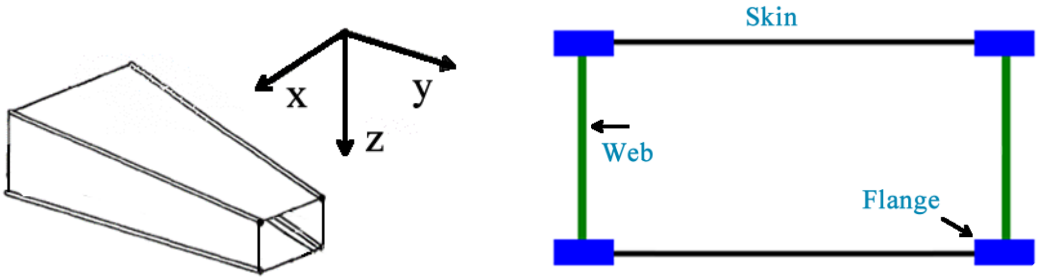


Figure 15: Simplified model of a wingbox

Calculating stations

Here are considered airplanes with a constant leading-edge sweep and a trailing edge with a single inflection, the latter called break station. Due to the wing structure complexity, the structural model is discretized into five distinct sections along the wingspan: wing root station, wing break station, and two sections beyond it (**Figure 16**). It is noteworthy that the number of stations can be changed by the user as well as their location.

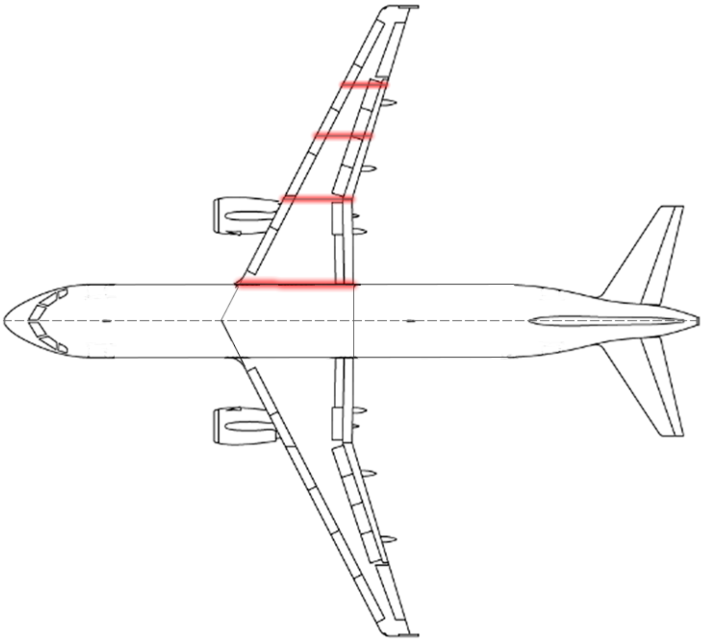


Figure 16: Calculating stations

External loading

Each section is sized individually. The loads are calculated by a wing-body full potential code and properly transferred to the sections. There are cases where the fuel weight is considered, and others are not. As the mission proceeds, some airplanes are continuously pumping fuel stored in the wings to a central tank in the fuselage to increase the fatigue life of the structure.

Another routine built the structural layout according to prescribed rib spacing and spar locations **(Figure 17**). The fuel storage capacities for the internal and external tanks are also calculated as well as the center of gravity of tanks considering them filled with fuel.

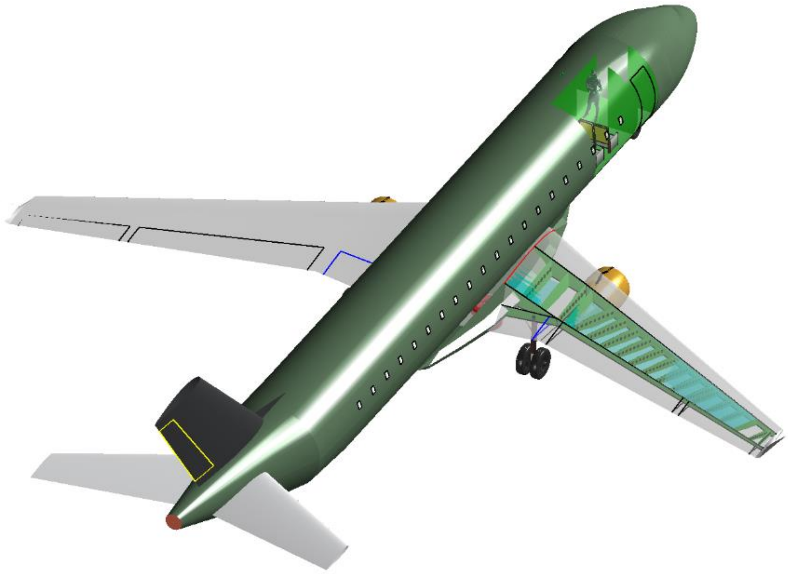


Figure 17: Example of wing layout for a 72-passenger airliner fitted with underwing **engines**

Internal stresses

The internal stress in each case is represented by the tensor of forces:

$$\tau = \{Rx, R_y, Rz, Mx, My, Mz\} \tag{17}$$

The reference system is indicated in **Figure 18**, and it has its origin in the geometric center of the section which due to the hypotheses of structural idealization that was adopted will coincide with the center of sharp stresses of the torsion box.

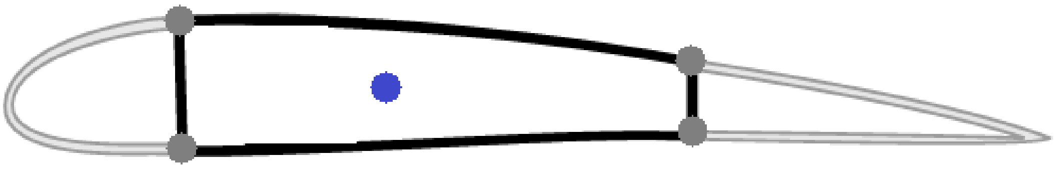


Figure 18: CEC Position (Cutting Stress Center)

Consequently, shear flows arise in the stringers and panel skins that need to be calculated for the very dimensioning of these components. Also, there will be axial forces, due to the actuation of bending moment, which, as already mentioned, are supported exclusively by the boons. Each section is treated as an isosceles trapezium to simplify structural calculations (**Figure 19**).

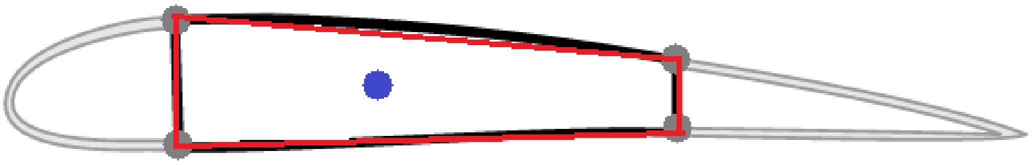


Figure 19: Approximation of the Wingbox Cross Section as an Isosceles Trapezium

The distance between the position of the geometric center of the trapezium and the front stringer is given by:

$$d_{CG} = \left(\frac{wbi}{3}\right)\frac{2h_{lweb} + h_{tweb}}{h_{lweb} + h_{tweb}} \tag{18}$$

Determination of the area of the idealized sections

For the center of mass to coincide with the geometric center of the section, the boom areas after idealization must be equal.

The moments of inertia of the section are given by:

$$I_{xx} = \frac{S_{Boom,i}h_{lweb}^2}{2} + \frac{S_{Boom,i}h_{tweb}^2}{2} \tag{19}$$

$$I_{zz} = \frac{S_{Boom,i}{d_{CG}}^2}{2} + \frac{S_{Boom,i}(wbi - d_{CG})^2}{2} \tag{20}$$

The axial stress at a generic point can then be expressed as:

$$\sigma(x,z) = \frac{M_x z}{I_{xx}} + \frac{M_z x}{I_{zz}} \tag{21}$$

According to **Eq. 21**, it is possible to deduce which of the four booms will present the highest axial stress and will consequently be the most critical for sizing. Considering that the rupture stress of material that constitutes the wing is known. the *critical* boom can be dimensioned. In this step, the idealized area of the boom and not its real area is employed. At the end of the sizing procedure is when the thicknesses of the stringers and the skin will be known.

Shear flow

Figure 20 shows an example of shear force due to aerodynamic loading for a load factor, *n*, equal to 1.

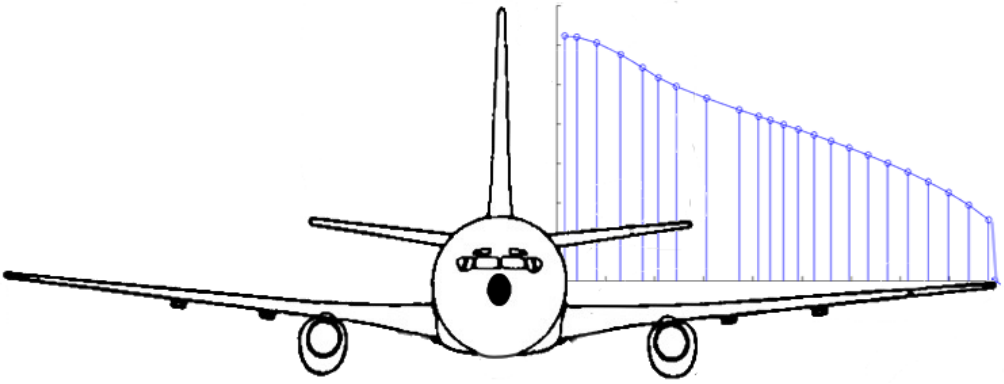


Figure 20: Example of shear force distribution along semispan of jet transport aircraft ((M∞ = 0.80, C$_L$=0.48, *n*=1)

To find the values of the shear flow of a closed section, the method presented by Megson [22]. The shear flow is divided into two parts:

$$q = q_B + q_{s,O} \tag{22}$$

In **Eq. 22**, q_B is the flow calculated after making a "cut" on one of the walls of the closed section. This flow varies depending on the path around the perimeter of the section:

$$\begin{aligned} q_B = &-\left(\frac{S_x I_{xx} - S_y I_{xy}}{I_{xx} I_{yy} - I_{xy}^2}\right)\left(\int_0^s t_D x ds + \sum_r^n S_{Boom,i} x_r\right) \\ &- \left(\frac{S_y I_{yy} - S_x I_{xy}}{I_{xx} I_{yy} - I_{xy}^2}\right)\left(\int_0^s t_D y ds + \sum_r^n S_{Boom,i} y_r\right) \end{aligned} \tag{23}$$

As the section is symmetrical concerning the axis x => $I_{xy} = 0$. Besides, the idealization assumes that the thickness is null, $t_D = 0$, so that the expression of **Eq. 23** can be simplified:

$$q_B = -\frac{S_x}{I_{yy}} \sum_r^n S_{Boom,i} x_r - \frac{S_y}{I_{xx}} \sum_r^n S_{Boom,i} y_r \tag{24}$$

Eq. 24 indicates that the shear flow is constant on the same wall and any variation is caused solely due to the presence of a boom. The contribution of $q_{s,O}$ is constant on all walls of the closed section. Its function is to equalize the torque produced by the external forces with the internal torque that will arise due to the internal shear flow. Therefore, knowing the external forces and the idealized area of the booms, shear flows can be determined and consequently size the thickness of the stringers and skin.

Shear stress is given by the division between shear flow and material thickness. Thus, smaller thicknesses result in higher shear stresses.

$$\tau = \frac{q}{e} \Rightarrow e = \frac{q}{\tau_{MAX}} \tag{25}$$

The last step of structural sizing is to revert the idealization of the structure to obtain the actual areas of the booms:

$$B_1 = S_{Boom,i} - \frac{e_{lweb} h_{lweb}}{6} - \frac{e_{upanel} wbi}{2} \tag{26}$$

$$B_2 = S_{Boom,i} - \frac{e_{tweb} h_{tweb}}{6} - \frac{e_{upanel} wbi}{2} \tag{27}$$

$$B_3 = S_{Boom,i} - \frac{e_{tweb} h_{tweb}}{6} - \frac{e_{lpanel} wbi}{2} \tag{28}$$

$$B_4 = S_{Boom,i} - \frac{e_{lweb} h_{lweb}}{6} - \frac{e_{lpanel} wbi}{2} \tag{29}$$

Rib sizing

Contrary to what was performed for the sizing of the coating, stringer, and booms, a statistical and historical approximation is used to determine the mass of the ribs. This is due to failed attempts to scale this component, whose main structural function is to resist buckling to ensure the shape of the ass when loaded.

Another difficulty presented was the fact that the ribs of the aircraft presented holes for the passage of hydraulic components and weight alleviation, for example, or to form the fuel tank inside the wings.

The solution adopted was to use an area density for the ribs, $\rho_{Nerv} = 9.6 kg/m^2$. The average spacing between the ribs on a passenger transport aircraft is 0,70 m. Thanks to the planform known data and the location of the rear and front spars, and other data a routine was built to define the structural layout and therefore all positioning and some dimensions of ribs in the wing. The mass of all ribs is given as:

$$M_{Nerv} = \rho_{Nerv} S_{Nerv} \tag{30}$$

Where is the sum of the areas of all cross-section sum of the areas of the cross-section where the ribs are located, S_{Nerv}.

Model for the secondary structure

In the previous chapter, a model for calculating the mass of the primary structure of the wing was elaborated. Now, it remains to do the same procedure for the secondary structure (flaps, spoilers, hydraulic systems, etc.) to obtain a complete model of the wing of an aircraft. However, due to the complexity of scaling the devices of the secondary structure, a statistical approach is used to estimate the mass.

The difficulty comes, among other factors, from the wide variety of types of hyper-support devices that an aircraft can have:

In addition, the sizing of systems such as hydraulics is a complex task, with a level of difficulty that escapes the objectives of this work and that are not compatible during the conceptual design stage of the aircraft.

Historically, the mass of the secondary structure contributes with 20-30% mass of wings. This enables a very simple first mass estimation:

$$M_{Wss} = 0.26 M_W \tag{31}$$

Torenbeek [24] proposed a method to estimate the mass of the secondary structure with more accuracy. This method requires that the area of high-lift devices be known. However, due to unsatisfactory results when compared to the actual mass of existing aircraft wings, its use was then not here considered [25]. The more accurate formula proposed by Roux [25] is to assume that the mass of the secondary structure is proportional to the power of the wing area.

$$M_W = K\, S^n \tag{32}$$

Being the factors and adjusted from historical values, minimizing the quadratic error between the model and the actual mass. $K = 25.9$ and $n = 0.97$ for airplanes with MTOW higher than 20 t.

Once the tensor of stresses is constructed, the design of the primary structure of the wings can be started. The sizing of the primary structure can be divided into two cases:

- Sizing of *continuous* structures along with the aircraft's wings, such as stringer, skin, and reinforcers. The structural module must be written in a routine that from the geometry of the wings and the force tensioner returns the area of each section to be sized.
- For the sizing of the *discontinuous* structures along the aircraft wing, more specifically the ribs, the same subroutine must provide the total volume of this structure throughout the whole wing.

Two test cases were run with aircraft like B737-200 and A320-200 (**Tables 3** and **4**). Since airfoil geometry and other relevant information are unknown; airfoils presenting the same maximum relative thickness as those of the real airplanes were utilized, instead. Overall wing mass estimated using the Torenbeek method [24] is also provided. The MZFW that is required by the Torenbeek approach was furnished based on the fuel capacity calculated by the present methodology considering that fuel is stored only on the wings.

Table 3: Airplanes selected to test the methodology

	Similar to Boeing 737-200	**Similar to A320-200**
Seating abreast in the Y-class	6	6
Single class passenger capacity (32-in pitch)	132	180
Wing aspect ratio	8.83	9.38
Wing taper ratio	0.266	0.240
Wing area [m^2]	91.04	122.4
Location of the front/rear spar	23/64% chord	23/64% chord
Rib spacing (in)	22	22
tc_root	15.4% at 19.6% chord	15.13%
MMO	0.82	0.82

Table 4: Mass estimation compared with actual on

	Similar to Boeing 737-200	**Similar to A320-200**
Estimated wing mass (primary/secondary/total)	3432/2060/5492	5950/2744/8694
Mass according to Torenbeek [24]	5346 kg	8285 kg
Actual overall wing mass [25]	5038 kg	8766 kg

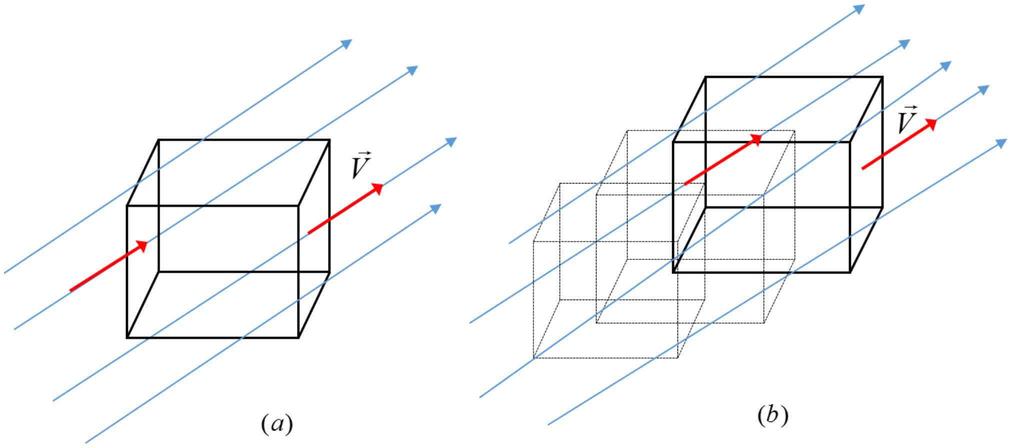
3. A fluid-structure simulation for capturing elastic effects on wing structure

The objective of this Section is to highlight the importance of considering aeroelastic effects on the aircraft conceptual design. For this purpose, aeroelastic effects observed in a military jet airplane were computed by a fluid-structure interaction (FSI) computer simulation procedure and compared to available experimental data. The effects of elastic characteristics of the configuration on pressure coefficient distribution were deeply investigated. The detailed aircraft wing structure of the KC-135 aerial refueller was built by using preliminary sizing procedures based on estimated main aerodynamic loadings, flight enveloped, V-n diagram, and detailed structural layout reports. The simulation results are in excellent agreement compared with wind tunnel and flight-test data.

3.1 Fluid-structure interaction models

The modeling of fluid-structure interaction can be classified under three major branches, following the coupling strategy: fully coupled, loosely coupled, and closely coupled analyses.

The fully coupled approach combines structural equations of motion and fluid dynamics equations in integral form and solves the related time-discretized equations. However, such a procedure requires constructions that produce matrices with different orders of magnitude for solids and fluids because they are written for different modeling reference frames (**Figure 21**). This hinders the discretization of the problem, limiting the construction of the grids and being expensive from the computational point of view, usually being used in two-dimensional problems [26].



(a) Eulerian control volume

(b) Lagrangian control volume moving and deforming with fluid flow

Figure 21: Reference frames

Loosely coupled approaches [27], unlike fully coupled strategy, structural dynamics and fluid dynamics are solved using separated solvers and have different computational grid topologies in which boundaries are non-coincident. An external connection between fluid and structure modules is made. The fluid solver, in general, has reasonable techniques to export forces to the interface with the structure solver.

The closed approach is one of the most widely used methods in the field of computational aeroelasticity. Fluid models and structures are solved in different solvers. However, it has an internal connection module that exchanges information between the two models, which intercommunicate iteratively (**Figure 22**).

The information exchanged are the surface loads, mapped on the CFD surface grid and transmitted to the structural dynamics (CSD) grid, which maps the displacement field, transferring this information to the CFD solver [28].

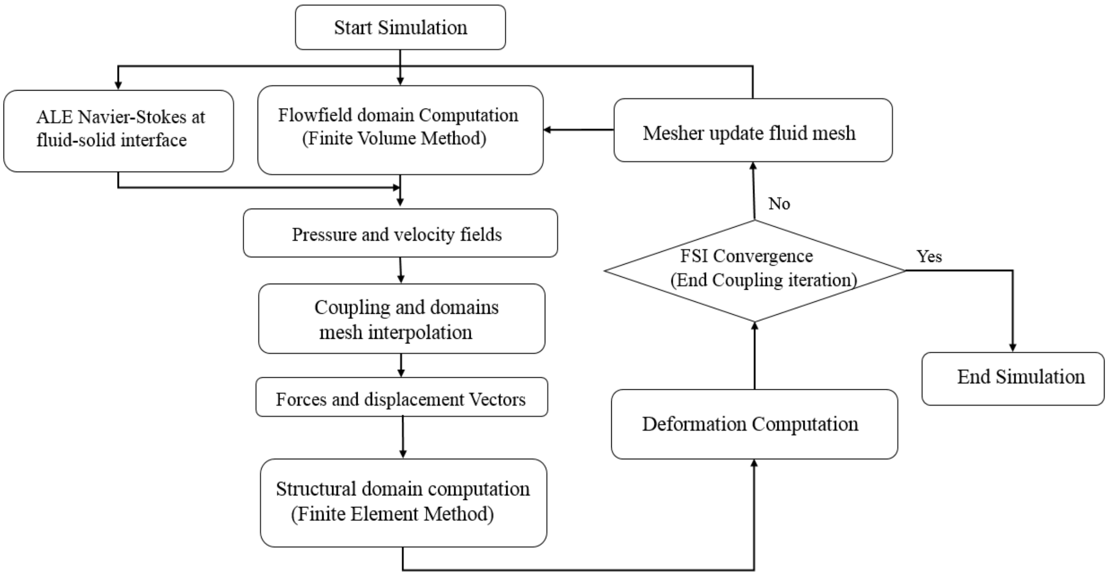

Start Simulation
ALE Navier-Stokes at fluid-solid interface
Flowfield domain Computation (Finite Volume Method)
Mesher update fluid mesh
No
Pressure and velocity fields
FSI Convergence (End Coupling iteration)
Yes
Coupling and domains mesh interpolation
End Simulation
Forces and displacement Vectors
Deformation Computation
Structural domain computation (Finite Element Method)

Figure 22: Fluid-structure coupling procedures

3.2 Fluid Flow Dynamics Fundamental Equations

In this section, an overview of the fluid dynamics equations for an Eulerian reference frame is given.

The fluid model assumes a linear relationship between viscous stress and strain:

$$\sigma_{ij} = \mu\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} - \frac{2}{3}\frac{\partial v_k}{\partial x_k}\delta_{ij}\right) - p\delta_{ij} \tag{33}$$

The laws of conservation of mass, motion, and energy may be written as:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho v_i)}{\partial x_i} = 0 \tag{34}$$

$$\frac{\partial(\rho v_i)}{\partial x_i} + \frac{\partial(\rho v_i v_j)}{\partial x_j} = \frac{\partial}{\partial x_j}\left[\mu\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} - \frac{2}{3}\frac{\partial v_k}{\partial x_k}\delta_{ij}\right)\right] - \frac{\partial p}{\partial x_i} + \rho f_i \tag{35}$$

$$\frac{\partial(\rho e)}{\partial t} + \frac{\partial(\rho v_i e)}{\partial x_i} = \mu\left[\frac{\partial v_i}{\partial x_j}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right) - \frac{2}{3}\left(\frac{\partial v_i}{\partial x_i}\right)^2\right] - p\frac{\partial v_i}{\partial x_i} + \frac{\partial}{\partial x_i}\left(\kappa\frac{\partial v_i}{\partial x_i}\right) + \rho q \tag{36}$$

where ρ is the density of the fluid,f_i are the external forces, and q represents heat sources.

The realizable $\kappa - \epsilon$ turbulence model is well known, and it was employed in the computations of the present Section. Enhanced wall treatment was also chosen, where the entire domain is subdivided into a viscous-affected near-wall region and a fully turbulent region determined by a wall-based turbulent Reynolds number and interpolated by enhanced wall functions that approximate expected boundary layer behavior and the region between viscous-affected and fully turbulent is represented by a blending function.

A finite volume method was used to solve numerically partial differential equations of fluid flow. It consists of the approximation of integral by applying Gauss's divergence theorem. Finite volume methods integrate the partial differential equations over each control volume to find discretized equations to each control volume, conserving physical quantities (e.g., mass, heat transfer) to each control volume.

This method is applied over a conservation equation reformulated into the general form, called as general transport equation [29], written in its differential form:

$$\frac{\partial \rho\phi}{\partial t} + div(\rho\phi u) = div(\Gamma_{\phi}\nabla\phi) + S^{\phi} \tag{37}$$

Eq. 38 was written in divergence form to allows application of Gauss's theorem, that follows:

$$\frac{\partial}{\partial t}\int_{\Omega} \rho\phi d\Omega + \oint_{\Gamma} J \cdot n d\Gamma = \int_{\Omega} S^{\phi} d\Omega \tag{38}$$

Integrating the generalized transport equation considering a time interval Δt in the control volume Ω, the following finite volume equation can be obtained:

$$\frac{(\rho\phi_p)^{t+\Delta t} - (\rho\phi_p)^t}{\Delta t}\Delta V + \sum_{f}^{N_{faces}} \rho_f V_f \phi_f \cdot A_f = \sum_{f}^{N_{faces}} \Gamma_{\phi}\nabla\phi_f \cdot A_f + S^{\phi}V \tag{39}$$

For spatial discretization, it is assumed that properties are constant on a given control volume at a given time in its center of gravity, varying these properties along with elements through interpolation functions in which the most known are centered, upwind (first-order or second-order accuracy), and hybrid, based on Péclet number.

The solvers generally used to handle these equations are pressure-based ones (SIMPLE, SIMPLE-C, PISO, FSM) [30] [31] and density-based solvers (ROE-FDS, AUSM, HLLC) [31] [32] [33].

The present simulation used is an implicit density-based with ROE-FDS scheme, with a second-order upwind scheme applied in the flow and on turbulence model (calculation of kinetic energy and dissipation rates). The condition of convergence is based on the Courant number for a given cell size and time step.

3.3 Structural dynamics approach

We consider the Lagrangian reference frame and a linear relationship between stress and strain in these models.

$$\varepsilon_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \tag{40}$$

The equations of linear elasticity are derived from linear stress and strain relationship and describe conservation of displacement and are known as equations of motion, in its generalized form, known as Navier-Cauchy equations for linear elasticity.

$$\rho_s \frac{\partial^2 u_i}{\partial t^2} - \nabla \cdot \sigma(u) = 0 \tag{41}$$

To solve numerically **Eq. 41** is used the finite element method (FEM) that divide the solid domain Ω_s (or Ω^s) into multiple finite elements Ω_e and the calculation of the displacement field U_e is performed on each finite element from values calculated on its nodes (points that are part of finite element edges) and interpolated by polynomials.

According to that was described in the preceding paragraph, a linear equation on the element Ω_e can be then obtained:

$$U(x_i, t) = N_e(x_i)U_e(t) \tag{42}$$

where $U_e(t)$ is the nodal displacements unknowns and $N_e(x_i)$ is the interpolation polynomial matrix, polynomials called shape functions (based on Hermite's polynomials).

Using the Galerkin method to discretize $U(x_i, t)$, we obtain the generalized equation of motion for each finite element as follows:

$$m_e \ddot{u}_e + k_e u_e = f_e \tag{43}$$

with elementary mass matrices m_e, stiffness elementary matrix k_e, and external forces vector represented by the element f_e.

Assembling the equations of all elements presented into the domain, we have the generalized equation of motion to the domain Ω_s:

$$[M]\{\ddot{u}\} + [C]\{\dot{u}\} + [K]\{u\} = [F(t)] \tag{44}$$

The equation can be solved using a modal approach in which the solution is composed by eigenvectors of the vibration subproblem, which can be written as n individual equations, corresponding one equation to each vibration shape mode, as follows:

$$\begin{cases} \ddot{u}_i(t) + 2\xi_i\omega_i\ddot{u}_i(t) + \omega_i^2 u_i(t) = f_i(t) & i = 1,2,3,\ldots,n \\ f_i(t) = \Phi_i^T[F(t)] \end{cases} \tag{45}$$

where ω_i is the natural frequency for i^{th} mode and ξ_i is the damping parameter for i^{th} mode. To solve this system is necessary to calculate numerically integrals in the mass and stiffness matrices using Gaussian quadrature and after that, solve the linear system to obtain displacement vector through Gauss-Seidel methods and perform modal extraction through Lanczos Method.

3.4 The fluid-structure interface

In the case of the FSI problem we consider a solid domain Ω^s and a fluid domain Ω^f in contact with each other along with the interface $\Gamma_{f/s}$ (as we can see in **Figure 23**). The density of contact forces on the fluid in the solid domain perspective is written as:

$$t_s = \sigma_s n_s \tag{46}$$

Similarly, the density of contact forces on the fluid in the fluid domain perspective is written as:

$$t_f = \sigma_f n_f \tag{47}$$

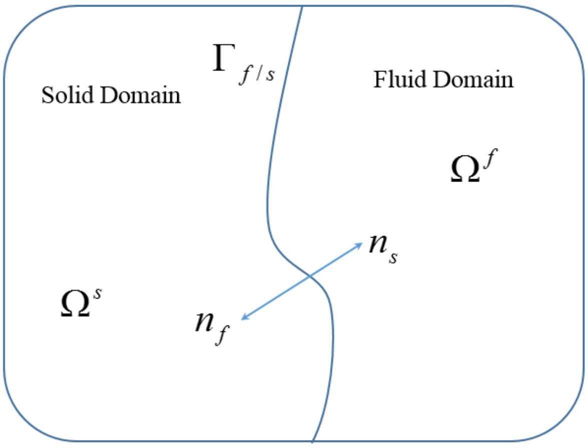


Figure 23: FSI domains

There are the Dirichlet displacement function u and Dirichlet data functions based on velocity vector v on boundary conditions regarding the solid domain. Therefore, two assumptions are made to solve this general problem. The first assumption is that the velocities are continuous along with the interface $\Gamma_{f/s}$, given by:

$$v = \dot{u} \tag{48}$$

The other assumption is relative to the mechanical equilibrium on $\Gamma_{f/s}$

$$t_s + t_f = 0 \tag{49}$$

The normal vectors are linked by the following relationship:

$$n = n_s = -n_f \tag{50}$$

Therefore, the interface equilibrium condition arises:

$$\left(\sigma_f - \sigma_s\right) n = 0 \tag{51}$$

To model meshing motion in the fluid domain is necessary to reconcile characteristics of both reference frames seen in **Figure 21**, to allow carry through a control volume quantity of mass and heat, and at the same time, this control volume may deform and move with time.

To meet these requirements, there is an Arbitrary Lagrangian-Eulerian (ALE) formulation that combines the regions in the fluid domain where we need to approximate a Lagrangian formulation to follow particle motion (for example close to the interface between the fluid and the solid domains) and the regions where an Eulerian description is sufficient to describe the flow (far away from solid domain). This formulation is succinctly described by **Eq. 52**.

$$\begin{cases} \text{Find } (v,p)\in V(\Omega_t^f)^d \times L^2(\Omega_t^f), t\in I \text{ such that} \\ \int\limits_{\Omega_t^f}\left[\left.\frac{\partial v}{\partial t}\right|_x + (v-w)\cdot\nabla v\right]\upsilon d\Omega = \int\limits_{\Omega_t^f}\left[p\nabla\cdot\upsilon - \frac{2}{\mathrm{Re}}\nabla v : d(\upsilon) + b_f\cdot\upsilon\right]d\Omega,\ \forall\upsilon\in V(\Omega_t^f)^d \\ \int\limits_{\Omega_t^f}(\nabla\cdot v)q d\Omega = 0,\ \forall q\in L^2(\Omega_t^f) \end{cases} \tag{52}$$

Applying the Galerkin Method, substituting velocity and pressure fields by its discretized versions and the original spaces by approximation spaces and its respective test functions, we have:

$$\begin{cases} \text{Find } (v_h, p_h)\in M_H \times P_H \text{ such that} \\ \sum_{i=1}^{2K} v_i \int\limits_{\Omega_t^f} q_l \nabla\cdot\upsilon_i d\Omega = 0 \\ \sum_{i=1}^{2K}\frac{\partial v_i}{\partial t}\int\limits_{\Omega_t^f}\upsilon_k\cdot\upsilon_i d\Omega + \sum_{i=1}^{2K} v_i \int\limits_{\Omega_t^f}\upsilon_k\cdot(v-w)\cdot\nabla\upsilon_i d\Omega = \\ \sum_{j=1}^{2K} p_j \int\limits_{\Omega_t^f} q_l\nabla\cdot\upsilon_k d\Omega - \frac{1}{\mathrm{Re}}\sum_{i=1}^{2K} v_i \int\limits_{\Omega_t^f}\nabla\upsilon_k : (\nabla\upsilon_i + [\nabla\upsilon_k]^T)d\Omega \\ \forall(\upsilon_k, q_l)_{k=1\ldots 2K; l=1\ldots L} \in M_H \times P_H \end{cases} \tag{53}$$

The spatial discretization of this model is realized by choosing special types of elements, Crouzeix–Raviart that allows a discontinuous approximation of the pressure or Taylor-Hood elements which allows continuous approximation of the pressure. The temporal discretization has the following form:

$$\begin{cases} \nabla\cdot v^{n+1} = 0 \\ \left(\dfrac{3v^{n+1} - 4v^n + v^{n+1}}{2\Delta t}\right) + \left(v^* - w\right)\cdot\nabla v^* = -\nabla p^{n+1} + \dfrac{1}{\mathrm{Re}}\nabla v^* \end{cases} \tag{54}$$

with

$$\nabla p^{n+1} = \frac{3}{2\Delta t}\nabla\cdot v^* - \nabla p^n \tag{55}$$

To determine the field velocities, it is necessary to treat the nonlinearities present in the convective term within ALE. This limitation is resolved using the iterative fixed-point method or Picard's method to minimize residues as assures the ALE algorithm convergence.

The position of mesh motion is determined using the spring analogy, suggested by Hartwich and Agrawal [34], in which the strategy was based on the master/slave node relationship between the moving surface points (masterpoints) and vertices located at the other blocks (slave points).

The movement of the masterpoints is based on the displacements obtained from the structural calculation solver. The movement of the slave points depends on the movement of its corresponding master point from point x_m to x_{m+1} and slave points move from x_s to x_{s+1}.

$$x_{s+1} = x_s + \theta(x_{m+1} - x_m) \tag{56}$$

where θ is a decay function that depends on stiffness factor β, in which larger values mean that meshes act like a rigid body and f_{min} ensures optimal remeshing behavior if master nodes present small displacements.

$$\theta = \exp\left\{-\beta \min\left[\left(f_{\min}, \frac{dv}{(\varsigma + dm)}\right)\right]\right\} \tag{57}$$

where dv and dm are spatial gradients and a ς number sufficient small to avoid division by zero.

$$\begin{gathered} dv = \sqrt{(x_v - x_m)^2 + (y_v - y_m)^2 + (z_v - z_m)^2} \\ dm = \sqrt{(x_{m+1} - x_m)^2 + (y_{m+1} - y_m)^2 + (z_{m+1} - z_m)^2} \end{gathered} \tag{58}$$

Figure 24 summarizes fluid-structure Interaction numerical calculation. CFD and CSD (FEM) solvers perform calculations normally in regions where the domain is purely solid or fluid. In the fluid-structure interface region, the ALE solver performs the fluid computation by emulating a Lagrangian fluid particle frame, interpolating information in the solid mesh of the interface region the virtual displacements resulting from the velocity field and the forces from the pressure field (**Figure 24**).

The FEM solver calculates the deformations, and the equilibrium condition of the interface is checked by the coupling module. If this condition is not attained, the coupling module sends the deformations calculated by the FEM solver to the dynamic mesher of the fluid domain, which deforms the mesh and the calculations restart, and the iterations continue until the equilibrium condition is reached.

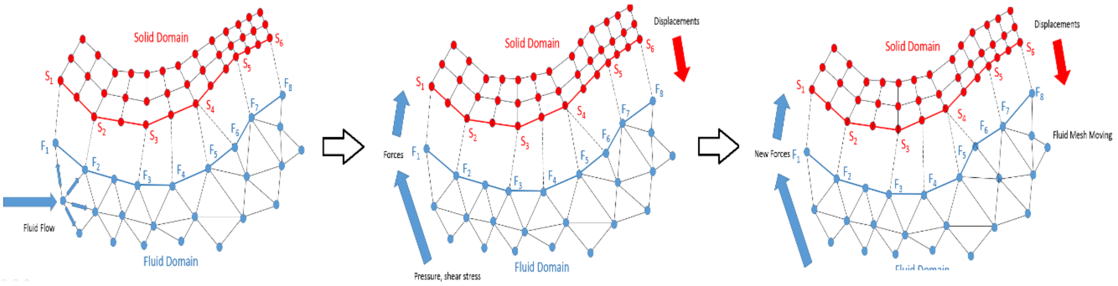


Figure 24: Mesh domains coupling and interpolation

3.5 Airplane model

From reports and other sources of information [35] [36], the external geometry (**Figure 25**) and structural layout of KC-135 could be obtained. The detailed structural layout could also be then defined, as well as the flight envelope of the airplane (**Figure 26**), and the *V-n* diagram (**Figure 27**). The highest load factor is 2.5 and the lowest is -1.

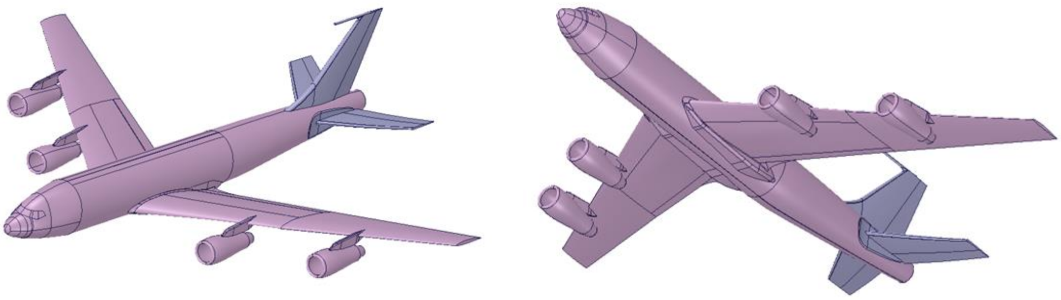


Figure 25: CAD model of the KC-135

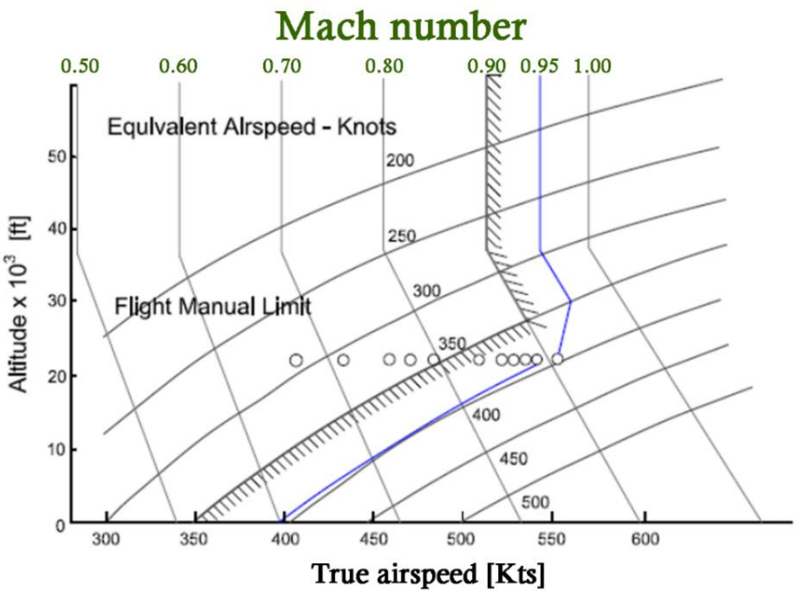


Figure 26: Boeing KC 135 Stratotanker flight envelope [35]

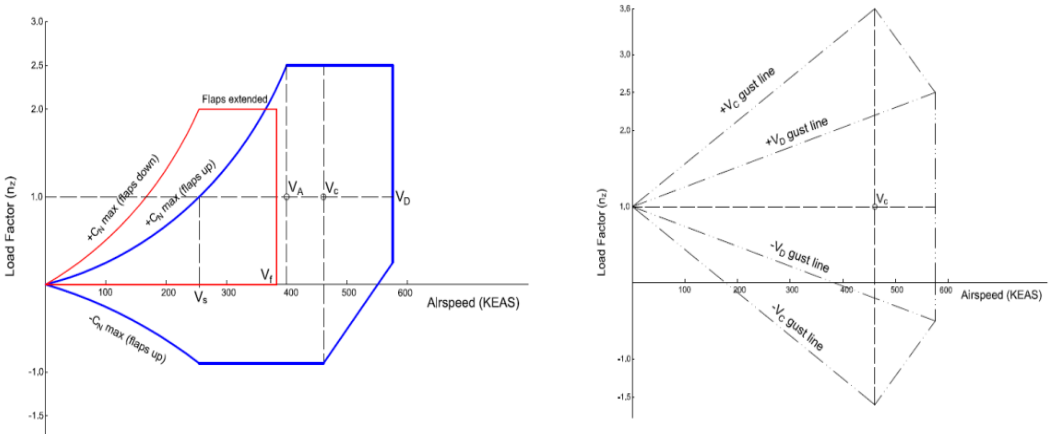


Figure 27: V-n diagram for the structural sizing of KC-135

Figure 28 shows the structural calculation procedure for the sizing of the KC-135 wing. Based on the collected performance data of the aircraft, the loads and load distributions along the wingspan and the chord could be calculated. The load calculation considers cases from maneuvers, gust loads, rolling acceleration, flap and aileron deflections, fuel tank, and engine weights. However, it is necessary to ensure proper safety margins to withstand unforeseen loads at adverse operational conditions.

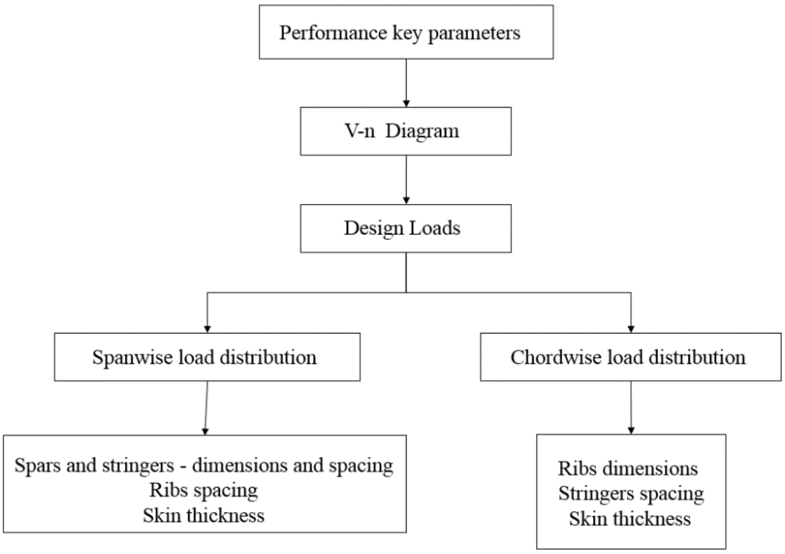

Performance key parameters
V-n Diagram
Design Loads
Spanwise load distribution
Chordwise load distribution
Spars and stringers - dimensions and spacing
Ribs spacing
Skin thickness
Ribs dimensions
Stringers spacing
Skin thickness

Figure 28: Calculation procedure of KC-135 wing structural elements

The spanwise distribution of bending moment and shear force is important for sizing components whose buckling, and shear stresses are significant, such as stringers and spar sections. The inertia moment of the structure also depends on the number of stringers and spacing of both stringers and spars. Chordwise distributions are important for sizing the local buckling of the stringers and the shearing stresses under the ribs. The thickness of the skin is influenced by the spanwise and chordwise loadings that affect the panel buckling at each wing station. Some dimensions of wing structural elements are shown in **Table 5**. The actual dimensions were obtained from [37], [38], and [39]. The calculated values agree very well with them. The rebuilt wing structural layout can be seen in **Figure 29**.

Table 5: Calculated wing structural elements and their comparison with available data

	Calculated	**Data**
The average thickness of wing skin [in]	0.050	0.038
Ribs minimum thickness range [in]	0.025-0.098	-
Maximum allowable spacing of ribs [in]	32.48	26.4-28.5
Maximum allowable spacing of stringers [in]	13.39	12.5
The chordwise relative position of spars [%]	23.5/65	23/65
The average thickness of spars [in]	0.285	0.32
Wing mass [kg]	11320	10859 [25] 11462 [24]
Fraction of wing weight in terms of MTOW* [%]	8.40	8.05-8.50
* *$MTOW_{Ref}$=134838 kg* [24]		

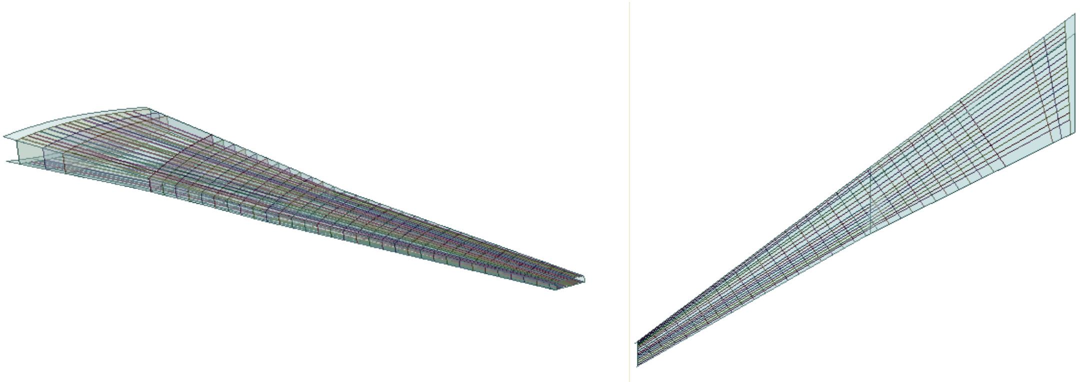


Figure 29: Structural layout of the rebuilt KC-135 wingbox

After the creation of the geometrical model, the CSD and CFD computational meshes were built. The finite element meshes contain quadrilateral shell elements to comply with the skewness criteria of elements as close as possible to 0, which guarantees the accuracy of the analysis (**Figure 30**). In the case of the structural model, the meshes referring to the geometry of the outer surfaces of the wing were defined as the fluid-structure interaction interfaces as well as the communication parameters with the Multiphysics coupling module, discretized in quadrilaterals shell elements with skewness values between 0.22-0.26. CFD meshes were generated using tetrahedral and hexahedral elements to meet the orthogonal quality criteria of cells close as possible to 1, to ensure the convergence and the use of inflation close to the walls, to allow the accurate calculation of the boundary layer in the region through interpolation of the wall functions. The mesh of the fluid model was discretized in tetrahedral elements with orthogonal mesh metrics values between 0.85-0.92.

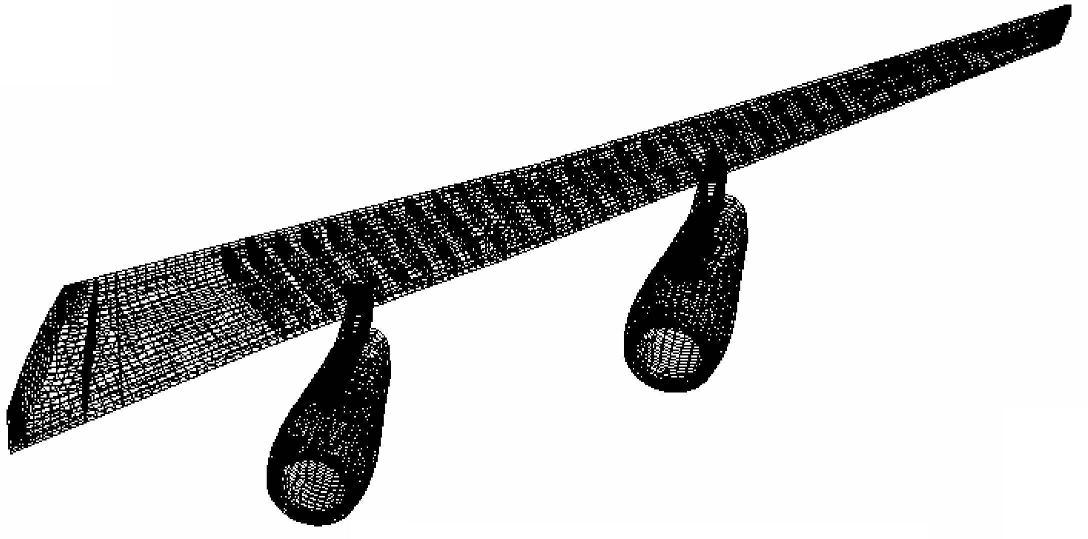


Figure 30: CSD finite element model

3.6 Results

Simulations considering rigid and elastic models were carried out. Their results were compared with wind tunnel data (rigid) and flight test ones (flexible). The wind tunnel runs were carried out for two scaled models: one featuring winglets; the other had no winglets. The flight test is only concerned with a version of KC-135 fitted with winglets. The winglet was also introduced into the computational models, and it features a 15^o dihedral angle and a -4^o twist.

Figure 31 shows the results of a transonic flow simulation for the configuration with no winglets. As expected, there is an excellent agreement between the calculated distribution and the wind tunnel data. **Figure 32** presents two distinct calculated *Cp* distributions for the configuration fitted with winglets: one obtained with a pure CFD simulation; the other considering a simulation encompassing FSI. Again, the fluid dynamic simulation, which considers that the model is rigid, agrees very well with wind tunnel data whereas the simulation considering aeroelastic effects is in perfect accordance with flight test data.

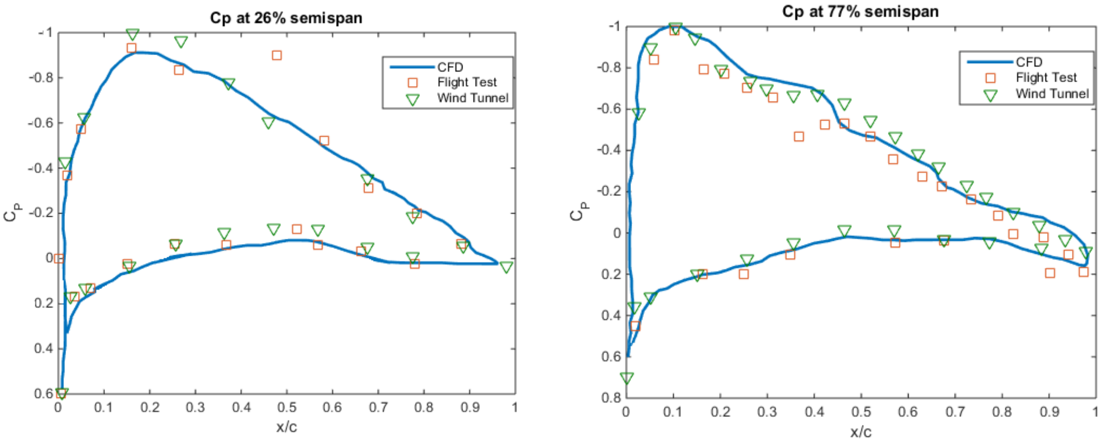


Figure 31: Calculated Cp distributions for the rigid model wing with no winglets compared with wind tunnel test and flight test data (Mach = 0.70, α = 3.5º)

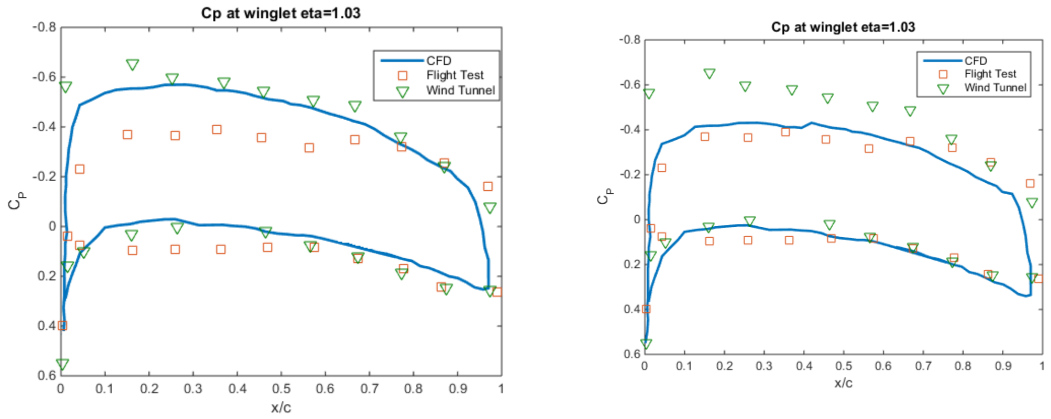


Figure 32: Calculated Cp distributions for the rigid and flexible models fitted with winglets. At left, Cp distribution in a winglet station for a CFD solution. At right, the FSI simulation results agree very well with flight test data (Mach = 0.70, α = 3.5º)

In **Figure 33**, calculated elastic displacements for the baseline configuration without winglets and the KC-135 wing fitted with winglets. The wing fitted with a winglet presents a greater displacement at the wingtip. A wider range of wingtip displacements at a freestream Mach number of 0.76 and for a wider range of lift coefficients can be seen in **Figure 34**. This includes some winglet twist variation for a fixed dihedral angle of 15º.

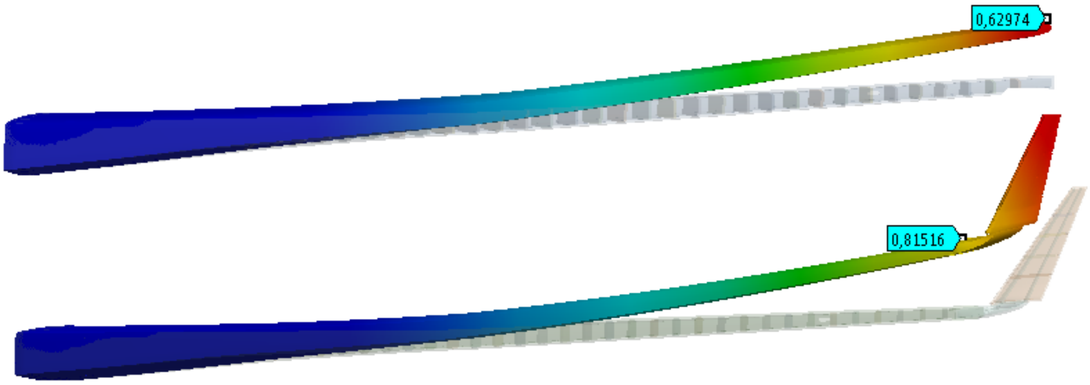


Figure 33: Calculated wingtip displacement in meters (Mach = 0.70, α = 3.5º)

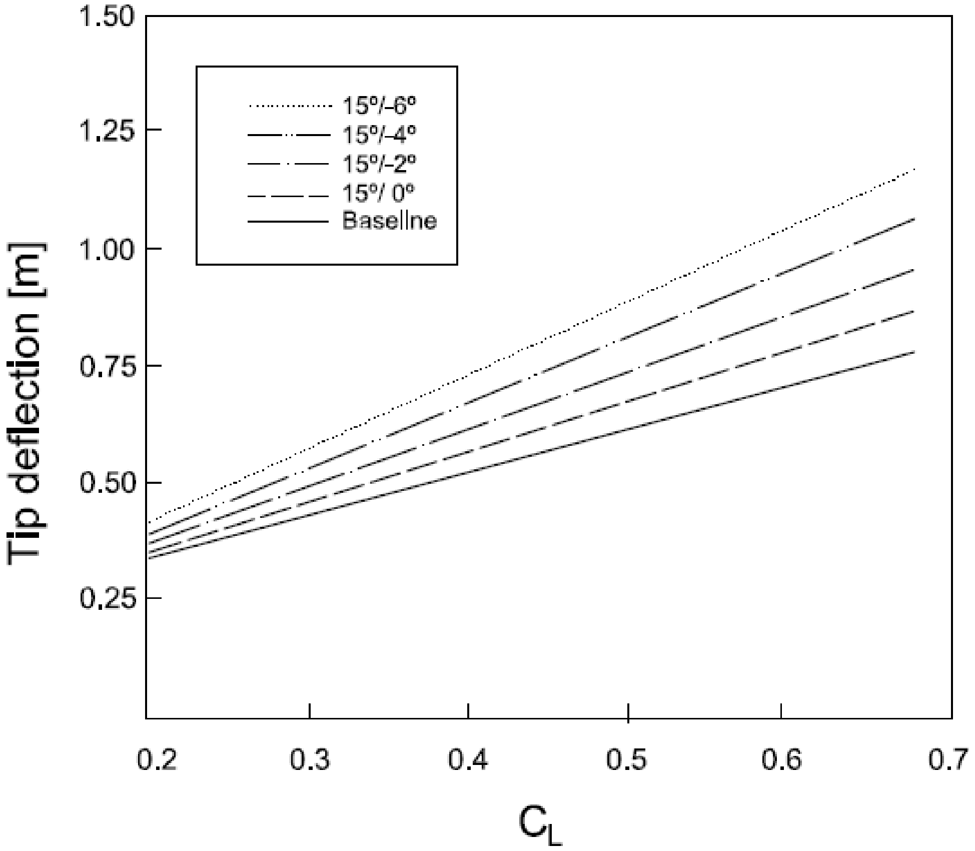


Figure 34: Wingtip deflection for some winglet configurations

Figure 35 reveals that the addition of winglets to the KC-135 configuration reduced the modal frequencies. The addition of winglet weight at the wingtip region, behind the elastic line, therefore, increases the moment of inertia there and may decrease flutter speed. The additional lift generated at the wingtip increases the bending moment at the wing root, demanding a slightly heavier structure. There is an extended aerodynamic effect due to the winglet presence on the configuration. Comparing the lift distribution to the wing without winglets, the unloading of the central and inner part of the wing may further reduce the drag coefficient at transonic conditions due to the eventual reduction of the strength of shock waves there.

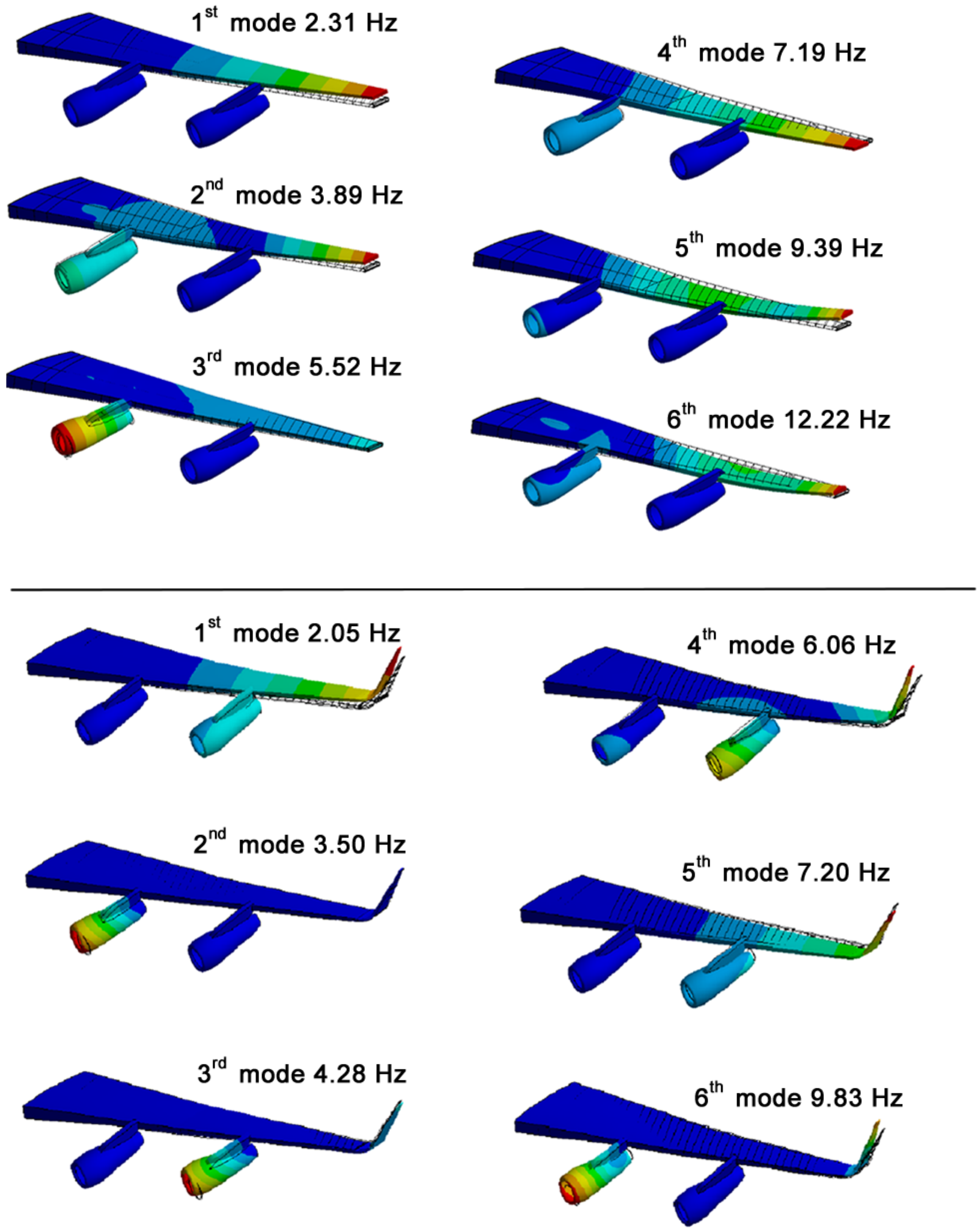


Figure 35: Mode shapes and frequencies of KC-135 wing with and without winglets

4. Considerations about flow modeling

In the applications, concerning optimization of transport aircraft that are shown in the following Sections, the flow model that was adopted for calculating aerodynamics was the full potential one. Because of this, it is important to address the advantages and drawbacks of such a formulation, as well as to justify its use here.

During the computation of airplane mission performance, where critical masses like the MTOW must be accurately calculated, aerodynamic coefficients are intensely required. For this purpose, aerodynamic databanks are usually built up with computational fluid dynamic analyses, wind tunnel data, or a combination of both. In this approach, only the total drag is modeled, and not its components, and this leads to inaccurate interpolations. The alternative to this is the call of aerodynamic codes in real-time. In the case of Euler or RANS formulations, this is not viable, because of the high computation time that is necessary for every flow condition. A realistic mission calculation may require more than one hundred runs for aerodynamic coefficient calculation at each step of the mission. This means that even for full potential codes, which require an average computation time of one minute for a transonic case calculation, an MDO task, where thousands of airplanes must be evaluated, can become a burden for a Linux cluster composed of few processors. A metamodel of aerodynamics offers an elegant and viable alternative to overcome those limitations. Thanks to its faster calculation time, full potential codes are an attractive formulation for the build-up of databases for the training of ANNs.

A viable alternative for MDO computations is the utilization of full potential codes with viscous and non-isentropic corrections [40]. Their cost-benefit makes this formulation extremely attractive for external aerodynamic calculations of aircraft configurations inside an MDO process, which requires extensive calls of the flow analysis codes for performance, load, and stability calculations. However, the extremely sensitive nature of transonic flow regarding airplane geometry, in which geometric variations of lifting surfaces in the order of boundary-layer thickness or lower lead to significant changes in pressure distribution, makes it mandatory that integral boundary-layer routines be coupled with the transonic potential codes. Besides delivering satisfactory zero-lift drag values, the viscous coupled calculation shall also be able to handle shock-boundary layer interaction. Tinoco affirms that the full potential equation offers a quick alternative for analyzing transonic flow with a good degree of accuracy [41]. **Eqs. 59** and **60** are the non-stationary three-dimensional equations of the full potential in three dimensions.

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho\phi_x)}{\partial x} + \frac{\partial(\rho\phi_y)}{\partial y} + \frac{\partial(\rho\phi_z)}{\partial z} = 0 \tag{59}$$

$$\rho = \left[1 - \frac{\gamma - 1}{\gamma + 1}\left(\phi_x^2 + \phi_y^2 + \phi_z^2\right)\right]^{\frac{\gamma}{\gamma - 1}} \tag{60}$$

In **Eq. 60**, γ is the isentropic expansion factor, the ratio of the heat capacity at constant pressure to heat capacity at constant volume. The velocity components on the x, y, and z axes are u, v, and w, respectively, and they can be obtained by deriving the potential for velocities ϕ:

$$\begin{aligned} u &= \phi_x \\ v &= \phi_y \\ w &= \phi_z \end{aligned} \tag{61}$$

For acceptable solutions of the full potential equation, it is recommended that the Cartesian mesh close to the body be normal to it. The full potential equation can be written in strong conservation-law form for a general, body-conforming, curvilinear coordinate system (**Figure 36**). As an example, for the bi-dimensional equation:

$$\left(\frac{\rho U}{J}\right)_{\xi}+\left(\frac{\rho V}{J}\right)_{\eta}=0 \tag{62}$$

Here, ξ is the chordwise coordinate, η is the coordinate in the normal direction regarding the airfoil. *J* is the Jacobian of the coordinate transformation. If velocities are made dimensionless concerning the critical speed of sound (a*), the density can be expressed as

$$\rho=\left[1-\frac{\gamma-1}{\gamma+1}\left(U\phi_{\xi}+V\phi_{\eta}\right)\right]^{1/(\gamma-1)} \tag{63}$$

Eq. 63 also assumes that density is made dimensionless concerning stagnation density. Moreover, the present formulation considers mass and momentum conservation, besides the isentropic hypothesis.

The nomenclature that is typically adopted is the standard one, such that ϕ is the full velocity potential, and the subscripts indicate partial derivatives concerning each of the spatial coordinates. The contravariant velocity components, *U* and *V* are given by

$$U=A_{1}\phi_{\xi}+A_{2}\phi_{\eta} \tag{64}$$

$$V=A_{2}\phi_{\xi}+A_{3}\phi_{\eta} \tag{65}$$

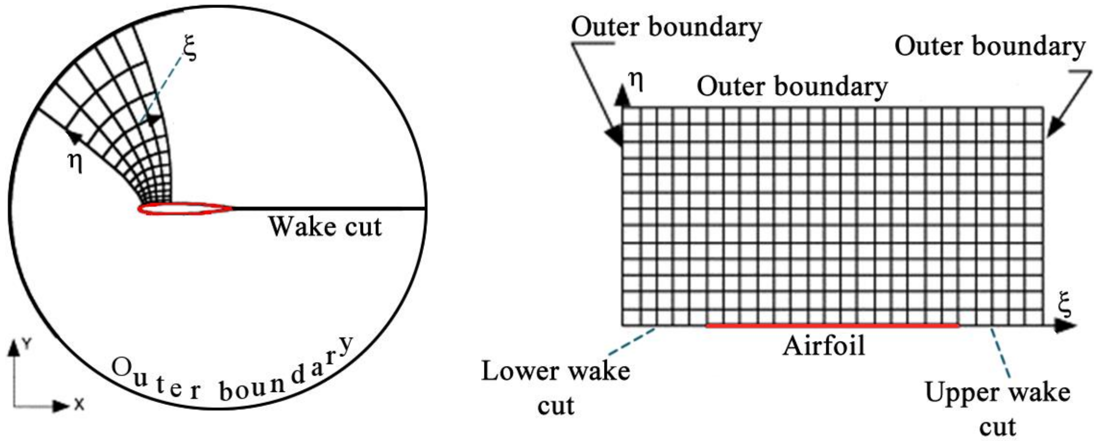


Figure 36: Coordinate transformation and mesh topology

Holst [42] reviewed various potential equation forms with emphasis on the full potential equation. The review also discussed applicable mathematical characteristics and all assumptions adopted for equation derivation. The impact of the derivation assumptions on simulation accuracy, especially concerning models for the capture of shock waves was analyzed. The extensive article also contains the description of key characteristics of all numerical algorithm types employed for solving nonlinear potential equations, including steady, unsteady, both space and time marching, and design methods. Both spatial discretization and iteration scheme characteristics were examined by Holst.

The full potential code employed here presents the following characteristics:

- It solves the three-dimensional full potential equation in the conservative form.
- The airplane configuration to be analyzed is comprised of the fuselage, low wing, and winglet.
- Both subsonic and transonic flow can be calculated.
- The implicit AF2 algorithm [1] is employed for the time-marching in the pseudo time [43].
- A fast viscous-inviscid coupling with a blowing technique is utilized for 3D calculations.
- The location of flow transition can be prescribed. In the present work, the value of 5% of correspondent reference length was used.

A non-isentropic correction is applied to the pressure coefficient (*Cp*) distributions [44].

To demonstrate the accuracy and cost-benefit for employing full potential codes in conceptual design, a comparison of *Cp* distributions from Euler and RANS codes with those from full-potential solutions was carried out for a transonic airfoil flow as well as for the airflow on a twinjet airliner with rear-mounted engines.

The two-dimensional FPE2D full potential code used the formulation described in Mattos [45] and the simulation was used to calculate a transonic flow over an 11.823%-thick airfoil, called here ITADX4. The convergence history is shown in **Figure 37** as well as the Mach number contours of the converged solution. The Euler code used to calculate the flow over the same airfoil at the same conditions is the NSC2KE developed by Bijan Mohammadi [46]. This is a finite-volume Galerkin program computing 2D and axisymmetric flows on unstructured meshes. To solve the Euler part of the equations, a Roe, Osher, and a kinetic solver are available. To compute turbulent flows a k-epsilon model is available [46]. **Figure 38** shows a comparison between the *Cp* distribution on airfoil from the full potential code and the Euler solution. As expected, the conservative full potential solution shock location lies behind that of the Euler solution and it turns out to be a stronger shock wave than that from the Euler formulation. However, if a non-isentropic correction is applied to the full potential solution, both distributions will match closely [44].

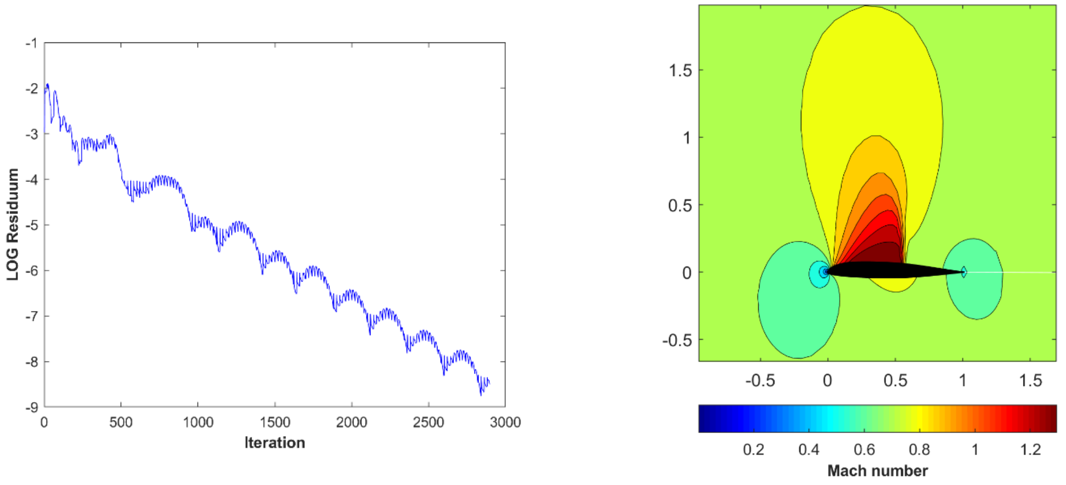


Figure 37: Solution of a transonic flow calculated over the airfoil ITADX4 with a full potential code (M_∞= 0.73 and 1.5° angle of attack)

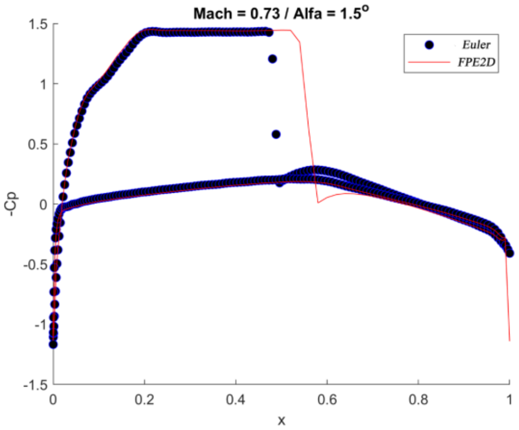


Figure 38: Comparison of *Cp* distributions for the ITADX4 airfoil obtained by an Euler solution and with the one from full potential code

An inverse technique was developed for this code [45] [47]. From a prescribed Cp distribution and an initial airfoil geometry, a new geometry that satisfies the prescribed distribution at the desired flight condition can be found just in a few interactions of the redesign process. The difference between the prescribed and existing pressure distribution is translated into a normal velocity to airfoil or wing geometry. Through an interactive process where the geometry is modified, this difference is progressively eliminated.

A test case that handles the design of a supercritical profile for a specific flow condition was chosen to demonstrate the capability of the inverse design technique. The pressure distribution calculated with the code FPE3D for the profile RAE2822 with Mach number = 0.730 and angle of attack 2° is characterized by a moderate compression shock located at 65% of the chord. The prescribed pressure distribution, shown in dashed lines in **Figure 39**, eliminates the shock wave on the upper surface. The design process converged after 24 cycles and **Figure 39** shows that there is a particularly good agreement between the calculated pressure distribution and the prescribed one. **Figure 39** also compares the RAE2822 profile to the obtained with the inverse technique that typically converges in a few interactions **[47]**. The RAE2822 airfoil was mainly modified on the upper surface after the shock location by a flat curve extension, with its nose geometry suffering almost no modifications. However, the inverse technique presents the disadvantage of undesirable behaviors off the design point. However, conjunctly with optimization techniques, it can be incredibly useful within a multi-objective optimization context.

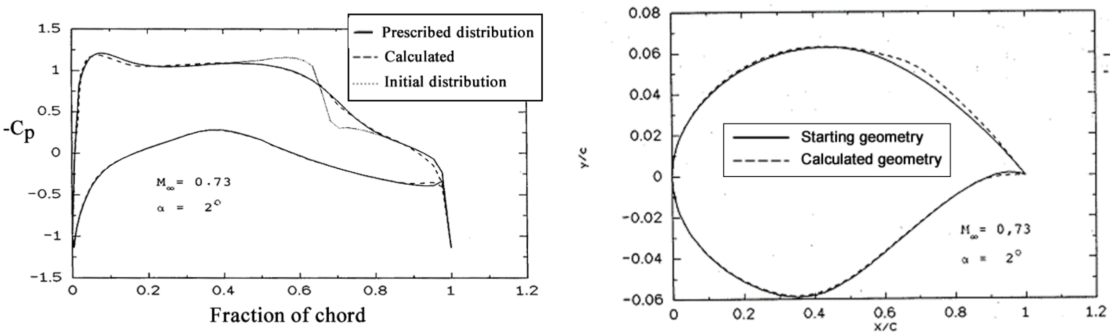


Figure 39: Example of the inverse technique application for airfoil design at a transonic condition (M_∞= 0.73, α=2°) [45] [47]

The TRE planform has a typical shape designed for the transonic regime: the twist between the wingtip and the wing root is 4°, the aspect ratio is 8.4, the leading-edge sweepback angle is 25.5°, a taper ratio of 0.22, and the root profile is 15% thick. The TRE planform can be seen in **Figure 40**. The computing mesh comprises 60x21x30 points in ξ-, η- or ζ-direction, respectively.

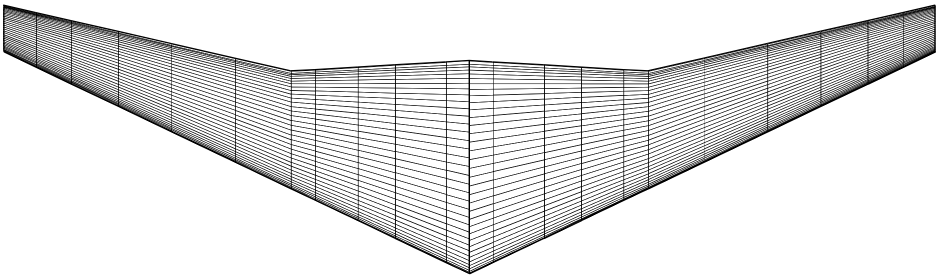


Figure 40: TRE-planform with surface mesh

A complete configuration with the TRE planform was investigated for a wide range of Mach numbers in Boeing's transonic wind tunnel [45]. The configuration includes fuselage, tail units, wings, winglets, and rear-mounted engine nacelles. The experimental measurement data used here were obtained for a variant without a vertical stabilizer. The numerical calculation was performed with the DWING full potential code for wing-alone configuration.

Figure 41 shows the calculated and measured pressure distributions for wing stations at 22% and 66% of the wingspan respectively at a Mach number of 0.70, α=0.365°. For the inboard station, the flow has a moderate compression shock in the front area and is slightly overcritical in the suction tip of the station at 66% of the wingspan. There is a very good agreement everywhere between the experimental Cp values and the pressure distribution calculated by DWING. Due to a missing pressure sample at the bottom of the trailing edge, it is not possible to make a direct comparison of both pressure distributions at this point. In this part of the wing, the viscous effects are of great importance and their effect on the geometry is expressed by a decambering effect of the airfoil geometry in this region.

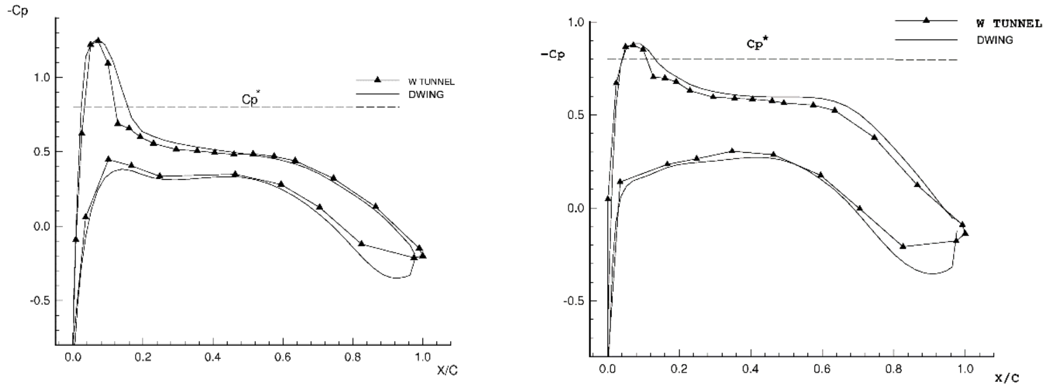


Figure 41: Pressure distributions for a station of the TRE planform at 22% (at left) and 66% (at right) of the wingspan. Mach = 0.700, α = 0.365°

For the analysis of the twinjet airliner designated ITA50SR (**Figure 42**), the RANS simulation [48] employed the SST k-ω turbulence model, and the hexahedral mesh that was created for the simulation was composed of approximately 2.1 million cells. Making use of an implicit scheme for time marching, the convergence was reached after all residuals dropped to values below 1x10$^{-4,}$ which consumed 5675 iterations and some mesh adaptions [49]. Each iteration running on three cores of a desktop computer fitted with an Intel® Core™ i5-11600K processor demanded on average 18 secs. The full potential solution for the wing-fuselage combination took 25 secs to converge on the same machine. **Figure 42** shows contours of velocity magnitude over the airplane and **Figure 43** shows a comparison between the results from the RANS simulation and the full potential code. There, *Cp* distributions for two wing stations of the airliner of ITA50SR are shown. The agreement between the related *Cp* distributions is particularly good. In addition, Mattos et al. show another comparison between a RANS code and a full potential one, where drag coefficients were computed for a wing-body-winglet configuration [50]. There is a particularly good agreement between both flow formulations.

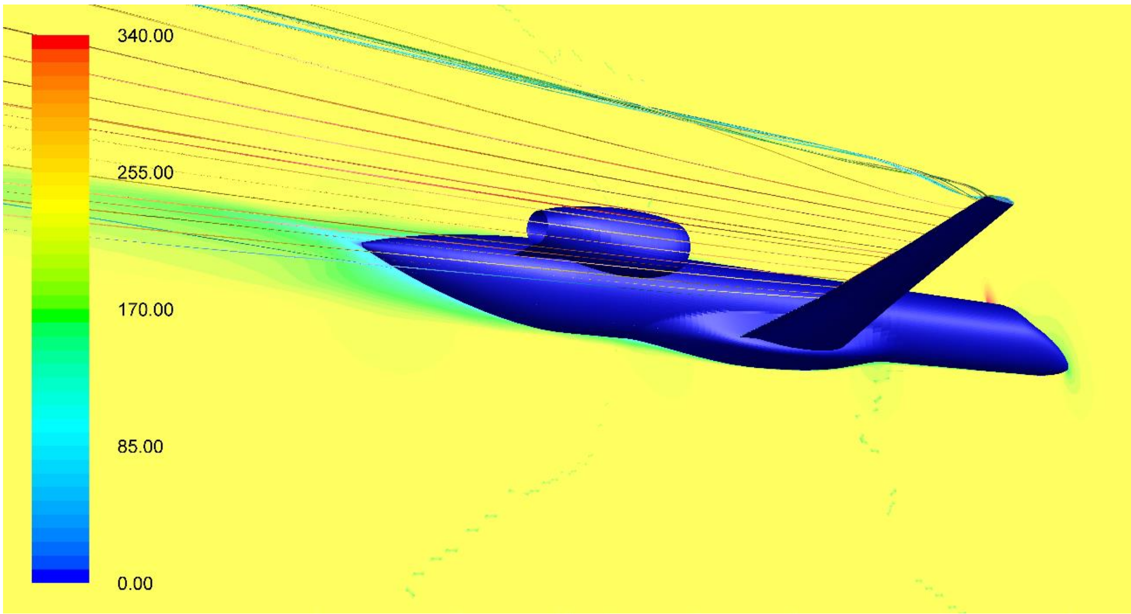


Figure 42: Contours of velocity magnitude for the ITA50SR airliner (M∞ = 0.775, α=1.21º)

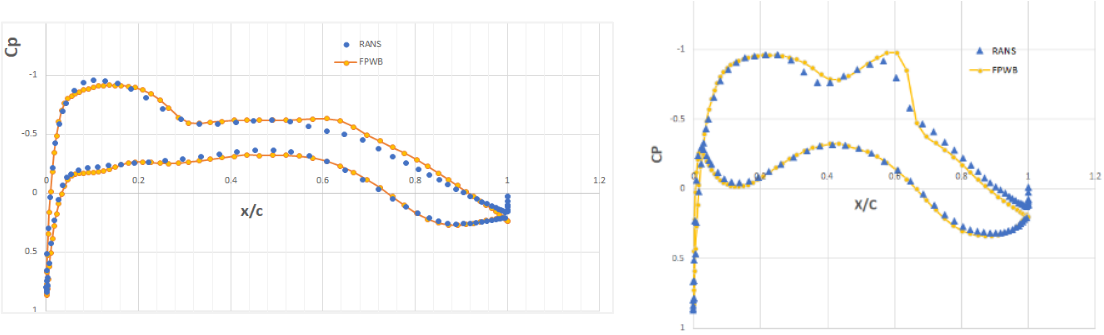


Figure 43: *Cp* distributions obtained with a RANS code and a full potential one for the ITA50ADV airliner. At left, station a 22% of semispan; right, station located at 50% of semispan (M∞ = 0.775, α=1.21º, Reynolds number of 15x10^6)

5. Machine Learning overview and applications

5.1 Introduction

Machine learning is the elaboration and application of computational techniques to discover patterns and trends contained in data (call it experience, to some extent). Machine learning algorithms can learn information directly from data, without relying on a predetermined equation as a model. Sometimes is almost impossible to derive equations or a system of equations to model a physic phenom. In other cases, even considering that a system of differential equations is available, it can be very costly to solve and therefore surrogate models are a good option.

Machine learning algorithms improve their performance adaptively as the number of samples available for learning increases and it can be split into two types of techniques (**Figure 44**): supervised learning, which trains a model from known input and output data enabling predictions when inputs of non-trained data are provided; and unsupervised learning that finds hidden patterns or intrinsic structures in the input data.

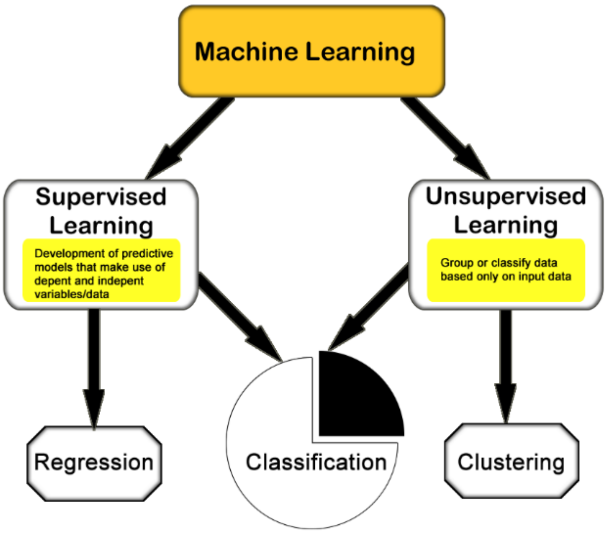

Machine Learning
Supervised Learning
Development of predictive models that make use of depent and indepent variables/data
Unsupervised Learning
Group or classify data based only on input data
Regression
Classification
Clustering

Figure 44: Machine learning branches

The main goal of supervised machine learning is to build a model that makes evidence-based predictions in the presence of uncertainty. A supervised learning algorithm employs a known set of input data and known responses to the data (output) to create a model that generates accurate predictions when new and untrained data is supplied. Supervised learning uses classification and regression techniques to develop predictive models. In other words, supervised learning can be defined by using labeled datasets to train algorithms that classify data or accurately predict outcomes.

Classification techniques properly provide for categorical answers. A simple example is whether an email can be categorized as genuine or as spam. Another example is predicting the imminence of a heart attack. Classification models sort input data into distinct categories. Typical applications naturally include medical imaging, speech recognition, customer behavior, and credit score. Classification can be precisely defined as the process of assigning a label to an object based on its characteristics according to a class. Problems can involve any number of them, which are defined as a group containing objects that share attributes. In binary classification, however, two classes are only allowed.

Regression analysis is widely used for prediction and forecasting, where its use has substantial overlap with the field of machine learning. Prediction of oil prices and stock prices are examples of application. In some situations, regression analysis can be used to infer causal relationships between the independent and dependent variables. Importantly, regressions by themselves only reveal relationships between a dependent variable and a collection of independent variables in a fixed dataset. To use regressions for prediction or to infer causal relationships, respectively, a researcher must carefully justify why existing relationships have predictive power for a new context or why a relationship between two variables has a causal interpretation. The latter is especially important when researchers hope to estimate causal relationships using observational data.

Unsupervised learning discovers hidden patterns or intrinsic structures in the data. It is used to make inferences from datasets that consist of input data without labeled responses. Grouping is the most common unsupervised learning technique. It is used for exploratory data analysis to find hidden patterns or data groupings. Applications for grouping include gene sequence analysis, market research, and object recognition.

There are plenty of supervised and unsupervised machine learning algorithms (**Figure 45),** and each has a different approach to the learning process [51]. Merely choosing a good one may prove difficult because there is no direct approach or the best method and therefore finding the right algorithm for a particular problem. Highly flexible representations tend to super-adjust data by modeling small variations that can be noised; simple ones are easier to interpret and faster to run but may inevitably have less accuracy. Therefore, choosing the right algorithm requires exchanging one benefit for another, including speed, accuracy, and complexity of the model. Trial and error are the essences of machine learning - if one used the approach or used algorithm does not work, the concerned user must merely try another.

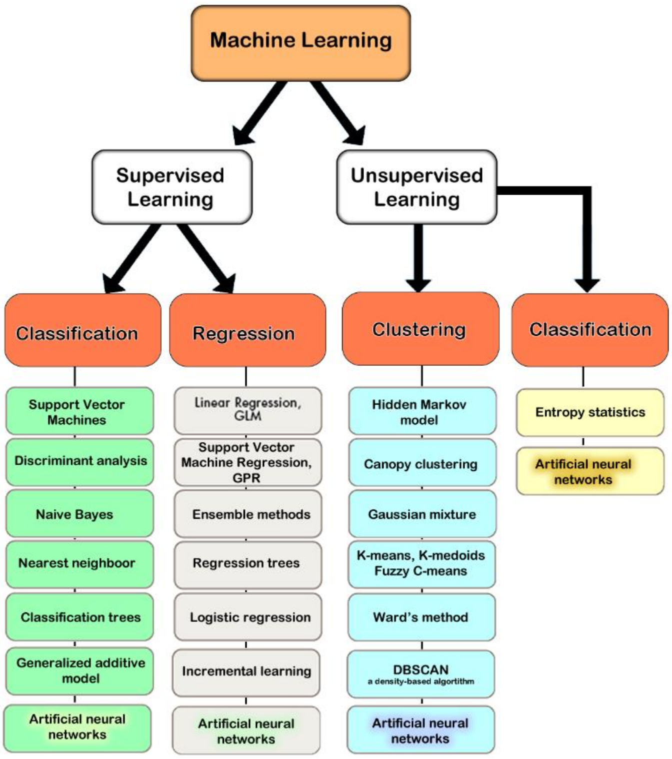


Figure 45: Supervised and unsupervised learning algorithms

Kireev et al. developed Self-Organizing Maps (SOM) to solve a classification problem of large databases of chemical reactants [52], i. e., an application of unsupervised learning with Korhonen neural networks. With the utilization of entropy statistics, Frenken and Leydesdorff [53] analyzed scaling patterns in terms of changes in the ratios among characteristics of 143 configurations of commercial transport aircraft. In their research, the piston-powered DC-3, and the jet-powered Boeing 707, were revealed to have triggered scaled trajectories. Some configuration characteristics of both airplanes have been scaled at different moments in time, which points to the versatility of a dominant design that allows a firm to react to a variety of customer requirements. Scaling at the level of the industry took off only after subsequently re-engineered models were introduced, like the piston propeller Douglas DC-4 and the twin-aisle Boeing 767. The two scaling trajectories in civil aircraft corresponding to the piston propeller and the turbofan paradigm can be compared with a single, less pronounced scaling trajectory in helicopter technology for data during the period 1940–1996 [53]. They state that management and policy implications can be specified in terms of the phases of codification at the firm and the industry level, thanks to the entropy statistics analysis.

5.2 Artificial neural networks

Artificial Neural Networks (ANNs) are a special type of machine learning algorithms that are modeled similarly as information is processed by the human brain. That is, just like how the neurons in our nervous system can learn from the past data, similarly, the ANN can learn from the data and respond to the form of predictions or classifications. ANNs are nonlinear statistical models which display a complex relationship between the inputs and outputs to discover a new pattern. They can be used for a broad range of applications such as image recognition, regression, speech recognition, machine translation as well as medical diagnosis. ANNS can handle a mix of many variables of different types (logical, real, integer). They can also deal with a large amount of data, whereas usual regression methods do not.

An important advantage of ANN is the fact that it learns from the example data sets. The most common usage of ANN is that of a random function approximation. With these types of tools, one can have a cost-effective method of arriving at the solutions that define the distribution. ANN is also capable of taking sample data rather than the entire dataset to provide the output result. With ANNs, one can enhance existing data analysis techniques owing to their advanced predictive capabilities.

A neural network is an interconnected structure of processing elements, called nodes, whose goal is to mimic the biological neuron. Weights are attributed to the inter-unit connections, and they are obtained by a process of adaptation from a set of training data. After the trained ANN is stimulated with a group of inputs, the signal processing through the neuron layers will produce an output, simulating an interpolation of the database.

Lipmann [54] performed a description of the evolution of ANN research and McCulloch [55] presented one of the first mathematical models of ANN. Rosenblatt [56] introduced and endorsed the single-layer perceptron (Linear Neurons) network for classification problems. However, Minsky [57] stated the weakness of perceptron architecture. This led to some disinterest in the ANN research field [58] until the development of new network architectures and learning methods, such as the back-propagation algorithm [59].

ANNs find applications in many areas such as pattern recognition, non-linear control, optimization, and decision making. The use of ANN also improved the techniques for computer speech and image recognition [60]. There are many neural network types and architectures. Below, short descriptions of some of the most known types are provided:

- *Feedforward Neural Network – Artificial Neuron*. This is the simplest type of ANN, where the data or the input flows from input to output layers, going through the intermediate layers to the output nodes. Hidden layers are optional in the architecture and there are not any loops in this network type. Feedforward networks are utilized for any input to output mapping [61].
- *Radial basis function Neural Network*. The radial basis function (RBF) is a real-valued function whose value depends only on the distance between the input and some fixed point, which can be the origin or some other fixed point. Sums of radial basis functions are used to approximate intricate functions. In an RBF network, radial basis functions perform the role of the activation function.
- *Kohonen Self Organizing Network*. A Kohonen network maps inputs of arbitrary dimension to a discrete neuron map. A Kohonen map is an unsupervised learning (self-organizing) model. It is a method for dimensional reduction, as high-dimension inputs are mapped into a lower-dimensional discretized representation and conserve the implicit structure of its input space. When training the map, the location of the neuron remains constant, but the weights will differ depending on the value [62]. This self-organization process has distinct steps: initially, every neuron weight is initialized with a small value in the input vector; in the second phase, the neuron closest to the point is then selected as the *winning neuron* and the neurons that are linked to it will also move towards that point. The distance between the point and the neurons can be calculated by the several distance approaches like, for instance, the Euclidean one, and the closest neuron is then taken as the winner. Through the interactive process, all points are clustered, with each neuron representing each kind of cluster.
- *Recurrent Neural Networks* are based on the principle of feeding the output of a layer back to the input to help in recalibrating neuron weights to better predict the outcome of the layer. Here, the first layer is formed like the feed-forward neural network with the product of the sum of the weights and the features [62]. Indeed, the recurrent neural network process starts once this is calculated. Thus, from one-time step to the next, each neuron will remember some information recorded in the previous time step. This way, each neuron acts as a memory cell in the computations. In this process, it is necessary to let the neural network work on the front propagation and save what information it needs for later use. Hopfield networks are recurrent or fully interconnected neural networks. There are two versions of Hopfield neural networks: in the binary version all neurons are connected but there is no connection from a neuron to itself, and in the continuous case all connections including self-connections are allowed. Hopfield neural networks are applied to solve many optimization problems. In medical image processing, they are applied in the continuous mode to image restoration, and the binary mode to image segmentation and boundary detection.
- *Convolutional Neural Network* (CNN) is an architecture for deep learning, in the field of machine learning. Here, a model learns to perform tasks directly from images, text, video, or audio. Examples of deep learning are computational vision, natural language processing, and audio recognition. CNN may have tens or hundreds of layers that each learns to detect different features of an image. Filters are applied to training images at different resolutions, and the output of each convolved image is used as the input to the next layer [63]. The filters can start with simple features, such as brightness, contrast, and edge detection, and then with increasing complexity, they capture unique features of the object of interest [63]. After the learning process in many layers of the image features is completed, the architecture of a CNN is then ready for classification.
- *Modular Neural Networks*. This sort of network encompasses an assortment of different networks working independently and contributing to delivering an output. Each neural network presents distinctive input sets when compared to other networks constructing and performing other derated assignments. These networks do not interact or signal each other for the accomplishment of the assignments. The advantage of a modular neural network is that it breaks down an outsized procedure into smaller parts decreasing its complexity

ANN consists of many simple computational elements denominated artificial neurons, which are densely interconnected and operate in parallel. They are many possible neuron combinations and therefore many ANN architectures. **Figure 46** shows a typical arrangement of an artificial neuron, which converts a group of inputs into a single output after being processed by an activation function. These activations can derive from external variables or other artificial neurons.

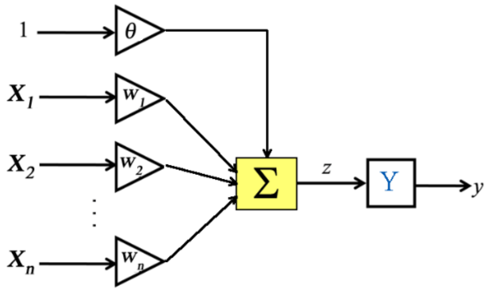
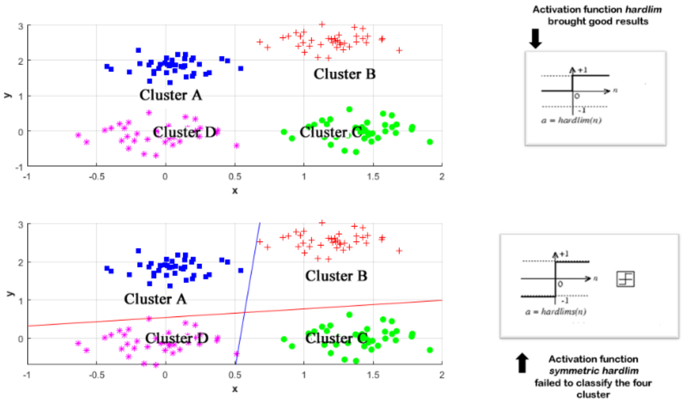


Figure 46: Artificial Neuron structure

Figure 47 shows an example of the application of an ANN to trace straight boundaries among four clusters of points, a classification problem, indeed. The neural network type perceptron of MATALB® was used in the code [64] [65]. The perceptron algorithm initializes all weights to zero and performs an iterative process for training using the cloud of points. It updates the weights only if the classification was wrong. The activation function *hardlim* (hard limit) was used, and the solution's convergence is very fast.


Activation function *hardlim* brought good results

Activation function *symmetric hardlim* failed to classify the four cluster

Figure 47: A simple cluster classification problem using an ANN

Another example of the application of the artificial neural network is the accurate representation of the following function, called here *Peaks*:

The shape of this function can be seen in **Figure 48**. It is characterized by a hill adjacent to a valley.

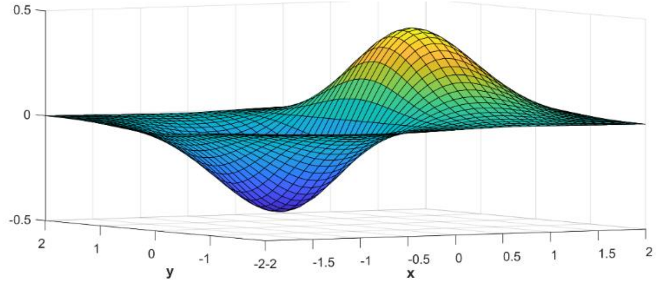


Figure 48: A graph of a function called here *Peaks:* $f_{peaks} = xe^{-x^2-y^2}$

Radial basis networks can be used to approximate functions. The MATLAB® command *newrb* adds neurons to a radial basis network's hidden layer until it meets the specified mean squared error goal [66]. A parameter called *spread* must also be provided to the *newrb* command. The larger *spread* is, the smoother the function approximation. Too large a *spread* means a lot of neurons are required to fit a fast-changing function. Too small a *spread* means many neurons are required to fit a smooth function, and the network might not generalize well [66]. It is recommended to use *newrb* with different *spreads* to find out the best value for a given problem.

Figure 49 shows the surrogate model of the *Peaks* function using the radial basis network of MATLAB®. The parameter spread was set to 1, the *goal* parameter received 1x10^{-7}, and 1681 points were used for training and validation of the ANN model. The number of neurons that was adopted for this model is 20, but it is recommended to test a set of them to obtain an optimal representation of the *Peaks* function. **Figure 49** also contains a graph of the estimation errors by the ANN, and it can be observed that they are higher close to the boundaries determined by the set of points that were used for the training and validation. As homework, it is suggested to the experienced reader to carry out two optimizations with MALAB®: the first directly utilizing the *Peaks* function as objective; the second one should use the estimated ANN values for function estimation. It is a good test to verify whether the surrogate model can be used in optimizations or not.

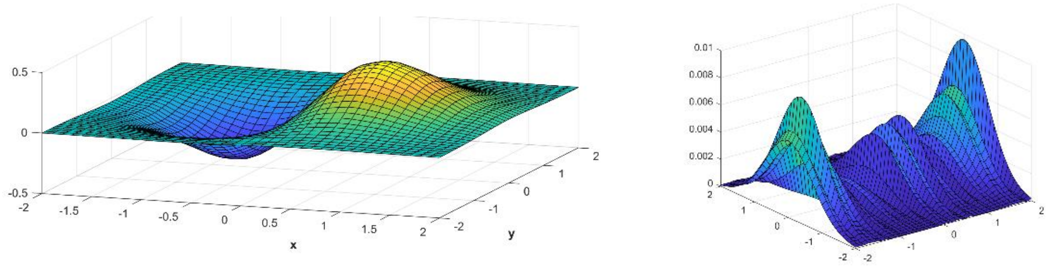


Figure 49: Peaks function as modeled by radial basis network and estimation errors

5.3 Extracting flight data from large databases

A. Improving mission profiles

Although the system-of-systems philosophy in aircraft design allows for tools for the inclusion of operational aspects during the optimization process, few existing design frameworks consider information about the actual aircraft mission profiles, their inherent constraints, and how they impact the airline economics. The increase of distance flown by a long-range aircraft when compared to a benchmark distance (Great Circle Distance), is often referred to as en-route inefficiency.

Figure 50 shows a route performed by a Boeing 777-300ER that flew from New York to Narita Airport in Japan in May 2020. The flight data were obtained from a flight tracking website [67]. Another route this time of a transatlantic flight is with a Boeing 747-8 in May 2020 [68] as illustrated in **Figure 51**. Taking off from São Paulo Guarulhos Airport the time spent to reach the initial cruise altitude was 23 min and 21 s (**Figure 52**). Two distinct marks for this flight are the step cruise and a speed overshoot at the beginning of the cruise phase. Although the Boeing 747-8 service ceiling is 43,100 ft, the maximum altitude this airplane flew in this long-range flight was 38,000 ft. For both flights, there is a considerable departure from the Great Circle trajectory.

Recent operational data have shown that depending on the region, en-route inefficiency may vary between 2 and 7 percent [69]. Such values are naturally associated with a proportional increase in fuel burn and emissions. Rios Cruz et al. [69] evaluated the consequences of such an increase in traveling distance in the design of a European airline network. Data data-driven mission profile definition was incorporated into an MDO framework for airplane conceptual design. Machine learning methods were applied to Automatic Dependent Surveillance-Broadcast (ADS-B) flight data considering ten major EU airports over six months. Thus, the definition of principal flight patterns was possible to feed an aircraft performance module whose output is the operational cost. A genetic algorithm optimizer is part of the design framework, which also contains an aircraft sizing module, a data-driven mission performance evaluator, and an embedded airline network optimizer. Altogether, the proposed algorithm aims to maximize the network profit. The results show that more conservative estimations in terms of profit and direct operational cost are achieved when accounting for realistic mission profiles.

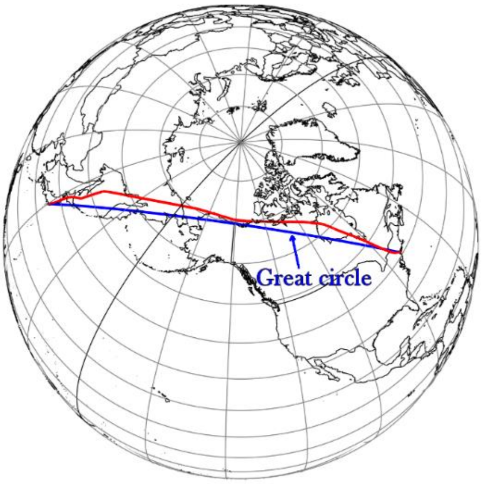


Figure 50: Three-dimensional view of the route from New York to Tokyo with a Boeing 777-300ER. Some data of the cruise phase are missing and were linearly interpolated

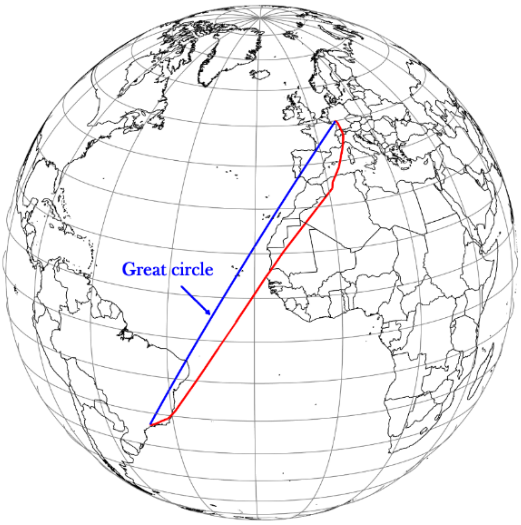


Figure 51: Three-dimensional view of the route from São Paulo to Frankfurt with a Boeing 747. Some data of the cruise phase are missing and were linearly interpolated

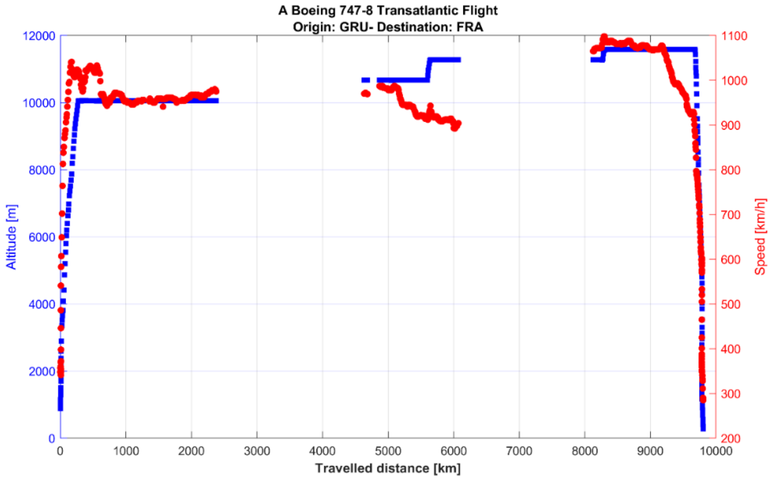


Figure 52: Flight profile of a flight GRU-FRA in May 2020. Ground speed and altitude are given as a function of traveled distance

To enhance the cluster detection performed by the DBSCAN algorithm [70] without losing flight information, a phase identification was added to the framework elaborated by Rios Cruz. This process employs the methodology elaborated by Sun et al [71], which applied fuzzy logic on the time series data following the theory from [72].

Fuzzy logic was applied to establish to which phase a certain point in the dataset pertains considering a set of membership functions to describe altitude, speed, and rate of climb, related to a certain flight phase. Four types of membership functions are used: Gaussian, Z-shaped, S-shaped, and Pi functions. Differently from [71], the *Pi* function was included to increase the accuracy in the cruise and level descent phase detection. Additionally, during the process of phase identification, it was observed that cruise phase detection is a function of the distance between airports. Because of that, the settings proposed by [71] for the definitions of high altitude, needed to be tuned for situations where the distance between airports was small than 200 nautical miles, considering a value of $H_{hi}(\eta)$ = G(η,30000,15000).

The membership functions related to the different characteristics of the flight stages are shown in **Figs. 53** to **55**. The x-axis represents the low and high extremes of each feature. Four membership functions were used for the altitude and RoC, and only three for the speed. This is justified considering that is expected that the same range of speeds takes place during the last part of the climb, the cruise, and the beginning of the descent phase.

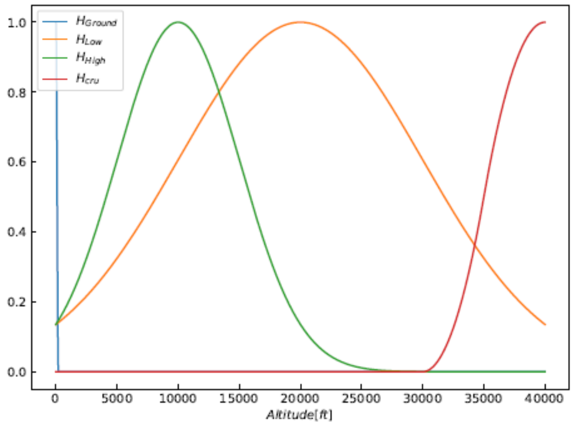


Figure 53:Altitude

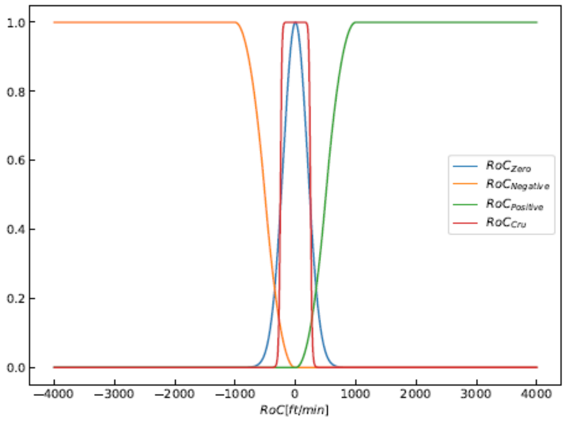


Figure 54: Rate of climb

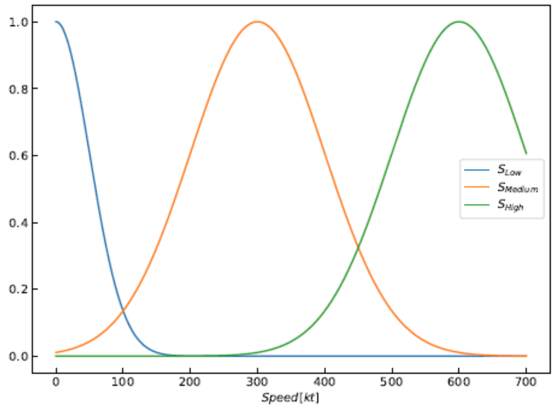


Figure 55: Speed

The following rules were used to identify the correct phase:

- if $Hgnd \land Vlo \land RoC0$ then Ground
- if $Hlo \land Vmid \land RoC+$ then Climb
- if $Hcr \land Vhi \land RoCcr$ then Cruise
- if $Hlo \land Vmid \land RoC-$ then Descent
- if $Hlo \land Hhi \land Vmid \land RoC0$ then Level flight

The fuzzy logic then uses this information for a given data point and all possible discrete flight phase states $P(0 \leq P_i \leq 6)$ (all represented by Gaussian functions). Finally, a defuzzification takes place, where the most likely flight phase state is found using **Eq. 66**.

$$\hat{P} = round(\arg \underset{P}{\overset{max}{}} S(P)) \tag{66}$$

In **Eq. 66**, $\hat{P}$ represents the output where the highest combined fuzzy value occurs. The labels generated in this step ('GND', 'CL', 'CR', 'DE', 'LVL') are included in a column of the data frame to be used in the following steps.

To detect the main horizontal and vertical flight patterns, a two-step cluster approach followed the preceding procedure. Only the cruise information of the trajectories was used to measure the distance between them (input for the selected algorithm), envisioning the reduction of outliers. Considering this and due to the spatial nature of the dataset, the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm was selected to perform the clustering task.

B. Estimating airliner mass from flight data

This is another good example of the use of inverse techniques. There have been several approaches to compute the mass at initial climb segments. These masses are used for fuel consumption calculation and trajectory prediction.

Bayesian inference has been used for the estimation of the mass of airliners from statistical flight data. Bayesian inference derives the posterior probability from two known data: a prior probability and a "likelihood function" derived from a statistical model for the observed data (**Figure 56**).

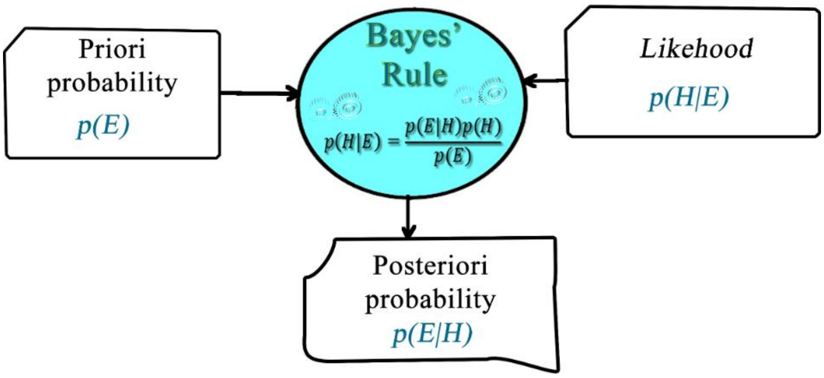


Figure 56: Illustration of application of Bayes' rule

After a study presented by Sun et al. [73], the estimation of the initial mass of airplanes not just at one specific flight phase, but a combination of all phases, derived an improvement of their methodology [74]. The mass estimation problem was considered as a single parameter for a Bayesian inference task, considering observational masses computed along with the entire flight. In addition to multiple observations, prior knowledge of weight was used to improve the estimation. Sun et al. used approximations to obtain the number of passengers, considering constraints for the variation of weight estimation according to the known payload for each airplane. The authors state that prior knowledge can be very valuable for the estimation of actual aircraft mass when applying Bayesian inference. Fuel consumption was modeled by using the ICAO aircraft engine emissions database, which is freely available. Using this data, the authors proposed a quadratic fuel flow model that is constructed setting the fuel flow as a function of the ratio between thrust at a given flight condition and maximum static thrust [74].

Silva [75] proposed three methods to estimate airliner takeoff mass from en-route climbing data, gathered from Automatic Detection Surveillance-Broadcast sources (ADS-B). The first method proposed by Silva is a deterministic one, consisting of the identification of flight procedures for climbing segments such as initial altitude for acceleration before the climb itself, and, for example, where the calibrated airspeed is constant. The second and the third approaches compute operational parameters directly from Automatic Detection Surveillance-Broadcast data and utilize them in each step for the performance calculation. Mass estimation process can be optionally carried out considering wind effects. Silva [75] made use of the BADA 3 performance model [76], calibrated with ADS-B data, and used a genetic algorithm for error minimization. The derivatives *dV/dH* and *dV/dt* (necessary for trajectory computations) are calculated from ADS-B data plots, providing more efficient results than on previous academic studies, which normally use the constant calibrated airspeed method.

The objective function utilized is summarized in **Eq. 67**.

$$Fobj = \left(\frac{1}{N} \sum_{i=1}^{Npoints} [(H_p)_{observed}(t_i) - (H_p)_{siml}(m, \delta_{red}, t_i)]^2 \right)^{0.5} \quad (67)$$

where:

- N represents the number of samples extracted from ADS-B at a specific time stamp t_i
- $(H_p)_{observed}$ is the observed altitude (from ADS-B string) at a specific time stamp t_i
- $(H_p)_{siml}$ is the simulated altitude, calculated via Aircraft Performance Model, related to the time stamp t_i.
- δ_{red} is a thrust setting coefficient.

Data from selected flights operated with A330-200 were employed to validate the three approaches employed for mass estimation. All flights departed from Medellin (Colombia) to Madrid (Spain). The methodology for mass estimation from an integrated simulation on several points along stable segments in the climb phase contributed to the good accuracy of the results. The simulations with and without wind models have shown equivalent results, with errors between 2% to 4% when compared with actual mass data for two of the approaches (**Table 6). S**ensitivity analyses by varying δ_{red} with altitude were performed. An improvement in mass estimation was recorded when compared with previous academic studies, which present errors ranging from 5% to 10% for masses at the initial climb phase.

Table 6: Estimated and actual masses for flights with A330 @ 11,500 ft [75]

Flight	Approach	F_{Obj} [m]	δ_{red}	Calculated mass [kg]	Actual mass [kg]	Error [kg]	Error [%]
1	dVdH	11.56 m	0.950	200,340	202,701	2361	<2%
	dVdt	14.54 m	0.989	208,192	202,701	5491	<4%
2	dVdH	26.54 m	0.942	204,205	206,420	2215	<2%
	dVdt	32.19 m	0.973	208,971	206,420	2551	<2%
3	dVdH	11.56 m	0.966	207,759	209,324	1564	<1.5%
	dVdt	14.54 m	0.966	213,495	209,324	4172	<2%

5.4 applications of multi-layer perceptron network

A. Multi-layer perceptron network

Multi-disciplinary airplane design frameworks, essential tools for today's aircraft design, require considerable computational power, and the computing cost grows when higher fidelity tools are used to model the associated disciplines. The use of surrogate models offers an efficient alternative to overcome this issue. In turn, artificial neural networks have been successfully used to generate auxiliary models in complex systems with many variables. In this context, the present work deals with the design and application of artificial neural networks to predict aerodynamic coefficients of transport airplanes with a high degree of accuracy. The neural networks are fed with calculations from computational fluid dynamic codes, and they can predict lift, pitch moment, and drag coefficients for wing-fuselage-winglet configurations of transport airplanes. The input parameters for the neural network control the wing planform, winglet geometry, fuselage geometric parameters, airfoil geometry, and flight condition. Airfoil geometry is modeled by attributing weights to 14 basic airfoils compounding a database. An aerodynamic database consisting of approximately 62,000 cases calculated with a full-potential code with computation of viscous effects is used for the neural network training, validation, and test. Networks with different numbers of neurons are evaluated and those with the lowest mean quadratic errors are selected.

A Multi-layer Perceptron (MLP) network consists of an input layer, several intermediate layers (to transform inputs into something that the output layer can use), and an output layer. Connecting several nodes in parallel and series, an MLP network is formed. The image shown in **Figure 57** reveals the schematic of a feedforward network, which consists of a layered structure with information flowing from the inputs to the outputs. The inputs are points or data collection that may differ in nature which receive structured information and repass them to the first layer containing neural (Perceptron) nodes. This first layer of functional nodes is known as the hidden layer because it is not targeted to inspect or control the output values on these nodes during the process of training when the network weights are obtained. The incorporation of hidden layers enables the network to model complex non-linear behavior [77] by using the usual transfer functions already mentioned before. The optimal number of hidden layers could be smaller than the number of inputs. Typically, just two hidden layers work fine with few data and a higher number of hidden layers can be fruitful for the difficult object. Several architectures must be evaluated and presented the lowest mean quadratic error should then be taken. The outputs that flew through the hidden layers are subsequently processed by an output layer and the results are compared to the known values associated with the input pattern.

Eq. 68 illustrates the simplicity of the calculation of output values for the two-hidden-layer network shown in **Figure 57** using the input set.

$$y_i = \emptyset\left(\sum_{j=1}^{3} w_{ji,2}\,\Upsilon\left(z_{j,1}\right) + \theta_{j,2}\right) = \emptyset\left(\sum_{j=1}^{3} w_{ji,2}\,\Upsilon\left(\sum_{k=1}^{K} w_{kj,1} x_K + \theta_{j,1}\right) + \theta_{j,2}\right) \qquad (68)$$

It is possible to generate a layer with neurons that share the same inputs. An artificial neural network is the juxtaposition of these layers. Several ANN architectures are described in recent literature. Among these architectures, the multi-layer feed-forward ANN is recommended for non-linear regression problems [78]. This type of network can approximate any function to any desired degree of accuracy, provided it has enough neurons [79]. When multi-layer feed-forward ANN is used in regression problems, the neurons of the last layer use linear transfer functions, as the output might be any real number. The neurons of other layers usually use the hyperbolic tangent transfer function.

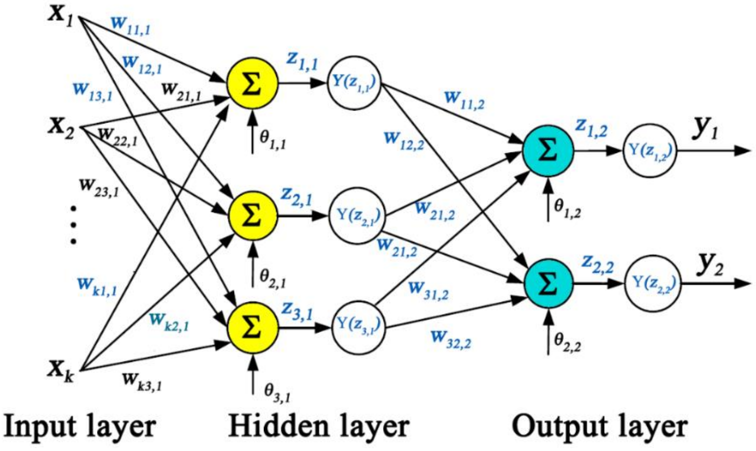


Figure 57: Exemplification of a multi-layer feed-forward network

All networks that were considered for the estimation of aerodynamic coefficients present two hidden layers. MATLAB® *fitnet* function fitting neural network [80] was employed in the present study. The *fitnet* is a specialized version of feedforward networks. The hyperbolic tangent sigmoid [81] was employed as transfer functions in the hidden layers. The network training function updates weight and bias values according to Levenberg-Marquardt optimization, also known as the Damping Least Square method [82] [83]. Like the quasi-Newton methods, the Levenberg-Marquardt algorithm was designed to approach second-order training speed without having to compute the Hessian matrix. The training/validation/test with the Levenberg-Marquart algorithm was fed with 57,000 cases. Training stops when any of these conditions occur [84]:

- The maximum number of epochs is reached.
- The maximum stipulated processing time is exceeded.
- Performance is minimized and reaches the goal.
- The performance gradient falls below a minimum value that was previously set.
- The damping factor exceeds the prescribed maximum value.

B. Forecasting oil price

Crude oil is a commodity found in geological formations below the Earth's surface, and at the bottom of the oceans. It can be refined into various kinds of consumer fuels and other substances, many of them finding applications in the petrochemical industry. Crude oil is amongst the most important energy resources on earth right now. So far, its derivatives remain the world’s leading fuel, providing nearly one-third of the global energy consumption. In addition, this information is crucial to methodologies that elaborate the market outlook of transport aircraft [85]

To forecast the oil price precisely, a two-hidden-layer perceptron network was chosen to train and validate data from crude oil production, demand, and stocks. The activation functions for the first and second layers are tangent sigmoid and log sigmoid, respectively. The input variables and their respective sources of information are as follows:

- World oil demand [86].
- U.S. oil stocks [87].
- Total oil production [88] [89].

The output variable, or the variable to be forecasted, is the West Texas Intermediate (WTI) crude oil price [90].

Figure 58 shows the crude oil output from three different sources and **Figure 59** is a two-vertical axis graph containing the WTI crude oil price and demand over time. The available data were compiled and adjusted for the 1987-2019 period.

A set of networks comprising different combinations of several neurons into two hidden layers was evaluated. The number of neurons in the first hidden layer varied from 20 to 120. The second layer could then receive neurons starting from 20. The maximum number possible to attain depends on the number of neurons existing in the first layer.

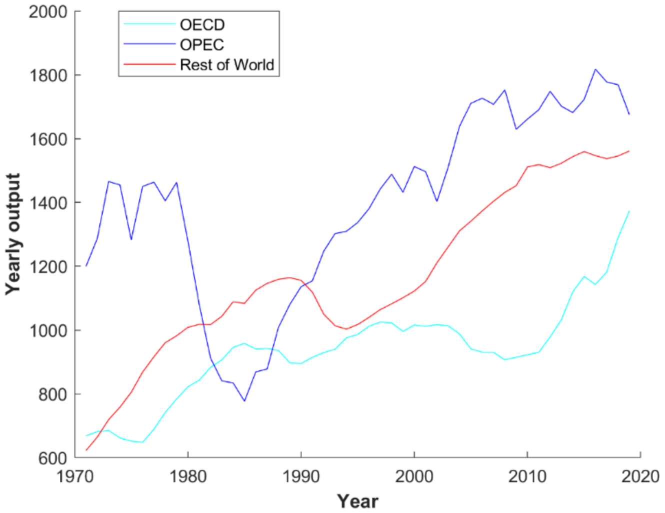


Figure 58: Crude oil yearly output

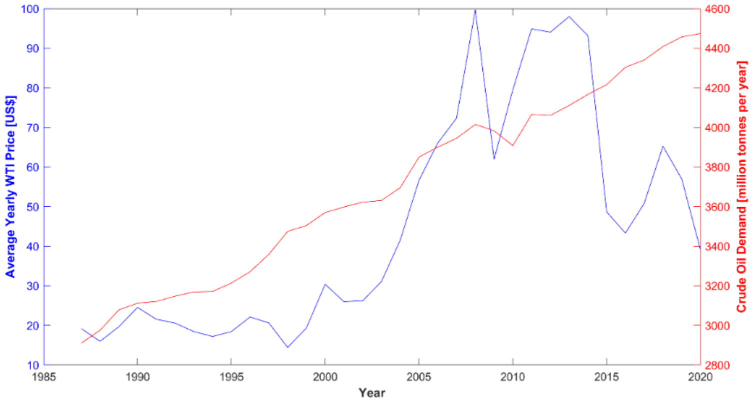


Figure 59: Average WTI crude oil price and demand

Ten years of the compiled data were normalized and thereafter separated for forecasting by ANN. The remaining data were then used for training and validation. The ANN with the lowest mean square error has 120 and 60 neurons in the first and second hidden layer, respectively. **Figure 60** compares the forecasted values by the selected ANN with actual data. The average error in the period analyzed was US$ 16.37.

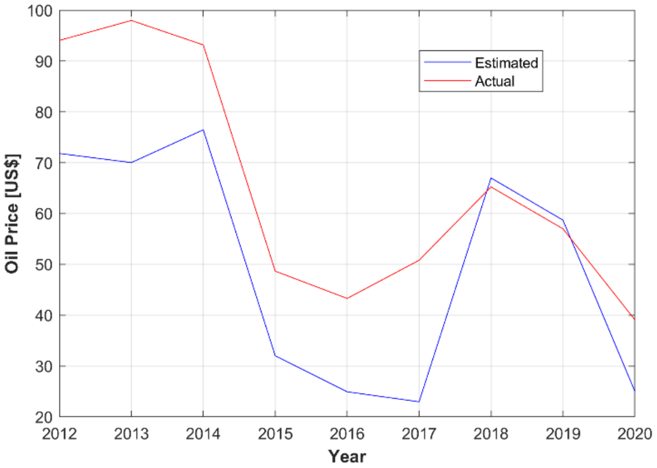


Figure 60: Forecasted and actual crude oil prices for the 2012-2019 period

C. Estimating drag coefficients with a surrogate model

Here, a surrogate model with ANN to replace a full-potential code in airplane MDO frameworks is described and some results are presented. A database with almost 72,000 cases to train and design the ANNs was generated. The data robustness of the database was checked with the learning curves method. The 59 design variables were used for the surrogate model with neural networks. **Figure 61** shows some range and boundaries for some of them, whose distribution in design space was performed by a Latin hypercube Design of Experiment (DOE) algorithm. Some additional runs at higher Mach numbers were enforced because of the non-linear relationship among the design variables and the aerodynamic coefficients in this part of the design space.

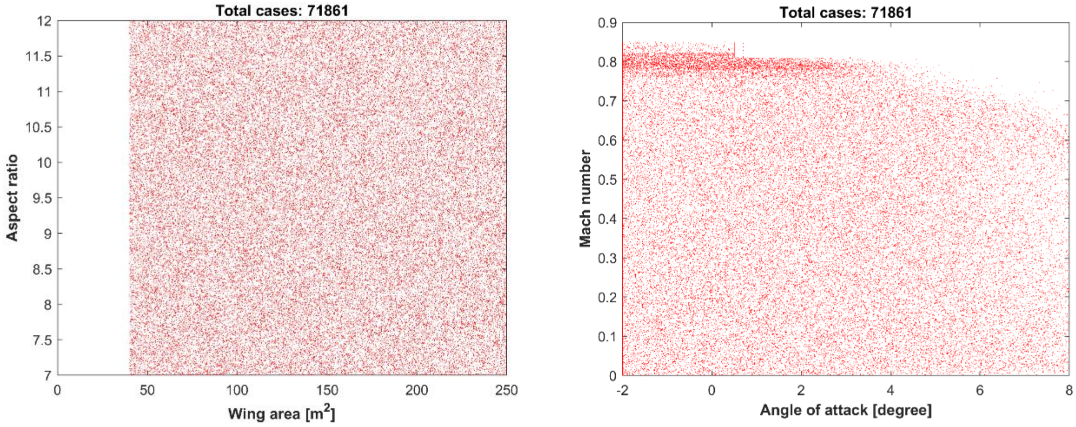
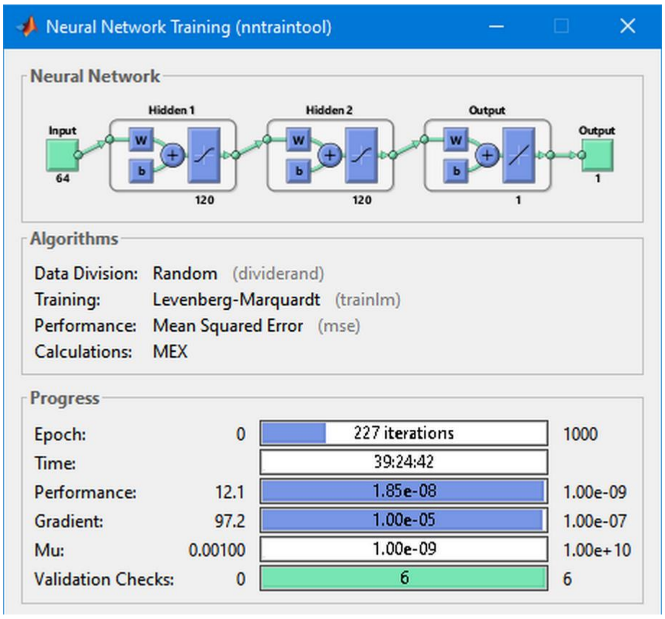


Figure 61: Some variables used for flow calculation with successful output. Notice the good coverage provided by a Latin hypercube DOE.

All training validation and testing of candidate networks ran on a desktop computer fitted with an Intel® Core™ i9-9900K CPU @ 3.60 GHz clock. **Figure 62** displays the main panel of the training/validation/test process for the 120x120 network. The process for this network took over 39 h of computing time.



Algorithms

Data Division: Random (dividerand)
Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)
Calculations: MEX

Progress

Epoch:	0	227 iterations	1000
Time:		39:24:42	
Performance:	12.1	1.85e-08	1.00e-09
Gradient:	97.2	1.00e-05	1.00e-07
Mu:	0.00100	1.00e-09	1.00e+10
Validation Checks:	0	6	6

Figure 62: A main panel of the training/validation/test process with MATLAB® for the 120x120 network targeted to estimate the pitching moment coefficient

The airplane with ID 165218 (**Figure 63**) was chosen to verify the capability and accuracy of drag coefficient prediction of the neural network system that was developed in the present work. This configuration was not used in the training/validation/test procedure. The features of this configuration are provided in **Table 7**. **Figs. 64** to **68** show a comparison among results from the surrogate ANN model and calculations performed with the full potential code at the same flow condition.

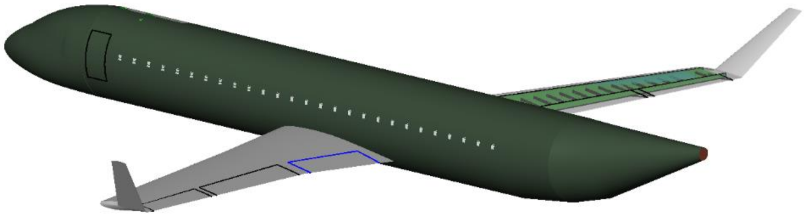


Figure 63: Configuration with ID 156218

Table 7: Data for the ID 156218 configuration

Wing aspect ratio	10.85
Wing taper ratio	0.433
Location of the wing break station	43% of semispan
Dihedral angle	2.60°
Root incidence angle	2.83°
Incidence of the break station	1.08°
Incidence of the tip station	-4.00°
Wing area	133.88 m^2
Quarter-chord sweepback angle	16.19°
Winglet aspect ratio	2.81
Winglet taper ratio	0.493
Winglet sweepback angle	30.04°
Winglet dihedral angle	48.20°
Winglet twist angle	-2.36°
Fuselage length	38.90 m
Fuselage width	3.331 m
Fuselage height	3.642 m
Fuselage wetted area	367.76 m^2

Figure 64 shows the comparison between the results for the lift coefficient at several numbers of Mach. All flow conditions were considered at a one-degree angle of attack and 13,000 m flight altitude. The Reynolds number with the wing MAC as reference length varies with the Mach number. The agreement between the CFD code and the surrogate model is excellent with errors lower than a fraction of one drag count, even at higher Mach numbers where strong shock waves are present.

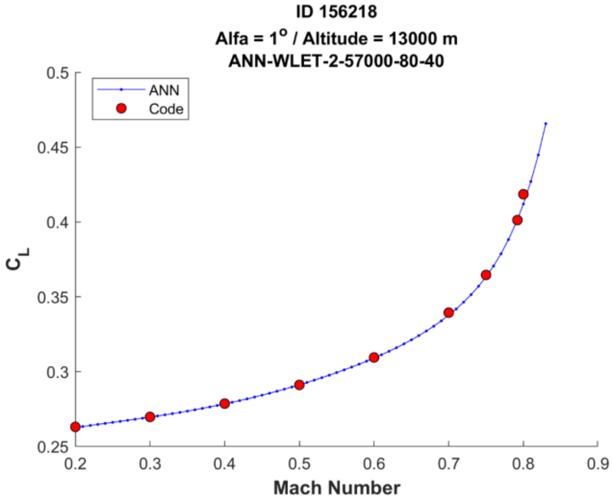


Figure 64: Comparison between lift coefficient estimation for airplane 156218 by the chosen ANN and those calculated with the full potential code

Figure 65 and **Figure 66** show graphs that compare calculations carried out by the full potential code with predictions from the surrogate models for Mach numbers varying from 0.20 to 0.83. The vertical scale of graphs was set in drag counts - one drag count corresponds to a drag coefficient of 1x10^{-4} - to facilitate readings. Only for reference purposes, the green symbols in both graphs are relative to the configuration with ID 156218 with no winglets. The winglet increased the configuration drag but the comparison shall be made at the same lift coefficient when its benefit regarding lower drag will emerge.

Figure 65 reveals that there is an exceptionally good agreement for the C_{D0} between the values estimated by ANN of size 100x60 and the results from the CFD code. According to **Figure 66**, the same is valid for the estimation of the induced drag, C_{DI}, with those from the chosen ANN of size 100x100. In both curves, the difference between the CFD code and the ANN lies below one drag count except for the Mach number of 0.80, where the discrepancy increased approximately to two drag counts. However, this is a region of drag divergence and this difference can be considered negligible and the capacity of the ANN system to predict viscous and induced drag coefficients is outstanding.

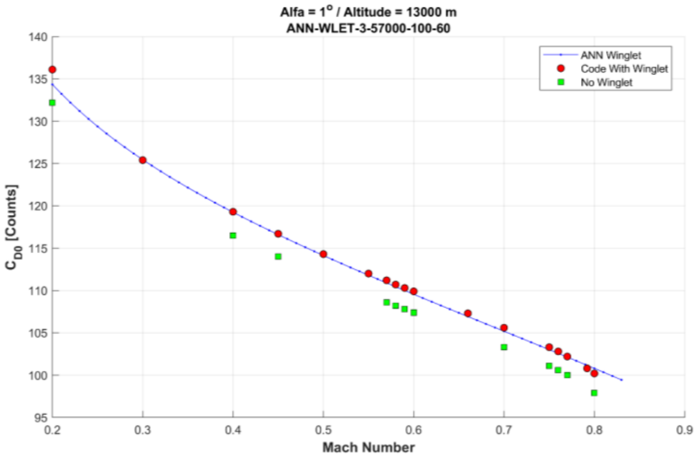


Figure 65: Comparison of zero-lift drag coefficient for airplane 156218 between ANN and those calculated by the full potential code

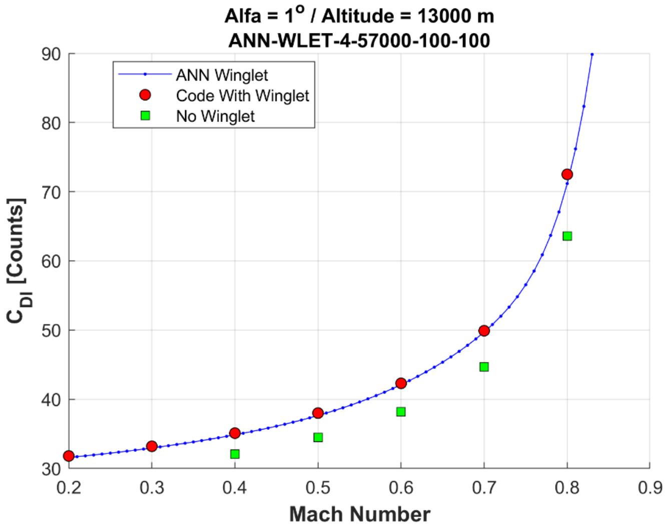


Figure 66: Induced drag coefficient for airplane 156218 at α=1° at an altitude of 13,000 m

Figure 67 and **Figure 68** show graphs that compare calculations carried out by the full potential code with predictions from the surrogate models for the wave drag, C_{DW}, and the wing pitching moment coefficient, C_M. **Figure 67** reveals that there is a good agreement for the C_{DW} between the values estimated by the dedicated ANN of size 100x100 and the results from the CFD code. The main difference between calculated and predicted values is in the order of four drag counts, taking place at curve foothills. **Figure 68** reveals an excellent agreement between the calculated and predicted C_M. The graph at left, the variation of pitching moment with the angle of attack, meaning that the slope, which corresponds to $C_{M\alpha}$, is constant as expected for a subsonic flow condition.

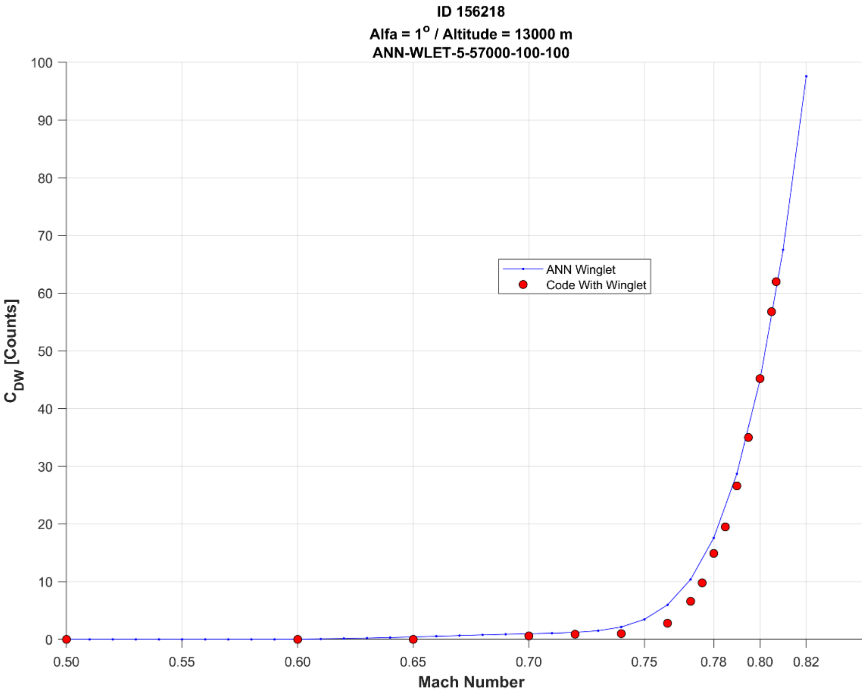


Figure 67: Comparison between wave drag coefficient estimation for airplane 156218 by the 100x100 ANN and that calculated by the full potential code

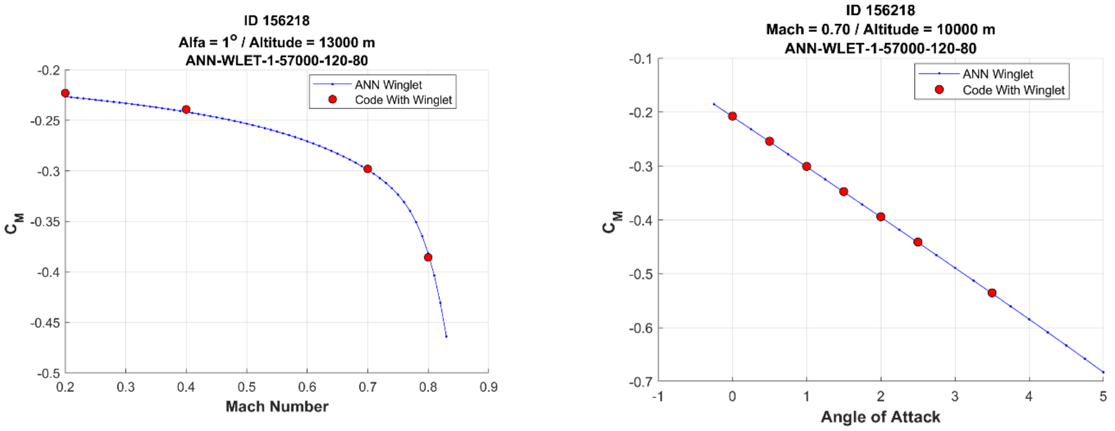


Figure 68: Comparison of pitch moment coefficient estimation for ID 156218 by the chosen ANN and that calculated by the full potential code. At left, variation with Mach number at a constant angle of attack; at right, variation with angle of attack at a Mach number of 0.70 and altitude of 10,000 m

To illustrate the utilization of the ANN surrogate model, an aerodynamic optimization of an airplane similar in capacity to the B757-200 was carried out. This is our baseline or reference airplane, and it is designated here BF-200LR. A top view of both airplanes is provided in **Figure 69**. Basic data for BF-200LR is given in **Table 8**.

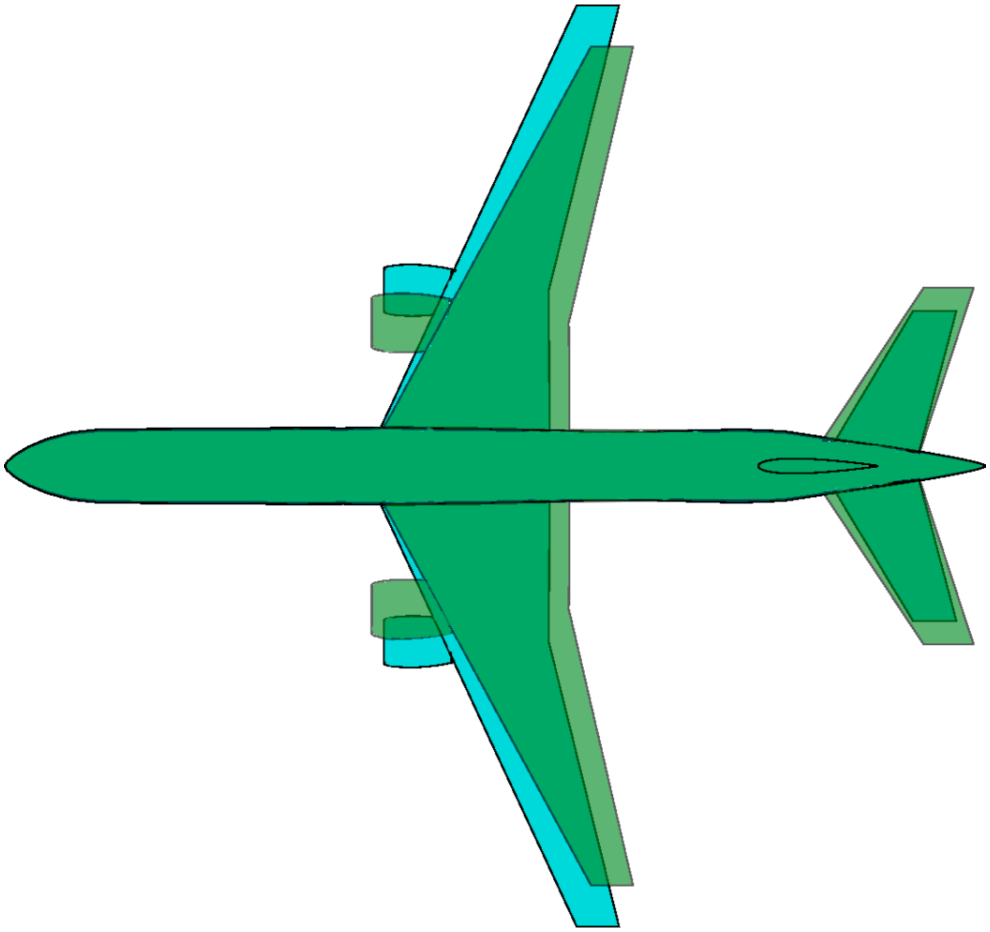


Figure 69: B757-200 and BF-200LR top-view juxtaposition for comparison purposes

Table 8: Basic data for BF-200LR

Wing Aspect ratio	Wing area	Wing sweep	Overall thrust	Engine BPR	Engine diameter	Two-class Accommodation @ 32-inch pitch in Y-class	Design range
9.97	175.35 m^2	22.29°	285.6 kN	6.00:1	1.93 m	192	3,420 nm
MTOW		**OEW**		**MMO**	**Fuel capacity**		**Range**
100,111 kg		53,573 kg		0.81	30,924 kg		3,470 nm$^\alpha$
α @ FL370 and FL390, Mach number of 0.80, payload of 19,200 kg, ISA+0 °C, takeoff with MTOW, 200 nm to an alternate, 30 min loiter @ 1,500 ft							

This additional study was intended to design the break- and tip-station airfoils, jointly to find out the optimal wing twist angle, and the incidence of the break station for a set of objectives and constraints.

Airfoil geometry impacts enormously the maximum lift coefficient. According to Ref. [91], the C_{Lmax} at landing for the B757-200 is 2.38, considerably lower than that of its computational representation airplane considered here, which presents a value of 3.06 for this coefficient. This is due to the different wing airfoil geometries that compose the actual airplane and its counterpart of the present work. There is no information available about the B757-200 airfoils and that utilized here are transonic airfoils that intend to match the maximum relative thickness of them as close as possible. On the other hand, the MMO of B757-200 is 0.86 [91], relatively high considering the moderate sweepback angle of its wing. Despite the typical mission established for the B757 was to fly domestic routes in the United States, even for medium populated cities, the North American manufacturer preferred a configuration with higher speed, which certainly worse field performance. In general, airfoils presenting good transonic characteristics like higher divergence Mach number tend to reveal some degradation of field performance. Thus, an optimization with these two conflicting objectives may produce interesting results.

The optimization task that was then carried out included two objectives:

- the maximum value of the Mach x Lift/Drag (MLD) in the 0.70-0.85 Mach number range,
- and the maximum lift coefficient at landing configuration.

The constraints for this problem are:

- the maximum relative thickness of break-station airfoil must be greater than that of tip-station airfoil.
- The maximum relative thickness of the root-station airfoil must be greater than that of the break-station airfoil.
- MMO must be higher than 0.80.
- Fuel capacity greater than 30,000 kg.
- C_{Lmax} @ landing must be higher than 2.5.
- Drag rise to MMO lower than 20 counts.

The aerodynamic coefficients at the transonic regime for this simulation were calculated by an ANN system with 64 input variables for wing-fuselage-winglet combinations [92] and the remaining aircraft components a Class-II approach was employed. The airfoils were generated by 14 weights applied to the geometry of 14 airfoil geometries composing a database. Thus, 42 input variables are necessary to define the geometry of three basic wing stations, root, break, and tip. The clean-wing maximum lift coefficient is calculated by a full-potential code in combination with a 2D panel code by the critical section method [93]. Utilizing this information and additional configuration characteristics, the Datcom method [94] is then employed for the estimation of the C_{Lmax} coefficients at landing and takeoff configurations.

The simulation for airfoil optimization was stooped after 20 generations with 1200 individuals being analyzed. The multi-objective *gamultiobj* genetic algorithm of MATLAB® was employed in all simulations that were carried out here [95]. **Figure 70** shows the Pareto front and the characteristics of feasible and unfeasible individuals that arose in the simulation. A considerable improvement of MLD for the reference airplane was obtained, the same cannot be said for the C_{Lmax}. **Figure 71** compares the original and optimized airfoils that resulted from the optimization run.

Table 9 shows the characteristics of a selected individual from the Pareto front. MMO of 0.82 was obtained with the utilization of the surrogate ANN system considering a lift coefficient of 0.50. The aircraft module of the present design framework calculated the MTOW based on the same mission as that for the BF-200LR of maximum takeoff thrust, and a value of 97,841 kg was obtained. This is considerably lower than that shown in Table XI, of our reference airplane. However, this be only credited to the new airfoils, because MMO was increased from 0.81 to 0.82. A lower MMO means lower structural loads, and this will lead to a lower OEW. In addition, an aircraft with a similar mission of B757-200 when fitted with new, high by-pass engines, higher aspect ratio wings, and optimized airfoils, recorded a 17-t decrease in MTOW.

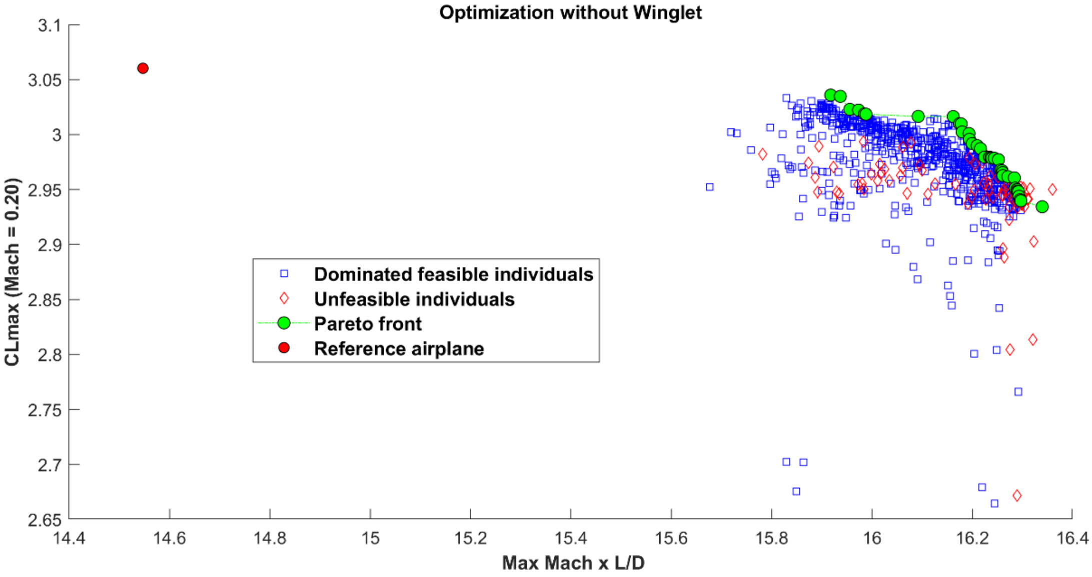


Figure 70: Optimization of wing airfoil geometry for a B757-200 similar aircraft

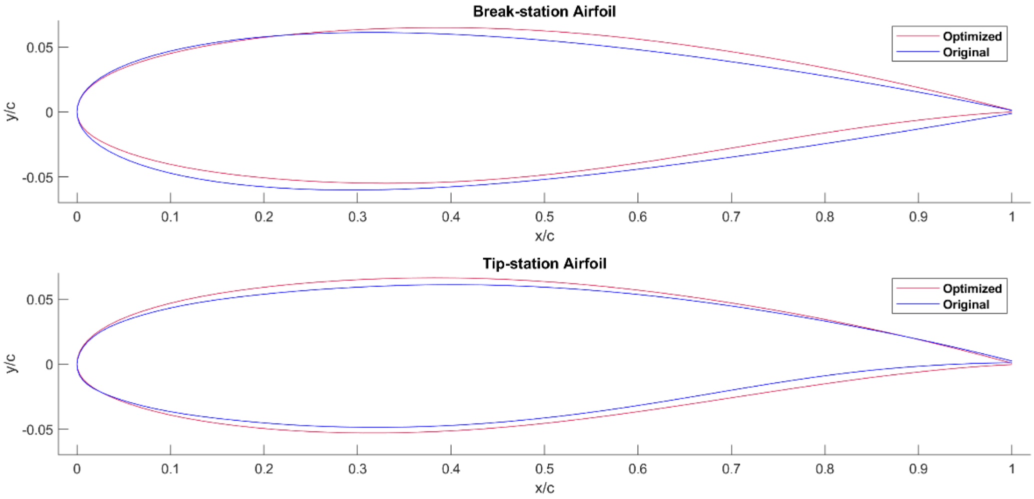


Figure 71: Break- and tip-station airfoils of the selected airplane optimal configuration

Table 9: Optimization of wing airfoils of BF-200LR

Parameter	Value
Max. MLD	16.21
C_{Lmax} landing configuration (Flap 40°)	2.987
C_{Lmax} clean wing	1.59
MMO	0.82
Fuel capacity	31,287 kg
Max. relative thickness of root airfoil	15.1%
Max. relative thickness of the break-station airfoil	12.0%
Max. relative thickness of tip airfoil	11,9%
Wing twist angle	-3.74°
Incidence of wing break station	0.276°
MTOW	97,841 kg
OEW	52,854 kg
ΔMTOW$^{\alpha}$	-2,270 kg
ΔOEW	-719 kg

α @ FL370 and FL390, Range of 3,470 nm @ Mach number of 0.80, payload of 19,200 kg, ISA+0 °C, takeoff at MTOW, 200 nm to an alternate, 30 min loiter @ 1,500 ft

D. Optimization with prescribed pressure coefficient distributions

An optimization task was carried out for the design of a 90-seat airliner wing. Two objectives were considered: the first one is the maximization of $M\,{}^{L}\!/_{D}$; the remained objective was the minimization of the square error of the difference between the calculated Cp distributions and the prescribed ones at six stations along wingspan ($M_\infty = 0.78, \alpha = 1.8^o$). An airfoil database consisting of 21 geometries was used to build up the root, break, and tip station airfoils. The airfoils belonging to this database vary from NACA airfoils to supercritical ones from NASA e RAE (Royal Aeronautical Establishment). A fixed wing area of 92.29 m^2 was adopted for the airplane configuration considered in the present optimization task. The design variables are as follows:

- Wing quarter-chord sweepback angle
- Wing twist
- Incidence of the break station
- Wing aspect ratio
- Wing taper ratio
- Location of the break station (fraction of semispan)
- Root airfoil geometry (defined by 21 weights)
- Break-station airfoil geometry (defined by 21 weights)
- Tip airfoil geometry (defined by 21 weights)

Two constraints were set:

- Clean wing $C_{L,max}$ higher than 1.65
- Wing fuel storage capacity greater than 8600 kg

The $C_{L,max}$ is calculated according to the critical section method [93] by a combination of a full potential code and the XFOIL panel code [96]. The aerodynamic coefficients were calculated with the ANN developed by Secco and Mattos [97]. The divergence Mach number ($MachDiv$) is computed when the drag coefficient increases 20 drag counts above the subsonic value considering a constant wing-body lift coefficient of 0.45. The maximum $M\,{}^{L}\!/_{D}$ is then obtained in the Mach number range of 0.65-*MachDiv*. The multi-objective *gamultiobj* genetic algorithm of MATLAB® was employed in all simulations that were carried out here [95].

Figure 72 shows the Pareto front composed of four configurations and the individuals that surfaced during just ten generations of the optimization task. The optimal configuration with the lowest $M\,{}^{L}\!/_{D}$ presents the better agreement with the prescribed Cp distributions (**Figure 73**). The divergence Mach number of this configuration is 0.786, which is slightly above the Mach number of the prescribed distributions ($M_\infty = 0.78$). This is a good result, but it may be not enough to guarantee that off-design undesired behavior arises. Considering that the computation of aerodynamic coefficients has become very fast with the utilization of ANNs, is highly recommendable to set up constraints for the divergence Mach number, in addition to those two already employed here. The divergence Mach number should be higher than the Mach number where the Cp distributions are prescribed. This constraint must also encompass not just one but some lift coefficients.

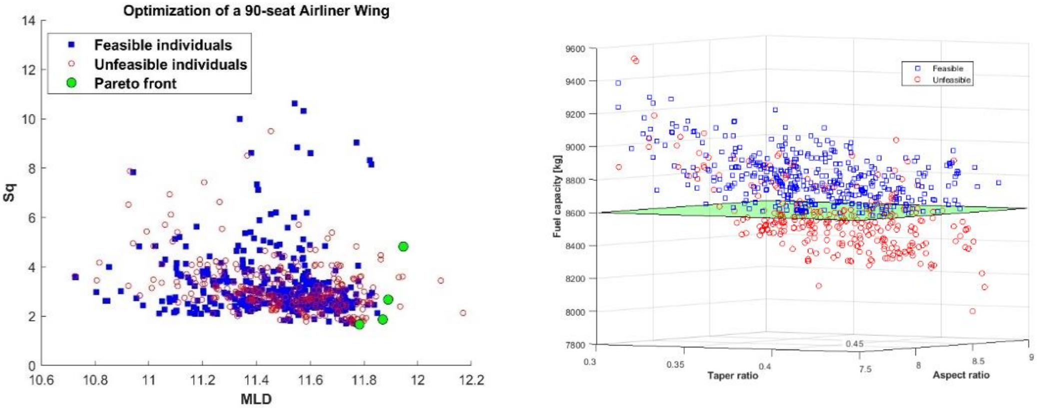


Figure 72: Simulation for the wing design of a 90-seat airliner. *Sq* is the square error that resulted after computing the difference between the prescribed and calculated *Cp* distributions at six stations along the wingspan

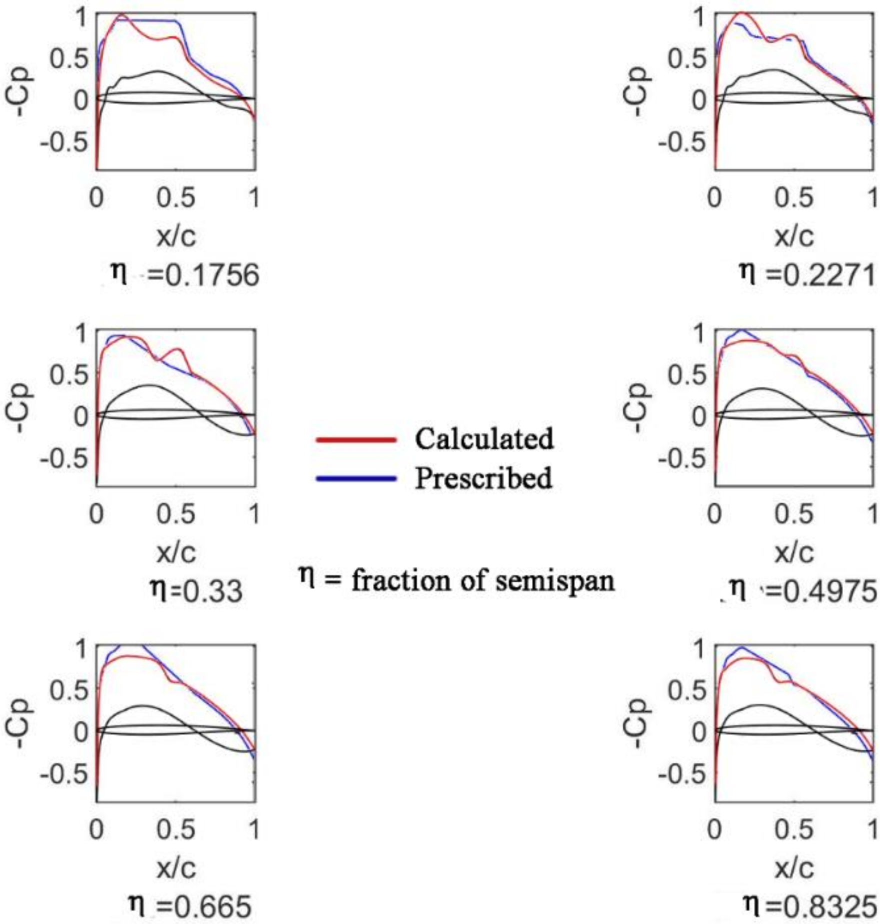


Figure 73: Prescribed and calculated Cp distributions for the optimal configuration with the lowest $M\,{}^{L}/_{D}$

6. Flutter Constraints into Conceptual Design of Transport Airplane

6.1 Introduction

Long-haul jet airliners usually present high aspect ratio wings because they perform the cruise phase at the highest possible $M \times {}^{L}\!/_{D}$. This is achieved at Mach number lower than the maximum operational one, meaning that the lift coefficients are moderately higher during a large part of the cruise phase and, naturally, during the climb phase. The induced drag coefficient is proportional to the square of the lift coefficient and the inverse of the aspect ratio. Besides a weight penalty, increasing the wing aspect ratio will lead to many adverse effects such as tip stall, lower flutter speed, and pitch up, the latter resulting from the combination of high aspect ratio wing with higher sweep angles. The increase in aspect ratio has a two-sided effect on induced drag because Oswald's factor [98] is playing an important role, as well. There is an additional increase in this drag component that is caused by the loading increase at the outboard wing, which can be mitigated by a combination of wing twist with chord distribution along the wingspan.

The flutter of a lifting surface involves the interaction of flexural and torsional motions. Separately, neither motion may cause flutter, but when considered conjunctly, at critical values of amplitude and phase angle, the forces produced by one motion excite the other; the two types of motion are then said to be *coupled* [22]. Inertial, aerodynamic, and elastic are usual sources of coupling. However, modern aircraft configurations with vortex-dominated flows may involve contained regions of separated flow, unsteady phenomena comprising of separation and subsequent flow reattachment, stalling conditions, besides various time-lag effects between the aerodynamic forces and the motion [99].

Even the addition of winglets must be carefully managed for preventing the appearance of flutter in the flight envelope. The transformation of the Boeing 737-800 airliner into the Boeing Business Jet (BBJ) involved the addition of winglets and for this reason, each wing received 75 lb of concentrated mass to mitigate flutter [100]. The quest for a more efficient aircraft has bumped into the limits of the aluminum-built cantilever wing configuration. Therefore, many approaches have been adopted for new possibilities to mitigate flutter for high-aspect-ratio wings such as the Truss-Braced Wing concept [101] or extremely flexible composite wings [102]. Another possibility is the increase in wing aspect ratio, which enables a reduction in induced drag. However, as mentioned before, cantilever wings of a higher aspect ratio are more likely to have flutter problems. Therefore, studies involving the flutter phenomenon become an even more relevant design consideration for new aircraft concepts and require those constraints to be included in the design as early as possible, preferably at the conceptual design stage. This way, costly changes are avoided at later stages of the aircraft development program.

Problems with torsional divergence affected aircraft in the First World War. They were solved by trial-and-error and stiffening of the wing without carrying for broader implications [103]. The first recorded and documented case of flutter in an aircraft occurred to a Handley Page O/400 English bomber during a flight in 1916 [103]. The airplane suffered a violent tail oscillation that led to an extreme distortion of the rear fuselage and the elevators to move asymmetrically [103]. Although the aircraft landed without injuries and fatalities, in the subsequent investigation some recommendations stated that left and right elevators should be rigidly connected by a stiff shaft. These recommendations subsequently became a design requirement. In addition, the National Physical Laboratory (NPL) was required to investigate the phenomenon theoretically.

The first formal flutter test was carried out by von Schlippe in 1935 in Germany [103]. The test aircraft was submitted to vibration at resonant frequencies at progressively higher speeds. Schlippe plotted amplitude as a function of airspeed and a rise in amplitude would suggest reduced damping with flutter occurring at the asymptote of theoretically infinite amplitude. This idea was applied successfully to several German aircraft until a Junkers Ju 90 military transport suffered flutter and crashed during a flight test in 1938 [103].

The distribution of heavy mass items in the wing can be optimized for flutter prevention. Wing structural design is driven by both strength and stiffness criteria. For example, if the torsion carrying structure of a wing is designed by a stiffness requirement, the wing would probably consist of a structure that carries its normal stresses in the wing skin with a minimum of stringers and flanges. This type of wing structure would require several spanwise webs to stabilize the heavily loaded skin. For a wing designed initially by strength criteria to withstand a specked load factor, it is straightforward that a higher torsional stiffness and hence a higher flutter speed will result if the ratio of stiffener area to skin area is reduced to a minimum. In addition, the use of higher strength alloys, which have no corresponding increase in modulus of elasticity, tends to make flutter more critical for wings designed for strength only. Wing aspect ratio should be treated as a design variable and not as objective as stated in Ref. [104]. Beyond certain limits of this design variable, flutter can no longer be avoided by mass distribution in the wing or by the addition of ballast. For non-linear systems, some design considerations were outlined in the first Section of this Chapter.

In the work of Opgenoord [105], the influence of aeroelasticity was investigated during the conceptual design of a transport aircraft by using a multi-disciplinary optimization tool. His article described an airfoil flutter model based on low-order physics. The flutter model handles the smallest vorticity moments of the flowfield and the volume source density perturbations. The model is calibrated using unstable 2D transonic CFD simulations. The resulting aeroelastic system combines the calibrated aerodynamic model with a beam model. The low computational cost of the model allows its incorporation into a conceptual design tool for state-of-the-art transport aircraft. The results presented in that work showed that the inclusion of flutter restrictions in the optimization of the aircraft design limits the wing aspect ratio, resulting in higher fuel consumption. Therefore, it was concluded that flutter limits performance gains obtained by using more advanced materials in the wing.

Jonsson et al. presented a review of the methods of development, implementation, and application of flutter and post-flutter constraints in the optimization of aircraft designs [106]. Furthermore, discussions about additional requirements associated with this type of project optimization, such as acceptable computational cost, smoothness of function, robustness, and derivative calculus, were introduced. In its conclusion, a summary of the current state of this field and the main open problems were presented.

A comparative sensitivity study for aeroelastic instability of aircraft wings in subsonic flow was carried out by Berci [107], which used analytical models and numerical tools with different multi-disciplinary approaches. His analyses were based on previous works and covered parametric variations of aero-structural properties, quantifying their effect on the aeroelastic stability frontier. Considering various degrees of fidelity both theoretical and computational calculations were evaluated, for possible practical applications in the preliminary design and optimization of aircraft. The results presented in that comparative study recommend the use of a hybrid strategy for the analyses, in which the flutter limit is obtained using a high-fidelity approach, while flutter sensitivity is calculated using a low fidelity approach.

The present work investigates the impact of flutter speed constraints in the configuration of a transport airplane. The package NeoCASS [108], a sophisticated suite developed for aircraft structural sizing and flutter speed calculation, was integrated into an existing MDO platform for aircraft conceptual design [109]. Two simulations tasks were carried out, one incorporating the flutter constraint and the other not. The results were analyzed and are discussed here.

6.2 Aircraft Conceptual Design Framework

A. Multi-disciplinary design framework

A MATLAB-based aircraft conceptual design framework for conceptual airliner design was developed at ITA over the years counting on contributions from several masters and doctoral thesis [109] [49] [110] [97] [111]. This framework is composed of several modules, which can handle and integrate aeronautical disciplines, manage airplane configuration, and control the optimization process. In addition, this MDO tool can calculate airplane noise signature at ICAO reference points [49], estimate engine emissions [49], and can be coupled with unsupervised learning algorithms to provide airplane classification, [112].

The workflow of the airplane design framework of the present work is shown in **Figure 74**. There are many possibilities in terms of fidelity of the discipline modeling that is used for airplane sizing. For example, lift-drag characteristics can be calculated from simpler models like Class-I formulations to CFD codes, or even by surrogate ANN models [97]. NeoCASS can be considered a medium-fidelity structural and aeroelasticity tool and will be described in detail in a dedicated section of the present study, as well as the multi-objective genetic algorithms were chosen to perform the optimization tasks.

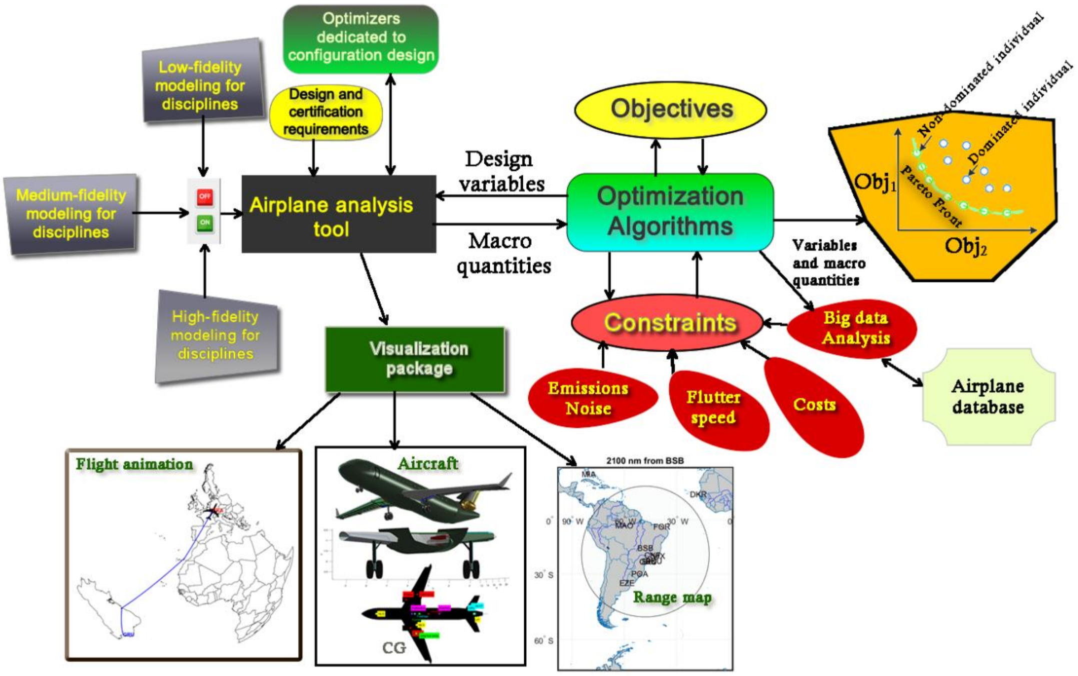


Figure 74: Overview of the present MDO platform

Important design requirements must be considered because they will generate a feasible airplane configuration when satisfied. Certification requirements such as climb gradient in the 2nd segment, cruise and missed approach thrust requirements, rate of climb at service ceiling, and balanced takeoff field length are some of them.

The multi-objective *gamultiobj* algorithm from MATLAB® was employed in all simulations that were carried out in this Section [95]. Two objectives were considered here: direct operating cost per nautical mile and an efficiency index for the configuration that was elaborated by the authors and will be discussed later in the following sections.

B. Airplane analysis tool

The airplane calculator workflow shown in **Figure 75** is an essential part of the MDO framework. From airplane geometry and topology, and mission requirements, an iterative process for MTOW calculation is performed. Designated here of airplane analysis tool (AAT), this component is the masterpiece of the framework. AAT provides an embracing description and characterization of the airplane. Many levels of fidelity for discipline representation are available. Noise and emissions footprint can be optionally calculated. Detailed mission performance and range evaluations are derived from flight profiles set by the user and obeying air traffic constraints, all based on the aerodynamics, mass, and engine characteristics of the airplane. It is possible to examine any design mission or off-design missions corresponding to specific takeoff weights or required block distances.

Realistic airplanes obey must obey some requirements:

- Enough thrust to fly in the service ceiling at MMO.
- Required 2nd segment climb gradient.
- Required FAR 24.119 landing climb gradient.
- Required FAR 24.121 landing climb gradient.
- Enough thrust to perform en-route climb.
- Takeoff field length at a specified atmospheric condition and altitude.
- Landing field length at a specified atmospheric condition and altitude.
- Fuel storage to perform a mission with a specified range and payload.

The airplane mission setup is highly detailed with a flight profile that makes accurate calculations of major airplane masses possible like MTOW, OEW, and MZFW. Air traffic constraints can be incorporated into the calculation, if required [113]. Several aerodynamic methods are available in the framework with different levels of fidelity, including an artificial neural network surrogate model. An in-house generic turbofan engine model is also part of the MDO framework [110]. It is possible to analyze the performance, emissions, and operational costs for missions inside a complex airline network with different takeoff weights or block distances [17]. The fuel storage capacity is precisely calculated considering the actual airplane geometry and structural layout.

All mission segments are analyzed, and optimal and transitional climb altitudes are calculated considering buffet margins, available engine thrust, rate of climb, and accurate airplane lift-to-drag ratio. Alternate airport, hold and descent patterns, reserve and maneuvering fuels, and weight allowances are additional parameters. Typically, after the second segment climb, the in-route climb uses airspeed schedules, namely, a calibrated airspeed (CAS) segment followed by a constant previously stipulated Mach number up to the initial cruise altitude. The cruise phase can be flown at a constant altitude or following a step-increasing altitude pattern. There are two cruise speed profiles, one with a constant Mach number and a long-range pattern.

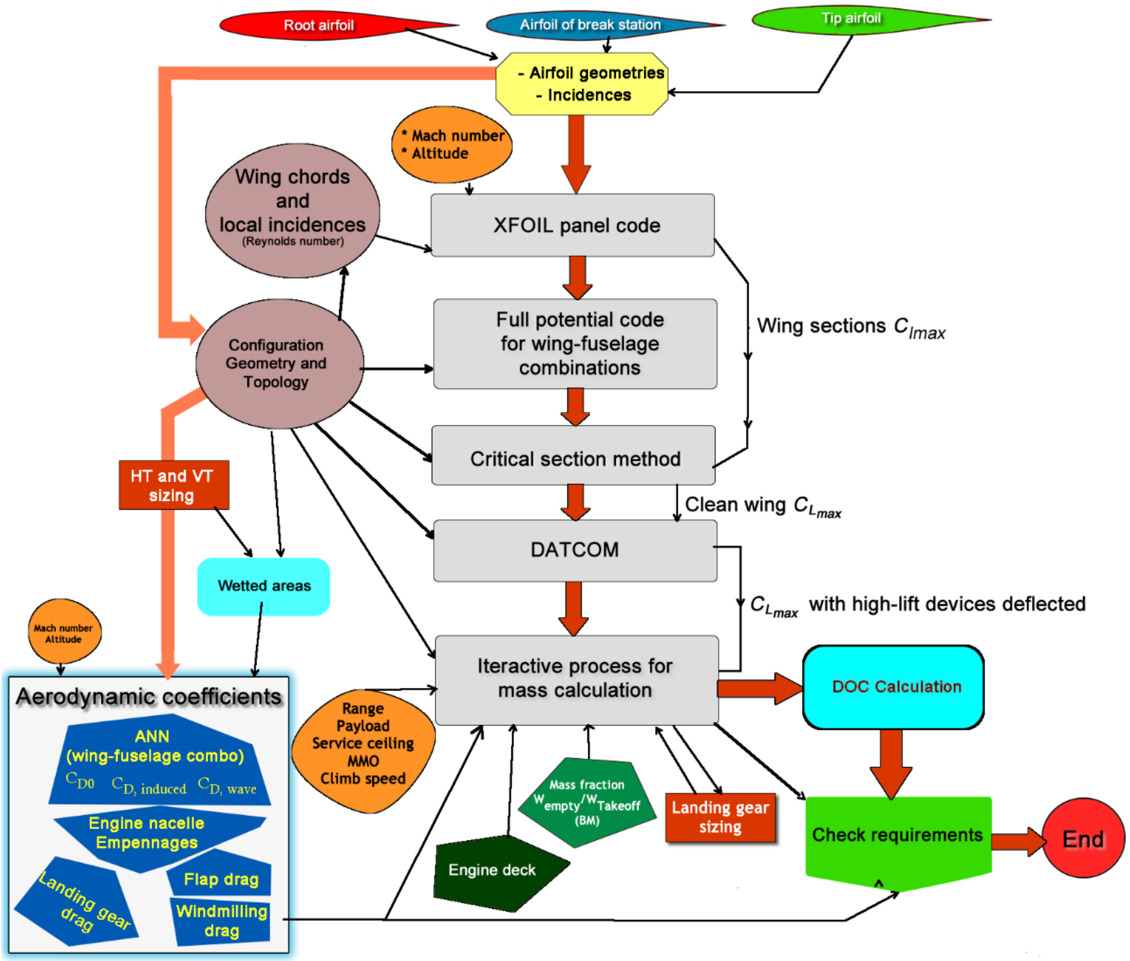


Figure 75: Workflow of the airplane analysis tool

C. Airframe weight and maximum takeoff weight

For the calculation of the maximum takeoff weight (MTOW), the estimation of the empty weight is needed, which can be supplied according to the MTOW (class I weight method) itself or it can be calculated by the sum of the individual structural weights and aircraft systems (Class II method). Within the Class I methodology, the following relationship between MTOW and OEW was developed at ITA:

$$\frac{OEW}{MTOW} = \left[a + b\left(\widehat{MTOW}\ \widehat{AR}\ TW\ \widehat{S_w}\ MMO\ \hat{L}_f\ F_w \Psi\ \widehat{SC}\right)^{C0} + c\left(\widehat{MTOW}\right)^{C1}\left(\widehat{AR}\right)^{C2}(TW)^{C3}\left(\widehat{S_w}\right)^{C4}(MMO)^{C5}(\hat{L}_f)^{C6}(F_d)^{C7}(\Psi)^{C8}(\widehat{SC})^{C9}\right] \tag{69}$$

Eq. 69 was tailored for a mix of units of the international and English systems. The MMO, *TW*, and F_w symbols refer to the maximum Mach of operation, the thrust-to-weight ratio, and the equivalent diameter of the cross-section of the passenger cabin, respectively.

Normalizations that were considered in **Eq. 69**, given in **Table 10**.

Table 10: Variable normalization for Eq. 69

$\widehat{MTOW} = \frac{MTOW}{50000}$	*MTOW* = Maximum Takeoff Weight [kg]
$\widehat{AR} = \frac{AR}{8}$	*AR* = Wing aspect ratio
$\widehat{WL} = \frac{WL}{100}$	*WL* = Wing loading [kg/m^2]
$\hat{L}_f = \frac{L_f}{30}$	L_f = Fuselage length [m]
$\widehat{\Psi} = \frac{\Psi}{20}$	*Ψ* = Quarter-chord sweepback angle [degree]
$\widehat{SC} = \frac{SC}{40000}$	*SC* = *Service Ceiling* [ft]

To obtain the exponents and multipliers of the terms of **Eq. 69**, the authors used a database of line aircraft containing information from 123 airliner models, with their entry into service from the 1950s up to others with commercial operation expected in 2020. **Figure 76** shows the relationship between the MTOW and the reference area of the wings for the aircraft in the database. An optimization algorithm was elaborated for the minimization of the mean quadratic error between the OEW/MTOW estimated by **Eq. 69** and the actual values of the airplanes in the database. The exponents and coefficients of **Eq. 69** are the design variables. The mono-objective genetic algorithm of MATLAB® 2019a [114] was used in optimization simulations and after 5500 generations the simulation was stopped. **Table 11** shows the coefficients and exponents that the optimization run provided. **Table 12** contains some estimation errors for the weight fraction of the present method. The agreement between the actual and estimated weights is very good.

Table 11: Values of the coefficients present in Eq. 69

a	b	c	C1	C2	C3	C4	C5	C6	C7	C8	C9
-0.2359	-0.3065	0.8900	-0.1731	0.0871	0.0857	0.1050	-0.0931	0.1220	0.2232	0.0857	0.0418

Table 12: Estimation error of weight fraction for some jet airliners

Airplane	Actual OEW/MTOW	Calculated OEW/MTOW	Percent error
Boeing 737-700	0.54157	0.53548	1.12
Airbus A320-200	0.54494	0.53829	1.22
B747-400	0.49077	0.4976	1.39
Boeing 757-300	0.53874	0.49639	7.86
Boeing 767-300	0.55681	0.55400	0.505
CRJ200ER	0.59806	0.59552	0.425
EMBRAER E170LR	0.55645	0.56798	2.07
Fokker 100	0.57074	0.55416	2.90
Tupolev Tu-104A	0.54737	0.56043	2.39
VFW 614	0.61048	0.60329	1.18

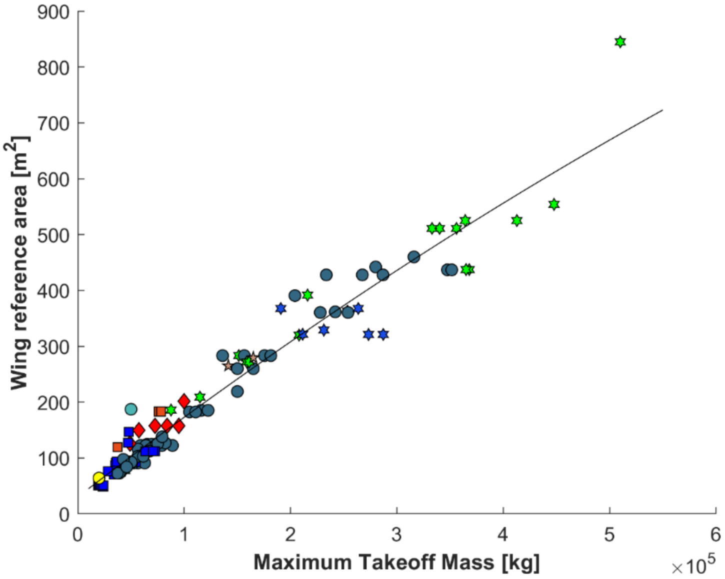


Figure 76: MTOW plotted against wing reference area considering airplanes utilized for the elaboration of Eq. 69. The different symbols in the graph reflect the different engine configurations

D. Engine weight

There are some simple methods as well as some more sophisticated methods for the estimation of turbofan engine weight [115]. But for WATE++, all of them are not accurate enough to be used in the optimization framework for airplane design. Thus, a methodology developed by Fregnani *et al.* [116] was adopted for turbofan engine weight estimation. This approach utilizes a formulation based on some engine geometric and thermodynamic parameters (**Eq. 70**).

$$W_E = \begin{matrix} T1 * (BPR/\overline{BPR})^a(OPR/\overline{OPR})^b(T_s/\bar{T})^c(D_E/\overline{D_E})^d(L_E/\overline{L_E})^e(\dot{m}/\bar{\dot{m}})^f + \\ T2 * (BPR/\overline{BPR})(OPR/\overline{OPR})(T_s/\bar{T})(D_E/\overline{D_E})(L_E/\overline{L_E})(\dot{m}/\bar{\dot{m}}) + T3 \end{matrix} \tag{70}$$

The coefficients and exponents of **Eq. 70** were obtained by an optimization using a genetic algorithm [114]. The objective was the minimization of the mean square error of known engine weights. A database, comprised of 20 engines, was elaborated which embraced a large variety of turbofan engines [14]. **Table 13** shows the T_1, T_2, and T_3 coefficients and a, b, c, …, f parameters obtained with the optimization process. **Table 14** shows the average parameters used for normalization in **Eq. 70**.

Table 13: Obtained exponents and coefficients for Eq. 70

Coefficient/exponent	Value
T1	2587.2461
T2	50.1920
T3	154.6179
a	-0.1965
b	-0.0718
c	1.0435
d	0.2493
e	-0.3444
f	-0.1455

Table 14: Parameters used for normalization in Eq. 70

Parameter	Value
$\overline{BPR}$	4.6911
$\overline{OPR}$	25.4000
$\overline{D}_E$ [m]	1.7906
$\overline{L}_E$ [m]	3.3276
$\overline{T}$ [kN]	148.1217
$\overline{\dot{m}}$ [kg/s]	464.7333

Table 15 contains weight estimation errors for some known turbofan engines.

Table 15: Weight error estimation for some known engines

Engine	Percent error
AE3007A1P	10.94
CF6-50C	2.55
JT8D-219	0.74
GE CF-34-10A	6.48
R&R RB211-535C	0.97
Trent 800-875	3.12
Williams FJ-44	4.29
Pratt & Whitney PW2040	0.44
GE-90/77B	0.31
R&R Tay 620	2.65

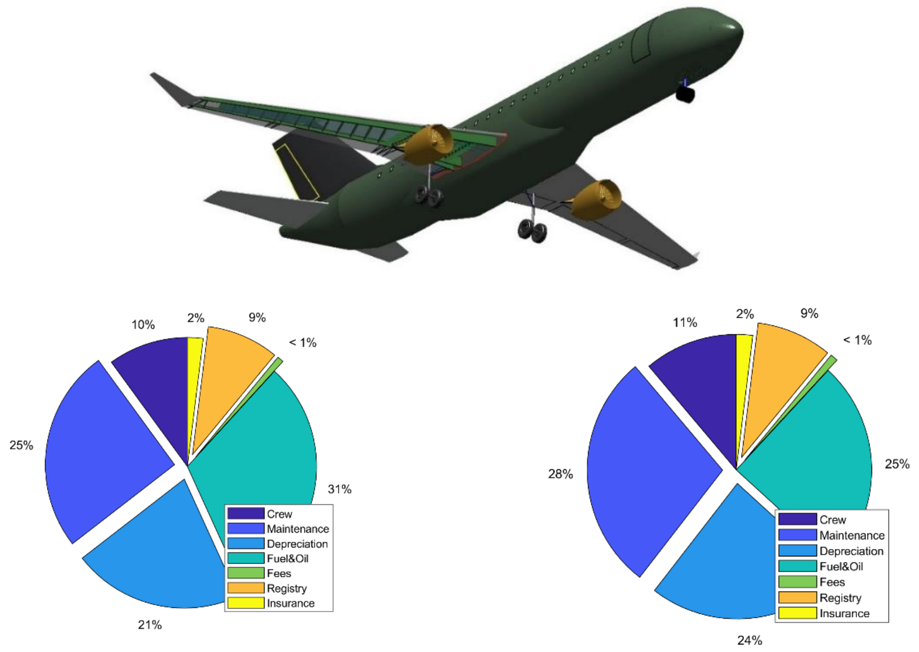
E. Direct operating cost

Although it is widely discussed which cost elements do belong to *Direct* Operating Costs and which do not, it is generally accepted that DOC includes those cost elements which depend on the equipment (aircraft) and crew that are necessary to perform a given flight. On contrary, *Indirect* Operating Costs (IOC) depend on the way an airline is administrated [117]. The ATA 67 DOC method [118], a good reference, considers as aircraft-dependent and hence part of DOC:

- cockpit crew costs,
- fuel costs,
- maintenance costs,
- depreciation,
- insurance (against hull loss).

Cockpit crew cost is heavily dependent on MTOW, and it is calculated in terms of flight hours. Training costs for the crew or maintenance personnel traditionally are considered as a percentage of the crew's fixed cost. For the engine, it is necessary to provide to the DOC calculation routine the number of engines, the engine weight, and the time between overhauls that is considered.

Figure 77 displays the result of a direct operating cost estimation performed for the hypothetical 76-seat ITA76ADV airliner with a design range of 2100 nm. The calculation considered two jet-fuel prices, US$ 1.70 and US$ 2.389, a 40.6% increase. The contribution to DOC increased from 25% to 31%, surpassing the crew slice related to that cost. Thus, as expected, the DOC is indeed sensitive to fuel price.



DOC: US$ 10.82/nm

Jet A1: US$ 2.387/gallon

DOC: US$7.85/nm

Jet A1: US$ 1.70/gallon

Figure 77: DOC breakdown for the ITA76ADV considering two Jet A1 fuel prices

The values here calculated for the DOC have estimated values and no comparison to actual values practiced by airlines is possible, because they handle this information as strategic and confidential. Nevertheless, it offers a relatively good basis for comparison among the airplanes that are an object of investigation. It is worth mentioning, that changing the fuel price will significantly alter the way the airplanes are evaluated, as shown in **Figure 77**, and eventually it could be part of a Pareto front. Ref. [119] provides graph information about the annual variation in jet fuel price.

Efficiency index

The authors elaborated a parameter that synthesizes the value of an airliner and that was designated here as efficiency index:

$$EI_{Liner} = \frac{\widehat{Rnm} \times MMO \times \widehat{NPAX}}{\left(TW \times \widehat{OEW}\right)} \tag{71}$$

The symbols of **Eq. 71** are:

$$\widehat{NPAX} = \frac{NPAX}{150} \quad \widehat{Rnm} = \frac{Rnm}{2500} \quad \widehat{OEW} = \frac{OEW}{40000}$$

An airplane presenting low efficiency may be an indication of a low range, a low cruise speed, or a high OEW for the mission it was conceived for. A low speed may appear not to be important, but it can be translated into a longer block time that impacts DOC. Besides, slower airplanes are obligated by air traffic control to fly at lower altitudes to not interfere with faster airplanes. Flying at lower altitudes, in turn, may cause a higher fuel consumption depending on the mission profile.

F. Passenger cabin sizing

Considering that the medium-range aircraft is the subject of the present research, a two-class cabin configuration was selected, where the cabin is configurated with 10% of the seats destined for first-class, and the other 90% are economy class, this proportion being suggested by [120]. The fuselage is modeled as a long cylinder with a constant ellipsoidal cross-section whose length is determined by the cabin length variable **(L_f)** and the shape of the ellipse by the height-to-width ratio variable (F_{hw}).

The cabin sizing module is divided into two steps. Initially, the layout of the cross-section of the passenger cabin is defined. The cargo hold space of the aircraft is also considered. In the present work, a container of type LD3-45W (2.44m x 1.53m x 1.14m) has been considered. The function makes the layout obey all the minimum clearance distances. A MATLAB® routine was written to design the cross-section of passenger cabins. After sizing the cross-section, the second step begins with the elaboration of the cabin floorplan. The cabin plan code calculates cabin length, toilet and galley positions, emergency exit sizing, and other relevant parameters. **Figure 78** shows an example of some cabin cross-sections. FAR 24.810 rules are employed in the incorporation of emergency exits. Additional considerations are carried out to comply with ditching regulations.

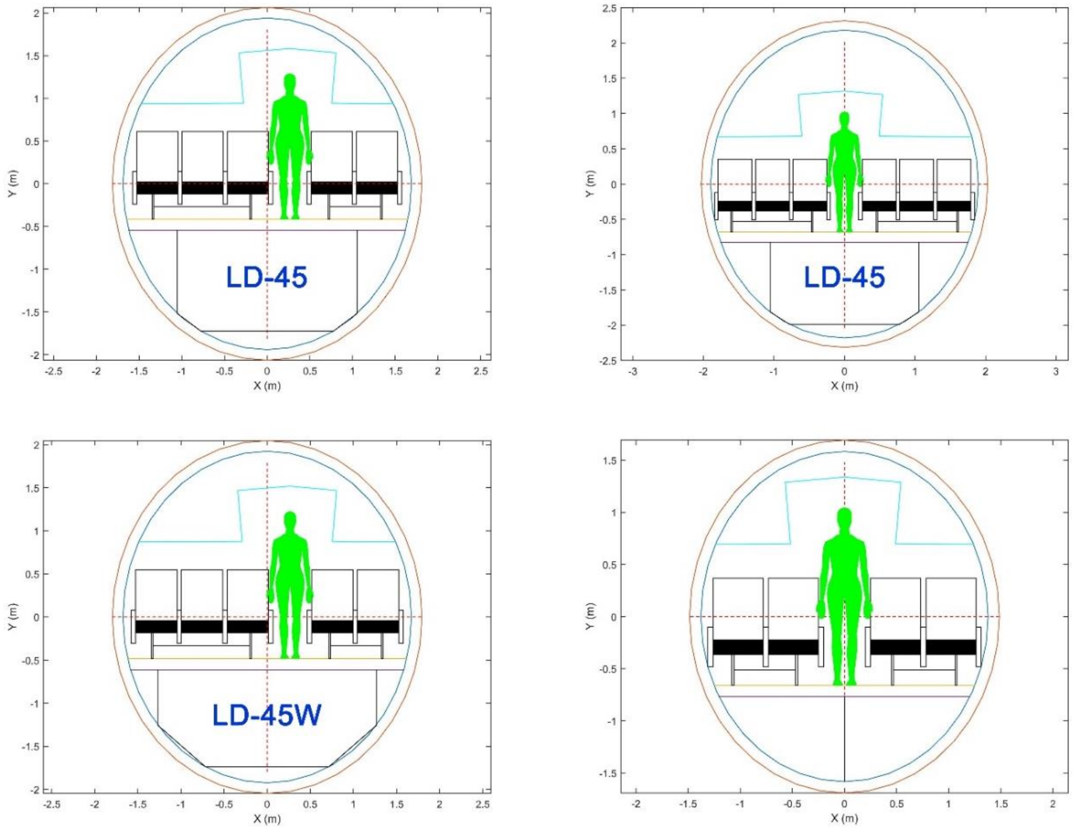


Figure 78: Examples of fuselage cross-sections

6.3 Verification of the B757-200 airliner

The B757 has a six-abreast cross-section and is powered by either RB211 or PW2030 and PW2040 Series turbofans. The B757-200 is the basic version, which entered service in 1983.

In August of 1978, Eastern Airlines and British Airways announced orders for the B757 and chose the RB211-535 to equip the airplane [121]. Designated RB211-535C, the 3-spool engine entered service in January of 1983 when the first B757-200 was delivered to Eastern Airlines. The engine has a nominal thrust of 37,400 lbf (166.36 kN) and an OPR of 21.2:1 [122]. In 1979, Pratt & Whitney launched its PW2000 engine, claiming 8% better fuel efficiency than the -535C for the PW2037 version [121]. The English engine manufacturer reacted and using the -524 core as a basis, the company developed the 40,100 lbf (178 kN) thrust RB211-535E4, which entered service in October of 1984. There are differences in appearance between the two versions like a mixed exhaust nozzle and a bigger fan cone for the RB211-535E4. BPR was slightly reduced to 4.40:1 from 4.46 for the 535C [122]. There is another version of the Rolls&Royce engine designated RB-211E4B, which has a takeoff thrust of 43,500 lb [123]. The 535E4 engine was also the first to use the wide chord fan which increased efficiency, reduced noise, and gave increased protection against damage from foreign object ingestion [121]. As a result, a relatively small number of -535Cs was installed on production aircraft in May of 1988. American Airlines ordered 50 B757s powered by the -535E4 emphasizing the engine's low noise as an important factor for their choice. The stretched version B757-300 entered service with Condor Flugdienst in 1999. With a length of 54.5 m, the type is the longest single-aisle twinjet ever built.

According to Ref. [124], The B757-200 has several sub-versions presenting different MTOW, OEW, and MZFW. Two configurations fitted with RB211-535E4B engines were selected as references for the ongoing work, and their payload-range diagrams are shown in **Figure 79** and some characteristics are given in **Table 16** [123]. Two ranges signaled by the dashed lines in the payload-range diagrams are related to a mission with a payload of 192 passengers.

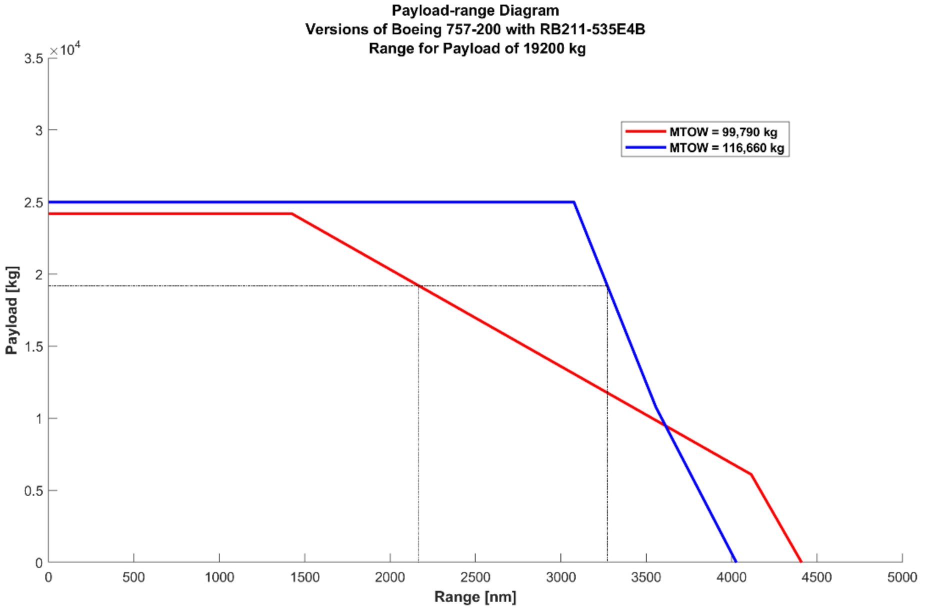


Figure 79: Payload-range diagram for some sub-versions of B757-200 [123]

Table 16: Characteristics of two B757-200 sub-versions fitted with RB211-535E4B engines [121] [122] [123] [125]

Sub-version	MTOW	OEW	MZFW	Usable fuel	Takeoff Thrust	TFL
1	99,790 kg	58,570 kg	83,460 kg	42,680 kg	37,400 lbf	1,660 m
2	115,650 kg	58,570 kg	84,360 kg	43,490 kg	43,500 lbf	2,070 m
L_f	**wAR**	**wS**	**wSw14**	**MMO**	**Aisle width of Y-class**	
46.97 m	7.82	185.25 m^2	25°	0.86	0.508 m	

Some of the B757-200 main characteristics calculated by the present methodology are given in **Table 17**. Again, the agreement between actual and estimated weights is particularly good. The year 1984 was set for the year of service entry. As expected, the airplane was categorized as a classic design (scaled trajectory). However, just for testing, if the year 1981 were adopted for that parameter, the airplane would be labeled by the entropy statistics classification program as a breakthrough, as also is expected. The reason behind this is simple: this way the configuration being analyzed passes its unique characteristics to the actual airplanes that entered service in 1983 and 1984, which are part of the database.

Table 17: Estimated values for the B757-200 fitted with RB211-535E4 by the present design

MTOW	OEW	Fuel Capacity	Range	Categorized as
115,100 kg	58,495$^{\alpha}$ kg	41,138$^{\beta}$ kg	3,272 nm$^{\alpha}$	Classic
α @ FL370 and FL390, Mach of 0.80, payload of 19,200 kg, ISA+0 ºC, takeoff with MTOW, RB211-535E4, 200 nm to an alternate, 30 min loiter @ 1,500 ft				
β wing capacity of 30,918 kg / ε Jet A1 price of US$ 2.387 US$/gal				

6.4 NeoCASS

NeoCASS is an open-source computational package written in MATLAB® that was conceived to be a structural analysis tool focused on the conceptual and preliminary design of aircraft [108]. It is also a structural module for the design framework CEASIOM, a project conceived by SimSAC. The CEASIOM project was started in 2006 with the support of several contributing institutions around the world. The software is divided into several modules, and Politecnico di Milano was responsible for developing a module for aeroelastic studies in conceptual phases, which was called NeoCASS.

The NeoCASS suite was developed with the following objectives:

- Ensure easy coupling with other codes.
- be relatively accurate for the conceptual phase and provide correct trend data to allow the project to move in the proper direction.
- be computationally efficient.
- be interchangeable between speed and accuracy, leaving the level up to the user required mesh discretization.
- do not need many model preparations.
- provide sensitivity derivatives by changing design variables.

Besides other capabilities, NeoCASS provides a method based on fundamental structural principles to estimate the structural weight of an aircraft, which is a hybrid method, because it uses both empirical calculations, based on the actual weights of existing aircraft, and analytical calculations, based on the utilization of simplified finite element models. In this way, it is possible to obtain good results even using unconventional aircraft configurations. For this, the software has an approach in which empirical calculations are first performed, so that, throughout the analyses, the structural weights are changed. For the analytical structural design, different operating conditions from standards or the user are used. The software can deliver both aircraft detail results, such as structural masses; static and dynamic aeroelastic behavior. The explanations about all the methods used in the software that will be presented below were taken from Cavagna and Ricci (2013), which is the software manual.

Figure 80 displays the NeoCASS architecture that was employed in the present study. The suite is composed of two main modules, which are GUESS (Generic Unknown Estimator in Structural Sizing) and SMARTCAD (Simplified Models for Aeroelasticity in Conceptual Aircraft Design), whose functions and operating principles will be explained in detail later. In gray are represented the preprocessing modules called AcBuilder (Aircraft Builder) and WB (Weight and Balance). These modules are considered pre-processing, as they are used only in the creation or preparation of the aircraft.xml file, which is in green color, which is the file that contains all the information of the aircraft and is used in all phases of the program. Also in green is another input file that is called smartcad.dat. This file merges data outputs from GUESS with some parameter settings for solving aeroelastic phenomena. In yellow are the files that must be configured by the user, where the characteristics of the maneuvers and analysis modes to be considered by the program are found. In light blue color are the program outputs, which are the analytical structural masses of the aircraft, the vibration modes, the *mesh.dat* file, which is a structural representation of the aircraft based on a beam model, and the file *smartcad.dat*, which is a file compatible with other software like NASTRAN for post-processing analyses.

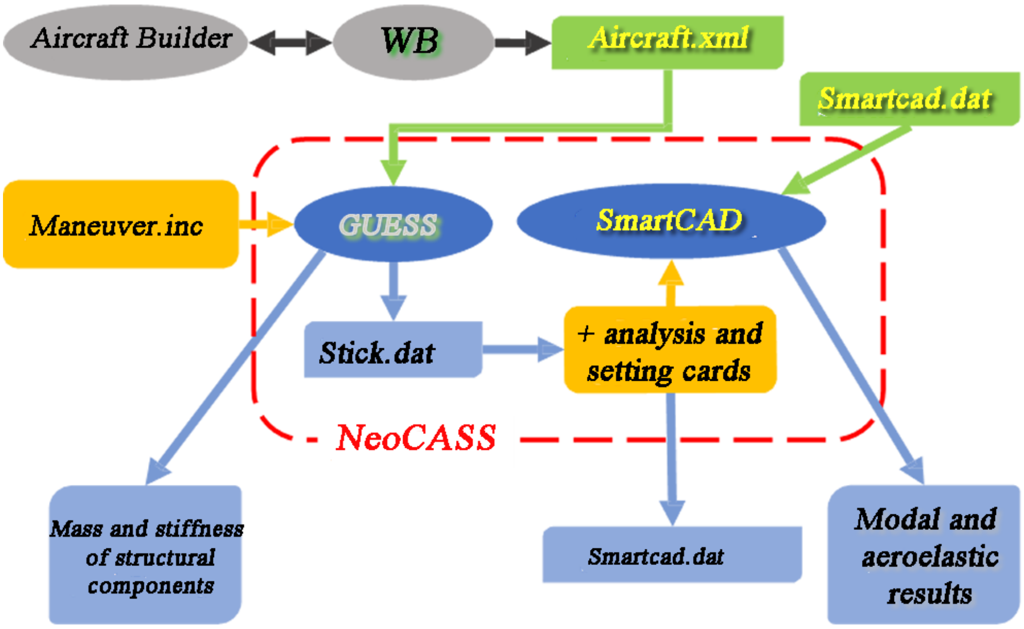

Aircraft Builder
WB
Aircraft.xml
Smartcad.dat
Maneuver.inc
GUESS
SmartCAD
Stick.dat
+ analysis and setting cards
NeoCASS
Mass and stiffness of structural components
Smartcad.dat
Modal and aeroelastic results

Figure 80: NeoCASS architecture

6.5 Sensitivity analysis

A. Influence of aspect ratio on the flutter speed

To measure the influence of wing geometric parameters on flutter speed, a sensitivity study was conducted by varying wing geometric parameters and determining flutter speed using the present design framework. Wing taper ratio, aspect ratio, and sweep were the geometric parameters that suffered variation.

The aspect ratio has a large influence on the flutter speed, as can be seen in **Figure 81**, which always poses a challenge for aircraft design teams. As expected, the results show that an aspect ratio increase leads to a significant reduction of flutter speed. This is due to the decrease of the wing inertia and the bending moment increase at the wing root. Another important aspect is the small effect of Mach number increase on the flutter speed of the wing analyzed here. The results shown in **Figure 81** agree with the results presented by Opgenoord [105].

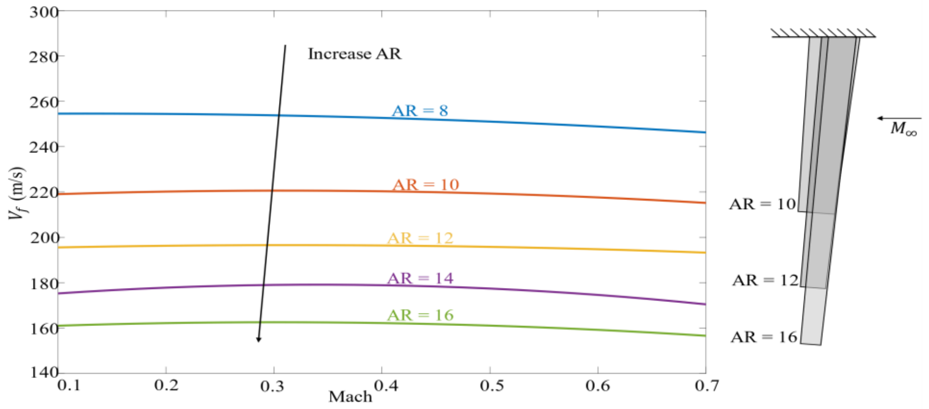


Figure 81: Influence of aspect ratio on flutter speed

B. Influence of wing taper ratio on the flutter speed

The taper ratio tends to decrease the flutter speed when it is increased, as can be seen in **Figure 82.** It can be observed that the Mach variation does not have much effect on the flutter speed and the taper ratio only determines the offset between the curves. This is to be expected since the increase in taper leads to an increase in the total bending and twisting moments, which increases the natural frequency of the wing, as shown in the work of Opgenoord [105].

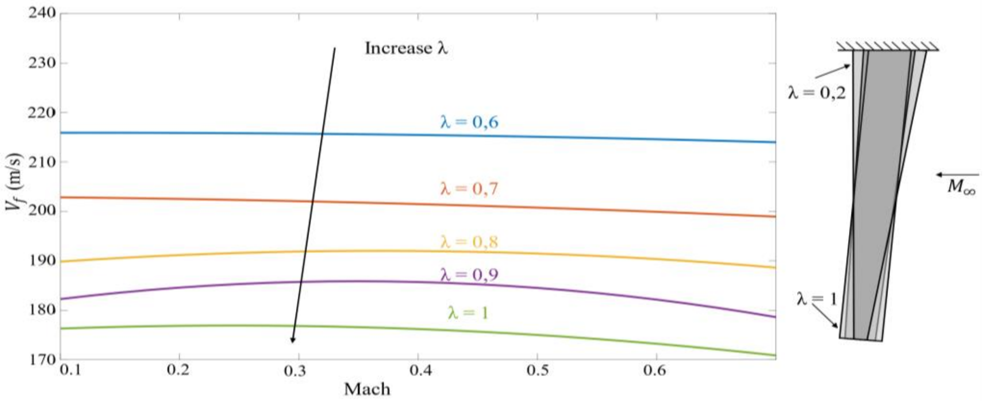


Figure 82: Influence of taper ratio on the flutter speed

C. Influence of sweep on the flutter speed

The wing sweepback angle also has a large influence on the flutter speed, as can be seen in **Figure 83**. It shows a similar behavior as for the aspect ratio as a function of Mach number. On the other hand, as the sweep increases, the flutter speed also increases, which is the opposite result of the aspect ratio. This result was also recorded by Barmby [126].

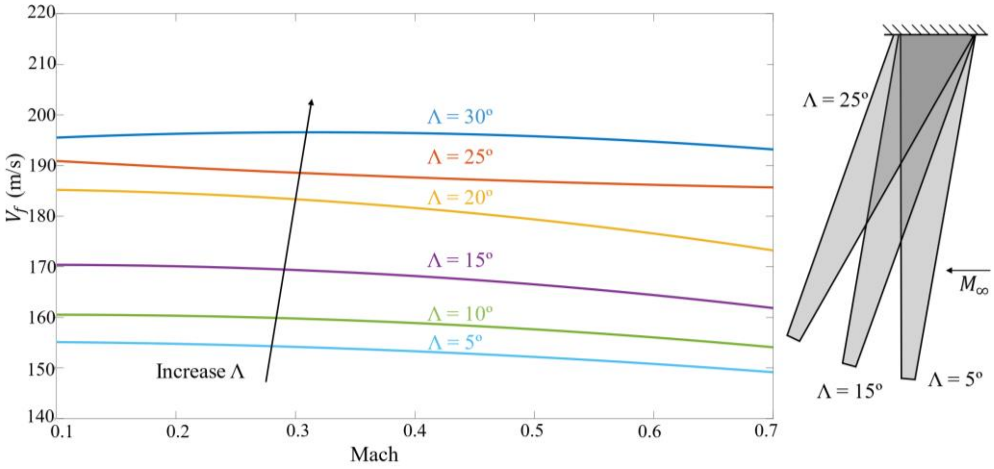


Figure 83: Influence of sweep on the flutter speed

6.6 Results

A. Design variables

The design variables that were used to generate the models of each aircraft are described in **Table 18**. It was necessary to establish lower and upper bounds for each of these variables, as this way it is possible to prevent the algorithm from searching for unconventional individuals during the optimization in regions where the models are not validated, generating misleading results or even in a region where these individuals are unfeasible. The boundaries for each of these variables are also shown in **Table 18**. The boundaries are based on typical values for airliners with passenger capacity varying from 130 to 220, with a small amount of tolerance being added to ensure design space freedom for the optimizer.

Table 18: Boundaries of variables of the present optimization problem

Variable	Lower boundary	Upper boundary
Wing Area ($\boldsymbol{S_w}$)	110 m²	210 m²
Aspect Ratio ($\boldsymbol{AR_w}$)	7.7	11.0
Taper Ratio ($\boldsymbol{\lambda_w}$)	0.24	0.45
Wing Quarter-chord Sweepback angle ($\boldsymbol{\Lambda_w}$)	22°	32°
Fuselage Length ($\boldsymbol{L_f}$)	36 m	50 m
Height/width of central fuselage ratio (F_{hw})	0.9	1.1
Design range starting with MTOW, ISA conditions ($\boldsymbol{R_{nm}}$)	2100 nm	3700 nm
Maximum Mach Operating (MMO)	0.78	0.83
Maximum aircraft certified altitude (SC)	35000 ft	41000 ft
Maximum Takeoff Thrust at sea level / ISA conditions (T)	220 kN	400 kN

B. flight envelope

To define the application of the flutter constraint, it was initially necessary to define the flight envelope of each aircraft being analyzed. The graph of altitude versus speed limit was then elaborated as an example (**Figure 84**).

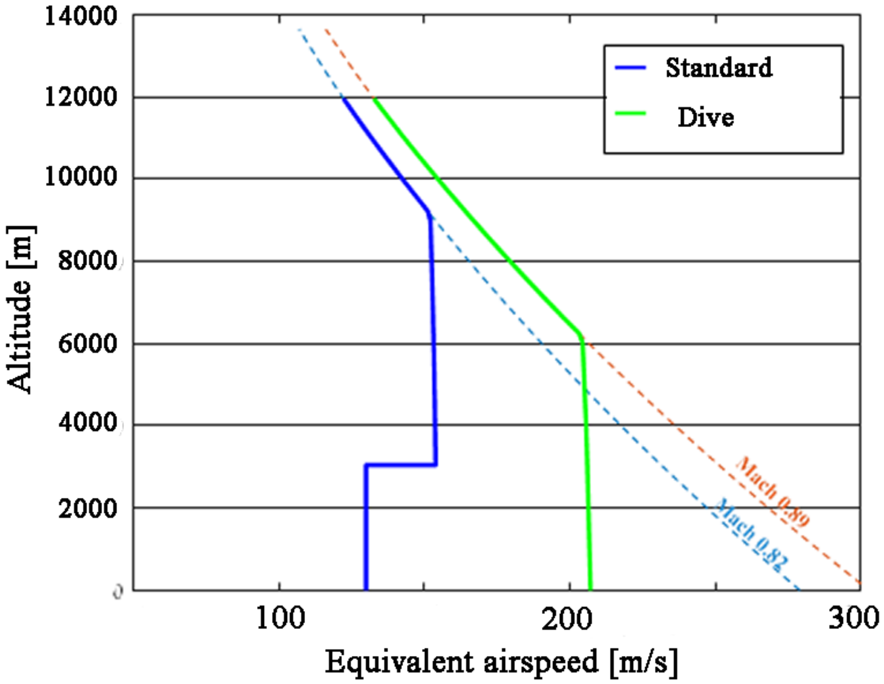


Figure 84: V-H diagram

The flight envelope shown in **Figure 84** restricts the maximum speed possible to be achieved by the aircraft at a certain altitude. The dark blue line represents the flight envelope for a typical mission, where a climb is performed at a constant calibrated speed of 250 kts up to an altitude of 10,000 ft. Afterward, a climb at a constant calibrated speed of 310 knots to crossover altitude is established. Finally, there is the cruise phase, in which the aircraft MMO was the cruise Mach number, which for the airplane of **Figure 84** was 0.82. The green line represents the aircraft's maximum dive speed. In the constant calibrated speed region, 350 kts were used for the envelope considering the dive speed, which was derived from the VMO of some aircraft in the category. VMO is usually associated with operations at lower altitudes and deals with structural loads and flutter. In the region at constant Mach number, the dive speed (Md) is given by:

$$M_d = MMO + 0.07 \tag{72}$$

The FAR 25.629, which determines the aeroelastic stability requirements of commercial aircraft, the minimum limit for the aeroelastic instability velocity is obtained from the V-H diagram of the aircraft's flight envelope. The aeroelastic instability threshold value is drawn from the aircraft's dive velocity curve and must be 15% greater at an equivalent velocity. If this value exceeds 1.00, in the constant Mach region, the value of 1 should be adopted for the Mach number. Based on the V-H diagram shown in **Figure 84**, the diagram of **Figure 85** was elaborated to comply with FAR 25.629 requirements.

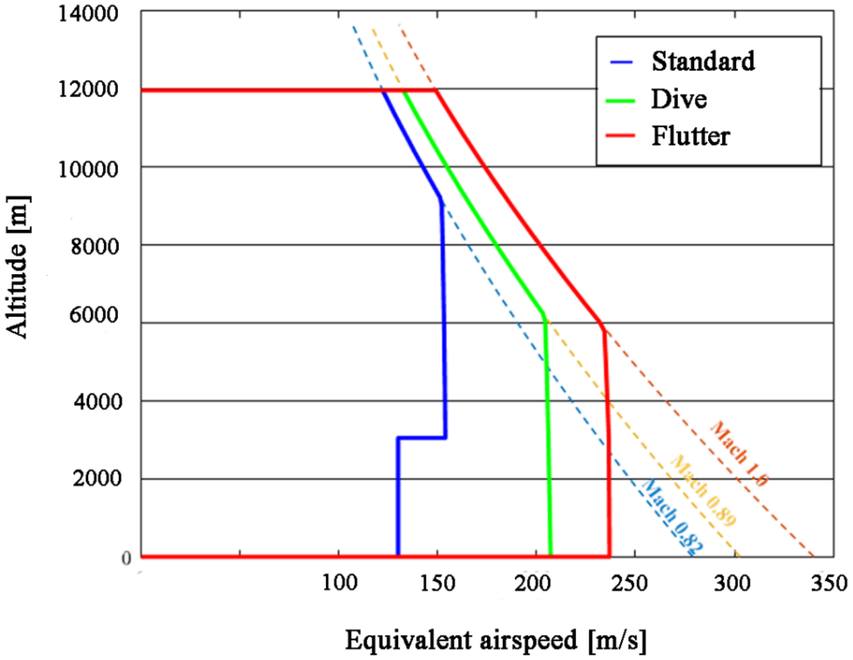


Figure 85: Flight envelope considering flutter speed constraints

C. Multi-objective optimization without flutter constraint

A multi-objective optimization design task was carried out without any flutter constraint. After 35 generations with 55 individuals using the MATLAB® multi-objective genetic algorithm, the number of feasible individuals that emerged from the computations was 1682, i. e., those not violating any constraint. (**Figure 86**) - out of a total number of 1925 configurations analyzed. The analysis of each plane took on average 30 seconds and the optimization task lasted about 16 hours on a Desktop PC fitted with Intel® I9-9800K processor and 32 GB of RAM.

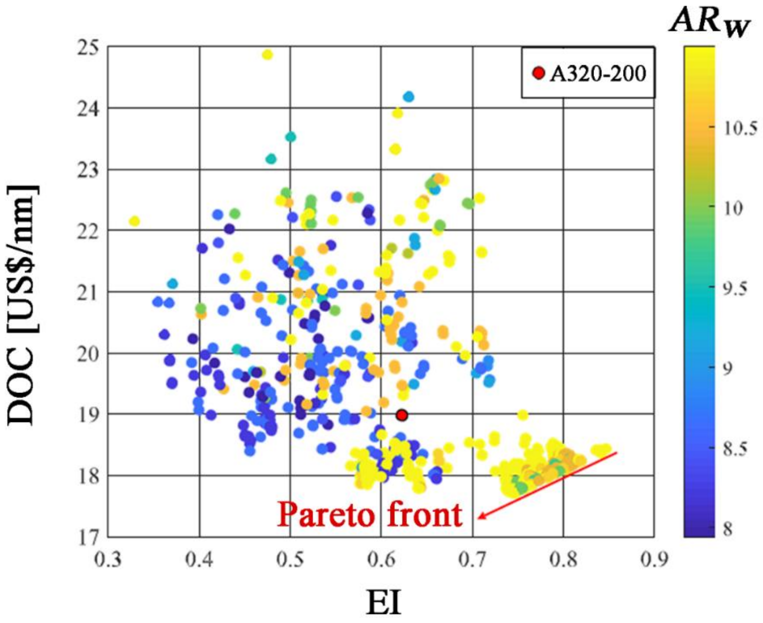


Figure 86: Optimization task with no flutter constraint

The individuals belonging to the Pareto front presented a wing aspect ratio higher than 9.92. **Table 19** contains some characteristics of the optimal airplanes.

Table 19: Some design variables of individuals belonging to the Pareto front of the optimization without flutter constraint

ID	S_w [m²]	AR_w	λ_w	L_f [m]	R_{nm}	DOC [US$/nm]	EI
1855	171.65	10.92	0,263	38.9	3470	18.46	0.846
1627	171.65	10.42	0,267	37.1	3470	18.04	0.790
1808	170.65	10.67	0,263	36.4	3470	17.85	0.765
1907	171.65	9.92	0,263	37.1	3471	18.07	0.795
1590	171.65	10.92	0,263	36.0	3470	17.73	0.744

All individuals in Pareto front for classic airliners share some features:

- MMO of 0.83.
- Wing quarter-chord sweepback angle of 22º.
- Height/width ratio of the central fuselage of 0.91.
- Maximum Takeoff Thrust at sea level of 303 kN.
- Maximum aircraft certified altitude 35,000 ft.

Table 20: Selected optimal airplanes compared to the A320-200 airliner

	A320-200	Highest *EI*	Lowest *DOC*
MTOW [kg]	76,653	89,579	84,771
OEW [kg]	43,298	49,514	47.,439
Npax	150	156	140
AR_w	9.50	10.92	10.092
Range [nm]	2550	3470	3470
Wing area [m²]	122.4	171.65	171.65
EI	0.622	0.846	0.745
***DOC* [US$/nm]**	18.99	18.46	17.73

Table 20 contains some estimated characteristics of the A320-200 airliner compared to those of two optimal individuals. The data for the airplanes presenting the highest *EI* and the lowest *DOC* is compared to those figures estimated for A320-200. To satisfy the maximization of the *EI*, the algorithm found solutions with larger wing areas to increase the fuel capacity, leading to an increase in range. Comparing the aircraft with the highest *EI* to the one with the lowest *DOC*, both present similar characteristics but different passenger capacity carried and the maximum take-off weight, characteristics that directly modify *DOC*.

The aircraft with the smallest *DOC* (#1590) presented a short fuselage and therefore a smaller number of passengers. This characteristic resulted in a reduction in empty weight of about 2 tons when compared with the aircraft with the higher *EI* (#1855). The smaller fuselage also leads to a reduction in the wetted area of the aircraft, a parameter directly related to drag. These factors are crucial for reducing the aircraft's fuel consumption and consequently *DOC*. On the other hand, the aircraft with the highest *EI* (#1855) had the longest fuselage length among the aircraft located at the Pareto front, which is expected, because the number of passengers is a parameter directly proportional to *EI*. The analyzes have shown that the characteristics that most affect the results of *EI* and *DOC* were aspect ratio, range, and wing area.

D. Multi-objective optimization with flutter speed constraint

The second task of the multi-objective optimization involved constraining the flutter speed within the flight envelope of the aircraft. Again, after 35 generations of 55 individuals, using the *gamultiob*j algorithm [95], a set of 1491 viable individuals, i.e., those not violating any constraint. (**Figure 87**) - out of a total of 1925 configurations analyzed. The analysis of each plane took an average of 180 seconds, and the total optimization runtime was about 96 hours. The Pareto front consisted of a total of 146 individuals. **Table 21** contains information on some individuals that belong to the Pareto front.

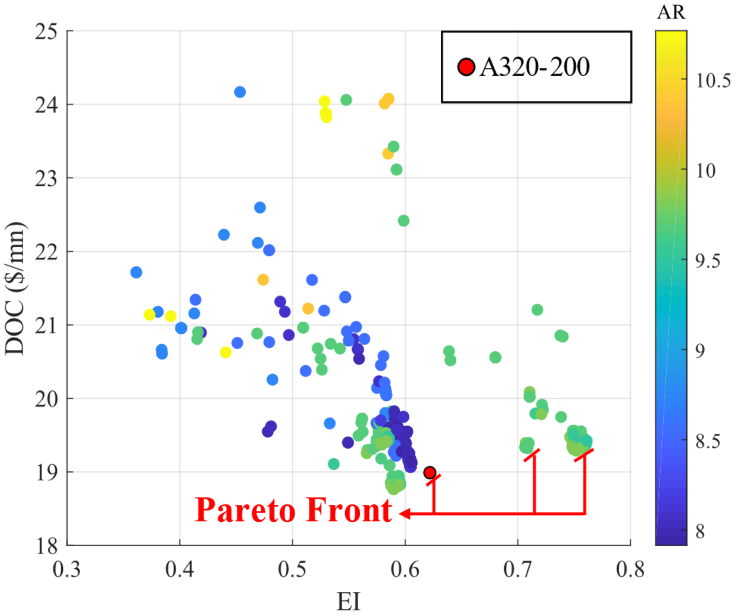


Figure 87: Optimization results with flutter constraint

The number of viable individuals has decreased with the inclusion of this new constraint. In this optimization run, the Pareto front is formed by three small, interspersed regions containing many individuals. This front happened at this time as the algorithm found two viable and distinct solutions which gave good results. Another interesting from the obtained results is the low occurrence of individuals with an aspect ratio higher than 10, which is due to the flutter constraint. In the present optimization, we note that one of the regions of the Pareto front is very close to the A320 plane, indicating results that are more in line with reality.

Table 21: Some design variables of individuals belonging to the Pareto front of the optimization with flutter constraint

ID	S_w [m²]	AR_w	λ_w	L_f [m]	F_{hw}	R_{nm}	SC [ft]	DOC [US$/nm]	EI
1462	177.97	9.49	0.342	39.3	0,90	3409	40,000	19.42	0.761
1107	177.97	9.68	0.258	39.2	0,90	3409	40,000	19.31	0.753
1839	144	8.03	0.256	39.2	0,90	2725	35,000	19.07	0.604
1273	144	7.94	0.258	39.2	0,92	2725	35,000	19.12	0.605
1870	144	9.81	0.258	39.2	0,90	2725	35,000	18.76	0.589

All individuals in the Pareto front for classic airliners share some features:

- MMO of 0.83.
- Wing quarter-chord sweepback angle of 22º.
- Maximum takeoff thrust at sea level of 339 kN.

Table 22 shows some characteristics of the A320-200 aircraft compared to those of optimal individuals with the highest *EI* and the one with the lowest *DOC*. Comparing the characteristics of the A320 with the characteristics of the higher EI aircraft, it is again noticeable that due to the objective of maximizing the *EI*, the algorithm has searched for solutions with larger wing areas to increase the fuel capacity, which leads to an increase in the range. However, comparing the characteristics of the A320 with those of the aircraft with the lowest *DOC*, it was found that at this time these aircraft presented very similar characteristics. This feature occurred because aircraft with high aerodynamic efficiency was no longer feasible by subtracting the flutter speed, thus the other parameters that compose the calculation of *DOC* became more important. As a result, smaller aircraft have become more attractive as they have lower maintenance, crew, and depreciation costs.

Table 22: Selected optimal airplanes compared to the A320-200 airliner

	A320-200	Highest *EI*	Lowest *DOC*
MTOW [kg]	76,653	91,554	82,343
OEW [kg]	43,298	50,424	46,342
Npax	150	160	158
AR_w	9.50	9.49	9.81
Range [nm]	2550	3409	2725
Wing area [m²]	122.4	178.0	144.0
EI	0.622	0.761	0.589
***DOC* [US$/nm]**	18.99	19.43	18.78

According to the figures contained in **Table 22**, range and passenger capacity are directly proportional to *EI*. Therefore, larger aircraft and consequently aircraft with greater range and number of passengers tend to have higher *EI* values. The aircraft found with the highest *EI* (#1462) meets this standard, as it combines the greater range and greatest number of passengers of the Pareto front.

6.7 Impact of aspect ratio on optimal design

The plots shown in **Figure 88** provide a good comparison between the results of the two optimization tasks carried out here. They show the feasible individuals that emerged in the optimization runs and related Pareto front at the same scale.

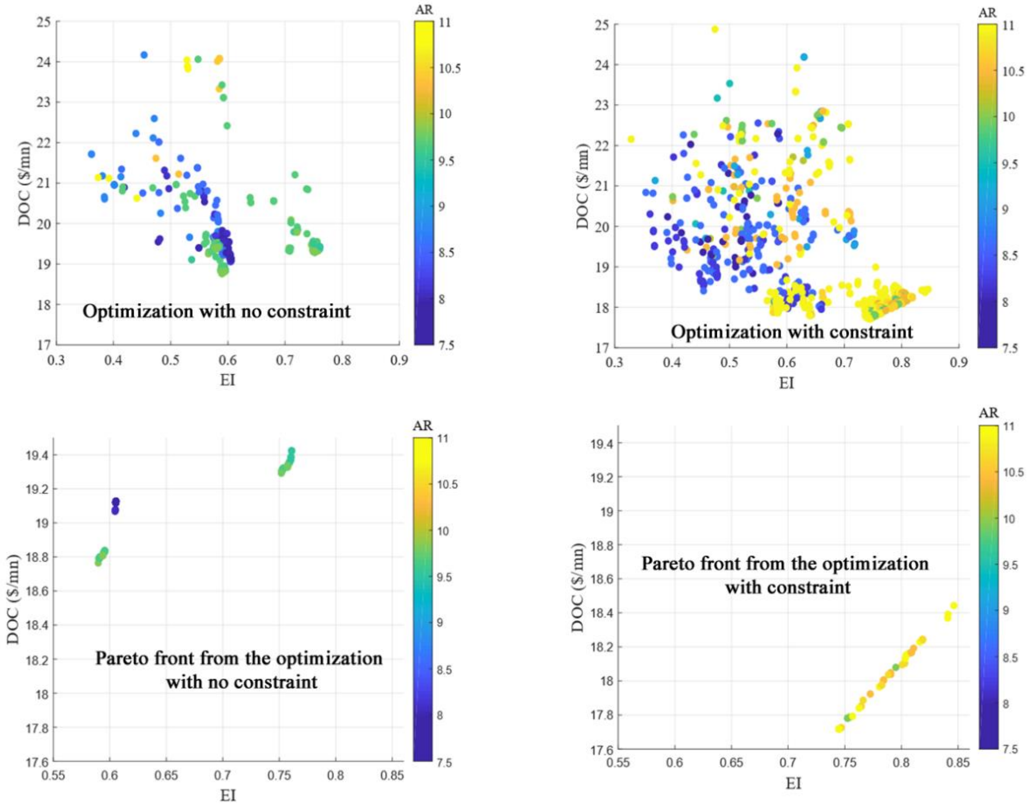


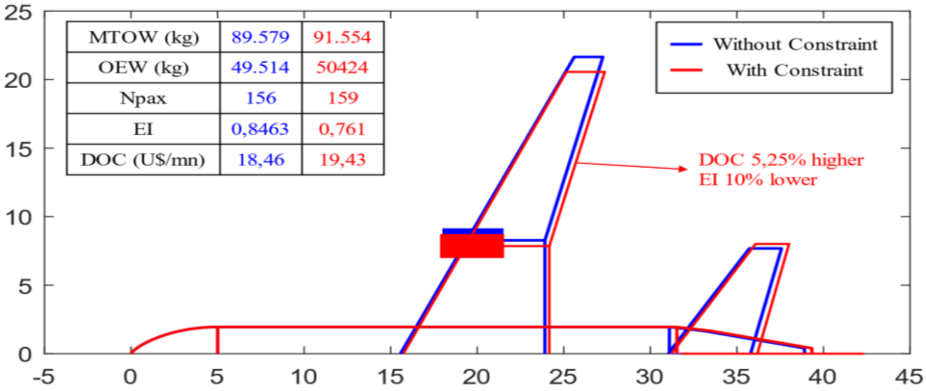
Figure 88: Comparison between the optimization tasks with and no flutter constraint

Comparing the results in **Figure 88 (a)** and **(b),** it is noticeable that optimization with no flutter speed constraint resulted in optimal individuals with an aspect ratio close to 11, in a much higher proportion than in the results obtained from the optimization task where the constraint was incorporated. This feature can be explained as individuals with a higher aspect ratio present lower flutter speed, as can be seen in **Figure 88**, which was penalized when the flutter constraint was added to the optimization procedure. The added constraint equation also led to a change in the position of the Pareto front, which is much more central in **Figure 88 (b)**, showing the significant degradation of the results found. When comparing the Pareto frontiers in **Figure 88 (c)** and **(d)**, it is noticeable that the Pareto frontier from the optimization with no flutter speed constraint is composed of individuals with a high aspect ratio than those from the optimization task where the constraint was imposed.

Table 23 contains the characteristics of the aircraft with the highest *EI* of the two optimizations and **Figure 89** shows a simplified top view of both. The two aircraft have similar solutions because in both cases maximizing the *EI* led the optimizer to look for aircraft with a larger wing area, passenger capacity, and range. However, due to the lower aspect ratio, the flutter-constrained aircraft has poorer aerodynamic performance and therefore a 10% lower *EI* and a 5.25% higher *DOC*. In addition, the poorer aerodynamic performance caused the optimizer to further increase some design variables of the flutter-constrained aircraft, which consequently has a larger wing area, cabin length, and overall traction. Despite the larger wing area, the constrained aircraft has a shorter range due to the degraded aerodynamic performance. Overall traction was increased because more traction was required to meet climb requirements due to increased drag.

Table 23: Comparison of design variables of higher EI aircraft

Aircraft	ID	S_w [m²]	AR_w	λ_w	Λ_w
With no constraint	1855	171.65	10.92	0.263	22
With constraint	1462	177.97	9.49	0.342	22
L_f [m]	**WR**	R_{nm}	**MMO**	**SC [ft]**	**T [kN]**
38.9	0.91	3470	0,83	35,000	303
39.3	0.90	3409	0.93	40,000	339



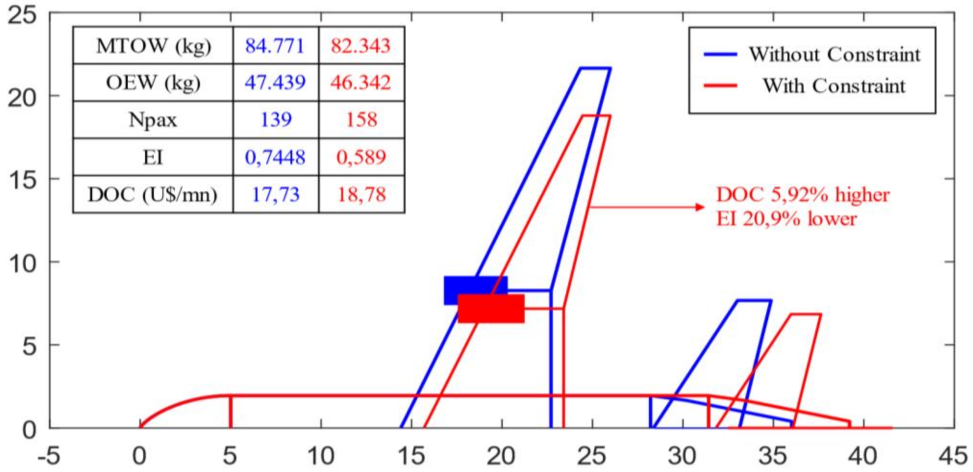
MTOW (kg)	89.579	91.554
OEW (kg)	49.514	50424
Npax	156	159
EI	0,8463	0,761
DOC (US$/mn)	18,46	19,43

Figure 89: The largest *EI* aircraft with and without restriction

Table 24 contains the design variables of the aircraft with the lowest *DOC* of the two optimizations, and **Figure 90** has their top views. In this case, the two aircraft presented significantly different characteristics, as the aircraft with the smallest DOC showed a longer range and a shorter fuselage length in the unconstrained optimization, while the range was reduced, and the cabin size increased in the constrained optimization. Despite this fact, the restricted aircraft presented poorer aerodynamic performance and consequently a 5.92% higher *DOC* and a 20.9% lower *EI*. This large difference in EI is justified by the different passenger numbers and ranges that place the aircraft in different market niches. Full traction was again increased to meet climb requirements.

Table 24: Comparison of design variables of the optimal aircraft with the lowest *DOC* that resulted from the two optimization tasks

Aircraft	ID	S_w [m²]	AR_w	λ_w	Λ_w
Without constraint	1855	171.65	10.92	0.263	22°
With constraint	1462	177.97	9.49	0.342	22°
L_f [m]	**WR**	R_{nm}	**MMO**	**SC [ft]**	**T [kN]**
38.9	0.91	3470	0.83	35,000	303
39.3	0.90	3409	0.93	40,000	339



MTOW (kg)	84.771	82.343
OEW (kg)	47.439	46.342
Npax	139	158
EI	0,7448	0,589
DOC (U$/mn)	17,73	18,78

Figure 90: The lowest DOC aircraft with and without restriction

Another important difference between the results is the time that was taken to fulfill the two optimizations. Although the two optimizations used the same computer configuration, the computational cost for the optimization with the flutter constraint was six times higher than that for the optimization without the flutter constraint. This shows that aeroelastic analyzes significantly increase the computational cost, especially considering that the analyzes only consider one case of takeoff weight, while normally this type of analysis considers different weights and Mach numbers.

7 Conclusions

This Chapter intends to discuss the importance of incorporating medium and high-fidelity aeroelastic tools into multi-disciplinary and multi-objective design platforms tailored to handle transport airplanes. In addition, the development of surrogate models for transonic flow representation with acceptable levels of accuracy is also the objective of this Chapter. For a better understanding of these topics alongside the three computational applications contained here, additional Sections are also part of the present Book Chapter, namely an overview of induced drag and new aircraft configurations, as well as the meaning of machine learning and its objectives.

The following topics were discussed in this Book Chapter:

The quest for high aspect-ratio wings has led to new airplane configurations to reduce induced drag. A dual-fuselage configuration was especially analyzed. Some pros and cons related to the adaptation of such configuration for passenger and cargo transport were raised. Structural considerations were also discussed for a strut-braced airliner.

A closely coupled FSI approach was used to investigate the aeroelastic behavior of the KC-135. Detailed structural and aerodynamic characteristics could be captured for this airplane, enabling the verification of main effects that arose from the winglet incorporation, verifying how the flexibility of the structures affects the aerodynamic behavior of the wing and the winglet. Besides the validation of the approach adopted here, corroborated by the excellent agreement with wind tunnel and flight test, an outstanding capacity of prediction of the aeroelastic phenomena is available, both qualitatively and quantitatively. The main utility of modeling these coupled physics is to obtain more accurate metamodels for preliminary studies, calibration of aeroelastic plant models for control purposes, and parametric models to perform multidisciplinary optimization studies. Thus, the objective of utilizing it from the design of morphing winglets is fulfilled.

Full potential flow is a very attractive approach for MDO platforms that are elaborated for aircraft conceptual design. Besides 3D applications, a computer code for the inverse transonic airfoil design was presented and its performance and accuracy are excellent. The design methodology presented at issue proved to be a simple and efficient tool for preliminary wing design, always converging towards unique solutions for each set of reasonable prescribed Cp distribution. After a few design iterations, it is possible to introduce large geometric modifications on the wing airfoil sections, and properly reproduce a suitable pressure distribution.

A surrogate model with ANN to replace a full-potential code in airplane MDO frameworks was presented. A database of approximately 72,000 cases to train and design the ANNs was generated. The size of the database was checked with the learning curves method. It is also important to visualize how the input variables of the database were distributed over the proposed domain, as this had a direct impact on the prediction errors distribution along with the domain of each input variable. Several multi-layer feed-forward ANNs were trained using the scaled Levenberg-Marquardt algorithm. ANN architectures in terms of several neurons were selected based on minimum mean squared error presented by approximately 5000 cases that did not take part in the training/validation/test of the networks. The ANNs dedicated to the three drag components, parasite, induced, and wave, give better predictions when compared with the single-network approach. The parasite drag coefficient presents the toughest patterns for ANN learning. A reduction of 1660 times in computational cost was recorded, with average absolute errors lower than one drag count for all kinds of drag coefficients. According to the results, it is possible to set up ANN to substitute CFD software to reduce the computational cost in a multidisciplinary optimization framework, with acceptable errors for the conceptual design phase.

A study on the influence of aeroelastic analysis on the configuration of transport airplanes obtained by a multi-disciplinary optimization procedure was carried out. For this purpose, two multi-objective optimization tasks were performed, with the main goal of minimizing the direct operational cost and maximizing the efficiency index for a medium-range wide-body aircraft, one without and the other with aeroelastic constraints. An MDO framework was integrated with the NeoCASS package for aeroelastic analysis. Thus, an aeroelastic constraint was implemented in which the maximum operating Mach number of the aircraft was scanned to find a flutter speed and an altitude at which the phenomenon occurs. If this flutter speed and altitude were inside the flight envelope of the aircraft, the individual was considered infeasible. The comparison of the optimization runs performed showed that the inclusion of flutter constraints in the optimization of the aircraft design has limited the wing aspect ratio, leading to poorer aerodynamic performance and, consequently, poorer optimization results. The comparison of aircraft with higher *EI* found in the two optimizations shows a 10% decrease in *EI* and a 5.25% increase in DOC, while for aircraft with lower DOC there was a 20.9% decrease in *EI* and a 5.92% increase in DOC. In the second case, the constraint completely changed the characteristics of the aircraft by reducing the range and increasing the number of passengers. The inclusion of aeroelastic constraints in conceptual design restrains the performance improvement and therefore significantly alters the optimal designs obtained. Moreover, it was found that these trends cannot be generalized for each aircraft project, since the occurrence of the flutter phenomenon is quite characteristic for each aircraft, responding to small changes in wing geometry, engine parameters, or aerodynamic forces. Therefore, the incorporation of a fast and accurate aeroelastic analysis at the beginning of the conceptual design phase is of utmost importance to increase the confidence of the optimizations.

Acknowledgments

Part of the computations used in the present work was carried out with hardware and software purchased with the support of the Brazilian federal agency Financiadora de Estudos e Projetos (FINEP) through the project CAPTAER II.

References

[1] W. F. Ballhaus Jr. and J. L. Steger, "Implicit Approximate Factorization Schemes for the Low-frequency Transonic Equation," NASA, TM X-73082, Washington, D. C., 1975.

[2] M. Niță and D. Scholz, "Estimating the Oswald Factor from Basic Aircraft Geometrical Parameters," in *German Aerospace Congress 2012*, Berlin, 2012.

[3] R. Bisplinghoff, H. Ashley and R. Halfma, Aeroelasticity, Cambridge, MA: Addison-Wesley, 1955.

[4] E. Dowell, J. Edwards and T. Strgana, "Nonlinear aeroelasticity.," *Journal of Aircraft,* p. 857–874, 2003.

[5] A. J. Eaton, C. Howcroft, E. Coetzee, S. Neild, M. Lowenberg and J. Cooper, "Numerical Continuation of Limit Cycle Oscillations and Bifurcations in High-Aspect-Ratio Wings," *Aerospace,* vol. 5, July 2018.

[6] B. Stanford and P. Beran, "Direct flutter and limit cycle computations of highly flexible wings for efficient analysis and optimization," *Journal of Fluids and Structures,* vol. 36, pp. 111-123, August 2012.

[7] Wikipedia, The Free Encyclopedia, "Boeing 787 Dreamliner," Wikipedia, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Boeing_787_Dreamliner. [Accessed 2018].

[8] Wikipedia, The Free Encyclopedia, "Airbus A350 XWB," Wikipedia, 2018. [Online]. Available: Airbus A350 XWB. [Accessed 2018].

[9] F. George, "Flying The A350: Airbus's Most Technologically Advanced Airliner," Aviation Week, 2015.

[10] Embraer, "Embraer 195 Airport Planning Manual," Embraer, São José dos Campos, 2015.

[11] Embraer, "E-Jets E2 Airport Planning Manual," Embraer, São José dos Campos, 2018.

[12] GE Aviation, "CF34-10E - GE Aviation," October 2018. [Online]. Available: https://www.geaviation.com/bga/engines/cf34-engine. [Accessed October 2018].

[13] Wikipedia, the Free Encyclopedia, "Embraer E-Jet E2 family," Wikipedia, October 2018. [Online]. Available: https://en.wikipedia.org/wiki/Embraer_E-Jet_E2_family. [Accessed October 2018].

[14] Jane's, "Jane's Aero Engines," IHS Markit, December 2017. [Online]. Available: https://janes.ihs.com/.

[15] J. M. Grasmeyer, P. A. Naghshineh-Pour, B. G. Tetrault, R. T. Haftka, R. K. Kapania, W. H. Mason and J. A. Schetz, "Multidisciplinary Design Optimization," Virginia Polytechnic Institute and State University, Blacksburg, 1998.

[16] P. Dees and M. Stowell, "SAE World Aviation Congress & Exposition," in *737-800 Winglet Integration*, 2001.

[17] J. Fregnani, B. Mattos and J. Hernandes, "Multidisciplinary and Multi-Objective Optimization Considering Aircraft Program Cost and Airline Network," *Journal of Air Transportation,* pp. 27-41, January 2021.

[18] MIT Global Airline Industry Program, "Airline Data Project," 2018. [Online]. Available: http://web.mit.edu/airlinedata/www/Aircraft&Related.html. [Accessed 2018].

[19] F. Gern, A. Naghshineh-Pour, E. Sulaeman, R. Kapania and R. Haftka, "Flexible Wing Model for Structural Sizing and Multidisciplinary Design Optimization of a Strut-Braced Wing," in *41st AIAA Structures, Structural Dynamics and Materials Meeting*, Atlanta, 2000.

[20] M. Bhatia, R. K. Kapania, M. v. Hoek and R. T. Haftka, "Structural Design of a Truss Braced Wing: Potential and Challenges," in *Proceedings of the 50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Palm Springs, CA, 2009.

[21] B. Grossman, R. Kapania, W. Mason and J. Schetz, "Multidisciplinary Design Investigation of Truss-braced Wing Aircraft:Phase 4," Virginia Polytechnic Institute, NAG-1-2217, Blacksburg, 2000.

[22] T. H. Megson, Aircraft Structures, 3rd ed., Oxford: Butterworth-Heinemann, 1999.

[23] P. L. Videiro, “Otimização Aeroestrutural na Fase de Projeto de Aeronave,” Instituto Tecnológico de Aeronáutica, Trabalho de Graduação, São José dos Campos, 2012.

[24] E. Torenbeek, Synthesis of Subsonic Airplane Design, Delft: Delft University Press, 1986.

[25] É. Roux, "Modèle de Masse de Voilure,," 2006. [Online]. Available: http://elodieroux.com/ReportFiles/ModelesVoilureVersionPublique.pdf. [Accessed February 2022].

[26] G. Guruswamy and C. Byun, "Fluid-structure Interaction using Navier-Stokes Flow Equations Coupled with Shell Finite Element Structures," in *23rd AIAA Fluid Dynamics, Plasmadynamics, and Lasers Conference*, Orlando, 1993.

[27] V. Harrand, J. Lucker, J. Siegel, V. Parthasarathy, G. Vijayan and P. Dionne, "Application of a multidisciplinary computing environment (MDICE) for loosely coupled fluid–structural analysis," in *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, St. Louis, 1998.

[28] M. Sadeghi, S. Yang, F. Liu and H. Tsai, "Parallel computation of wing flutter with a coupled Navier–Stokes/CSD method," in *41st AIAA Aerospace Sciences Meeting and Exhibit*, Reno, 2003.

[29] R. Bird, W. Stewart and .. Lightfoot, Transport Phenomena, New York: Wiley&Sons, 2007.

[30] H. Versteeg and W. Malalasekra, An introduction to computational fluid dynamics, New Delhi: Pearson India, 2010.

[31] Ansys, Inc., "Ansys Fluent User Guide," Ansys, Inc., 2015.

[32] P. L. Roe, "Approximate Riemann solvers, parameter vectors and difference schemes," *Journal of Computational Physics,* pp. 357-372, 1981.

[33] M. S. Liou and C. J. Steffen Jr., "A new flux splitting scheme," *Journal of Computational Physics,* pp. 23-39, 1993.

[34] P. Hartwitch and . S. Agrawal, "Method for perturbing multiblock patched grids in aeroelastic and design optimization applications," in *13th AIAA Computational Fluid Dynamics Conference*, Snowmass Village, 1997.

[35] M. R. Barber and D. Selegan, "KC-135 Winglet Program Review," NASA CP-2211, Edwards, California, 1982.

[36] B. Almojuela, "The Development of Boeing’s 367-80 or charging Into the Jet Age Armed with Only a Slide Rule and Spline," in *Pacific Northwest AIAA Technical Symposium*, Seattle, 2009.

[37] M. C.-Y. Niu, Airframe Structural Design: Practical Design Information and Data on Aircraft Structures, 2nd ed., Adaso/Adastra Engineering Center, 2011, p. 600.

[38] M. D. Sensmeier and J. A. Samaresh, "A Study of Vehicle Structural Layouts in Post-WWII Aircraft," in *45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics & Materials Conference*, Palm Springs, 2004.

[39] A. T. Loyd, Boeing 707 & AWACS, in detail and scale., Shrewsbury: Airlift Publishing, 1987.

[40] A. L. Bolsunovsky, N. P. Buzoverya, O. V. Karas and S. I. Skomorokhov, "An Experience in Aerodynamic Design of Transport Aircraft," in *28th International Congress of Aeronautical Sciences*, Brisbane, 2012.

[41] E. N. Tinoco, "CFD codes and applications at Boeing," *Sadhana,* pp. 141-163, 1991.

[42] T. L. Holst, "Transonic Flow Computations Using Nonlinear Potential Methods," *Progress in Aerospace Sciences,* pp. 1-61, 2000.

[43] C. Fletcher, Computational Techniques for Fluid Dynamics 2: Specific Techniques for Different Flow Categories, vol. 2, Springer, 2003.

[44] H. Goetz and D. Nixon, "Nonisentropic Potential Formulation for Transonic Flows," *AIAA Journal,* vol. 22, pp. 770-776, June 1984.

[45] B. S. Mattos, "Numericher Entwurf von transonischen Flügeln und Trageflügelprofilen unter Anwendung des AF2-Algorithmus zur Lösung de vollständigeb Potentialgleichung," Institut für Aerodynamik und Gasdynamik der Universität Stuttgart, Sttutgart, 1995.

[46] B. Mohammadi, "Fluid dynamics computation with NSC2KE: a user-guide: release 1.0," INRIA, Rocquencourt, 1994.

[47] B. Mattos and S. Wagner, "Numerical design of transonic wings in curvilinear coordinates," in *12th AIAA Applied Aerodynamics Conference*, Colorado Springs, 1994.

[48] ANSYS, Inc., "Ansys Fluent Theory Guide Release 15," 2013. [Online]. Available: http://www.pmt.usp.br/ACADEMIC/martoran/NotasModelosGrad/ANSYS%20Fluent%20Theory%20Guide%2015.pdf. [Accessed 2017].

[49] B. S. Mattos, J. A. Fregnani and P. C. Magalhães, Conceptual Design of Green Transport Airplanes, Sharjah, UAE: Bentham Books, 2018.

[50] B. S. Mattos, P. J. Komatsu and J. T. Tomita, "Optimal wingtip device design for transport airplane," *Aircraft Engineering and Aerospace Technology,* vol. 90, pp. 743-763, 2018.

[51] Wikipedia, The Free Encyclopedia, "List of Algorithms," Wikipedia, The Free Encyclopedia, February 2022. [Online]. Available: https://en.wikipedia.org/wiki/List_of_algorithms#Statistics. [Accessed February 2022].

[52] D. Kireev, F. Ros, P. Bernard, J. Chrétien and N. Rozhkova, "Non-supervised Neural Networks: A New Classification Tool to Process Large Databases," in *Computer-Assisted Lead Finding and Optimization: Current Tools for Medicinal Chemistry*, Wiley Online Library, Zurich, 1997, pp. 253-264.

[53] K. Frenken and A. Nuvolari, "Entropy Statistics as a Framework to Analyse Technological Evolution," in *Applied Evolutionary Economics and Complex Systems,*, J. F. &. W. Hölzl, Ed., Edward Elgar Publishing, 2004.

[54] R. Lipmann, "An Introduction to Computing with Neural Nets," *IEEE ASSP Magazine,* vol. 4, pp. 4-22, 1987.

[55] W. S. McCulloch and W. Pitts, "A Logical Calculus of the Ideas Immanent in Nervous Activity," *The Bulletin of Mathematical Biophysics,* vol. 5, pp. 115-133, 1043.

[56] F. Rosenblatt, "he Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain," *Psychological Review,* pp. 386-408, 1958.

[57] M. Minsky and P. Seymour, "Perceptrons," MIT Press, Cambridge, 1969.

[58] J. B. Pollack, "No Harm Intended: A Review of the Book entitled Perceptrons: An Introduction to Computational Geometry," *Journal of Mathematical Psychology,* pp. 358-356, 1989.

[59] D. E. Rumelhart, G. E. Hinton and R. J. Willians, "Learning Representations by Back-propagating Errors," *Cognitive Modeling,* vol. 1, 2002.

[60] T. Kohonen, "An Introduction to Neural Computing," *Neural Network,* vol. 1, pp. 3-16, 1988.

[61] Mathworks, "feedforwardnet," Mathworks, 2019. [Online]. Available: https://www.mathworks.com/help/deeplearning/ref/feedforwardnet.html. [Accessed May 2020].

[62] K. Maladkar, "6 Types of Artificial Neural Networks Currently Being Used in Machine Learning," 2018. [Online]. Available: https://analyticsindiamag.com/6-types-of-artificial-neural-networks-currently-being-used-in-todays-technology/. [Accessed May 2019].

[63] Mathworks, Inc., "Convolutional Neural Network," Mahtworks, Inc., 2020. [Online]. Available: https://www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html. [Accessed 2020].

[64] Mathworks, Inc., "Perceptron," Mathworks, Inc., February 2022. [Online]. Available: https://www.mathworks.com/help/deeplearning/ref/perceptron.html. [Accessed February 2022].

[65] P. Potočnik, "Neural Network Course - Laboratory of Synergetics, Faculty of Mechanical Engineering, University of Ljubljana," 2022. [Online]. Available: https://web.fs.uni-lj.si/lasin/en/pedagoski-proces/nevronske-mreze/. [Accessed February 2022].

[66] Mathworks, Inc., "newrb - Radial Basis Function," Mathworks, Inc., February 2022. [Online]. Available: https://www.mathworks.com/help/deeplearning/ref/newrb.html?searchHighlight=newrb&s_tid=srchtitle_newrb_1. [Accessed February 2022].

[67] FlightAware, "Flight JFK to Narita," Flight Aware, May 2020. [Online]. Available: https://pt.flightaware.com/. [Accessed May 2020].

[68] Flight Aware, "Flight GRU-FRA," Flight Aware, May 2020. [Online]. Available: https://pt.flightaware.com/. [Accessed May 2020].

[69] A. R. Cruz, M. C. Murça, B. S. Mattos and J. T. Fregnani, "Identification of Actual Mission Profiles and their Impact on Integrated Aircraft-Network Optimization," in *AIAA Aviation Forum*, Virtual Event, 2021.

[70] Wikipedia, The Free Encyclopedia, "DBSCAN," Wikipedia, The Free Encyclopedia, 2022. [Online]. Available: https://en.wikipedia.org/wiki/DBSCAN. [Accessed 2022].

[71] J. Sun, S. Ellerbroek and J. M. Hoekstra, "Flight Extraction and Phase Identification for Large Automatic Dependent Surveillance–Broadcast Datasets," *Journal of Aerospace Information Systems,* pp. 566-572, October 2017.

[72] L. A. Zadeh, "Fuzzy Sets," *Information and Control,* pp. 338-353, June 1965.

[73] E. Sun and J. Hoekstra, "Modeling aircraft performance parameters with," in *12th USA/Europe Air Traffic Management Research and Development*, Seattle, 2017.

[74] J. Sun, J. Ellerbroek and J. Hoekstra, "Aircraft initial mass estimation using Bayesian inference method," *Transportation Research Part C: Emerging Technologies,* pp. 59-73, May 2018.

[75] O. M. Silva, " Modelo de desempenho bada e estimativa de massa de aeronave com aplicação em tráfego aéreo," Instituto Tecnológico de Aeronáutica, Undergraduate Thesis, São José dos Campos, 2019.

[76] Eurocontrol, "User Manual for the Base of Aircraft Data (BADA) Revision 3.7," March 2009. [Online]. Available: https://www.eurocontrol.int/sites/default/files/library/003_BADA_3_7_User_manual.pdf. [Accessed February 2022].

[77] K. Gurney, "Neural networks for perceptual processing:from simulation tools to theories," *Philosophical Transactions of the Royal Society B,* pp. 339-353, 8 January 2007.

[78] S. J. Russel and P. Norvig, Artificial Intelligence, A Modern Approach, New Jersey: Prentice-Hall, 2010.

[79] K. Hornick, M. Stinchcombe and H. White, "Multilayer Feedforward Networks are Universal Approximators," in *Neural Networks,* Pergamon Press, 1989, pp. 359-366.

[80] Mathworks, "fitnet," Mathworks, 2020. [Online]. Available: https://www.mathworks.com/help/deeplearning/ref/fitnet.html. [Accessed 2019].

[81] T. P. Vogl, J. K. Mangis, A. K. Rigler, W. T. Zink and D. L. Alkon, "Accelerating the convergence of the backpropagation method," *Biological Cybernetics,* vol. 59, p. 257–263, 1988.

[82] D. Marquardt, "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," *SIAM Journal on Applied Mathematics,* p. 431–441, June 1963.

[83] M. T. Hagan and M. Menhaj, "Training feed-forward networks with the Marquardt algorithm," *IEEE Transactions on Neural Networks,* vol. 5, p. 989–993, 1999.

[84] Mathworks, Inc., "trainlm," Mathworks, Inc., 2020. [Online]. Available: https://www.mathworks.com/help/deeplearning/ref/trainlm.html. [Accessed 2020].

[85] P. V. Camarotti, " Estudo de drivers de mercado, metodologia e desenvolvimento de ferramenta semi-automática para elaboração de projeções de mercado de aviação de linha," ITA, Undergraduate Thesis, São José dos Campos, 2014.

[86] IEA, "Oil Information: Overview, IEA," 2020. [Online]. Available: https://www.iea.org/reports/oil-information-overview. [Accessed 2020].

[87] EIA - U. S. Energy Administration Information, "Petroleum & Other Liquids Weekly Stocks," 2020. [Online]. Available: https://www.eia.gov/dnav/pet/hist/LeafHandler.ashx?n=PET&s=WCESTUS1&f=W. [Accessed 2020].

[88] EIA - U.S. Energy Information Administration, "Crude Oil Production, OPEC Total, Annual," 2020. [Online]. Available: https://www.eia.gov/opendata/qb.php?sdid=STEO.COPR_OPEC.A. [Accessed 2020].

[89] EIA - U.S. Energy Information Administration, "Non-OPEC + OPEC non-Crude Production, Annual," 2020. [Online]. Available: https://www.eia.gov/opendata/qb.php?sdid=STEO.PAPR_NONOPEC_I_OPECNC.A. [Accessed 2020].

[90] Macrotrends, "WTI Crude Oil Prices," 2020. [Online]. Available: https://www.macrotrends.net/2516/wti-crude-oil-prices-all-years. [Accessed 2020].

[91] L. Jenkinson, P. Simpkin and D. Rhodes, "Civil Jet Aircraft design - Boeing Aircraft," Butterworth-Heinemann, 2001. [Online]. Available: https://booksite.elsevier.com/9780340741528/appendices/data-a/table-3/table.htm. [Accessed June 2021].

[92] M. Gumes e L. D. Andrade Pereira, "Utilização de Redes Neurais para Estimação de Coeficienets Aerodinâmicos e Derivadas de Estabilidade," Undegraduation thesis, Instituto tecnológico de Aeronáutica,, São José dos Campos, 2019.

[93] B. Singh, "A Medium-Fidelity Method for Rapid Maximum Lift Estimation," Delft University, Delft, 2017.

[94] S. Gudmundsson, "Maximum Lift Coefficient," Elsevier, 2018. [Online]. Available: https://www.sciencedirect.com/topics/engineering/maximum-lift-coefficient. [Accessed September 1029].

[95] Mathworks, Inc., "gamultiobj," Mathworks, Inc, 2018. [Online]. Available: https://www.mathworks.com/help/gads/gamultiobj.html. [Accessed August 2019].

[96] M. Drela, "XFOIL - Subsonic Airfoil Development System," October 2018. [Online]. Available: http://web.mit.edu/drela/Public/web/xfoil/. [Accessed October 2018].

[97] N. Secco and B. Mattos, "Artificial neural networks to predict aerodynamic coefficients of transport airplanes," *Aircraft Engineering and Aerospace Technology,* pp. 211-230, March 2017.

[98] M. Niţă and D. Scholz, "Estimating the Oswald Factor from Basic Aircraft Geometrical Parameters," in *German Aerospace Congress 2012,* Berlin, 2012.

[99] C. Panda and Venkatasubramani, "Aeroelasticity- In General and Flutter Phenomenon," in *Second International Conference on Emerging Trends in Engineering and Technology, ICETET-09,* Nagpur, Maharashtra, 2009.

[100] P. Dees and M. Stowell, "737–800 Winglet Integration," in *World Aviation Congress & Exposition,* Seattle, 2001.

[101] M. Bradley and K. Droney, "Ultra Green Aircraft research: Phase I Final Report," National Aeronautics and Space Administration, NASA CR2011-216847, Huntington Beach, CA, 2011.

[102] W. Zhinqiang and C. Cesnik, "Geometrically Nonlinear Aeroelastic Scaling for Very," in *54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference* , Boston, Massachusetts, 2013.

[103] M. Kehoe, "A Historical Overview of Flight Flutter Testing," NASA, Dryden Flight Research Center, 1985.

[104] B. Baldwin, "The Efficiency Of Wing Technology On Crossover Narrowbody Jets," Aviation Week Network, 21 October 2021. [Online]. Available: https://aviationweek.com/special-topics/crossover-narrowbody-jets/efficiency-wing-technology-crossover-narrowbody-jets. [Accessed October 2021].

[105] M. Opgenoord, M. Drela and K. Willcox, "Influence of Transonic Flutter on the Conceptual Design of Next-generation Transport Aircraft," in *AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Kissimmee, 2018.

[106] E. Jonsson, C. Riso, C. Lupp, C. Cesnik, J. Martins and B. Epureanu, "Flutter and Post-flutter Constraints in Aircraft Design Optimization," *Progress in Aerospace Sciences,* vol. 109, May 2019.

[107] M. Berci and F. Torrigiani, "Multifidelity Sensitivity Study of Subsonic Wing Flutter for Hybrid Approaches in Aircraft Multidisciplinary Design and Optimisation," *Aerospace,* vol. 7, November 2020.

[108] L. Cavagna, S. Ricci and L. Travaglini, "Aeroelastic Analysis and Optimization at Conceptual Design Level Using NeoCASS Suite," in *52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, Denver, 2011.

[109] J. A. Fregnani, B. Mattos and J. A. Hernandes, "Multidisciplinary and Multi-Objective Optimization Considering Aircraft Program Cost and Airline Network," *Journal of Air Transportation,* pp. 21-41, January 2021.

[110] V. Loureiro and B. S. Mattos, "Powerplant as Design Variable for Multi-disciplinary Design Optimization of Transport Airplane," in *27th AIA Applied Aerodynamics Conference*, San Antonio, 2009.

[111] D. A. Muraro and B. S. Mattos, "Estimation of Noise of Installed Turbofan Engines," in *3th DCTA-DLR Workshop on Data Analysis and Flight Control*, São José dos Campos, 2009.

[112] A. Bosquê, B. Mattos and R. Girardi, "Entropy Statistics Applied to Airplane Evolution," in *11th Brazilian Congress of Thermal Sciences and Engineering -- ENCIT 2006*, Curitiba, 2006.

[113] A. Rios, J. Fregnani, B. Mattos and M. Condé, "Identification of the Actual Mission Profiles and Their Impact on the Integrated Aircraft and Airline Network Optimization," 2021.

[114] MathWorks, "Genetic Algorithm," MathWorks, December 2017. [Online]. Available: https://www.mathworks.com/discovery/genetic-algorithm.html. [Accessed December 2017].

[115] R. C. Quintero, "TECHNO-ECONOMIC AND ENVIRONMENTAL Techno-Economic and Environment Risk Assessment of Innovative Propulsion Systems for Short-Range Civil Aircraft," Cranfield, 2009.

[116] J. A. Fregnani, B. S. Mattos and J. A. Hernandes, "Airline Network-Airplane Integrated Optimization Considering Manufacturer's Program Cost," in *ICAS 2020*, Shangai, 2021.

[117] G. W. van Bondergraven, "Commercial Aircraft DOC Methods," in *AIAA/AHS/ASEE Aircraft Design Systems and Operations Conference*, Dayton, 1990.

[118] Air Transport Association of America, "Standard Method of Estimating Comparative Operating Direct Costs of Turbine Powered Transport Airplanes," Air Transport Association of America, Washington, D. C., 1967.

[119] "Jet Fuel Price Monitor," IATA, May 2021. [Online]. Available: https://www.iata.org/en/publications/economics/fuel-monitor/. [Accessed May 2021].

[120] P. Montarnal, "Presto-Cabin A Preliminary Sizing TRool for Passenger Aircraft Cabins," Hamburg University of Applied Sciences, Hamburg, 2010.

[121] Wikipedia, The Free Encyclopedia, "Rolls-Royce RB211," Wikipedia, The Free Encyclopedia, 2021. [Online]. Available: https://en.wikipedia.org/wiki/Rolls-Royce_RB211. [Accessed May 2021].

[122] European Union Aviation Safety Agency, "Type-Certificate Data Sheet for Rolls&Royce for RB211-535 Series Engine," EASA, Cologne, 2020.

[123] Boeing Commercial Airplanes, "Boeing Resources," 2021. [Online]. Available: https://www.boeing.com/resources/boeingdotcom/company/about_bca/startup/pdf/historical/757_passenger.pdf. [Accessed May 2021].

[124] Boeing Commercial Airplanes, "Boeing 757 Airplane Characteristics for Airport Planning," Boeing, Seattle, 2002.

[125] Boeing Commercial, "777-200/300 Airplane Characteristics for Airport Planning," Boeing Co., Seattle, 1998.

[126] J. G. Barmby, H. J. Cunningham and I. E. Garrick, "Study of Effects cited on," NACA, Washington, 1951.

Chapter 6: On a Bio-Inspired Method for Topology Optimization via Map L-Systems and Fractone Modeling

Chapter details

Chapter DOI:

https://doi.org/10.4322/978-65-86503-71-5.c06

Chapter suggested citation / reference style:

Kobayashi, Marcelo H. (2022). “On a Bio-Inspired Method for Topology Optimization via Map L-Systems and Fractone Modeling”. *In* Jorge, Ariosto B., et al. (Eds.) *Model-Based and Signal-Based Inverse Methods*, Vol. I, UnB, Brasilia, DF, Brazil, pp. 237–268. Book series in Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity.

P.S.: DOI may be included at the end of citation, for completeness.

Book details

Book: Model-based and Signal-Based Inverse Methods

Edited by: Jorge, Ariosto B., Anflor, Carla T. M., Gomes, Guilherme F., & Carneiro, Sergio H. S.

Volume I of Book Series in:

Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity

Published by: UnB City: Brasilia, DF, Brazil Year: 2022

DOI: https://doi.org/10.4322/978-65-86503-71-5

On a Bio-Inspired Method for Topology Optimization via Map L-Systems and Fractone Modeling

Marcelo H. Kobayashi[1]

[1]University of Hawaii at Manoa, Department of Mechanical Engineering, 2540 Dole Street–Holmes Hall 302, Honolulu, HI 96822. Email: marcelok@hawaii.edu

Abstract

Nature has long been a source of inspiration to both engineers and designers alike. This chapter describes a bio-inspired, topology optimization method for engineering design optimization. The method is based on evolutionary developmental processes in biology and employs a cellular division model to develop topologies. Concretely, a Map L-System, a graph based, parallel rewriting method, is used to model the cellular division process that generates the topology of the design, and a genetic algorithm is then used to evolve a population of designs. In the developmental process of the structure, fractones, elements that play a fundamental role in the regulation of cellular divisions, are modeled and incorporated as an additional control parameter for the formation and optimization of topologies. The performance of the resulting method is illustrated with an aeroelastic flapping membrane wing optimization problem in which the supporting structure is optimized for power, lift, and thrust requirements.

1 Introduction

Topology optimization seeks the economical distribution of material within a design domain, while satisfying a set of constraints. There exists myriad computational methods for topology optimization, each employing a variety of techniques and with unique advantages as well as limitations—see the monographs Rozvany [1994], Allaire [2002], Bendsøe and Sigmund [2003], Osher and Fedkiw [2003] and review papers Gain and Paulino [2013], van Dijk et al. [2013], Deaton and Grandhi [2014], Wang et al. [2021] for a detailed description and study of the existing methods.

The most popular method for topology optimization, namely, the solid isotropic material with penalization (SIMP), involves a "pixelation" of the design domain. The color of each pixel is determined by a density function, ρ, that can vary from zero (void) to one (solid). The addition of non-physical "gray" material ($0 < \rho < 1$) is penalized by raising the density function to a power $p \in \mathbb{N}, p > 1$, and defining the stiffness tensor at a point as the product $\rho^p E_0$, where E_0 is the elasticity tensor of the structural material. Since ρ^p, the addition of gray material is made less effective as p increases, thus reducing the regions with gray material in the domain. Though simple to implement and intuitive in its formulation, the designs obtained with SIMP require post-processing to interpret the optimized layout, since inevitably they contain regions of artificial, gray material. In addition, filtering techniques and high resolution are often required to achieve meaningful results from these methods—higher resolution in turn greatly increases the computational demand to explore the design domain. To alleviate these demands, a genetic algorithm (GA) may be employed. In this approach complex problems may converge toward optimal designs without exploring every possibility, by directly translating the discretized domain into a genome and simulating a natural selection process to evolve the topologies. In addition, given the independence of the individuals in the population, perfectly parallelizable algorithms can be easily devised to speed up the computation of the population of designs in each generation.

The biologically inspired method described in this chapter takes advantage of the evolutionary algorithm, but it generates topologies without a discretization of the domain, thus avoiding the mesh dependency of the results that arise from the pixelation resolution. In place of a pixelated approach, the map L-system follows the sequence of rules to generate a vast pool of maps, which can be translated into engineering structures.

Previous studies have proven this biologically inspired method capable of successfully performing complex optimizations—see, for instance, Kobayashi et al. [2009], Stanford et al. [2012], Stanford et al. [2012], Kolonay and Kobayashi [2015]. Here we explore a mechanism of cellular division in neurosciences to enhance the search effectiveness and efficiency in discovering and refining of optimized solutions to engineering problems.

2 Map L-Systems

Introduced in 1968 by biologist Aristid Lindenmayer, the Lindenmayer system (or L-system) is a method of rewriting a series of character/grammar strings in a parallel fashion. The versatility of the L-system to represent the parameters of any starting element with a character string and evolve the structure by implementation of the governing production rules has proven the system to be useful for a number of applications. Some applications include the ability to produce interesting fractal geometries, model the branching growth of plant structures, and simulate cellular divisions—see figure 1. The map L-system proposed by Nakamura, Lindenmayer, and Aizawa (Prusinkiewicz and Lindenmayer [2004] Nakamura et al. [1986]) is an extension of L-systems for graphs that are maps, and was originally develop to model single layer cellular divisions. This class of methods forms the basis for modeling the topologies of structures in this work.

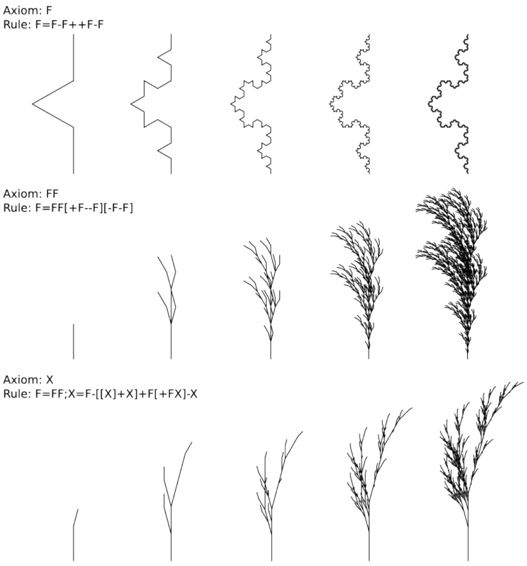


Figure 1: Sample images generated by L-systems: fractal pattern (top), development of plant structures (mid and bottom)

Maps are planar graphs defined by a finite set of regions in which each region is enclosed by a string of edges which meet at vertices. Every edge has one or two associated vertices, all edges are a part of a region's boundary, and all edges are connected (such that there are no 'islands' within the domain). A map may be representative of a single cellular layer where the edges are the cell wall, the enclosed regions are the intracellular space within the cells and the extracellular spaces lay within the walls. Simultaneous cell divisions are modeled with a binary propagating map 0L system with markers (or mBPM0L-system). The 0L system is the context-free Lindenmayer parallel rewriting system in which there are no interactions between cells. The map L-system is binary and propagating because cells are always divided into two daughter cells and in this model cells are never destroyed or joined. The markers play the functional role of flagging the boundary edges at potential vertices where the cell may divide and new edges may form (see Prusinkiewicz and Lindenmayer [2004]).

Informally, a two dimensional map L-system is first initiated with an axiom that defines the edges of the domain. Once initiated, the closed map undergoes a series of subdivisions by the addition of straight internal edges. The creation and location of these edges are governed by a predefined set of production rules. The algorithm begins each iteration with a discretization of the existing edges of the map and a placement of markers at selected nodes. The discretization patterns of the edges and the location and orientation of markers are based on the production rules. Once all edges have been divided and markers placed, the new edges are created by matching the markers. When two markers are present along the boundary of a common region, have the same label, and appropriate orientations, they will form a new wall between the two markers.

Mathematically, the system is defined by an alphabet, Σ, which is a finite set of characters (letters and symbols) and may be represented as $\Sigma = \{A, B, C, ..., [,], +, -\}$. From this alphabet, characters are selected to create an axiom, Ω, which is the string that initiates the rewriting process. For example, an axiom would be $\Omega = ABAB$. The third and final item required is the finite set of production rules or rewriting rules, P. The production rules are also limited to the characters of the alphabet, Σ, and must take the form $\alpha \to \chi$, where α is a single character of the alphabet which serves as the predecessor and χ is the word (or string) called the successor. Note that if there are multiple rules with identical predecessors and differing successors, a probability may be assigned to each rule to determine the frequency at which each are utilized. As an illustrative example, we consider the following two production rules:

$$\begin{aligned} A &\to B[-A]x[+A]B \\ B &\to A \end{aligned}$$

To this point, the example is nearly a simple D0L-system (deterministic because the predecessors are non-repeating and context free) which is not specific to a map generation. The example put forth, however, may be applied to a mapping problem with the addition of a few rules.

In a mapping scenario, each letter of the axiom represents an edge of the initial map. Therefore the length of the axiom must be equal to the number of edges in the initial structure. It is also customary to include the special characters [,], +, and - when using a map L-system. When included in the production rules, the matching brackets, [and], indicate the inclusion of a marker which is labeled by the enclosed character. The orientation (+ or -) preceding the label in the closed brackets indicates the directionality of the marker, in this case the + symbol correlates to a counterclockwise placement of the edge. The remaining characters of the successor, which are not enclosed by brackets, indicate the number of segments the preceding edge is divided into. The new edges will be labeled as prescribed by the rules and nodes are placed between the edges. For example the first edge, A, which is the lower boundary of the map in figure 2, is discretized into three equal segments. The new edge segments are labeled, B, x, and B respectively. Between the first and second segments a marker oriented downward is placed and an upward oriented marker is placed between the second and third segments. Since the first marker is oriented outward from the map it is discarded while the second marker is eventually paired with the other inward facing marker on the upper boundary.

After all the pertaining edges are subdivided and markers are placed, the markers are checked for any possible pairings. As mentioned earlier, a pair of markers are matched if they belong to the same cell (are on the boundary of the same region), are not located on the same edge, if their labels are the same, and if they are oriented toward each other. There may be more than one potential set of marker pairs in a given cell, but the first pair to be found determines the location of the cell division. After cells are scanned for matching markers and the locations of the cellular divisions are determined, the remaining markers are discarded. The resulting maps generated in this example, for an initial square map with equal subdivisions of the edges, are shown in figure 2.

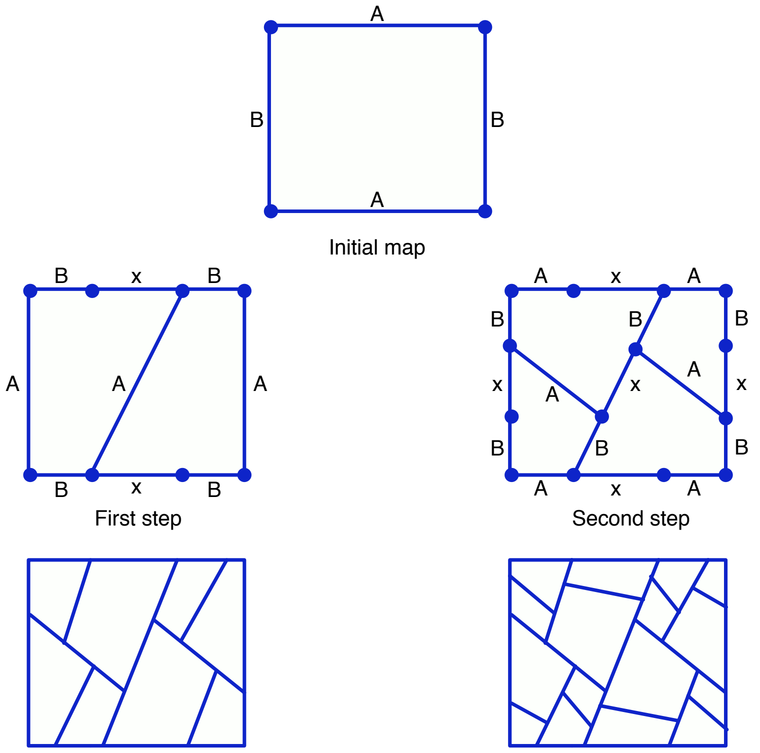
For the purpose of generating and optimizing topological maps in engineering applications it is useful to enforce a few additional constraints. After satisfying the previous conditions to create a cell wall, an eligible marker pair and its respective new edge must also meet certain limits prescribed by the user:

1. Prevention of small angles: the angles between the adjacent edges in the divided cells must be larger than a prescribed lower limit; this prevents the creation of cells with narrow angles.

2. Prevention of small areas: each newly formed cell must have a regional area which is greater than a prescribed percentage of the original map to avoid the formation of excessively small regions.

For practical problems, the structure must also have a finite number of edges. In some cases the cell divisions will cease when all edges of the map are labeled with a terminal token such (such as x in the previous example). Otherwise there exists a maximum number of iterations to be completed. Since this value is highly variable and dependent on the problem at hand, a number of iterations may be prescribed at the start of the map generation program or optimized in the genetic algorithm.

This section has provided a brief introduction to L-systems and an overview of the map L-system as is pertinent to topological map generations. For more details and information on Lindenmayer systems, the reader is referred to Prusinkiewicz and Lindenmayer [2004] and



Initial map

First step

Second step

Third step

Fourth step

Figure 2: Example of the mBPMOL-systems process for the first four iterations where $\Omega = ABAB$and P: $A \to B[-A]x[+A]B$ and $B \to A$.

the references there in. The next section will further investigate the biological aspects of cellular division and their associated regulation mechanisms.

3 Fractones

In cellular biology the mitotic phase, or period of active cellular division, is generally a minor component of the overall cell cycle. A majority of the cell's time is rather spent in preparation for a cellular division, where accumulation of mass and nutrients and synthesis of DNA occurs. Cell cycles may also be dormant at times when inactive time gaps are included Becker [2009]. While rapid cellular division may take place in the early stages of life to promote growth of the individual, the main purpose of cellular divisions in adult organisms is for the maintenance of the body. Adult cellular divisions are reduced to the replacement of damaged and aged cells and to meet the overall functional needs of the individual. As a result, the frequency of cellular divisions is highly variable and dependent on the type of cell and the physical state of the body. To accommodate these variable rates of demand, the cells will typically enter periods of arrest or dormancy until an external indicator is presented and initiates the division process.

For this engineering application, our interest lies in the regulation of the cell cycle. Many cells do not replicate very frequently and often branch from the cell cycle into a dormant phase until signaled to resume active division. In these cases the cell only begins to actively divide when initiated by an external source, most often by cell-type specific molecules known as growth factors. Our focus is in better understanding these control and regulation processes which govern the cell cycle and initiate cellular divisions. Recent findings in the area of neurosciences have introduced new concepts for better understanding such regulatory processes. The results of these studies, which will be reviewed in this section, are the inspiration for the model presented in this work.

In adult neurogenesis, neurons are generated from neural stem and progenitor cells (NSPC). NSPC are localized in specific area of the brain, most notably the subependymal layer of the left ventricle Kerever et al. [2007] Mercier et al. [2003]. Neural stem cell differentiation and proliferation in these areas are also known to be governed by the presence of growth factors, but the details of this regulatory process have remained unknown.

It has been hypothesized that the extracellular matrix in the adjacent regions of the left ventricle wall also plays a contributing role in the initiation of cellular divisions. Close examination of these regions have brought attention to branched structures in the extracellular matrix which come in direct contact with the NSPC. This branched (stem and bulb) structure which binds to the NSPC has assumed the term, fractonesKerever et al. [2007] Mercier et al. [2003]. Fractones are believed to be associated with the material of basement membranes which is the surface tissue containing high concentrations of extracellular molecules (ECM). These extracellular molecules include heparan sulfate proteoglycans (HSPG) which is a known binding cofactor of growth factorsKerever et al. [2007].

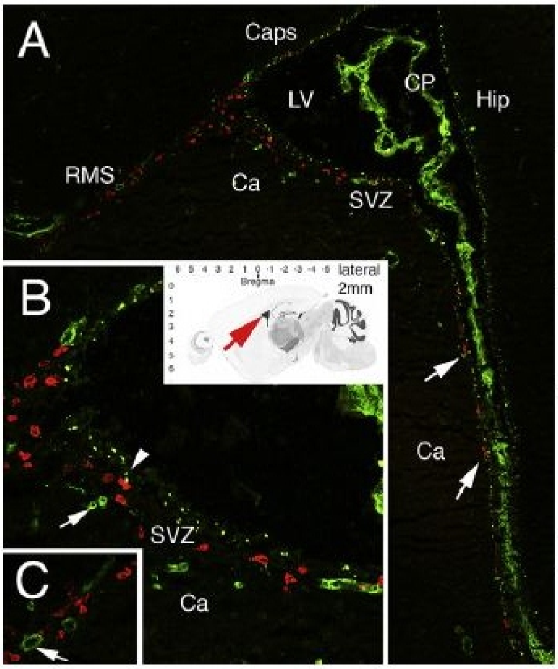


Figure 3: Cell proliferation and Fractones in the lateral ventricle: Fractones (arrows, green labeled N-sulfated HS) and proliferating cells (red labeled BrdU+) near the N-sulfated heparan sulfate fractone structures.

Imaging and statistical analysis methods have produced many new findings and evidence supporting the relationships theorized above. Molecules comprising the basement membranes have been identified in fractones structures, confirming that the two materials are indeed affiliated. A high density of cell proliferation was also observed near fractones, especially those containing N-sulfate HSPGKerever et al. [2007]. Immunolabeling techniques were performed to identify cells recently entering mitosis in several dissection samples. It was found that the majority of cells initiating cellular division were in the areas adjacent to fractones and capillaries, however statistical analysis revealed that cells initiating mitosis were generally in closer proximity to fractonesKerever et al. [2007]. These results indicate that fractones store the fibroblast growth factor 2 (FGF-2) via binding with HSPG and they are the main structures in relaying FGF-2 (growth factors) to the neural stem cells which then undergo mitosis.

The introduction to fractones presented here express the basic concepts that are of interest in this work. For further details on the analysis of fractones refer to Kerever et al. [2007] and Mercier et al. [2003].

4 The Fractone Map L-system

Thus far the map L-system and fractones have been introduced individually. In this next segment, the map L-system will undergo modifications to incorporate the idea of fractones and generate the resulting fractone map L-system.

It was previously demonstrated that fractones play a vital role in initiating division of their associated cells through the capture and delivery of growth factors from the extracellular space to the eligible cell. The model proposed here utilizes a similar and slightly simplified concept. For the purposes of this study, fractones are modeled as small and stationary structures which passively capture and consume simply diffusing growth factors. The fractones initiate mitosis in its neighboring cell once a threshold quantity of the growth factors is accumulated. The rates of growth factor accumulation in these structures are governed by a constant diffusion coefficient and the initial distribution of the growth factor molecules.

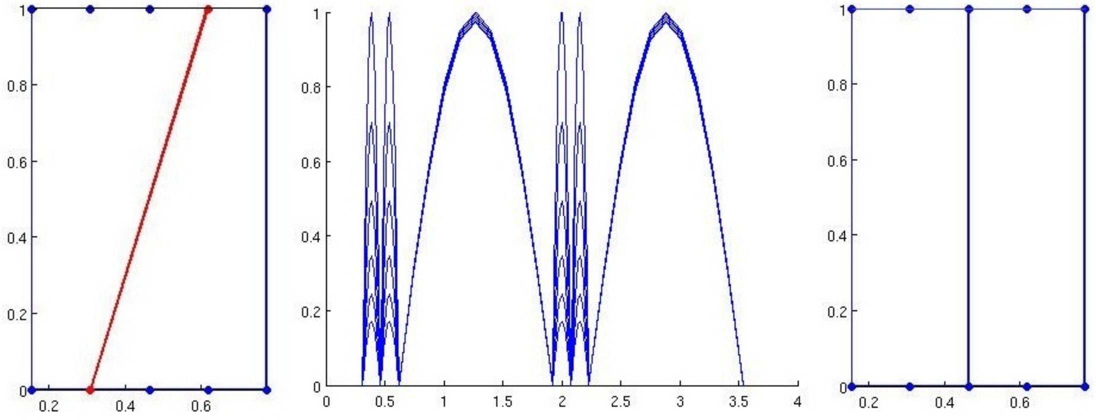
Integration of the fractones into the map L-system is accomplished by the assumption that all of the previously introduced markers in the map L-system represent fractones. All boundary and internal nodes also indicate active fractones (corner nodes are neglected). For simplification it is assumed that all markers (i.e. fractones) are actively consuming growth factors and remain active for the remaining iterations of the map generation. The fractones also have the same affinity for growth factors, and all fractones have the same threshold value. There is only one type of growth factor present in this model and the diffusivity of the molecule is constant throughout the system.

The diffusion of the growth factors along the edges is approximated with a one dimensional, piecewise linear finite element scheme. Diffusion is modeled along each line segment between markers and each segment is discretized with a constant number of uniform nodes. The source term of the diffusion equation is set to zero and the initial distribution of the growth factor is prescribed. All segment ends (fractone locations) are prescribed Dirichlet boundary conditions with a fixed concentration of zero.

The diffusion model is also prescribed a finite number of uniform time steps and after each time step, the amount of growth factors consumed at each node is computed. All eligible markers must accumulate the threshold value of growth factors in addition to satisfying all other requirements to form a pair and initiate a new cell wall formation. One example for such a case where the fractones influence the topology can be seen in figure 4.

After each iteration of the map generation when additional markers are placed and new walls may be formed, the growth factors are redistributed. For each existing segment that undergoes new subdivisions, the total quantity of growth factor is conserved and distributed among the new subsections. The existing growth factors of the edge are accumulated and dispersed with a weighted distribution governed by the length of each new segment. When a new wall is formed, an initial concentration of growth factors must also be assigned. In this case the average growth factor concentration of all existing segments is computed and assigned to the newly formed wall segments. This averaging is performed to remove any bias of wall formations at this member due to extremely low or high concentrations of growth factors relative to the remaining edges.

The growth factor concentrations for these calculations are carried over from the previous iteration in the map generation. If a pair of markers were matched in the preceding iteration, the growth factor concentrations of all edges during the first time step in which both markers reached the threshold value, serve as the initial conditions for the next iteration of the map. If there were no matching pairs in the previous iteration, the growth factor concentrations along all the edges during the last time step are the values carried over to the next iteration of the map generation.



(a) (b) (c)

Figure 4: First iteration of a mapping with and without fractones with axiom: ABAB and production rules:A → B[+A]x[+A]x[+A]B and B→A: a) First subdivision using original mapping system b) Diffusion of growth factors along linearized edges of the map (starting at bottom edge): GF concentration vs. x c) First subdivision using fractone mapping system

While this method is more complex and requires greater computational time to generate maps compared to the original map L-system, it has the potential to improve the overall performance of the optimization with the addition of the diffusion parameters in the genetic algorithm. This scheme may prove to be beneficial when applied to complex optimization problems such as the application presented in the next section.

5 Flapping Wing Optimization

An aeroelastic flapping membrane wing model Stanford et al. [2011] will serve as the test application of the above mentioned optimization methods. This problem analyzes the performance of a forward flight flapping membrane wing for a micro air vehicle. These bio-inspired wings are comprised of thin, flexible membranes reinforced with a rigid beam network, similar in form to the veined wings of insects. The performance of the wing structures are influenced by the venation patters of the supporting members and it is this topology which will undergo optimization. Both the original and fractone inspired map L-systems will be applied independently and the resulting maps will be used to generate the Pareto curves parameterized by the designs power requirement and thrust generation.

Evaluation of the wings under the given flight conditions are to be accomplished as follows. The structural modeling of the wing will be completed using a finite element analysis of the membranes with a triangular mesh (figure 5). The respective governing equation is applied to each element of the membrane:

$$N_x \cdot w,_{xx} + 2N_x y \cdot w,_{xy} + N_y \cdot w,_{yy} + f_z(x,y,t) = \rho \cdot w,_{tt} \tag{1}$$

where N is the pre-stressed resultants, w and f are the out-of-plane displacement and applied force per area, and ρ is the membrane density per length. The Euler-Bernoulli equation is used to analyze the beam members (battens, leading edge, tip, and root) Stanford et al. [2011].

The governing equations are converted into the usual finite element matrix form:

$$M \cdot u'' + C \cdot u' + K \cdot u = F \tag{2}$$

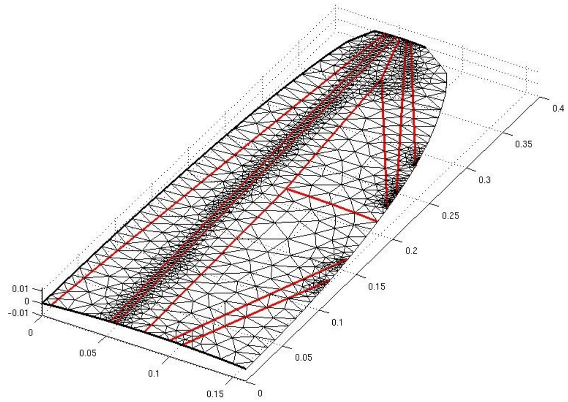


Figure 5: Finite element mesh of a sample wing design: triangular mesh of membrane, battens (red)

where M is the mass matrix, C is the damping matrix, K is the stiffness matrix, F is the accumulated load vector, and u is the total deformation of the structure Stanford et al. [2011] Eric B. Becker and Oden [1981]. Here the solution, u, is also approximated with a linear combination of modes:

$$u = \Phi \cdot \eta \tag{3}$$

where Φ is the modal matrix of natural vibrations and η is the modal amplitudes.

Prior to evaluating the performance of the wing designs, a few parameters must be defined. The flight kinematics are characterized by two angles of the wing, the first is the static angle of attack with respect to the external flow, α, and the second value, β, prescribes the range of the sinusoidal flapping. A constant velocity is also defined for the external air flow and a body attached coordinate system is utilized for the remaining computations.

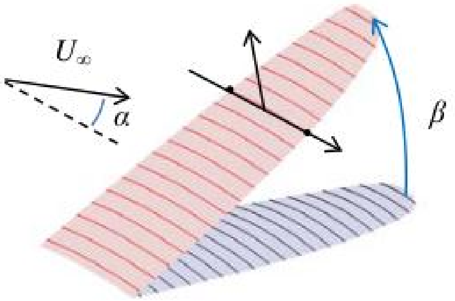


Figure 6: Characteristic angles and span-wise stations of the flapping wing with an attached coordinate system

The aerodynamic loads as opposed to the structural analysis are evaluated with a number of span-wise cross-sections of the wing. The pressure across the wing is determined by applying the no-penetration condition at each span-wise station:

$$\overline{\nu} + \lambda = u_o \cdot \frac{\partial h}{\partial x'} + \frac{\partial h}{\partial t} + \nu_o + \nu_1 \cdot x'/b \tag{4}$$

where h is the shape of the wing, u_o is the horizontal velocity, the last three terms are the vertical velocity (where b is the local semi-chord), and $\overline{\nu}$ and λ are the induced flow from the bound and trailing circulations Stanford et al. [2011]. The terms h, $\overline{\nu}$ and λ are transformed using the Glauert space $\varphi = acos(x'/b)$ for computations henceforth and the resulting integrations are computed using a defined set of Gaussian integration points.

The loads acting over each span section are computed using:

$$F_{y'} = \int_{-b}^{b} \Delta P \cdot dx' + F_{y'}^{\nu} \tag{5}$$

$$F_{x'} = \int_{-b}^{b} \Delta P \cdot \frac{\partial h}{\partial x'} \cdot dx' - 2\pi \cdot b \cdot \rho_\infty \cdot (nu_o + h_o - \lambda_o + u_o \cdot \Sigma n \cdot \frac{h_n}{b})^2 + F_{x'}^{\nu} \tag{6}$$

and the respective viscous terms are computed as follows ($F_{y'}^{v}$ is determined similarly):

$$F_{x'}^{v} = b \cdot \rho_i nfty \cdot U_\infty^2 \cdot (C_{D0} \cdot cos^2\alpha_s + C_{D\pi/2} \cdot sin^2\alpha_s) \cdot u_o/\sqrt{u_o^2 + v_o^2} \tag{7}$$

$$\alpha_s = atan(\frac{h(-b) - h(b)}{2b}) + atan(\frac{v_o}{u_o}) \tag{8}$$

where α_s is the local angle of attack, and C_{D0} and $C_D\pi/2$ are the drag coefficients at angles 0 and $\pi/2$ respectively.

A coupling of the two previous models (structural and airload) is employed to solve for the wing response at each temporal state. The loads are solved for at each cross-sectional segment and interpolated into the structural finite element mesh. The value of the pressure is evaluated at the center of each finite element and considered to be constant over the entire element. The deformation of the wing is determined and the wing shape is updated.

When the air vehicle is subjected to time-periodic flight conditions, the above solution may also be assumed to be time-periodic upon degradation of any transient terms. Each complete flapping cycle may therefore be discretized in time and the set of time-monolithic solutions are approximated using a finite element method.

Further details and information on evaluation of this model may be found in Stanford et al. [2011] and the references therein.

5.1 Genetic Algorithm

Once the topological maps have been generated by the previously discussed scheme, there is then the need for a system to evaluate, analyze and optimize the maps to produce useful results. Here the topologies are optimized using a genetic algorithm (GA). Just as biological evolution continues to progress by the process of natural selection, the nature of this method is to mimic the evolutionary process by preserving the most fit individuals. Genetic algorithms also allow for mutations and hybridization of individuals to produce offspring, however the advantage here is that computational process is greatly accelerated compared to the conventional evolution process.

The genetic algorithm generally begins with strings of numbers (equal in length) which are analogous to genomes of a common species. Each numerical string represents one individual and the number of strings represents the population size, which is constant through each generation. The genomes are representative of the structures which are to be optimized; in this case each genome may be translated into a topological map by the (fractone) map L-system. The individual genes or elements of the string are used to generate the axiom and production rules associated with the map L-system. The first set of numbers in the genome is translated into the axiom, generating a character label and directionality for each initial edge of the map. The majority of the genes used occur in the second extraction from the genome. This set generates the production rules which again prescribe a number of character strings and associated orientations for the markers. In the case of the fractone map L-system, an additional three genes are placed at the tail of the genome. These last three genes contain the values for the threshold, diffusivity, and initial concentration of the growth factors in the fractone model. The only constraint placed on these values is that the threshold value must be less than the initial concentration.

Once the genomes are translated into their associated maps, they are evaluated by a function unique to the problem at hand and each member of the population is ranked based on their "fitness". The most fit individuals are retained to repopulate the next generation of individuals by a combination of random mutation and cross-over of two parent genomes. The probabilities of crossover and mutation shall be cautiously selected by the user to ensure that the desirable characteristics are retained while allowing enough variation to avoid convergence at local optima. Once the offspring are generated, they are also evaluated for fitness and pooled together with the parent genomes. The combined population is ranked and the best genomes are selected to begin the next iteration. The process is completed when a prescribed number of iterations are achieved.

While a single objective optimization problem would be a straightforward example for determining "fitness" in a genetic algorithm, it is often desired that a multiobjective optimization be performed. In this case there is a defined constraint:

$$g(x) \leq 0 \tag{9}$$

and a vector of objective functions to be minimized:

$$\{f_1(x), f_2(x), \cdots, f_n(x)\} \tag{10}$$

This problem is solved using a non-domination ranking system in which designs are preferred if they perform superior to other designs in one or more of the objective functions. Generally there is no single optimum design, and rather a Pareto front is formed. The points along the Pareto optimum are characterized such that one objective function cannot be improved without compensation in another target function. A niching scheme based on the proximity of points is also employed and promotes a greater spread of the Pareto front. The topological designs contained in or closest to the Pareto front are ranked higher and favored in the selection of parents for the next generation.

The previous section presented the methodologies for performing a topology optimization using a biologically inspired fractone model. In this section the results are presented for an optimization of the venation pattern for a flapping membrane wing of a micro air vehicle. Both the original map L-system topology generation and the fractone map L-system methods are employed and the resulting performances are compared. Some optimal wing designs and their performances are introduced here along with some analysis of the Pareto fronts generated during the optimizations.

6 Wing Design

The wing structure was composed of a thin latex membrane and a carbon fiber lattice structure. The membrane was characterized with an isotropic pre-stress condition and the carbon fiber beams were prescribed a rectangular cross-section. These details, along with the remaining material and geometric properties of the wing are presented in Table 1.

The wing shape was defined with a root chord of 0.16 meters, a wing length of 0.4 meters, and a tip chord of 0.04 meters. A parabolic camber was prescribed with a maximum value

property	membrane	battens	leading edge
elastic modulus, E	2 MPa	300 GPa	300 GPa
Poisson's ratio, ν	0.5	0.34	0.34
density, ρ	1200 kg/m^3	1600 kg/m^3	1600 kg/m^3
thickness	0.1 mm	0.8 mm	2 mm
width	-	3 mm	5 mm
pre-stress, N_x, N_y	10 N/m	-	-

Table 1: Material and geometric properties of the membrane wing

of 2% the local chord length. There was no twist of the original wing, the dihedral angle was zero, and the angle of attack (α) was set to 4°.

The kinematics of flight were parameterized with a flapping frequency (ω) of 40 rad/s and a 30° amplitude of sinusoidal flapping (β). The flow velocity (U_∞) was 10 m/s and the density of air (ρ_∞) was 1.225 kg/m^3. The drag coefficients, C_{D0} and $C_{D\pi/2}$, were 0.05 and 2 respectively.

A number of parameters were also prescribed for the evaluation process which computed the fitness of each wing design. The structural analysis of the wings was performed with a finite element method using 20 modes. Airload analysis was performed with 20 span wise wing stations with 20 Gauss points and 6 inflow states. The flight dynamics were computed using a total of 100 timesteps per flapping cycle and 5 full cycles.

Three coefficients were defined for the multidisciplinary optimization of the wings. The lift generation, thrust generation, and power requirements were evaluated as follows:

$$\begin{aligned} C_L &= -F_x/(0.5 \cdot \rho_\infty \cdot U_\infty^2 \cdot S) \\ C_P &= P/(0.5 \cdot \rho_\infty \cdot U_\infty^3 \cdot S) \\ C_T &= -F_y/(0.5 \cdot \rho_\infty \cdot U_\infty^2 \cdot S) \end{aligned}$$

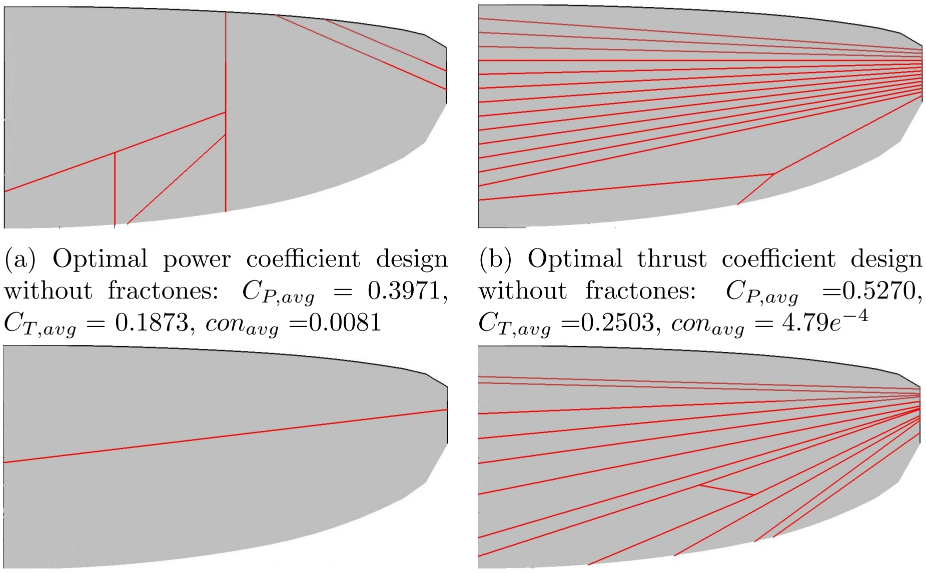
where F represents the respective forces, P is the accumulated power, and S is the area of the wing. For this optimization all coefficients were averaged over the flapping cycle and the lift coefficient was selected for the constraint function. The critical C_L value had a magnitude of 0.4892 (corresponding to a recorded average lift required for maximum thrust) and the constraint function was defined as:

$$g(x) = C_L - 0.4892 \tag{11}$$

The thrust and power coefficients were retained as the two objective functions (f_1(x), f_2(x)) to generate the Pareto fronts. A population size of 200 individuals and 200 generations were used in the genetic algorithm. The crossover probability of the genomes was 0.8 and the probability of mutation was 0.1.

The map L-system was constrained to a 20 letter alphabet and a maximum of 8 letters per production rule. A total of 4 iterations in the map generation were allowed. Growth factor diffusion was assessed in the fractone map L-system with a fixed number and length of timesteps. A total of 5 timesteps were used in each iteration of the map generation and each step was uniform with a length of 0.1.

A large variety of skeletal designs were generated from the optimization schemes parameterized here. A glimpse at the diverse pool of the resulting topologies can be seen in figure 7. Here the most optimal designs for the power and thrust objective functions are displayed. Each set of structures were randomly selected from a fractone and non-fractone based optimization. The drastic differences in the power and thrust optimums are apparent as well as the similarities of results generated within each set. Both skeletal structures for optimal power coefficients contain very few members. The optimal design produced using the fractone system is especially sparse, while the design generated without fractones includes some reinforcement of the trailing edge. Similarly both designs for optimal thrust closely resemble each other with higher densities of the lattice structures and the majority of the members oriented span-wise.


(a) Optimal power coefficient design without fractones: $C_{P,avg} = 0.3971$, $C_{T,avg} = 0.1873$, con_{avg} =0.0081

(b) Optimal thrust coefficient design without fractones: $C_{P,avg}$ =0.5270, $C_{T,avg}$ =0.2503, $con_{avg} = 4.79e^{-4}$

(c) Optimal power coefficient design using fractones: $C_{P,avg}$ =0.4085, $C_{T,avg}$ =0.1975, con_{avg} =0.0105

(d) Optimal thrust coefficient design using fractones: $C_{P,avg}$ =0.5142, $C_{T,avg}$ =0.2380, con_{avg} =0.0046

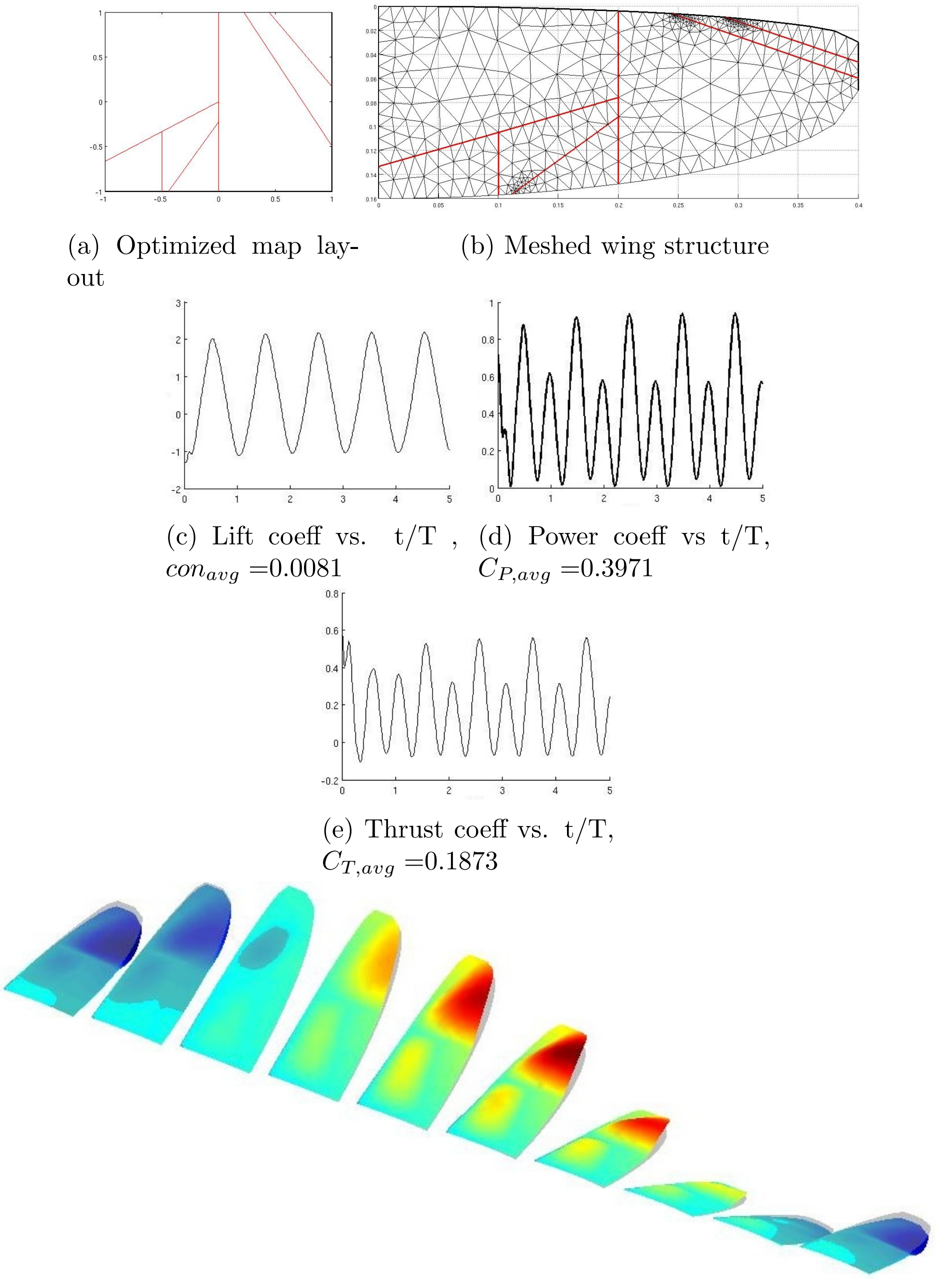
Figure 7: Optimal power and thrust coefficient designs

A selection of the resulting wings and their performances are presented in greater detail in figures 8 to 21. This collection includes a minimum power coefficient wing design resulting from the original map and fractone mapping systems in figures 8 and 15 respectively. An optimal thrust coefficient layout is also presented for the original and fractone mapping cases in figures 9 and 16 respectively. A total of five intermediate structures were also selected at random from each of the mapping scenarios and are displayed in figures 10 thru 14 and 17 through 21.

These results illustrate the structural dependence of the performances and correlate to the results found in Stanford et al. [2011]. As seen in the cases of the minimum power requirement designs, the lack of reinforcement at the trailing edge allowed for steeper gradients of displacement in the membranes. This allowed the wing a greater contour to the flow conditions and minimized the total power demand by reducing the aerodynamic resistance during both the upstroke and down-stroke of the flap. The compensation of this design however, was a reduction in peak thrust and lift performance. These designs (especially the fractone generated design with only one batten) had inferior lift performance to all other designs; while the lack of reinforcement allowed the wing to contour to more to flow, it reduced the occurrence of increased cambering and inflation.

In contrast to the power optimal designs, the stiffened thrust optimal designs exhibited reduced gradients with respect to the structures out-of-plane deformation. In both thrust optimal designs, the out-of-plane deformations were due to span-wise bending of the structures. During the down-stroke of these designs, when the lift was increased, it is seen that the angle of attack was also slightly reduced and this favored the generation of thrust.

Interesting results are also observed in the inspection of the random designs. In many of these scenarios, inflation occurred where venation was sparse and large deformation gradients were formed. Presumably these occurrences favored a lift optimal design. An interesting design is also seen in figure 19, in which chord wise reinforcements were incorporated. This design resulted in intermediate performance in all three design variables between the thrust and power optimal designs.



(a) Optimized map layout

(b) Meshed wing structure

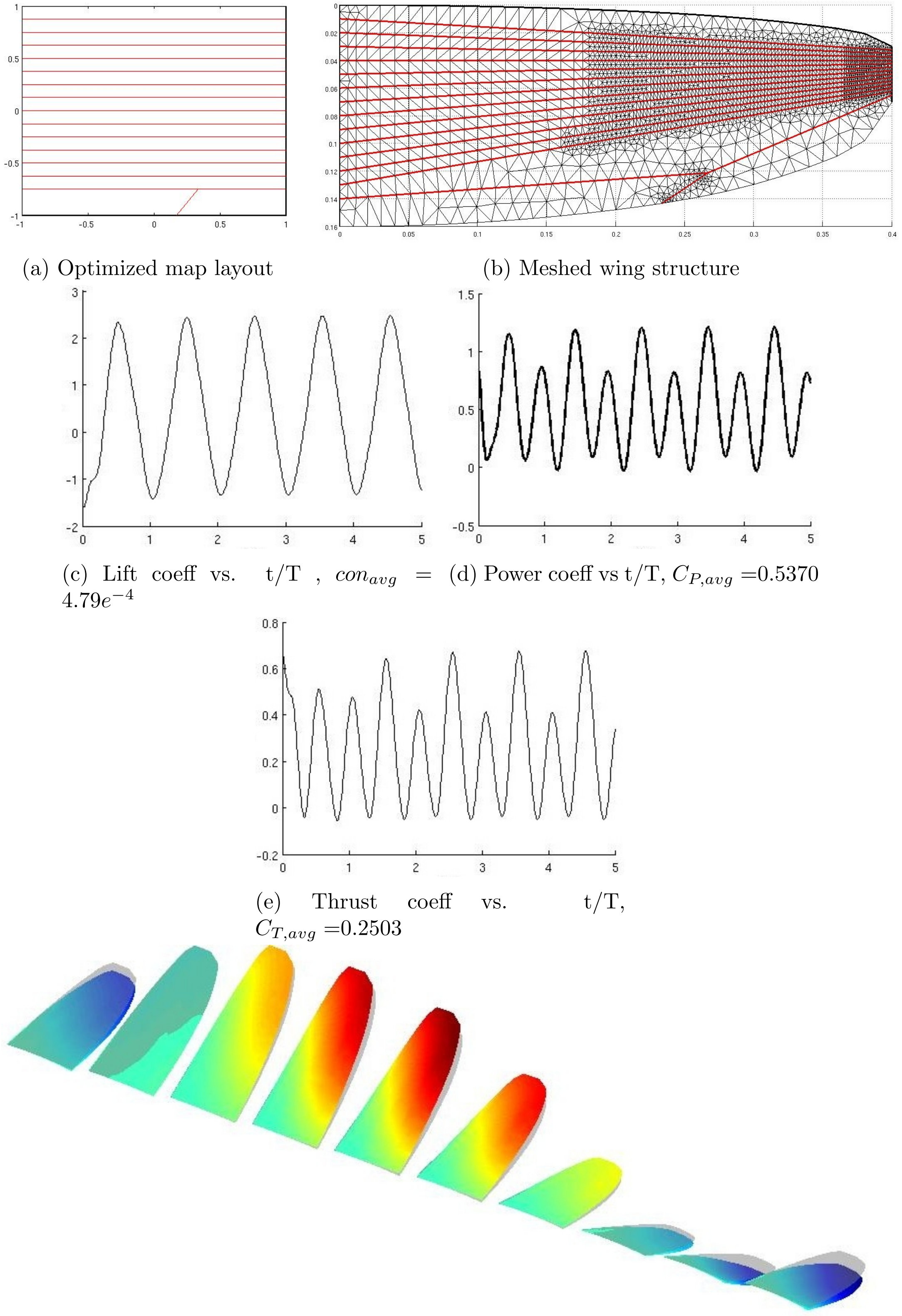
(c) Lift coeff vs. t/T , con_{avg} =0.0081

(d) Power coeff vs t/T, $C_{P,avg}$ =0.3971

(e) Thrust coeff vs. t/T, $C_{T,avg}$ =0.1873

(f) Deformation of wing over a flapping cycle

Figure 8: Optimal power coefficient design: Original mapping system Run 1



(a) Optimized map layout

(b) Meshed wing structure

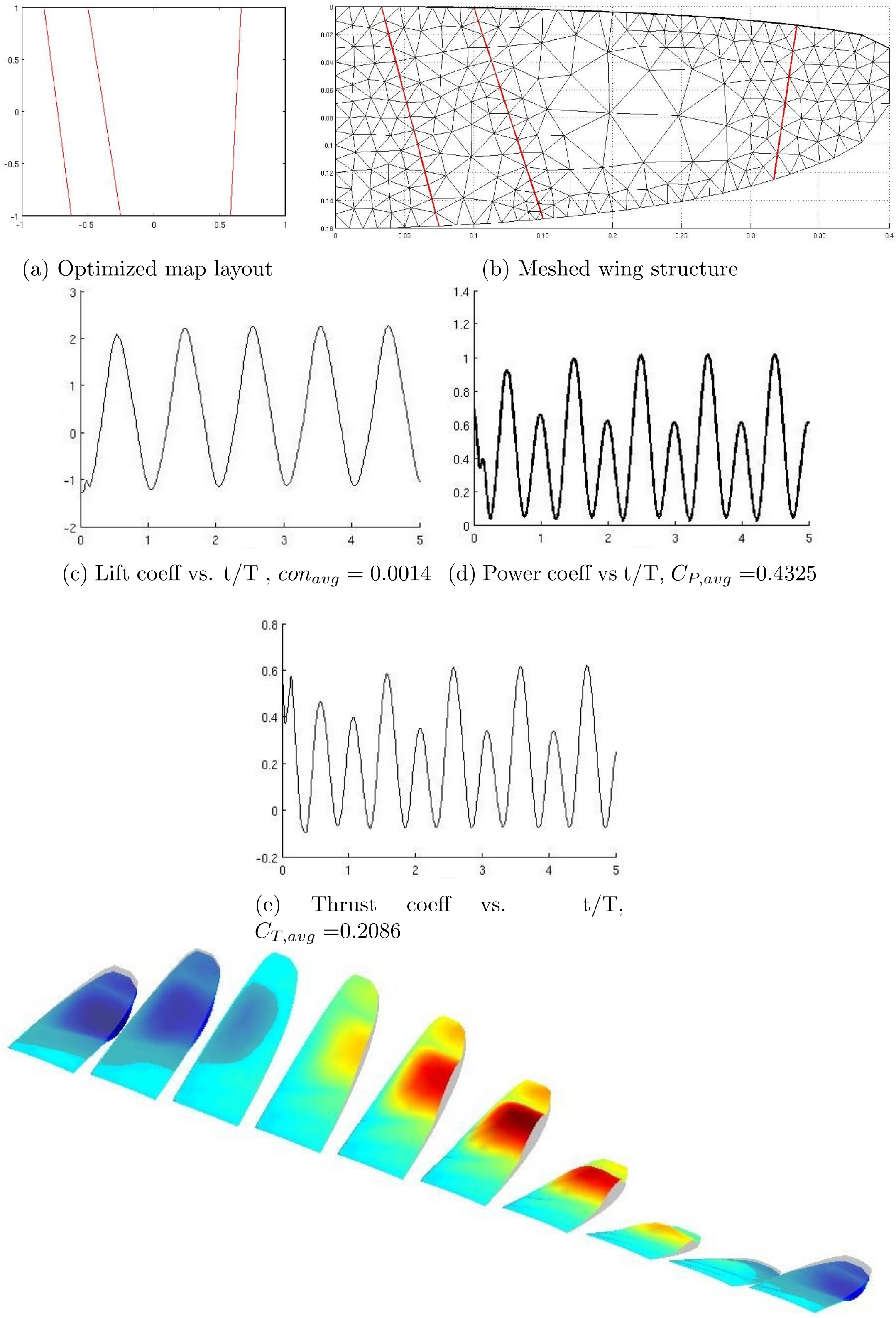
(c) Lift coeff vs. t/T , $con_{avg} = 4.79e^{-4}$

(d) Power coeff vs t/T, $C_{P,avg}$ =0.5370

(e) Thrust coeff vs. t/T, $C_{T,avg}$ =0.2503

(f) Deformation of wing over a flapping cycle

Figure 9: Optimal thrust coefficient design: Original mapping system Run 1



(a) Optimized map layout

(b) Meshed wing structure

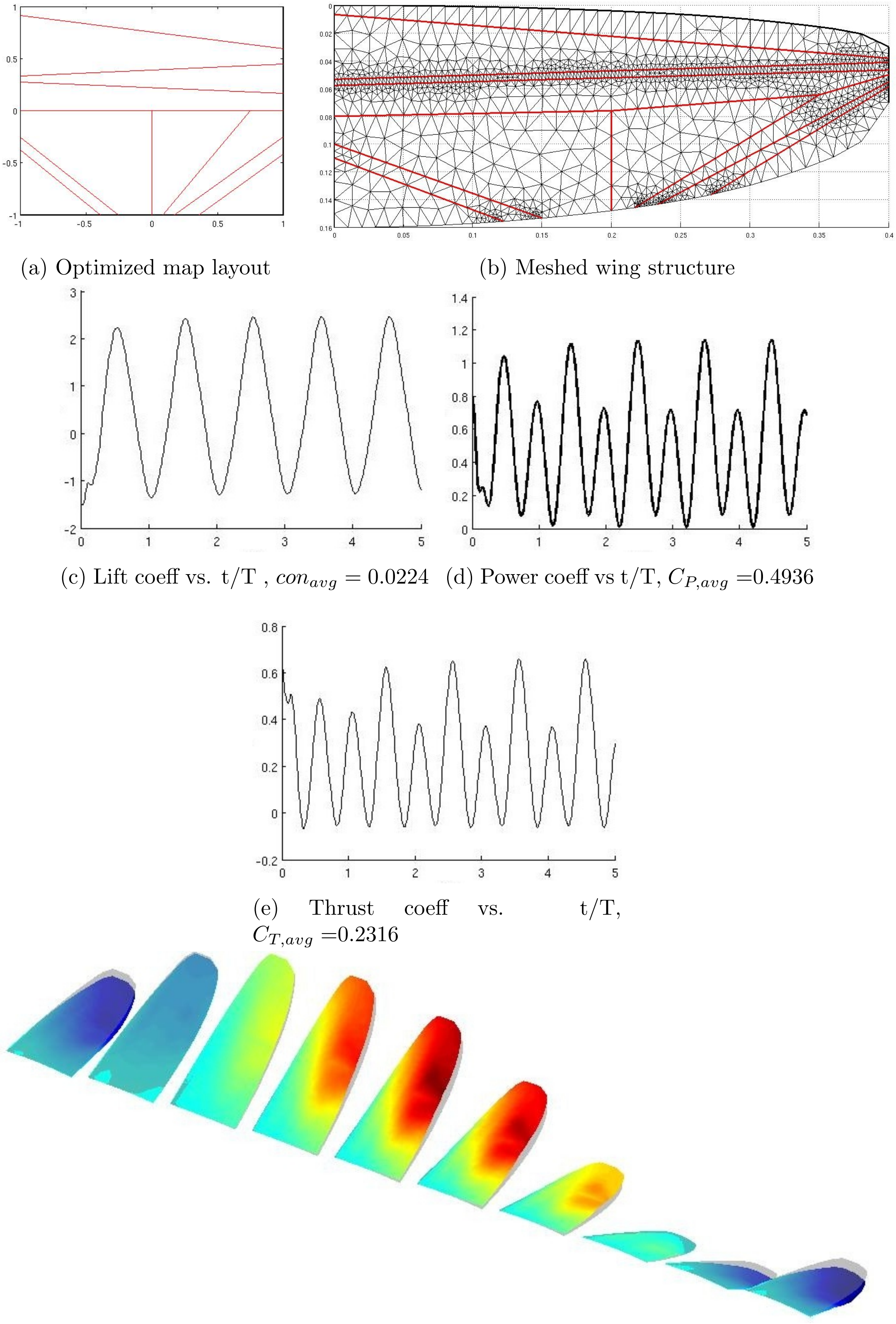
(c) Lift coeff vs. t/T , $con_{avg} = 0.0014$

(d) Power coeff vs t/T, $C_{P,avg}$ =0.4325

(e) Thrust coeff vs. t/T, $C_{T,avg}$ =0.2086

(f) Deformation of wing over a flapping cycle

Figure 10: Random design #1: Original mapping system Run 1



(a) Optimized map layout

(b) Meshed wing structure

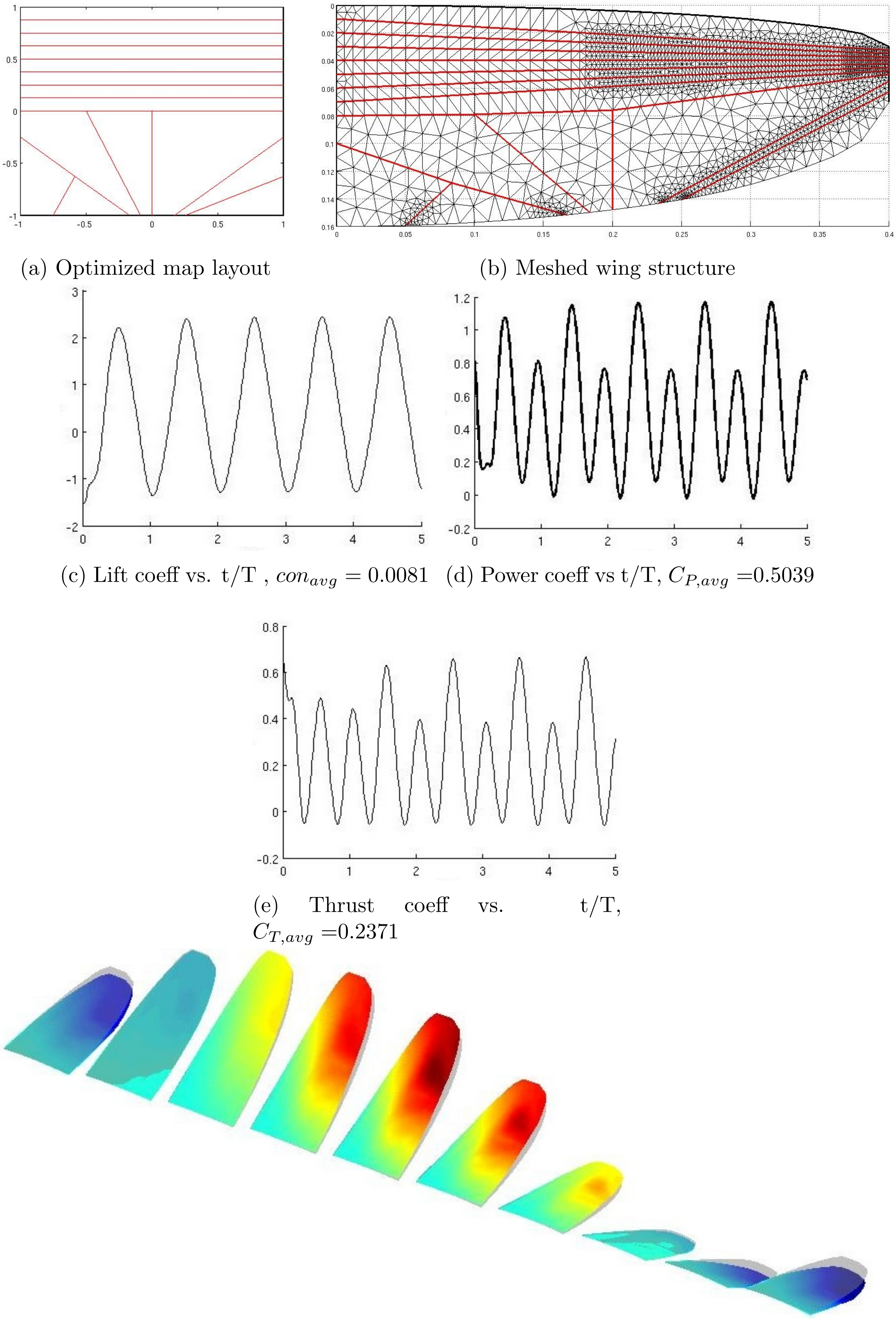
(c) Lift coeff vs. t/T , $con_{avg} = 0.0224$

(d) Power coeff vs t/T, $C_{P,avg}$ =0.4936

(e) Thrust coeff vs. t/T, $C_{T,avg}$ =0.2316

(f) Deformation of wing over a flapping cycle

Figure 11: Random design #2: Original mapping system Run 1



(a) Optimized map layout

(b) Meshed wing structure

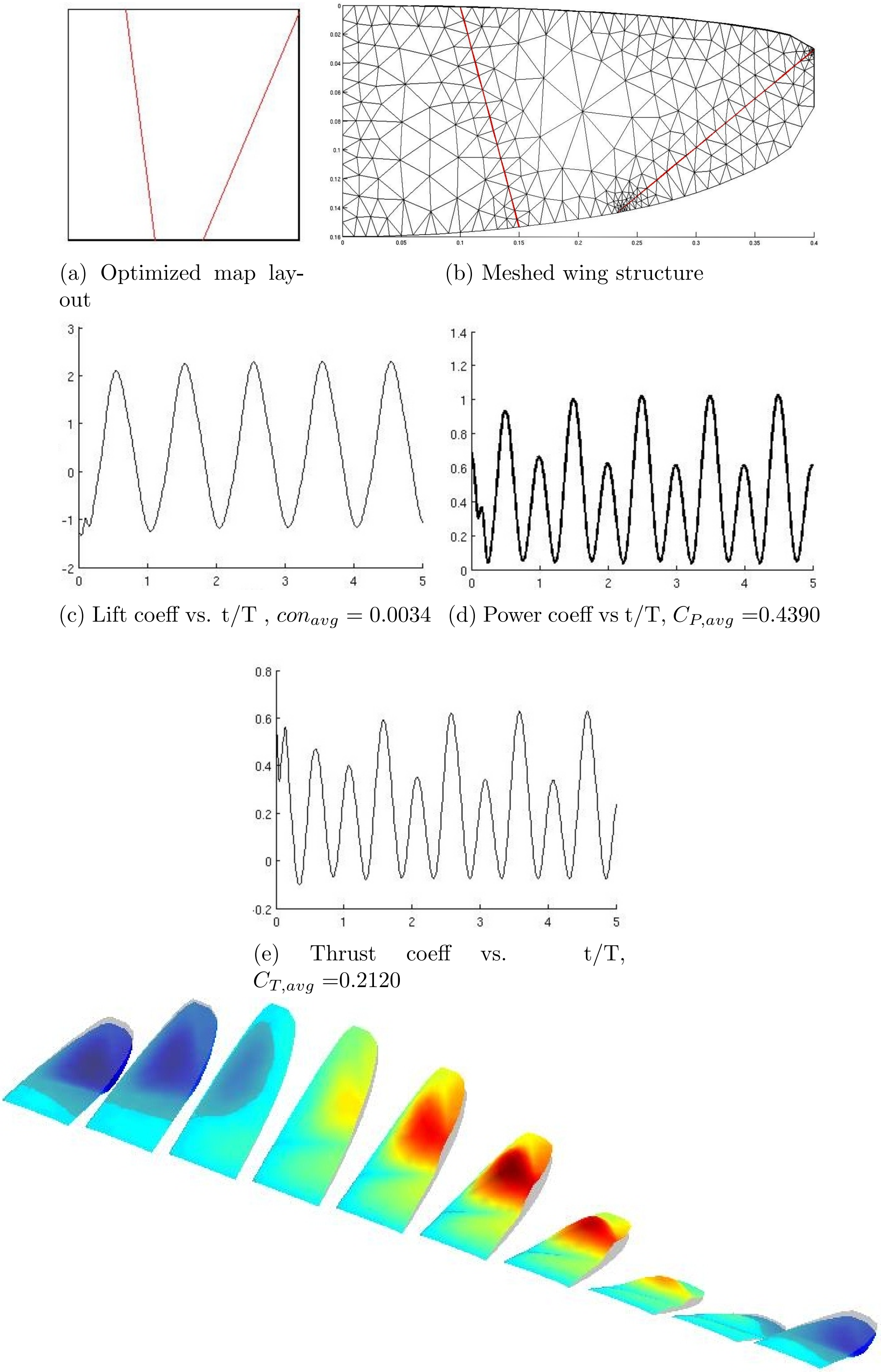
(c) Lift coeff vs. t/T , $con_{avg} = 0.0081$

(d) Power coeff vs t/T, $C_{P,avg}$ =0.5039

(e) Thrust coeff vs. t/T, $C_{T,avg}$ =0.2371

(f) Deformation of wing over a flapping cycle

Figure 12: Random design #3: Original mapping system Run 1



(a) Optimized map layout

(b) Meshed wing structure

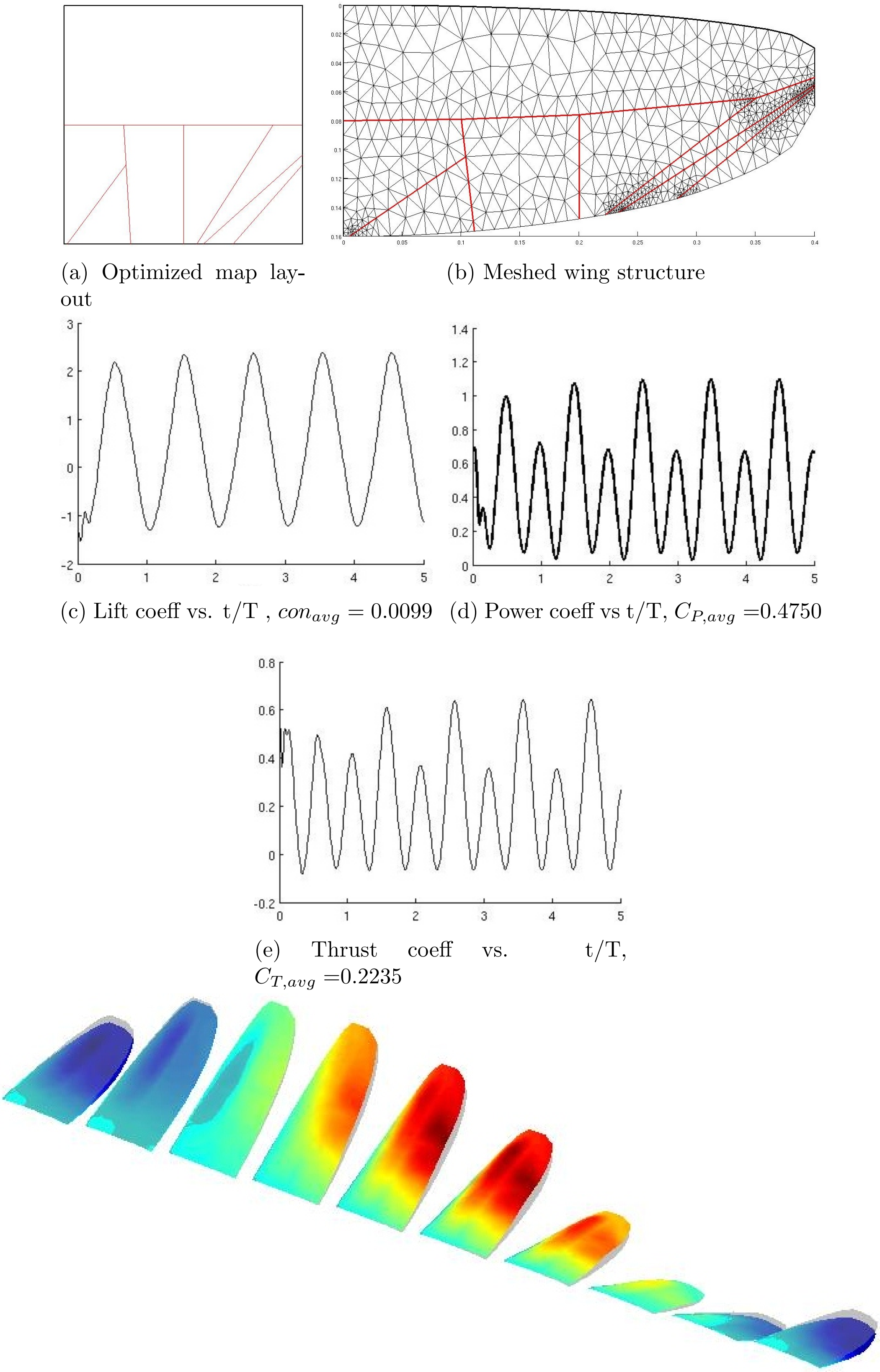
(c) Lift coeff vs. t/T , $con_{avg} = 0.0034$

(d) Power coeff vs t/T, $C_{P,avg}$ =0.4390

(e) Thrust coeff vs. t/T, $C_{T,avg}$ =0.2120

(f) Deformation of wing over a flapping cycle

Figure 13: Random design #4: Original mapping system Run 1



(a) Optimized map layout

(b) Meshed wing structure

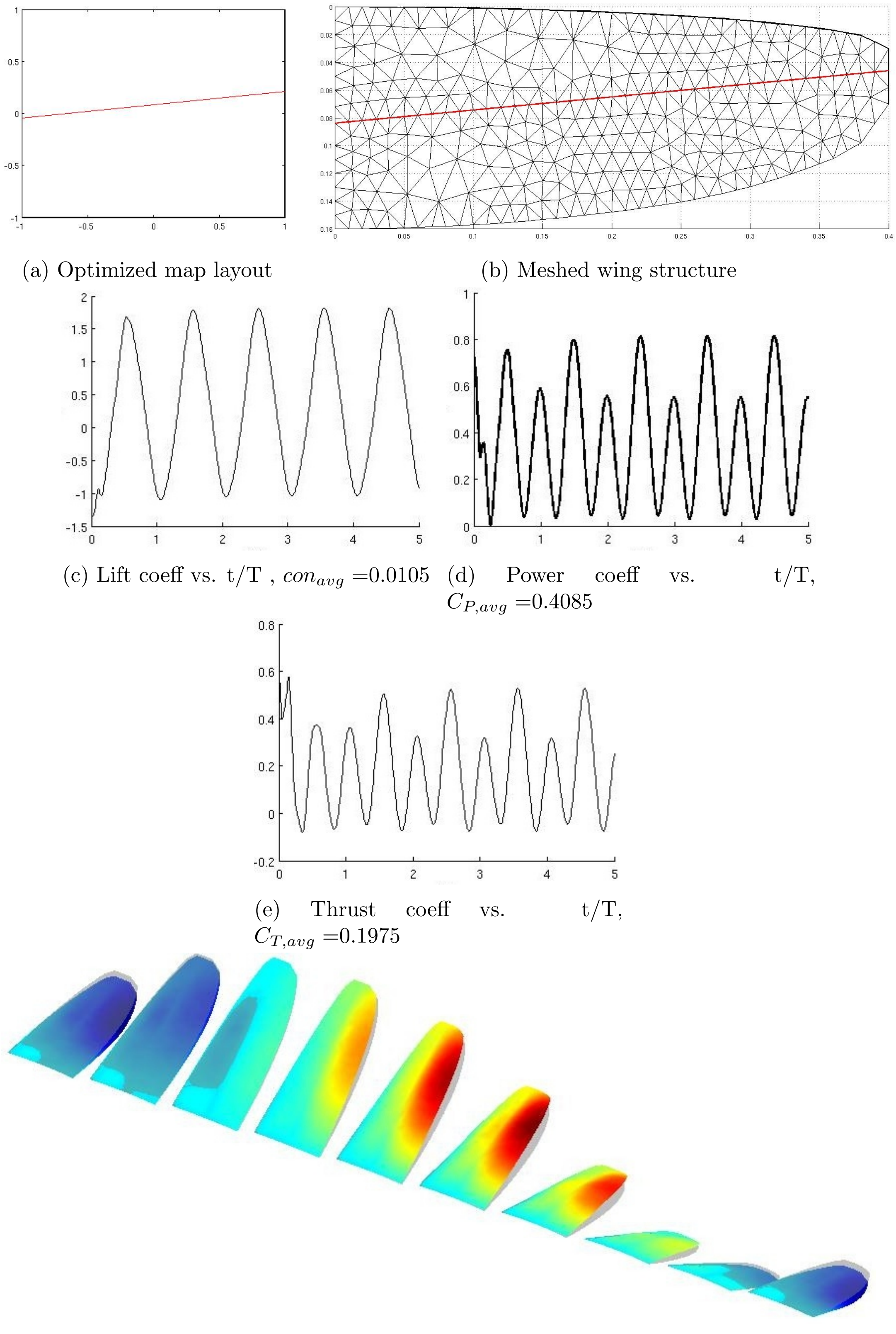
(c) Lift coeff vs. t/T , $con_{avg} = 0.0099$

(d) Power coeff vs t/T, $C_{P,avg}$ =0.4750

(e) Thrust coeff vs. t/T, $C_{T,avg}$ =0.2235

(f) Deformation of wing over a flapping cycle

Figure 14: Random design #5: Original mapping system Run 1



(a) Optimized map layout

(b) Meshed wing structure

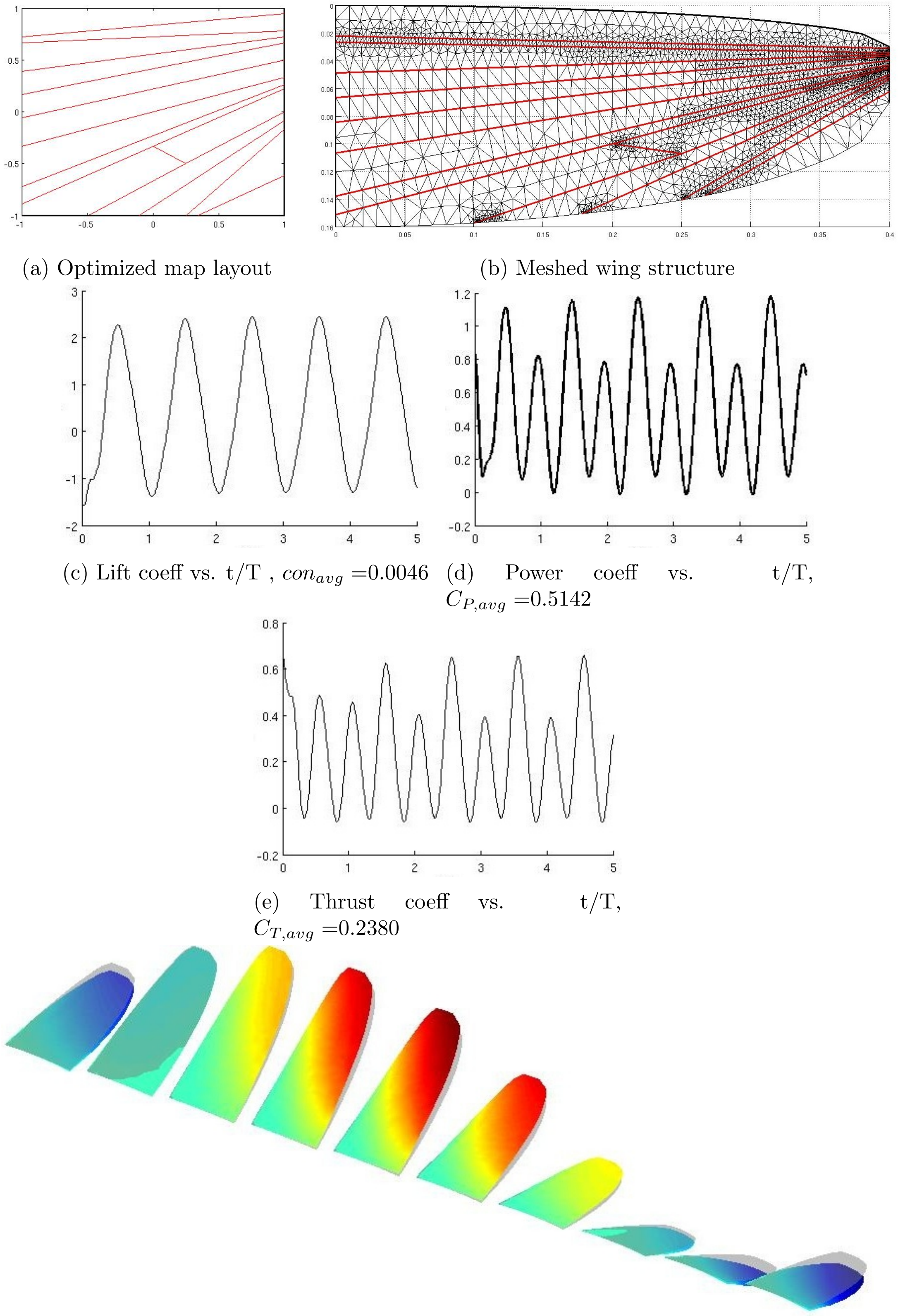
(c) Lift coeff vs. t/T , con_{avg} =0.0105

(d) Power coeff vs. t/T, $C_{P,avg}$ =0.4085

(e) Thrust coeff vs. t/T, $C_{T,avg}$ =0.1975

(f) Deformation of wing over a flapping cycle

Figure 15: Optimal power coefficient design: Fractone mapping system Run 2



(a) Optimized map layout

(b) Meshed wing structure

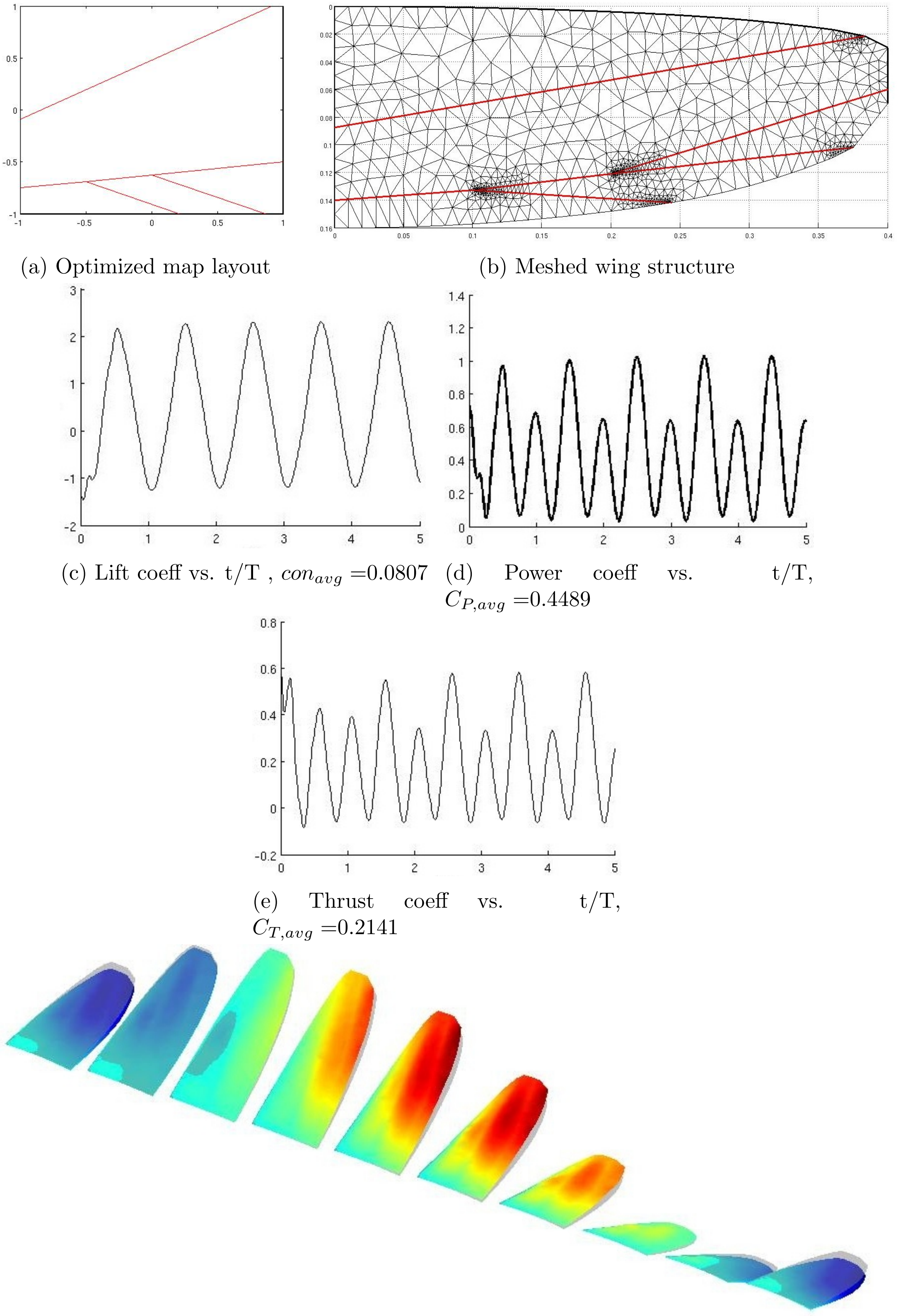
(c) Lift coeff vs. t/T , con_{avg} =0.0046

(d) Power coeff vs. t/T, $C_{P,avg}$ =0.5142

(e) Thrust coeff vs. t/T, $C_{T,avg}$ =0.2380

(f) Deformation of wing over a flapping cycle

Figure 16: Optimal thrust coefficient design: Fractone mapping system Run 2



(a) Optimized map layout

(b) Meshed wing structure

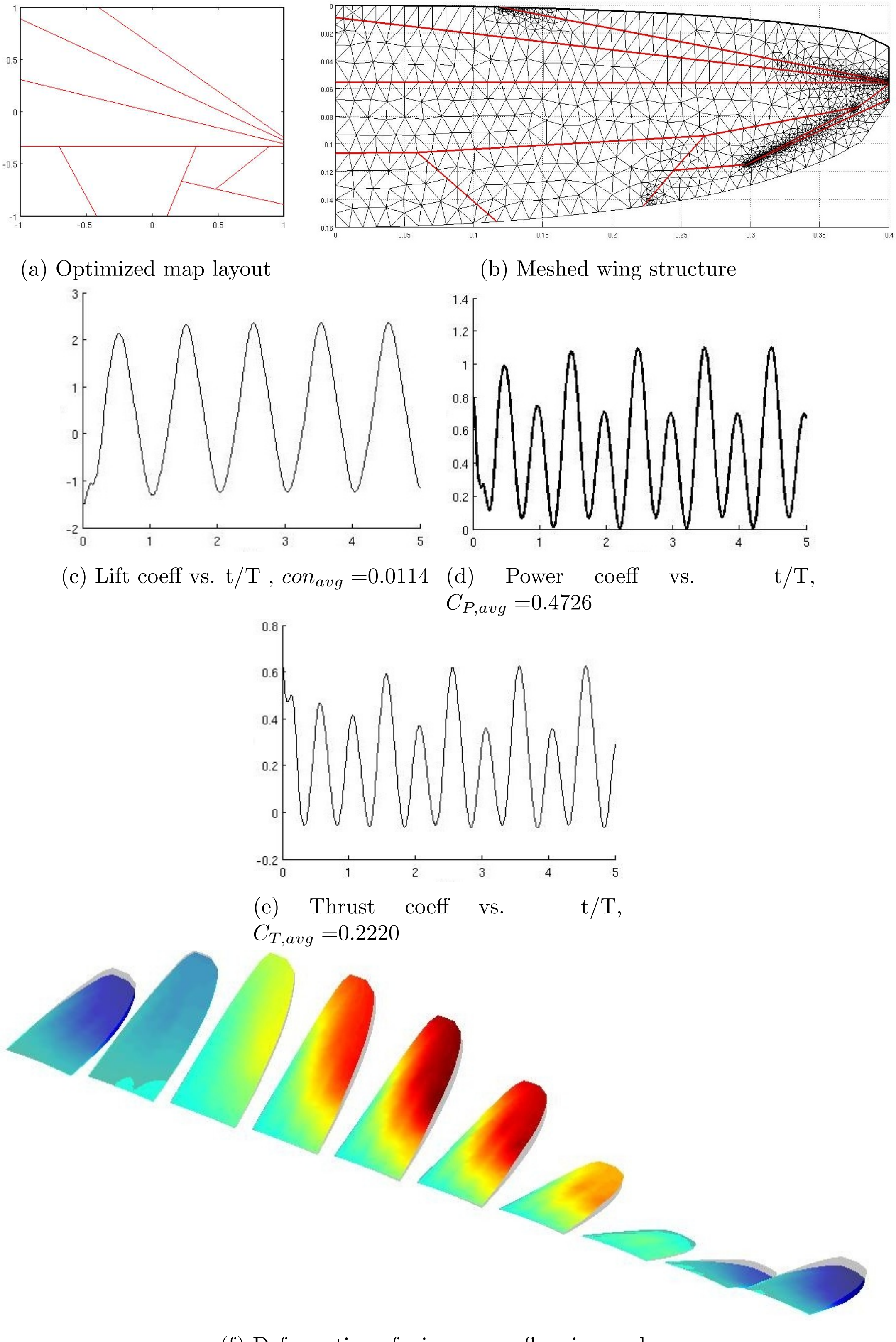
(c) Lift coeff vs. t/T , con_{avg} =0.0807

(d) Power coeff vs. t/T, $C_{P,avg}$ =0.4489

(e) Thrust coeff vs. t/T, $C_{T,avg}$ =0.2141

(f) Deformation of wing over a flapping cycle

Figure 17: Random Design #1: Fractone mapping system Run 2



(a) Optimized map layout

(b) Meshed wing structure

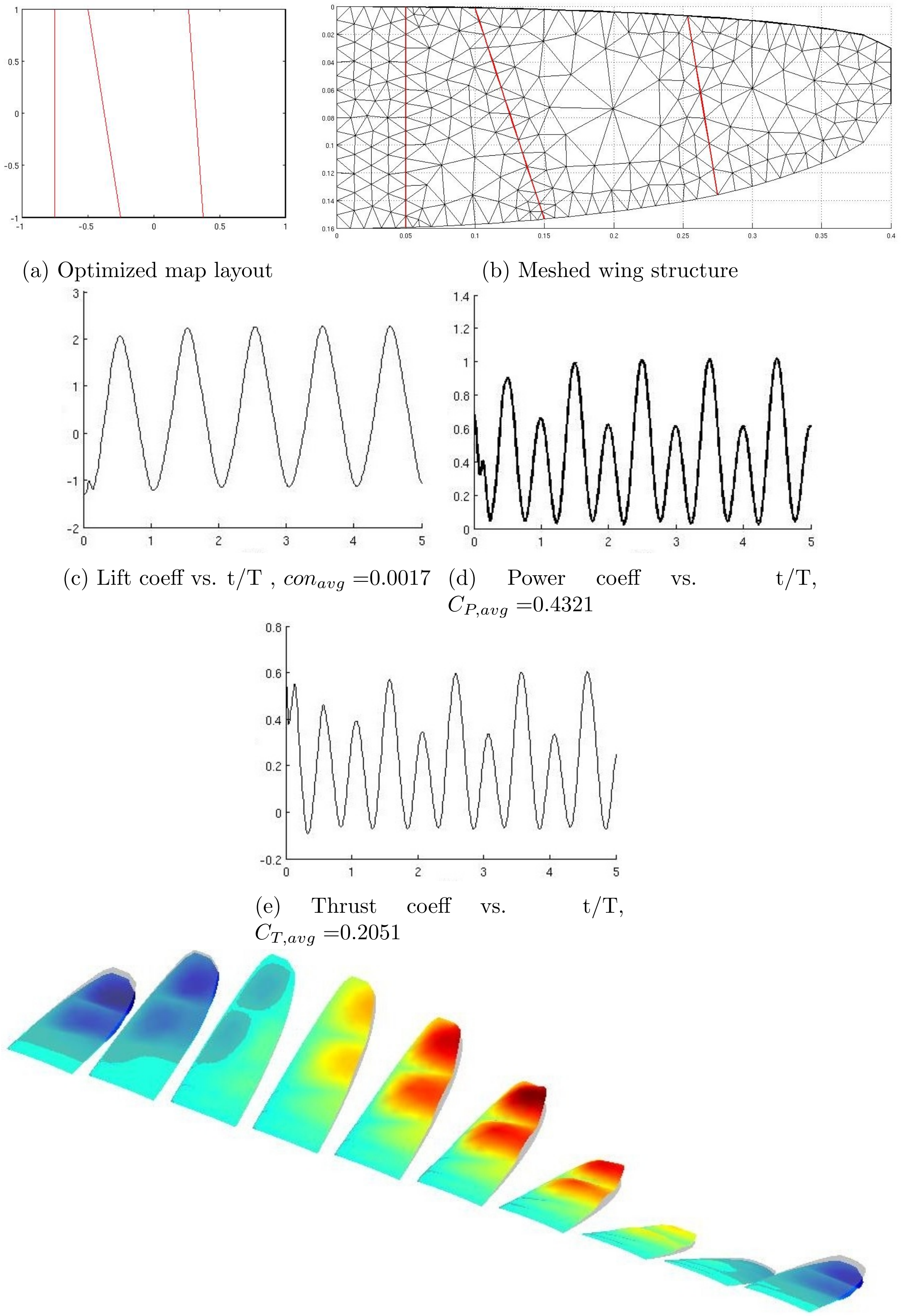
(c) Lift coeff vs. t/T , con_{avg} =0.0114

(d) Power coeff vs. t/T, $C_{P,avg}$ =0.4726

(e) Thrust coeff vs. t/T, $C_{T,avg}$ =0.2220

(f) Deformation of wing over a flapping cycle

Figure 18: Random Design #2: Fractone mapping system Run 2



(a) Optimized map layout

(b) Meshed wing structure

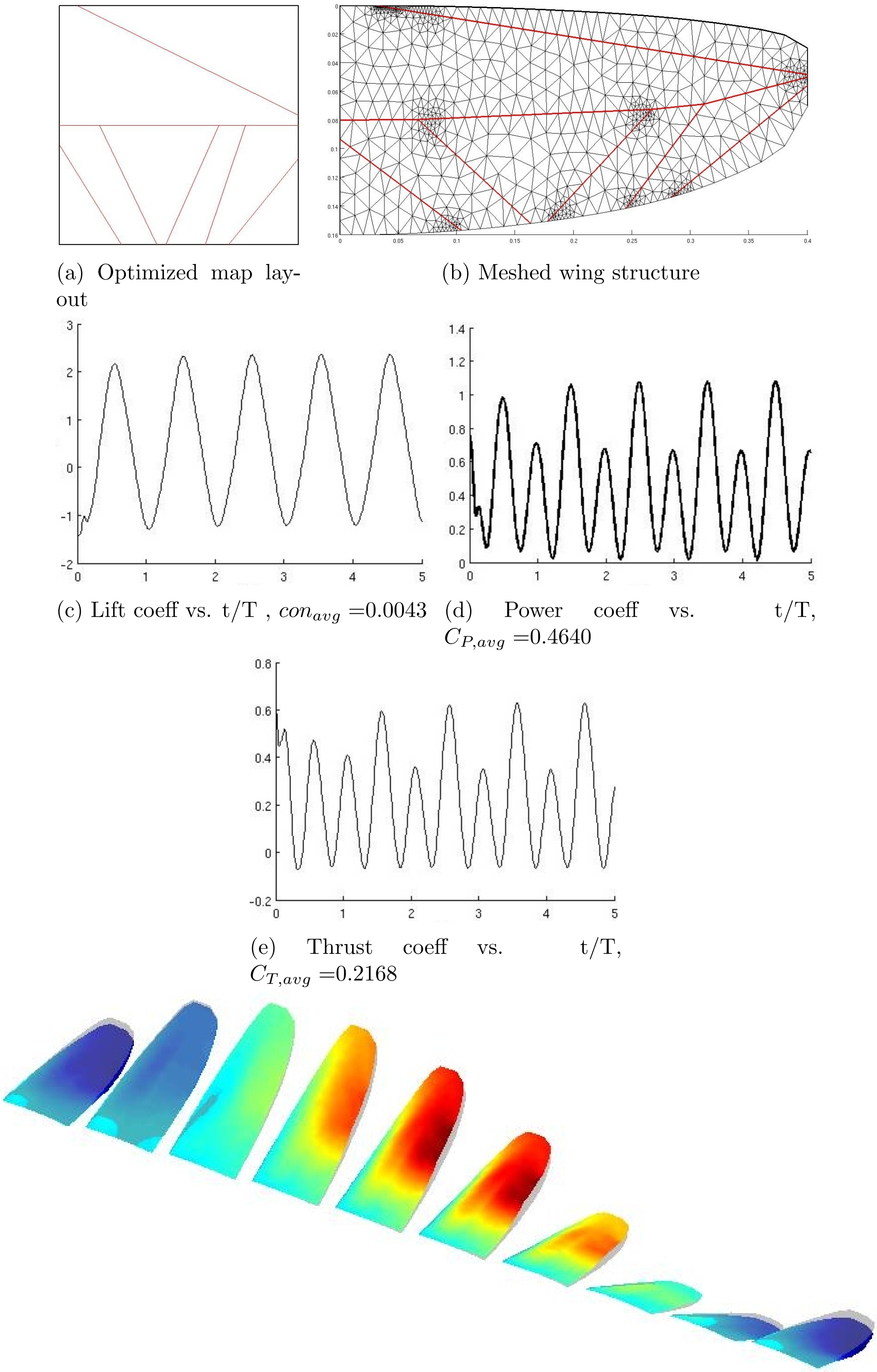
(c) Lift coeff vs. t/T , con_{avg} =0.0017

(d) Power coeff vs. t/T, $C_{P,avg}$ =0.4321

(e) Thrust coeff vs. t/T, $C_{T,avg}$ =0.2051

(f) Deformation of wing over a flapping cycle

Figure 19: Random Design #3: Fractone mapping system Run 2



(a) Optimized map layout

(b) Meshed wing structure

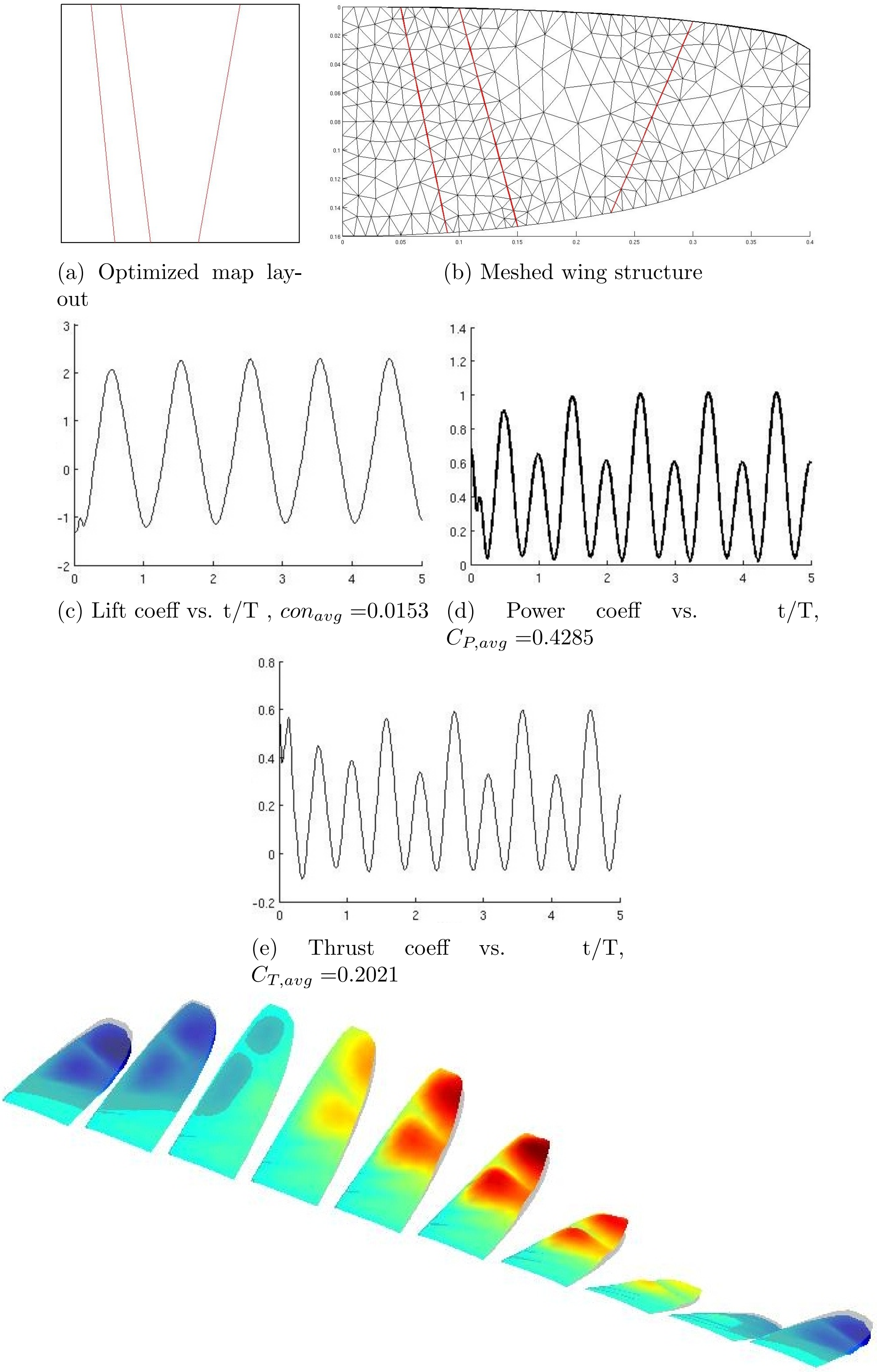
(c) Lift coeff vs. t/T , con_{avg} =0.0043

(d) Power coeff vs. t/T, $C_{P,avg}$ =0.4640

(e) Thrust coeff vs. t/T, $C_{T,avg}$ =0.2168

(f) Deformation of wing over a flapping cycle

Figure 20: Random Design #4: Fractone mapping system Run 2



(a) Optimized map layout

(b) Meshed wing structure

(c) Lift coeff vs. t/T , con_{avg} =0.0153

(d) Power coeff vs. t/T, $C_{P,avg}$ =0.4285

(e) Thrust coeff vs. t/T, $C_{T,avg}$ =0.2021

(f) Deformation of wing over a flapping cycle

Figure 21: Random Design #5: Fractone mapping system Run 2

7 Pareto Front Analysis

A total of ten trials were performed for each of the fractone and non-fractone optimizations. A typical Pareto front resulting in the final (200th) generation can be seen in figure 22. The Pareto front exhibited here illustrates a wide distribution of points as biased by the niching technique and superior performance of individuals compared to the other random designs.

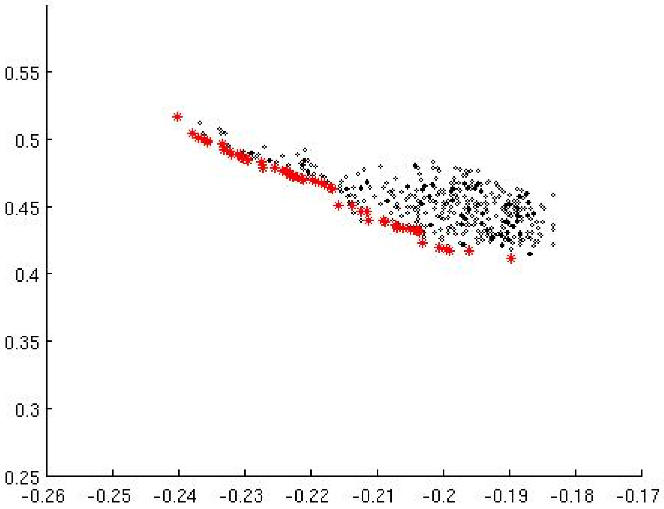


Figure 22: Sample Pareto Front (obtained from Run 2 without fractones):- Power coefficient vs. Thrust Coefficient, Pareto front (red) and other random designs (black)

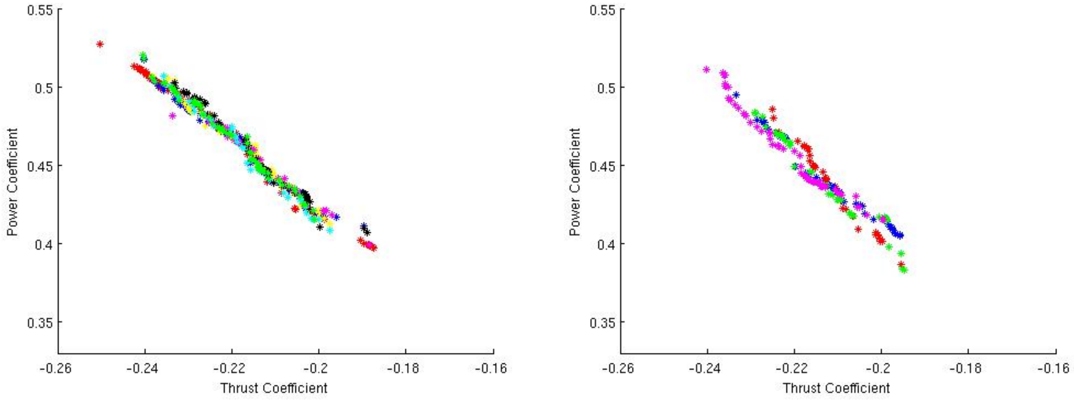
7.1 Repeatability and Performance

The final Pareto fronts obtained from of each of the ten trials were collected to observe the repeatability of the algorithm. Figure 23 displays the repeatability of results from both the fractone and original mapping schemes. The two different methods appear to have comparable performances in terms of consistency as seen from the clustering of points and relatively narrow band-widths of the collected fronts.

The collection of Pareto fronts from the two methods were further combined in figure 24 to compare the fitness of the two methods. The dispersion of points belonging to the two cases are fairly even and consistent. While both methods seem to perform equally well on the thrust optimal end of the spectrum, the fractone mapping designs exceed the capabilities of the original mapping system for the majority of the spectrum as it progresses to the minimal power requirement designs. This is observed by the dominance of the fractone designs along the leading edge of the Pareto Front, especially along on the right hand side of the curve.

7.2 Convergence

To determine the convergence of the GA, the gradual nearing of the Pareto fronts were numerically approximated by computing an averaged norm. The distance from the highest ranked points in each iteration to the set of points in the final (200th generation) Pareto front were estimated and averaged for each approaching generation. The resulting convergence rates were compiled and averaged and can be seen in figure 25. Here the convergence rates



(a) Original mapping (no fractones)

(b) Fractone mapping

Figure 23: Final Pareto fronts collected from runs 1 through10 (each test batch displayed in different colors and markers)

appear to be comparable, however, the rate of convergence with the fractone mapping scheme continuously improves over the 200 generations while the original methods quickly converges during the first few iterations and then levels off. The convergence of the fractone model is more consistent and it should be noted that the convergence rates are with respect to the final Pareto Fronts of each case and the final results of the fractone model were generally superior.

8 Conclusion

The results of these preliminary tests indicate that the performance of the fractone modified mapping system are in fact superior to the conventional mapping system used for topology optimization. Since the inclusion of fractones results in a competitive method for generating maps, new methods have the potential to further improve the performance and convergence of topology optimizations should the simple system be revised.

Future studies may test the effects of more complex fractone systems that incorporate multiple and competing growth factors, source terms, variations in the initial distributions, and removal or deactivation of fractones themselves. Inclusion of these additional control parameter could provide greater control of the final designs and improve the overall efficiency and capabilities of the process.

References

G. Allaire. *Shape Optimization be the Homogenization Method*. Springer, New York, 2002.

W. Becker. *The World of the Cell*. Pearson/Benjamin Cummings, 2009. ISBN 9780805393934. URL `http://books.google.com/books?id=dV-8PwAACAAJ`.

M. P. Bendsøe and O. Sigmund. *Topology Optimization: Theory, Methods and Applications*. Springer, New York, 2nd edition, 2003.

J. Deaton and R. Grandhi. A survey of structural and multidisciplinary continuum topology optimization: post 2000. *Structural and Multidisciplinary Optimization*, 49(1):1–38, 2014.

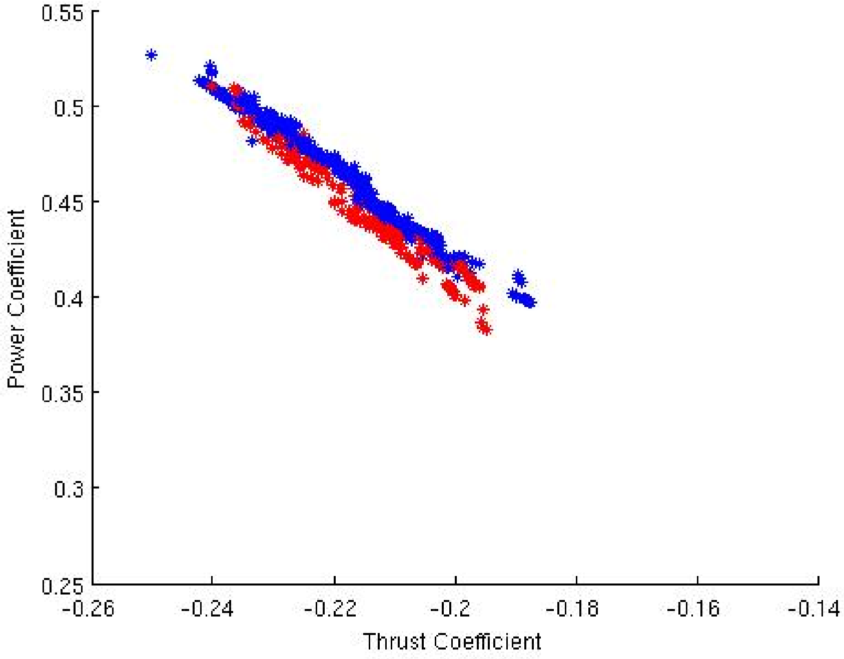


Figure 24: Collection of final Pareto fronts from 10 runs: fractone mapping in red, original mapping in blue

G. F. C. Eric B. Becker and J. T. Oden. *Finite Elements An Introduction : Volume I*. Prentice Hall, New Jersey, 1981.

A. L. Gain and G. H. Paulino. A critical comparative assessment of differential equation-driven methods for structural topology optimization. *Structural and Multidisciplinary Optimization*, 48:685–710, 2013. doi: 10.1007/s00158-013-0935-4.

A. Kerever, J. Schnack, D. Vellinga, N. Ichikawa, C. Moon, E. Arikawa-Hirasawa, J. T. Efird, and F. Mercier. Novel extracellular matrix structures in the neural stem cell niche capture the neurogenic factor fibroblast growth factor 2 from the extracellular milieu. *STEM CELLS*, 25(9):2146–2157, 2007. ISSN 1549-4918. doi: 10.1634/stemcells.2007-0082. URL `http://dx.doi.org/10.1634/stemcells.2007-0082`.

M. H. Kobayashi, H.-T. C. Pedro, R. Kolonay, and G. Reich. On a cellular division method for aircraft structural design. *The Aeronautical Journal of the Royal Aeronautical Society*, 113:821–831, 2009.

R. M. Kolonay and M. H. Kobayashi. Optimization of aircraft lifting surfaces using a cellular division method. *Journal of Aircraft*, 52(6):2051–2063, 2015.

F. Mercier, J. T. Kitasako, and G. I. Hatton. Fractones and other basal laminae in the hypothalamus. *The Journal of Comparative Neurology*, 455(3):324–340, 2003. ISSN 1096-9861. doi: 10.1002/cne.10496. URL `http://dx.doi.org/10.1002/cne.10496`.

Nakamura, A., A. Lindenmayer, and K. Aizawa. *The Book of L*. Springer, Berlin, 1986.

S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, New York, 2003.

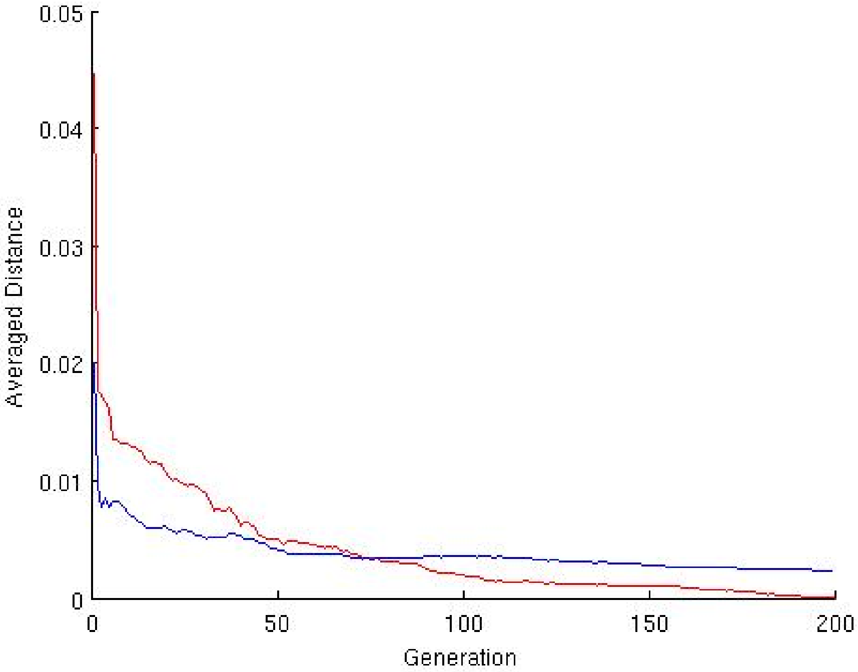


Figure 25: Averaged convergence rates for 10 runs: fractone mapping in red, original mapping in blue

P. Prusinkiewicz and A. Lindenmayer. *The Algorithmic Beauty of Plants.* Springer, New York, 2004.

G. Rozvany. *Topology optimization in structural mechanics.* Springer–Verlag, Berlin, 1994.

B. Stanford, P. Beran, and M. Kobayashi. Aeroelastic optimization of flapping wing venation: A cellular division approach. *AIAA journal*, 50(4):938–951, 2012.

B. K. Stanford, P. S. Beran, and M. H. Kobayashi. Aeroelastic optimization of flapping wing venation: a cellular division approach. 52nd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Denver, Colorado, 2011.

N. P. van Dijk, K. Maute, M. Langelaar, and F. van Keulen. Level-set methods for structural topology optimization: a review. *Structural and Multidisciplinary Optimization*, 48:437–472, 2013. doi: 10.1007/s00158-013-0912-y.

C. Wang, Z. Zhao, M. Zhou, O. Sigmund, and X. S. Zhang. A comprehensive review of educational articles on structural and multidisciplinary optimization. *Structural and Multidisciplinary Optimization*, 64:2827–2880, 2021.

Chapter 7: Fundamentals on the Topological Derivative concept and its classical applications

Chapter details

Chapter DOI:

https://doi.org/10.4322/978-65-86503-71-5.c07

Chapter suggested citation / reference style:

Carvalho, Fernando S., et al. (2022). “Fundamentals on the Topological Derivative concept and its classical applications”. *In* Jorge, Ariosto B., et al. (Eds.) *Model-Based and Signal-Based Inverse Methods*, Vol. I, UnB, Brasilia, DF, Brazil, pp. 269–300. Book series in Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity.

P.S.: DOI may be included at the end of citation, for completeness.

Book details

Book: Model-based and Signal-Based Inverse Methods

Edited by: Jorge, Ariosto B., Anflor, Carla T. M., Gomes, Guilherme F., & Carneiro, Sergio H. S.

Volume I of Book Series in:

Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity

Published by: UnB City: Brasilia, DF, Brazil Year: 2022

DOI: https://doi.org/10.4322/978-65-86503-71-5

Fundamentals on the Topological Derivative concept and its classical applications

Fernando Soares de Carvalho[1], Carla Tatiana Mota Anflor[2*],
Ariosto Bretanha Jorge[2], Adrián Pablo Cisilino[3], Rogério José Marczak[4].

[1]Math, Federal University of Tocantins, Arraias, Tocantins, Brazil, e-mail: fscarvalho@uft.edu.br
[2*]Post-Graduate Program - Integrity of Engineering Materials, University of Brasilia, Brazil. E-mail: anflor@unb.br, ariosto.b.jorge@gmail.com
[3]Department of Mechanical Engineering INTEMA, Universidad Nacional de Mar del Plata, Argentina, e-mail: cisilino@fi.mdp.edu.ar
[4] Department of Mechanical Engineering, Federal University of Rio Grande do Sul, Brazil, e-mail: rato@mecanica.ufrgs.br

*Corresponding author: anflor@unb.br

Abstract

The concept of the topological derivative has been derived for several engineering problems during the last years. In this chapter, the fundamentals and the resulting closed formulae of topological derivative for some of the most classical problems are addressed. A brief review of the mathematical statements used in the topological derivative concept is given. The programming strategies regarding implementing the main routine for the topology optimization are pointed out. Some numerical examples concerning classical applications are introduced to demonstrate the application of the topological derivative concept for topology problems.

1 Introduction

Topological sensitivity analysis was presented as a technique that allows obtaining simultaneously the optimal shape and topology, being proposed originally by (Schumacher [1996], Sokołowski and Żochowski [1997], Sokolowski and Zochowski [1999], Garreau et al. [1998], Garreau et al. [2001]). This sensitivity calculation results in a scalar function called a topological derivative (D_T). The D_T provides for each point in the domain the sensitivity of the cost function when creating a small hole at that point.Garreau et al. [1998]), proposed the truncated domain method to calculate the topological derivative. The proposed method was based on some simplifying assumptions, the most severe of which consisted in the fact that the cost function should not explicitly depend on the domain. The works of Sokolowski and Zochowski [1999] and Céa et al. [2000] presented the calculation of D_T, via shape sensitivity analysis, particularized only for cases where the homogeneous Neumann boundary condition was prescribed in the holes. The other boundary conditions, such as non-homogeneous Neumann, Dirichlet and Robin do not validate their applicability in this calculation hypothesis. Novotny et al. [2003]

precisely establish the concepts of derived topology and shape change sensitivity analysis for isotropic materials. This last methodology does not present any limitation regarding the cost function or the type of boundary condition prescribed in the holes. Since then several classes of engineering and physics problems have been solved by employing the D_T concept, for instance, topology optimization ([Amstutz and Novotny [2010], Novotny et al. [2007]]), inverse analysis ([Carpio and Rapún [2008], Rocha and Novotny [2017]]), and image processing ([Hintermüller and Laurain [2009], Larrabide et al. [2008]]). The D_T was also computed using the Boundary Element Method (BEM) for topology optimization of potential (Anflor [2007],Anflor and Marczak [2009], Anflor et al. [2014]) and elasticity (Marczak [2008], Bertsch et al. [2008], Anflor et al. [2018]) problems as an alternative to the Finite Element Method (FEM) employed as the standard numerical solver. All advantages provided by BEM as a boundary method were taken into account showing the efficiency of the developed methodology for optimization problems. Another class of problem of great interest concerns the damage identification in structures. The identification of flaws by the inverse problem was generally solved by heuristic algorithms where the information about sensitivity or gradient of the cost functional with respect to design parameters are not needed. Despite the success of the use of these algorithms, the computational cost was still high because a large number of direct problems has to be evaluated and solved. The computational time can be drastically reduced by using the cost functional topological sensitivity instead of the full functional (Comino et al. [2008]). In addition, the use of topological sensitivity coupled to heuristic algorithms increases the accuracy for estimating the location and size of defects. The concept of D_T becomes naturally attractive and suitable for problems concerning damage detection, once the D_T measures the sensitivity of a functional shape with respect to an infinitesimal singular domain perturbation. The perturbation may be represented as damages in the structure in the shape of holes, inclusions, sources terms, or even cracks. The topological sensitivity analysis was carried out for the Laplace equation to identify arbitrary shaped cracks in two-dimensional domains (Amstutz et al. [2005]). A method based on the multi-frequency D_T was developed as an alternative to standard guided-waves-based Structural Health Monitoring (SHM) methods and used for locating the presence of flaws in thin plates (Martinez Dominguez et al. [2018]). The damage identification based on the D_T method was addressed in (da Silva and Novotny [2022]) for problems governed by the elastodynamic Kirchhoff and Reissner-Mindlin plate bending models in the frequency domain. According to this brief review, the reader can have an idea about the wide range of the use of D_T in physical phenomena modeled by partial differential equations. In the remainder of this chapter, the fundamentals on D_T for classical problems of topology optimization are introduced, and some numerical examples are presented, showing the efficiency and applicability of this concept for generating optimal geometries.

2 Topological derivative considering the insertions as voids

The theory behind the D_T is the evaluation of a given cost function when a small hole of radius is open inside the domain, as shown in 1.

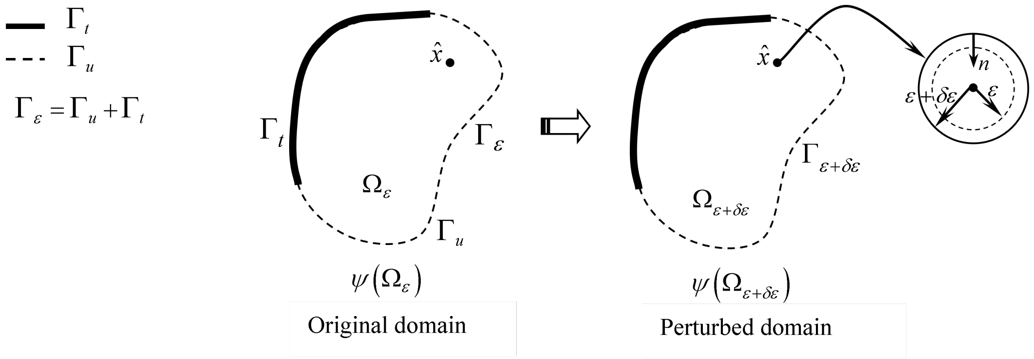

Γ_t
Γ_u
$\Gamma_\varepsilon = \Gamma_u + \Gamma_t$
$\hat{x}$
Γ_t
Γ_ε
Ω_ε
Γ_u
$\psi(\Omega_\varepsilon)$
Original domain
$\hat{x}$
n
$\varepsilon+\delta\varepsilon$
ε
$\Gamma_{\varepsilon+\delta\varepsilon}$
$\Omega_{\varepsilon+\delta\varepsilon}$
$\psi(\Omega_{\varepsilon+\delta\varepsilon})$
Perturbed domain

Figure 1: The new concept of the topological derivative and the boundary conditions

In this sense, the concept of D_T consists of determining the sensitivity of a given function cost (ψ) when this small hole is increased or decreased. The local value of the D_T at a point $\hat{x}$ inside the domain for this case is given by eq. 1:

$$D_T(\hat{x}) = \lim_{\varepsilon \to \infty} \frac{\psi(\omega_\varepsilon) - \psi(\omega)}{f(\varepsilon)}, \tag{1}$$

where $\psi(\omega)$ and $\psi(\omega_\epsilon)$ are the cost function evaluated for the original domain and the perturbed domain, respectively, and (f) is a problem-dependent regularizing function. It is important to highlight that it is not possible to establish an isomorphism between domains with different topologies using eq.1. A new concept regarding the D_T was introduced by Novotny et al. (2003) that allowed the non-isomorphism between the original and the modified domains to be overcome. The mathematical idea was based on the creation of a hole that can be accomplished by a single perturbation to an existing hole with radius tending to zero. This allows the restatement of the such a way that it is possible to establish a mapping between them, as presented in eq.2:

$$D_T(\hat{x}) = \lim_{\varepsilon \to \infty} \frac{\psi(\omega_{\varepsilon+\delta\epsilon}) - \psi(\omega_\varepsilon)}{f(\Omega_{\varepsilon+\delta_\varepsilon}) - f(\Omega_\varepsilon)}, \tag{2}$$

where δ is a small perturbation of the hole's radius. It's worth mention that eq.2 is a general definition for D_T. This section introduces the D_T particularized for some of the classical engineering problems.

2.1 Potential problems

In the case of isotropic linear heat transfer problems, the direct problem is stated as: Find u_ε, such that

$$\left\{\begin{array}{rcll} -k\nabla u_\varepsilon & = & b & \text{in } \ \Omega \ , \\ u_\varepsilon & = & u & \text{on } \ \Gamma_D, \\ k\partial_n u & = & q & \text{on } \ \Gamma_N, \\ k\partial_n u_\varepsilon & = & h_\varepsilon(u_\varepsilon - u_\infty) & \text{on } \ \Gamma_R, \\ h(\alpha,\beta,\gamma) & = & 0 & \text{on } \ \Gamma_\varepsilon, \end{array}\right. \tag{3}$$

Γ_ϵ stands for the holes boundary and

$$h(\alpha,\beta,\gamma) = \alpha(u_\varepsilon - \bar{u}^\varepsilon) + \beta(k\frac{\partial u_\varepsilon}{\partial n} + \bar{q}^\varepsilon) + \gamma(k\frac{\partial u_\varepsilon}{\partial n} + h_c(u - \bar{u}_\varepsilon)), \tag{4}$$

is a function which takes into account the type of boundary condition on the perimeter of holes to be created. In eq. 4, u_ϵ and $\frac{du_\varepsilon}{dn} = q_\epsilon$ are the temperature and the flux on the hole boundary, while u^ϵ_{inf} and h^ϵ_c are the holeâs internal convection parameters, respectively. Suitable choices of α, β and γ define the type of boundary condition on the hole. One may impose $\alpha = 1$ and $\beta = \gamma = 0$ in eq. 4 if the Dirichlet b.c. is applied to the holes that are being opening during the iterative process. Using asymptotic expansions to include the effects of a hole inserted in Ω it is possible derive analytic expressions for $\Psi(\Omega_\varepsilon$ and $\Psi(\Omega_{\varepsilon+\delta\varepsilon})$, which are used to generate the final expressions for eq.2.

A general form for the cost function can be written as the total potential energy function,

$$\Psi(\Omega_\tau) = \frac{1}{2}\int_{\Omega_\tau} \phi_{\Omega_\tau}(u_\tau)d\Omega_\tau + \int_{\Gamma_\tau} \phi_{\Gamma_\tau}(u_\tau)d\tau, \tag{5}$$

where τ is a parameter associate to the shape change velocity, i.e., $x_\tau(x) = x + \tau v(x)$. The sensitivity of the cost function with respect to τ can be derived from the *Gâteaux derivative* as,

$$\frac{d}{d\tau}\Psi(\Omega_\tau)_{\tau=0} = \lim_{\tau\to 0}\frac{\Psi(\Omega_\tau - \Omega_{\tau=0})}{\tau}h(\alpha,\beta,\gamma) = 0, \tag{6}$$

In this case the problem can be re-stated as,

Evaluate: $\frac{d}{d_\tau}\Psi(\Omega_\tau) = 0$

Subject to,

$$a_\tau(u_\tau, n_\tau) = l_\tau(n_\tau), \ \forall \ n_\tau \in \beta_\tau \, \text{and} \, \forall \ \tau \geq 0, \tag{7}$$

where a is a continuous, coercive bilinear form, l_τ is a continuous linear functional and β_τ is the space of the admissible perturbation functions for the perturbed domain $\Omega\tau$. Using the total potential energy as a cost function $(\Psi_\tau(u_\tau) := \frac{1}{2}a_\tau(u_\tau, u_\tau) - l_\tau(u(\tau))$, the a_τ and l_τ functional are written as:

$$a_\varepsilon(u_\varepsilon, n_\varepsilon) := \int_{\Omega_\varepsilon} k\nabla u_\varepsilon \cdot \nabla \eta_\varepsilon \, d\Omega + \int_{\Gamma_\varepsilon} h_c u_\varepsilon \eta_\varepsilon \, d\Gamma + \int_{\partial\Lambda\varepsilon} h_c^\varepsilon u_\varepsilon \eta_\varepsilon \, d\partial\Lambda \tag{8}$$

$$l_\varepsilon(n_\varepsilon) := \int_{\Omega_\varepsilon} bn_\varepsilon d\Omega - \int_\Gamma \overline{q}\eta_\varepsilon d\Gamma - \int_{\Gamma_c} h_c u_\infty \eta_\varepsilon d\Gamma + \int_{\partial\Lambda_\varepsilon \overline{q}_\varepsilon} \eta_\varepsilon d\partial\Lambda + \gamma \int_{\partial\Lambda_\varepsilon} h_c^\varepsilon u_\infty \eta_\varepsilon d\partial\Lambda \tag{9}$$

Equation 7 can be derived and the D_T particularized according to the boundary condition prescribed on the holes.

Neumann Boundary condition

When considering Neumann boundary condition eq.4 is set as ($\alpha = 0, \beta = 1, \gamma = 0$) and the D_T is obtained by taking the limit as,

$$D_T(\hat{x}) = -\lim_{\varepsilon\to 0} \frac{1}{2f'(\varepsilon)} \int_{\partial\Omega\varepsilon} [k(\frac{\partial u_\varepsilon}{\partial t}) - k(\frac{\partial u_\varepsilon}{\partial n}) - 2bu_\varepsilon - \frac{2}{\varepsilon}\overline{q}_\varepsilon u_\varepsilon] \, d\Omega\varepsilon, \tag{10}$$

where the variables t and n stand for the tangencial and normal directions, respectively.

In case of Neumann boundary conditions both cases can be considered, the homogeneous and non-homogeneous as introduced by eqs.11 and 12 , respectively

$$\overline{q_\varepsilon} = \frac{\partial u_\varepsilon}{\partial n}|_{\partial\Omega_\varepsilon} = 0 \text{ with } f'(\varepsilon) = -\pi\varepsilon^2, \tag{11}$$

$$\overline{q_\varepsilon} = \frac{\partial u_\varepsilon}{\partial n}|_{\partial\Omega_\varepsilon} \neq 0 \text{ with } f'(\varepsilon) = -2\pi\varepsilon^2. \tag{12}$$

Dirichlet Boundary Condition

For this case eq.5 can be particularized by setting the variables as ($\alpha = 1, \beta = 0, \gamma = 0$) and the D_T is obtained by taking the limit as,

$$D_T(\hat{x}) = -\lim_{\varepsilon\to 0} \frac{1}{2f'(\varepsilon)} \int_{\partial\Omega\varepsilon} [k(\frac{\partial u_\varepsilon}{\partial t}) - k(\frac{\partial u_\varepsilon}{\partial n}) - 2bu_\varepsilon] \, d\Omega\varepsilon, \tag{13}$$

being the conditions $u_\varepsilon = \overline{u_\varepsilon}$ and $\frac{\partial u_\varepsilon}{\partial t} \neq 0$, which are employed along with $f'(\varepsilon) = -\frac{2\pi}{\varepsilon \ln(\varepsilon)^2}$.

Robin Boundary Condition

In this case one has ($\alpha = 0$, $\beta = 0$, $\gamma = 1$) and the D_T is obtained taking the limit as,

$$D_T(\hat{x}) = -\lim_{\varepsilon\to 0} \frac{1}{2f'(\varepsilon)} \int_{\partial\Omega\varepsilon} [k(\frac{\partial u_\varepsilon}{\partial t}) - k(\frac{\partial u_\varepsilon}{\partial n}) - 2bu_\varepsilon - \frac{2}{\varepsilon} h_c^\varepsilon (u_\varepsilon - 2u_{\varepsilon\infty})] d\Omega\varepsilon \tag{14}$$

being the regularization function as $f'(\varepsilon) = -2\pi\varepsilon$.

The obtained closed formulae for the D_T are summarized in Table 1, considering the three classical cases of boundary conditions on the holes.

Table 1: Analytical expressions for D_T depending on the b.c. applied on the holes

B. C. Type	$D_T(\widehat{x})$	$\widehat{x}$
Neumann ($\alpha = 0, \beta = 1, \gamma = 0$)	$k\nabla u \nabla u - bu$	$\widehat{x} \in \Omega \cup \Gamma$
	$-q_\varepsilon u$	$\widehat{x} \in \Omega \cup \Gamma$
Dirichlet ($\alpha = 1, \beta = 0, \gamma = 0$)	$-\frac{1}{2}k(u - \overline{u}_\varepsilon)$	$\widehat{x} \in \Omega$
	$k\nabla u \nabla u - b\overline{u}_\varepsilon$	$\widehat{x} \in \Gamma$
Robin ($\alpha = 0, \beta = 0, \gamma = 1$)	$h_c^\varepsilon(u_\varepsilon - 2u_{\varepsilon\infty})$	$\widehat{x} \in \Omega \cup \Gamma$

*It is important to take attention that D_T is evaluated by different expressions for interior and boundary points.
**Topological optimization considering anisotropic media for potential problems were considered in Anflor and Marczak [2009].

2.2 Linear elasticity

The direct problem for elasticity is stated as,

$$\text{Find: } \{u_\epsilon | div\sigma_\varepsilon = b\} \text{ on } \Omega_\varepsilon \tag{15}$$

A general form for the cost function can be written as the total strain energy function:

$$\Psi(u_\tau) = \frac{1}{2}\int_{\Omega_\tau} C\nabla_\tau u_\tau \cdot \nabla_\tau u_\tau d\Omega_\tau - \int b \cdot u_\tau d\Omega_\tau - \int_\Gamma \bar{t} \cdot u_\tau d\tau = \frac{1}{2}a_\tau(u_\tau, u_\tau) - l_\tau(u_\tau), \tag{16}$$

where τ is the perturbation form for the cost function with respect to the shape, C is Hookeâs tensor, b is the body force, $\bar{t}$ is the traction boundary condition, and u_τ denotes the displacement vector field. Equation (17) refers to the sensitivity of the cost function with respect to τ and can be obtained from the *Gâteaux* derivative of the perturbed configuration given by Equation (13):

$$\frac{d}{d\tau}\Psi(\Omega_\tau)_{\tau=0} = \lim_{\tau \to 0} \frac{d}{d\tau}\Psi(\Omega_\tau). \tag{17}$$

In the absence of body forces, the D_T results:

$$D_T(\hat{x}) = -\lim_{\epsilon \to 0} \frac{1}{f'(\varepsilon)} \int_{\Gamma\varepsilon} \frac{1}{2\rho E}\sigma_\varepsilon^{tt} \, d\Gamma_\varepsilon. \tag{18}$$

If the limit of $\varepsilon \to 0$ in Equation 16, eq. 18 results:

$$D_T(\hat{x}) = \frac{1}{2\rho E}[(\sigma_1 + \sigma_2)^2 + 2(\sigma_1 - \sigma_2)^2], \tag{19}$$

where σ_1, σ_2 are the principal stresses of the stress tensor $\sigma|\hat{x}$ computed in $\hat{x} \epsilon \omega$. The principal stresses are given by

$$\sigma_{1,2} = \frac{1}{2}[tr\sigma \pm \sqrt{2\sigma^D\sigma^D}], \tag{20}$$

and σ^D is the deviatoric stress tensor:

$$\sigma^D = \sigma - \frac{1}{2}tr(\sigma)I. \tag{21}$$

Computing σ_1,σ_2 using Equations 20 and 21 and substituting in Equation 19 results in

$$D_T(\hat{x}) = \frac{1}{2\rho E}[4\sigma\sigma - (tr\sigma)^2]). \tag{22}$$

After some algebraic manipulation using Equation (19) and the constitutive relation, the D_T for plane stress problems stands as

$$D_T(\hat{x}) = \frac{2}{1+\nu}\sigma \cdot \varepsilon + \frac{(3\nu - 1)}{2(1-\nu^2)} tr\sigma\, tr\varepsilon \tag{23}$$

In eq.23, σ and ϵ are computed in the original domain, i.e. without voids. For the plane strain, the D_T results as

$$D_T(\hat{x}) = 2(1-\nu)\sigma \cdot \varepsilon + \frac{(1-\nu)(4\nu - 1)}{2(1-2\nu)} tr\sigma\, tr\varepsilon, \tag{24}$$

where ν denotes the Poisson ratio, while $tr\sigma$ and $tr\epsilon$ stand for the trace of the stress and strain tensors, respectively. A complete derivation for obtaining Equations 23 and 24 can be found in Novotny and Sokolowski [2013].

3 Topological Derivative considering the insertion of inclusions

In this section, the mathematical models for the diffusive-convective-reactive problem, Heat Exchanger, Eigenvalue of the Laplace problem, Kirchhoff Plate, Reissner-Mindlin Plate and Compliance. The original unperturbed and topologically perturbed problems are stated as well as the topological derivatives associate the shape functionals we are dealing with, are introduced.

3.1 Diffusive-Convective-Reactive Problem

The mathematical model for the diffusive-convective-reactive problem, as well as the shape functionals we are dealing with, are introduced. The original unperturbed and topologically perturbed problems are stated, together with arguments on the existence of the associated topological derivative (see Carvalho [2020]).

The original unperturbed problem is stated as:

$$u \in H_0^1(\Omega) : \int_\Omega \alpha \nabla u \cdot \nabla \eta + \int_\Omega \beta(\nabla u \cdot V)\eta + \int_\Omega \rho k u \eta = \int_\Omega f\eta \quad \forall \eta \in H_0^1(\Omega), \tag{25}$$

where α, β, ρ and k are positive and bounded functions, f is a distributed source and V is a given vector field, such that, $\operatorname{div}(V) = 0$ in Ω and $V \cdot n = 0$ on $\partial\Omega$. The quantities α, β, ρ, k and f are assumed to be piecewise constant functions as described in Table 2, with $\omega \subset \Omega$. Precise physical meaning of (25) is given in Sections 3.1.2 and 3.1.3.

Table 2: Values of α, β, ρ and f

	α	β	ρ	f
$\Omega \setminus \omega$	α_0	β_0	ρ_0	f_0
ω	α_1	β_1	ρ_1	f_1

In Figure 2 is presented a scheme in which it is possible to remove or add material according to the domain sensitivity.

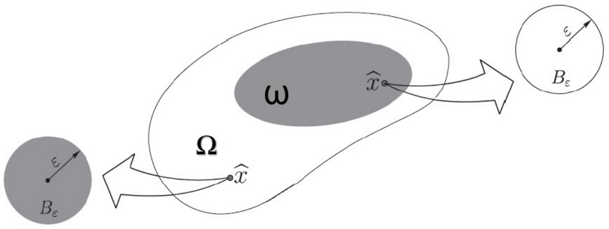


Figure 2: scheme of adding/removal material

The auxiliaries shape functionals are defined by,

$$\mathcal{G}(u) = \int_\Omega \rho k u^2 \qquad \text{and} \qquad \mathcal{J}(u) = \int_\Omega \alpha \|\nabla u\|^2. \tag{26}$$

In order to simplify further analysis, we introduce the adjoint problems

$$q \in H_0^1(\Omega) : \int_\Omega \alpha \nabla q \cdot \nabla \eta - \int_\Omega \beta (\nabla q \cdot V)\eta + \int_\Omega \rho k q \eta = \\ - 2\int_\Omega \rho k u \eta, \quad \forall \eta \in H_0^1(\Omega), \tag{27}$$

$$p \in H_0^1(\Omega) : \int_\Omega \alpha \nabla p \cdot \nabla \eta - \int_\Omega \beta (\nabla p \cdot V)\eta + \int_\Omega \rho k p \eta = \\ - 2\int_\Omega \alpha \nabla u \cdot \nabla \eta, \quad \forall \eta \in H_0^1(\Omega). \tag{28}$$

3.1.1 Perturbed problem

The topological perturbation is defined according to Tables **??** and 4, where $B_\varepsilon(\widehat{x}) = \{\|x - \widehat{x}\| < \varepsilon\}$ for $\widehat{x} \in \Omega$ and $\omega \subset \Omega$. From these elements, the topologically perturbed problem is stated as,

$$u_\varepsilon \in H_0^1(\Omega) : \int_\Omega \alpha_\varepsilon \nabla u_\varepsilon \cdot \nabla \eta + \int_\Omega \beta_\varepsilon (\nabla u_\varepsilon \cdot V)\eta + \int_\Omega \rho_\varepsilon k u_\varepsilon \eta = \int_\Omega f_\varepsilon \eta \quad \forall \eta \in H_0^1(\Omega), \quad (29)$$

with $V \cdot n = 0$ on ∂B_ε. The auxiliary shape functionals in perturbed domain are defined by

$$\mathcal{G}_\varepsilon(u_\varepsilon) = \int_\Omega \rho_\varepsilon k u_\varepsilon^2 \qquad \text{and} \qquad \mathcal{J}_\varepsilon(u_\varepsilon) = \int_\Omega \alpha_\varepsilon \|\nabla u_\varepsilon\|^2. \quad (30)$$

The contrasts of materials in the perturbed domain are shown in the tables 3 and 4.

Table 3: Values of α_ε, β_ε, ρ_ε and f_ε

	α_ε	β_ε	ρ_ε	f_ε
$\Omega \setminus B_\varepsilon$	α	β	ρ	f
B_ε	$\gamma_\alpha \alpha$	$\gamma_\beta \beta$	$\gamma_\rho \rho$	$\gamma_f f$

Table 4: Values of γ_α, γ_β, γ_ρ and γ_f

	γ_α	γ_β	γ_ρ	γ_f
$\Omega \setminus \omega$	α_1/α_0	β_1/β_0	ρ_1/ρ_0	f_1/f_0
ω	α_0/α_1	β_0/β_1	ρ_0/ρ_1	f_0/f_1

Before stating the two main results, let us introduce the following second-order polarization tensors

$$\mathbf{P}_\alpha = \frac{1-\gamma_\alpha}{1+\gamma_\alpha}\mathbf{I} \quad \text{and} \quad \mathbf{P}_{\alpha\beta} = \frac{1-\gamma_\beta}{1+\gamma_\alpha}\mathbf{I}, \quad (31)$$

associated with the contrast on the diffusive γ_α and convective γ_β terms. From the problems presented in (25) and (29) two results are formulated, related to the topological derivative.

Theorem 1 *Let $\mathcal{G}(u)$ be the shape functional defined in* (26)*-left, then its associated topological derivative is given by*

$$D_T\mathcal{G} = -2\alpha \mathbf{P}_\alpha \nabla u \cdot \nabla q - 2\beta(\mathbf{P}_{\alpha\beta}\nabla u \cdot V)q - \rho k(1-\gamma_\rho)u(u+q) + (1-\gamma_f)qf, \quad (32)$$

where q is the adjoint state solution of (27)*.*

Theorem 2 *Let $\mathcal{J}(u)$ be the shape functional presented in* (26)*-right. Then, the topological derivative of $\mathcal{J}$ is given by*

$$D_T\mathcal{J} = -2\alpha \mathbf{P}_\alpha \nabla u \cdot \nabla(u+p) - 2\beta(\mathbf{P}_{\alpha\beta}\nabla u \cdot V)p - \rho k(1-\gamma_\rho)up + (1-\gamma_f)pf, \quad (33)$$

where p is the adjoint solution of problem (28)*.*

3.1.2 Heat Exchanger

We are interested in the diffusion-convection (Eq. (25) with $k = 0$) problem which can be stated as: Find u, such that

$$\left\{ \begin{array}{rcll} -\mathrm{div}(\alpha \nabla u) + \beta(\nabla u \cdot V) & = & f & \text{in} \quad \Omega \,, \\ u & = & 0 & \text{on} \quad \Gamma_D, \\ \partial_n u & = & 0 & \text{on} \quad \Gamma_N. \end{array} \right. \tag{34}$$

Therefore, u represents the temperature field, whereas α is the diffusion coefficient, β is the convection coefficient and V is a given velocity field.

Let us consider the following shape functional

$$\mathcal{F}(u) = \tau \int_\Omega \alpha \|\nabla u\|^2 + (1-\tau) \int_\Omega \rho |u|^2, \tag{35}$$

with $0 \leq \tau \leq 1$ and u solution to (34). Then, its associated topological derivative, by taking into account contrasts on α and ρ (and not on β as well as on f), is given by (see Ruscheinsky et al. [2020b]),

$$D_T \mathcal{F} = -2\alpha \mathbf{P}_\alpha \nabla u \cdot (\tau \nabla u + \nabla p + \nabla q) - (1-\tau)(1-\gamma_\rho)\rho |u|^2, \tag{36}$$

where p and q are respectively solutions of the following adjoint problems

$$p \in \mathcal{U}(\Omega) : \int_\Omega \alpha \nabla p \cdot \nabla \eta - \int_\Omega (\nabla p \cdot V)\eta = -2\tau \int_\Omega \alpha \nabla u \cdot \nabla \eta \quad \forall \eta \in \mathcal{U}(\Omega), \tag{37}$$

$$q \in \mathcal{U}(\Omega) : \int_\Omega \alpha \nabla q \cdot \nabla \eta - \int_\Omega (\nabla q \cdot V)\eta = -2(1-\tau) \int_\Omega \rho u \eta \quad \forall \eta \in \mathcal{U}(\Omega), \tag{38}$$

with the space $\mathcal{U}(\Omega) = \{\varphi \in H^1(\Omega) : \varphi_{|_{\Gamma_D}} = 0\}$.

3.1.3 Eigenvalue of the Laplace problem

The eigenvalue of the Laplace problem modeling a membrane under free vibration can be stated as: Find u and λ, such that

$$\left\{ \begin{array}{rcll} -\mathrm{div}(\alpha \nabla u) & = & \lambda \rho u & \text{in} \quad \Omega, \\ u & = & 0 & \text{on} \quad \partial\Omega, \end{array} \right. \tag{39}$$

so that u represents the transverse displacement field, α is the stiffness coefficient and ρ is the density.

The associated first eigenvalue is defined as

$$\lambda_1 = \frac{\int_\Omega \alpha \|\nabla u\|^2}{\int_\Omega \rho |u|^2}, \tag{40}$$

with u solution of (39). The topological derivative for simple eigenvalues of the Laplacian can be found in Ammari and Khelifi [2003]. The extension to multiple eigenvalues and other types of singular domain perturbations has been derived in Nazarov and Sokolowski [2008]. In particular, the topological derivative of

$$\mathcal{F}(u) = \lambda_1^{-1} \tag{41}$$

is given by:

$$D_T\mathcal{F} = \frac{2\alpha\mathbf{P}_\alpha\nabla u \cdot \nabla u - (1-\gamma_\rho)\rho\lambda_1|u|^2}{\lambda_1^2\int_\Omega \rho|u|^2}, \tag{42}$$

which can be formally derived from Theorems 1 and 2. The rigorous justification for this result can be found in the book by Novotny and Sokolowski [2013]. As observed by Haftka and Gürdal [1992], standard sensitivities of eigenvalues hold only in the case of distinct eigenvalues. According to Seyranian et al. [1994] symmetric and complex structures that depend on many design parameters often present multiple eigenvalues. A numerical method of solution was developed by the authors to determine an ascent direction in the design space for the smallest eigenvalue. More recently, a simple strategy proposed by Zhang et al. [2015] can be used in order to deal with multiplicity of eigenmodes, which consists in select the closest eigenmode to the current one. See also the paper by Torii and Rocha de Faria [2017] for more sophisticated approach based on a smooth p-norm approximation for the smallest eigenvalue.

3.2 Kirchhoff Plates

Before starting the main results of this section, let us introduce the following fourth-order polarization tensor associated with the plate bending model

$$\mathbb{P} = -\frac{1-\gamma_\alpha}{1+\gamma_\alpha\delta_2}\left((1+\delta_2)\mathbb{I} + \frac{1-\gamma_\alpha}{2}\frac{\delta_1-\delta_2}{1+\gamma_\alpha\delta_1}\mathrm{I}\otimes\mathrm{I}\right), \tag{43}$$

where constants δ_1 and δ_2 will be defined later according to the model problem we are dealing with, namely Kirchhoff or Reissner-Mindlin. In (43), the symbols I and $\mathbb{I}$ are used to denote the second and fourth order identity tensors, respectively

The theory of Kirchhoff bending plates is based on the following kinematic assumption:

> *The normal fibers to the middle plane of the plate remain normal during deformation and do not suffer variations in their length. Consequently, both transversal shear and normal deformations are null.*

Therefore, the original unperturbed problem can be stated as: Find $u \in \mathcal{V}(\Omega)$, such that

$$\int_\Omega \alpha M(u)\cdot\nabla\nabla v + \int_\Omega \rho k u v = \int_\Omega f v, \quad \forall v \in \mathcal{V}(\Omega), \tag{44}$$

where $\mathcal{V}(\Omega) = H_0^2(\Omega; R)$. The coefficients α, ρ and f are given in Table 5. In addition, $M(u) = \mathbb{C}\nabla\nabla u$ is the moment tensor, $u : \Omega \mapsto R$ the transverse displacement and k a positive function. The constitutive tensor $\mathbb{C}$ is given by

$$\mathbb{C} = \frac{Eh^3}{12(1-\nu^2)}\left((1-\nu)\mathbb{I} + \nu\mathrm{I}\otimes\mathrm{I}\right), \tag{45}$$

being ν is the Poisson ratio, E is the Young modulus and h the plate thickness. The L^2 and energy norms shape functionals, we are dealing with, are respectively defined as

$$\mathcal{G}(u) = \int_\Omega \rho k|u|^2 \qquad \text{and} \qquad \mathcal{J}(u) = \int_\Omega \alpha M(u)\cdot\nabla\nabla u. \tag{46}$$

In order to simplify bluethe form of the topological derivatives, we introduce the adjoint problems for displacements q and p, as

$$q \in \mathcal{V}(\Omega) : \int_\Omega \alpha M(q) \cdot \nabla\nabla v + \int_\Omega \rho k q v = -2 \int_\Omega \rho k u v, \quad \forall v \in \mathcal{V}(\Omega), \tag{47}$$

$$p \in \mathcal{V}(\Omega) : \int_\Omega \alpha M(p) \cdot \nabla\nabla v + \int_\Omega \rho k p v = -2 \int_\Omega \alpha M(u) \cdot \nabla\nabla v, \quad \forall v \in \mathcal{V}(\Omega). \tag{48}$$

The topologically perturbed counterpart of problem (44) is written as: Find $u_\varepsilon \in \mathcal{V}(\Omega)$, such that

$$\int_\Omega \alpha_\varepsilon M(u_\varepsilon) \cdot \nabla\nabla v + \int_\Omega \rho_\varepsilon k u_\varepsilon v = \int_\Omega f_\varepsilon v, \quad \forall v \in \mathcal{V}(\Omega), \tag{49}$$

where the coefficients α_ε, ρ_ε and f_ε are defined through Table 3 and Table 4. The associated shape functionals are then defined as

$$\mathcal{G}_\varepsilon(u_\varepsilon) = \int_\Omega \rho_\varepsilon k |u_\varepsilon|^2 \qquad \text{and} \qquad \mathcal{J}_\varepsilon(u_\varepsilon) = \int_\Omega \alpha_\varepsilon M(u_\varepsilon) \cdot \nabla\nabla u_\varepsilon. \tag{50}$$

3.2.1 Topological sensitivities

By setting the constants δ_1 and δ_2 in the definition of the polarization tensor (43) as follows

$$\delta_1 = \frac{1+\nu}{1-\nu} \quad \text{and} \quad \delta_2 = \frac{1-\nu}{3+\nu}, \tag{51}$$

we can state the two main results of this Section, whose proofs are completely analogous to the presented by Amstutz and Novotny [2011]:

Theorem 3 *Let $\mathcal{G}(u)$ be the shape functional defined by* (46)*-left, then its associated topological derivative is given by*

$$D_T\mathcal{G} = \alpha \mathbb{P} M(u) \cdot \nabla\nabla q - (1-\gamma_\rho)\rho k u(u+q) + (1-\gamma_f) f q \quad a.e. \ in\, \Omega \tag{52}$$

where q is the adjoint state solution of (47)*.*

Theorem 4 *Let $\mathcal{J}(u)$ be the shape functional presented in* (46)*-right, then its topological derivative is given by*

$$D_T\mathcal{J} = \alpha \mathbb{P} M(u) \cdot \nabla\nabla(u+p) - (1-\gamma_\rho)\rho k u p + (1-\gamma_f) f p \quad a.e. \ in\, \Omega \tag{53}$$

where p is the adjoint solution of problem (48)*.*

The eigenvalue problem for the Kirchhoff model of a clamped thin plate under free vibration can be stated as: Find u and λ, such that

$$\left\{ \begin{array}{rcll} \operatorname{div}\operatorname{div}(\alpha M(u)) & = & \lambda \rho u & \text{in} \quad \Omega, \\ u = \partial_n u & = & 0 & \text{on} \quad \partial\Omega. \end{array} \right. \tag{54}$$

The associated first eigenvalue can be defined as

$$\lambda_1 = \frac{\int_\Omega \alpha M(u) \cdot \nabla\nabla u}{\int_\Omega \rho |u|^2}, \tag{55}$$

being u solution of (54). The topological derivative of $J(\mathcal{D}) := \lambda_1^{-1}$ is given by (see Carvalho et al. [2020]),

$$D_T J = -\frac{\alpha \mathbb{P} M(u) \cdot \nabla\nabla u + (1-\gamma_\rho)\rho\lambda_1 |u|^2}{\lambda_1^2 \int_\Omega \rho |u|^2}. \tag{56}$$

3.3 Reissner-Mindlin Plates

The theory of Reissner-Mindlin bending plates is based on the following kinematic assumption:

> *The normal fibers to the middle plane of the plate remain straight during the deformation process and do not suffer variations in their length, but they do not necessarily remain normal to the middle plane. Consequently, the transversal shear deformations are not negligible and the normal deformations are null.*

Therefore, the unperturbed problem is stated as: Find $(\theta, u) \in \mathcal{H}(\Omega)$, such that

$$\int_\Omega \alpha M(\theta) \cdot (\nabla \eta)^s + \int_\Omega \beta Q(\theta, u) \cdot (\eta - \nabla v) + \int_\Omega \rho k u v = \int_\Omega f v, \quad \forall\, (\eta, v) \in \mathcal{H}(\Omega), \tag{57}$$

where $\mathcal{H}(\Omega) = H_0^1(\Omega; R^2) \times H_0^1(\Omega; R)$. The coefficients α, β, ρ and f are given in Table 2. In addition, $\theta : \Omega \mapsto R^2$ is the rotation, $u : \Omega \mapsto R$ is the transversal displacement, $M(\theta) = \mathbb{C}(\nabla\theta)^s$ is the generalized bending moment tensor and $Q(\theta, u) = D(\theta - \nabla u)$ is the generalized shear tensor. The constitutive tensor $\mathbb{C}$ is defined by (45) whereas the second order tensor D is given by

$$D = \frac{\sigma E h}{2(1+\nu)} \mathrm{I}, \tag{58}$$

with shear correction factor $\sigma = 5/6$. The L^2 and energy norms shape functionals, we are dealing with, are defined as

$$\mathcal{G}(\theta, u) = \int_\Omega \rho k |u|^2 \quad \text{and} \quad \mathcal{J}(\theta, u) = \int_\Omega (\alpha M(\theta) \cdot (\nabla\theta)^s + \beta Q(\theta, u) \cdot (\theta - \nabla u)). \tag{59}$$

In order to simplify bluethe form of the topological derivatives, we introduce the adjoint problems for displacements (q, p) and the rotations (φ, ϕ), as

$$(\varphi, q) \in \mathcal{H}(\Omega) : \int_\Omega \alpha M(\varphi) \cdot (\nabla \eta)^s + \int_\Omega \beta Q(\varphi, q) \cdot (\eta - \nabla v) + \int_\Omega \rho k q v = - 2 \int_\Omega \rho k u v, \quad \forall (\eta, v) \in \mathcal{H}(\Omega), \tag{60}$$

$$\begin{aligned}(\phi, p) \in \mathcal{H}(\Omega) : \int_\Omega \alpha M(\phi) \cdot (\nabla\eta)^s + \int_\Omega \beta Q(\phi, p) \cdot (\eta - \nabla v) + \int_\Omega \rho k p v = \\ - 2\int_\Omega (\alpha M(\theta) \cdot (\nabla\eta)^s + \beta Q(\theta, u) \cdot (\eta - \nabla v)), \quad \forall (\eta, v) \in \mathcal{H}(\Omega). \quad (61)\end{aligned}$$

The topologically perturbed counterpart of problem (57) is written as: Find $(\theta_\varepsilon, u_\varepsilon) \in \mathcal{H}(\Omega)$, such that

$$\int_\Omega \alpha_\varepsilon M(\theta_\varepsilon) \cdot (\nabla\eta)^s + \int_\Omega \beta_\varepsilon Q(\theta_\varepsilon, u_\varepsilon) \cdot (\eta - \nabla v) + \int_\Omega \rho_\varepsilon k u_\varepsilon v = \int_\Omega f_\varepsilon v, \quad \forall\, (\eta, v) \in \mathcal{H}(\Omega), \quad (62)$$

where the coefficients α_ε, β_ε, ρ_ε and f_ε are reported in Tables **??** and 4. The associated shape functionals are then defined as

$$\mathcal{G}_\varepsilon(\theta_\varepsilon, u_\varepsilon) = \int_\Omega \rho_\varepsilon k |u_\varepsilon|^2 \quad \text{and} \quad (63)$$

$$\mathcal{J}_\varepsilon(\theta_\varepsilon, u_\varepsilon) = \int_\Omega (\alpha_\varepsilon M(\theta_\varepsilon) \cdot (\nabla\theta_\varepsilon)^s + \beta_\varepsilon Q(\theta_\varepsilon, u_\varepsilon) \cdot (\theta_\varepsilon - \nabla u_\varepsilon)). \quad (64)$$

3.3.1 Topological sensitivities

Let us introduce the following second-order tensor

$$P = -2\frac{1-\gamma_\beta}{1+\gamma_\beta}\,\mathrm{I}. \quad (65)$$

Now, by setting the constants δ_1 and δ_2 in the definition of the polarization tensor (43) as follows

$$\delta_1 = \frac{1+\nu}{1-\nu} \quad \text{and} \quad \delta_2 = \frac{3-\nu}{1+\nu}, \quad (66)$$

we can state the two main results of this section, whose proofs are completely analogous to the paper by Sales et al. [2015]:

Theorem 5 *Let $\mathcal{G}(\theta, u)$ be the shape functional defined by (59)-left, then its associated topological derivative is given by*

$$\begin{aligned}D_T\mathcal{G} = \alpha\mathbb{P}M(\theta) \cdot (\nabla\varphi)^s + \beta P Q(\theta, u) \cdot (\varphi - \nabla q) \\ - (1-\gamma_\rho)\rho k u(u+q) + (1-\gamma_f) f q \quad a.e.\ in\,\Omega \quad (67)\end{aligned}$$

where (φ, q) is the adjoint state solution of (60).

Theorem 6 *Let $\mathcal{J}(\theta, u)$ be the shape functional presented in (59)-right, then its associated topological derivative is given by*

$$\begin{aligned}D_T\mathcal{J} = \alpha\mathbb{P}M(\theta) \cdot (\nabla(\theta+\phi))^s + \beta P Q(\theta, u) \cdot ((\theta+\phi) - \nabla(u+p)) \\ - (1-\gamma_\rho)\rho k u p + (1-\gamma_f) f p \quad a.e.\ in\,\Omega \quad (68)\end{aligned}$$

where (ϕ, p) is the adjoint solution of problem (61).

the eigenvalue problem of a Reissner-Mindlin model of a clamped thick plate under free vibration can be stated as: Find (θ, u) and λ, such that

$$\left\{ \begin{array}{rcll} -\mathrm{div}(\alpha M(\theta)) + \beta Q(\theta, u) & = & 0 & \text{in} \quad \Omega \, , \\ \mathrm{div}(\beta Q(\theta, u)) & = & \rho \lambda u & \text{in} \quad \Omega \, , \\ \theta = 0, \ u & = & 0 & \text{on} \quad \partial\Omega \, . \end{array} \right. \tag{69}$$

The associated first eigenvalue is defined as

$$\lambda_1 = \frac{\int_\Omega (\alpha M(\theta) \cdot (\nabla\theta)^s + \beta Q(\theta, u) \cdot (\theta - \nabla u))}{\int_\Omega \rho |u|^2}, \tag{70}$$

being (θ, u) solution of (69). The topological derivative of $J(\mathcal{D}) = \lambda_1^{-1}$ is given by (see Carvalho et al. [2020]),

$$D_T J = -\frac{\alpha \mathbb{P} M(\theta) \cdot (\nabla\theta)^s + \beta P Q(\theta, u) \cdot (\theta - \nabla u) + (1 - \gamma_\rho) \rho \lambda_1 |u|^2}{\lambda_1^2 \int_\Omega \rho |u|^2}. \tag{71}$$

3.4 Compliance Problem

The compliance of the plate under bending effects is obtained as the sum of the shape functionals given by (46) for Kirchhoff problem and by (59) for Reissner-Mindlin problem. The zero order term in both problems (see eqs. (44) and (57)) can be interpreted as an elastic support, so that we define the quantity $s = \rho k$, where s represents the stiffness of the support. The transverse load f is assumed to be fixed, so that its associated contrast $\gamma_f = 1$.

In the case of Kirchhoff plate bending problem, the shape functional to be minimized is defined as $J(\mathcal{D}) := \mathcal{J}(u) + \mathcal{G}(u)$, with $\mathcal{J}(u)$ and $\mathcal{G}(u)$ given by (46), where u is the solution to: Find u, such that

$$\left\{ \begin{array}{rcll} \mathrm{div}\,\mathrm{div}(\alpha M(u)) + su & = & f & \text{in} \quad \Omega, \\ u = \partial_n u & = & 0 & \text{on} \quad \partial\Omega. \end{array} \right. \tag{72}$$

Therefore, from Theorem 3 and Theorem 4, we have that the associated topological derivative of the compliance shape functional $J(\mathcal{D})$ is given by (see, Carvalho et al. [2020]),

$$D_T J = -\alpha \mathbb{P} M(u) \cdot \nabla\nabla u + (1 - \gamma_\rho) s |u|^2. \tag{73}$$

Analogously, in the case of Reissner-Mindlin plate bending problem, the shape functional to be minimized is defined as $J(\mathcal{D}) := \mathcal{J}(\theta, u) + \mathcal{G}(\theta, u)$, with $\mathcal{J}(\theta, u)$ and $\mathcal{G}(\theta, u)$ given by (59), where (θ, u) are the solutions to: Find (θ, u), such that

$$\left\{ \begin{array}{rcll} -\mathrm{div}(\alpha M(\theta)) + \beta Q(\theta, u) & = & 0 & \text{in} \quad \Omega \, , \\ \mathrm{div}(\beta Q(\theta, u)) + su & = & f & \text{in} \quad \Omega \, , \\ \theta = 0, \ u & = & 0 & \text{on} \quad \partial\Omega \, . \end{array} \right. \tag{74}$$

Thus, from Theorem 5 and Theorem 6, we have that the associated topological derivative of the compliance shape functional $J(\mathcal{D})$ is given by

$$D_T J = -\alpha \mathbb{P} M(\theta) \cdot (\nabla\theta)^s - \beta P Q(\theta, u) \cdot (\theta - \nabla u) + (1 - \gamma_\rho) s |u|^2. \tag{75}$$

4 Numerical strategy framework

In this section the illustrative scheme presented in fig. 3 shows the numerical strategy implemented for the methodology aforementioned. The steps of implementation can be listed as,

- Step 1 - Generates the geometry, boundary conditions, and set the mechanical properties;
- Step 2 - Solves the direct problem using the chosen numerical solver (FEM, BEM, or other);
- Step 3 - Applies the respective D_T closed formula to the problem under consideration to get the domain's sensitivity;
- Step 4 - Select those points with low efficiency (low D_T) for being removed. Remark: The user must set the percentage of volume to be removed per iteration;
- Step 5 - Applies an auxiliary routine for material removal at the candidate internal points. Remark: This routine is based on pure geometry depending on the numerical method employed and must be able to detect new frontiers reapplying the b.c. as well as to detect the possible detached material (islands) from the main domain (mainly for BEM).

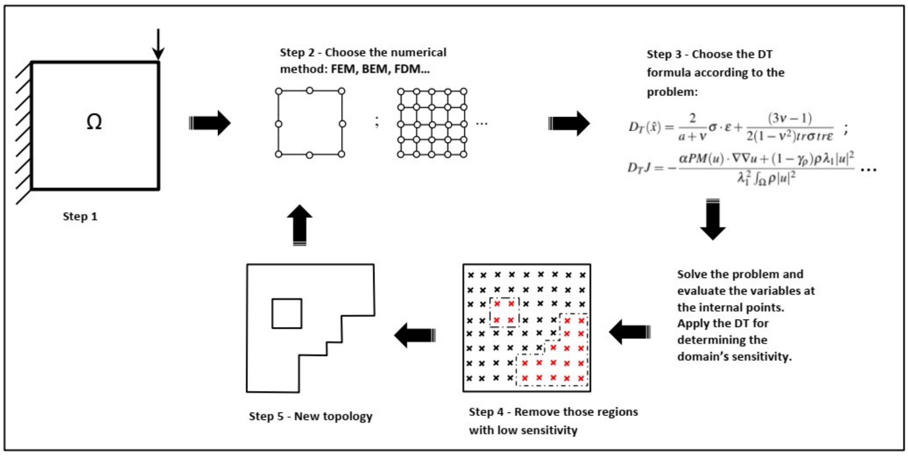

Ω
Step 1
Step 2 - Choose the numerical method: FEM, BEM, FDM...
;
...
Step 3 - Choose the DT formula according to the problem:
$D_T(\hat{x}) = \frac{2}{a+v}\sigma\cdot\varepsilon + \frac{(3v-1)}{2(1-v^2)tr\sigma tr\varepsilon}$;
$D_TJ = -\frac{\alpha PM(u)\cdot\nabla\nabla u + (1-\gamma_\rho)\rho\lambda_1|u|^2}{\lambda_1^2\int_\Omega\rho|u|^2}$ ...
Solve the problem and evaluate the variables at the internal points. Apply the DT for determining the domain's sensitivity.
Step 4 - Remove those regions with low sensitivity
Step 5 - New topology

Figure 3: Numerical methodology scheme for implementation

It is important to highlight that the strategy of material removal depends on the numerical approach and the methodology employed by the user to deal with the geometry. In the case of FEM, generally, one can set the domain fixed and suppress (in case of voids) or impose different values of mechanical properties (in case of inclusion insertion) to those elements with low sensitivities. In this kind of approach, no concerns with the boundary conditions or even islands arising up are needed, because the elements are not rearranged, see for instance the work of Ruscheinsky et al. [2020a]. It is also important to reinforce that the concept of D_T is derived considering the singular perturbation as voids or inclusions. In this sense, one must have attention to using the appropriate D_T closed formula to implement the strategy of removal/add material accordingly. When considering the BEM, one can deal with a fixed domain or moving frontiers. The first strategy is similar to the FEM procedure but in this case the technique of multiple regions must be considered, as implemented in Anflor et al. [2014]. For the moving frontiers, special treatment must be given in attention to the new geometry resulted from the previous iteration. When considering moving frontiers, the material is removed and new elements are added to redesign the domain needing the rearrangement on the discretization process. At this point, islands may arise (fig. 4 and the subroutine developed for Step 5, must be able to detect and discard them.

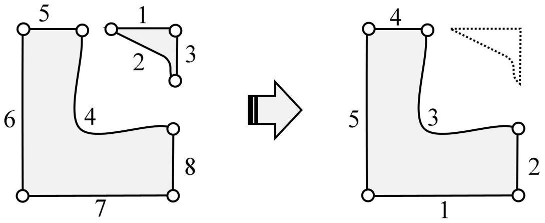


Figure 4: Detail of island detection, deletion and the renumbering of the elements

Additionally, several resources such as offset of internal points (switch off the entire grid of internal points), shape and size of stamps used to remove material are examples of strategies to improve in fast and efficiency of the iterative process when using BEM (fig.5). Further details about these strategies can be consulted in Marczak [2008] and Anflor et al. [2018].

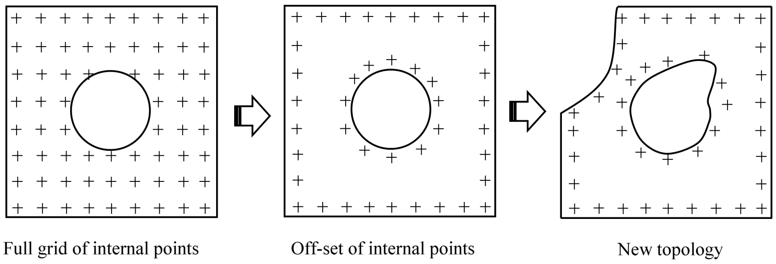


Figure 5: Strategies based on the BEM particularities

5 Numerical examples

This section presents some numerical examples in the context of topological optimization for classical problems of engineering, such as: Potential Problem, Linear Elasticity, Heat Exchanger, Maximization of the First Eigenvalue in Membrane, Plates (Kirchhoff and Reissner-Mindlin) and the Compliance for Kirchhoff and Reissner-Mindlin problem. The algorithm based on the D_T is implemented using BEM for the first two examples and FEM for the remaining examples. The resulting analytical formulae (see formulas (36), (42), (56),(71), (73), (75)) are used together with a level-set domain representation method to devise a simple topology design algorithm (for more details see Amstutz and Andrä [2006]). The obtained final topologies show the efficiency of the topological derivative method.

5.1 Potential Problem: Printed Circuit Board

This example concerns to a printed circuit board (PCB) substrate. The characteristics of of good PCB designs is the efficiency to dissipate the maximum amount of thermal energy with the minimum possible volume of material. In this sense the topology optimization becomes attractive for this class of problem. Figure 6 introduces the initial layout for this case, where four heat sources are used to simulate the heat generated by major electronic components mounted on the PCB. The hatched areas are not allowed changes because they are used for clamping the PCB. The domain is discretized with 32 linear boundary elements (BEs) and the holes opened during the optimization process with 6 linear BEs. All the cavities opened as the iterative process evolves have prescribed Neumann homogeneous as boundary condition. For this problem the optimization procedure is halted was halted when a volume ratio of 70% between the final and the original designs is achieved. The domain's sensitivity was computed using the first equation introduced in Table 1. The evolution history is introduced according Fig. 7. It is worth mentioning that in the PCB case, new cavities are created during the process near the corners, characterizing truly topological changes in the domain.

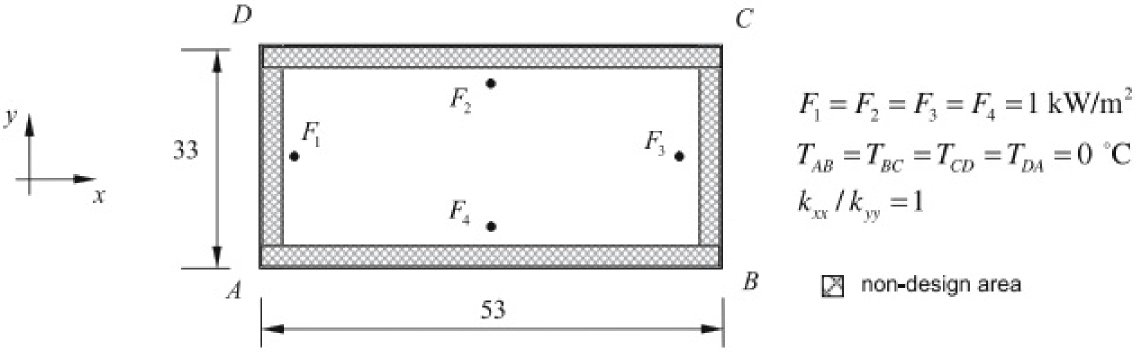
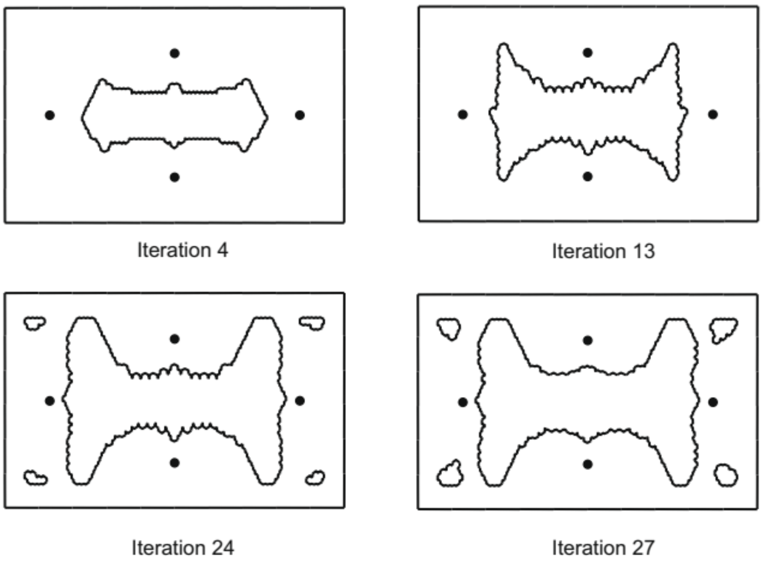

D
C
y
x
33
F_1
F_2
F_3
F_4
A
B
53
$F_1 = F_2 = F_3 = F_4 = 1$ kW/m^2
$T_{AB} = T_{BC} = T_{CD} = T_{DA} = 0$ °C
$k_{xx} / k_{yy} = 1$
▨ non-design area

Figure 6: Initial design for the PCB



Iteration 4

Iteration 13

Iteration 24

Iteration 27

Figure 7: Topology evolution for the PCB

5.2 Linear Elasticity

This example consists of a traditional beam, as shown in 8. A rectangle with dimensions of 5 units x 10 units is subjected to a total vertical load P = 1 kN applied at the middle of the bottom side. The first and the last element of the bottom side are pinned and bolted, respectively. The radius of the holes was set to 0.013125a. The stop criterion was set as the final volume reaching approximately 54% of the initial volume and the domain's sensitivity was computed using eq.23. The percentages of internal points selected to be removed during the optimization procedure are presented in Table 1. As can be seen, the amount of material removed during the iterative process is variable, based on the domain's sensitivity. As a comparison, the amount of material removed using linear and quadratic BEs is also presented. Figure 6 shows the evolution of the iterative process using quadratic elements, where the final topology results after only six iterations. The final topology results for linear and quadratic BEs are quite similar, as expected. The main difference is that with increasing accuracy of the BE solutions, the domain sensitivity isolines become more evident, allowing the removal of a greater amount of material per iteration.

It is important to highlight that the final topology resulted in the shape of a truss structure, as shown in Figure 9. Using a mirror-image effect procedure on this final topology (iteration 6), the result is a geometry similar to a wheel with radial supports (Figure 11). Based on the literature, a resulting shape of a wheel ensures that the developed optimization routine is capable of generating feasible topologies. In Figure 10 it is possible to observe the amount of material being removed, taking into account the linear and quadratic BEs, as the iterative process evolves.

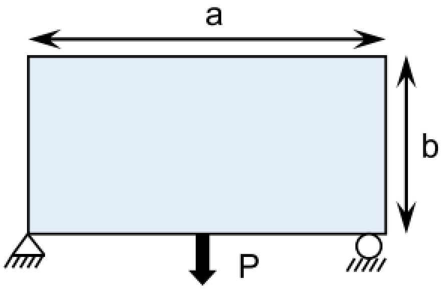


Figure 8: Beam boundary conditions

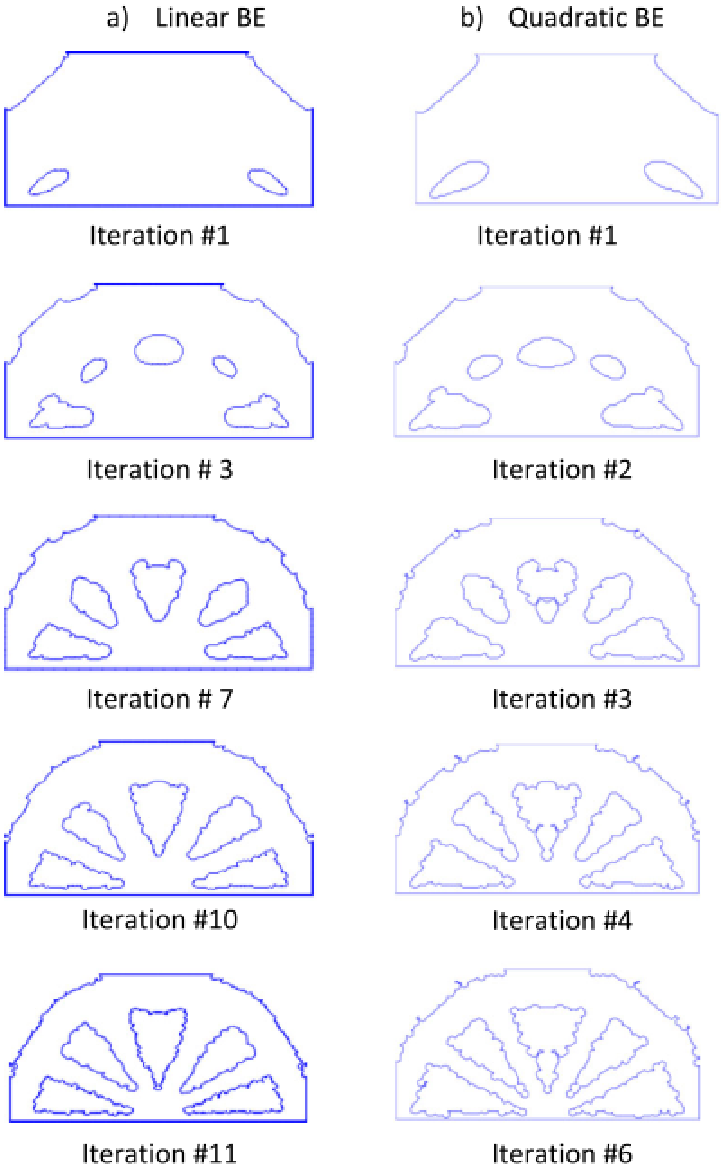


Figure 9: Beam topology evolution: a) linear and b) quadratic boundary elements

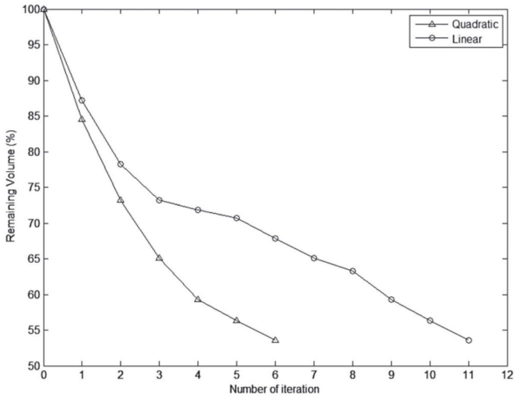


Figure 10: Volume x number of iterations history

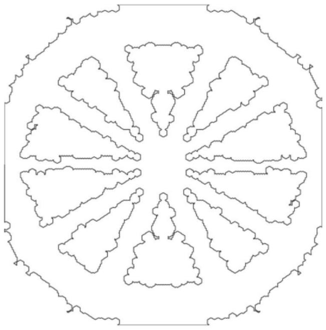


Figure 11: Final topology after a mirror-image effect

5.3 Heat Exchanger Design

The hold-all domain $\mathcal{D}$ is given by a unit square of size $(0,1) \times (0,1)$ with a distributed uniform heat generation of intensity $f = 10^4\,W$ over the domain. All the boundary are thermally insulated, with exception of the regions T_L and T_H of lengths 0.2. The temperature at T_L is prescribed as $u = 273\,K$ and T_H is prescribed as $u = 373\,K$. Fig. 12 shows the initial domain and the initial temperature map. The penalty parameter is set as $\mu = 4$, the weight as $t = 1$ and $\alpha = 1$. During the optimization procedure two material are used, the first one is the aluminum ($\alpha = 205W/mK$) and the second one is a material with low thermal conductivity $\gamma_\alpha \ll \alpha$. The initial domain consists of aluminum only ($\Omega = \mathcal{D}$). As the optimization process iterativelly evolves the aluminum is replaced by the second material. The domain's sensitivity is computed according to eq.36. Figures 13a-d show the evolution of topologies in the j^{th} iteration. In the j=56 iteration, we have the optimized topology. Figure 14, illustrates the shape function. The final topology (see Fig. 13(d)) has 60% volume of high thermal conductivity material.

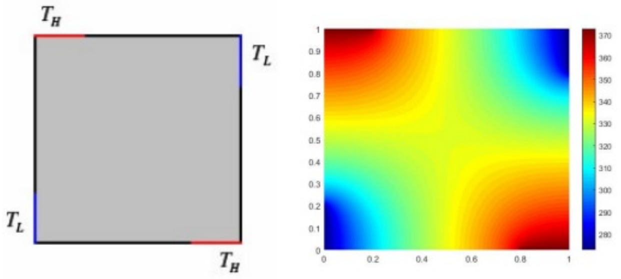
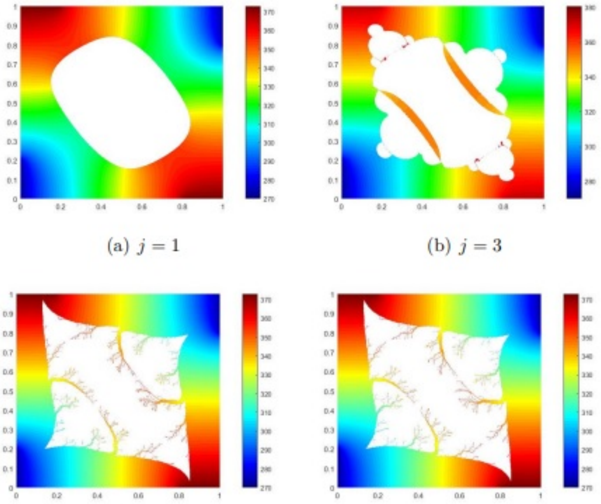


Figure 12: initial domain (left) and the initial temperature map (right)



(a) $j = 1$

(b) $j = 3$

(c) $j = 30$

(d) $j = 56$

Figure 13: Topologies evolution (j^{th}) iteration (a)-(d) and Final topology (d)

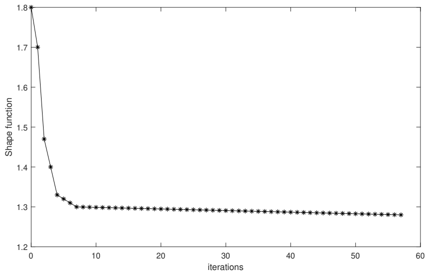


Figure 14: Shape function history

5.4 Membrane Problem: First eigenvalue maximization

The membrane is clamped in the four vertices and free in the rest of the contour. The non-structural concentrated mass m is applied at the plate's center $(0.5, 0.5)$, as depicted in Figure 15. Four cases are considered, namely, cases M1, M2, M3 and M4. The values of the penalty parameters μ and the non-structural mass m as depicted in table 5. The domain's sensitivity is computed according to eq. 42.

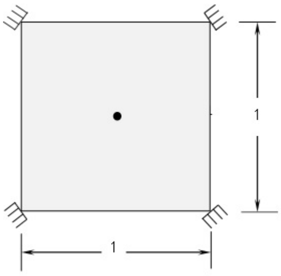
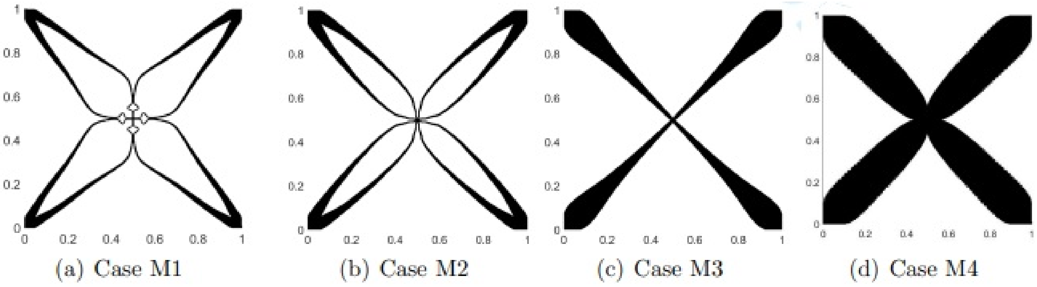


Figure 15: Initial domain

Table 5: Penalty values and concentrated mass

	Case M1	Case M2	Case M3	Case M4
μ	0.4	0.2	0.1	0.4
m	0.02	0.03	0.04	0.7

The final topologies for each case are presented in Figures 16a-d. Fig. 17 introduces the normalized first eigenvalue history λ_1/λ_0 (where λ_0^1 is its initial value) as the iterative process has evolved. The normalized first eigenvalues history λ_1/λ_2 are introduced in figure 18. Note that they are completely separated, so that multiple eigenvalues phenomenon was not observed in this particular example. The evolution histories for the volume fraction and shape funcion are presented in Fig. 19 and 20, respectively. The initial domain is discretized by using linear triangular finite elements resulting in an initial uniform mesh with 10,000 elements and 5,101 nodes. In order to increase the accuracy as well as the topology smoothness 4 steps of mesh refinement during the iterative process are allowed. After the fourth refinement the resulting mesh presents 2,560,000 elements and 1,281,601 nodes.



(a) Case M1 (b) Case M2 (c) Case M3 (d) Case M4

Figure 16: Optimized topologies for Cases M1, M2, M3 and M4

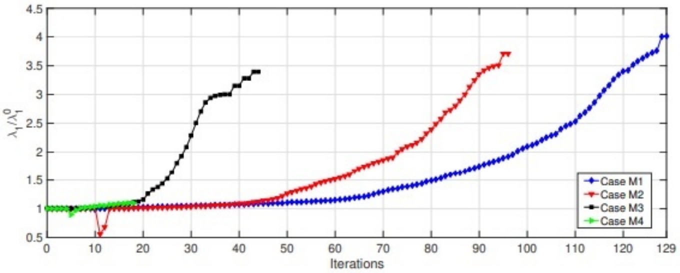


Figure 17: Normalized first eigenvalue λ_1/λ_1^0 history

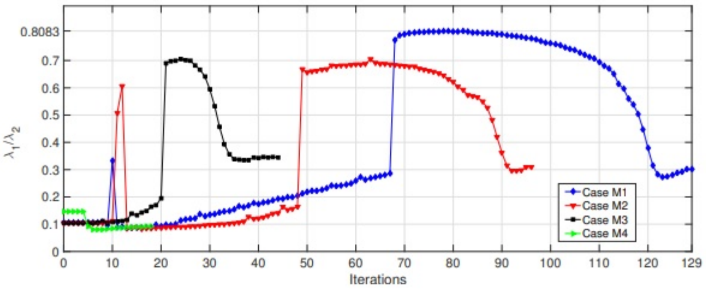


Figure 18: Normalized first eigenvalue λ_1/λ_2 history

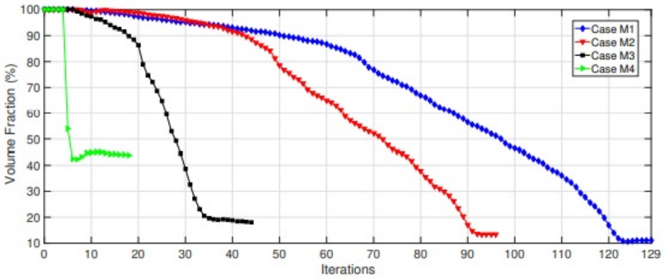


Figure 19: Shape Function history

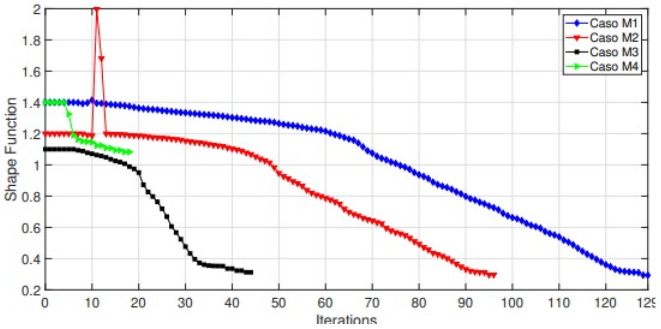


Figure 20: Volume Fraction history

5.5 Kirchhoff and Reissner-Mindlin Plates: First Eigenvalue Maximization

For the eigenvalue problem we will also consider for both problems (Kirchhoff and Reissner-Mindlin) a hold-all domain Ω given by a clamped square on the left and right sides and simply supported on the top and bottom sides of dimensions $(0,1) \times (0,1)\text{m}^2$. The non-structural concentrated mass is represented by black dot (see Figure 25). The Young modulus is $E = 210\text{GPa}$, Poisson ratio $\nu = 0.3$ and the plate thickness is $h = 0.05\text{m}$. The contrasts are given by $\gamma_\alpha = \gamma_\rho = 10^{-3}$ and the penalty parameter is set as $\mu = 1.2$. The domain's sensitivity is computed using eq. 56 for Kirchoff plate and eq. 71 for Reissner-Mindlin plate. The experiments are labeled as Cases E1 and E2 for Reissner-Mindlin and Kirchhoff plates, respectively. The final topologies are presented in Fig. 22(a)-(b). Finally, the history of the normalized first eigenvalue λ^1/λ_0^1 (with $\lambda_0^1 = 311.38$ and $\lambda_0^1 = 286.52$ for Kirchhoff and Reissner-Mindlin cases, respectively), volume fraction and shape function obtained during the iterative process are presented in Fig. 23 to Fig. 24. The domain is discretized by using linear triangular finite elements resulting in an initial uniform mesh with $10,000$ elements and $5,101$ nodes. In order to increase the accuracy as well as the topology smoothness 3 steps of uniform mesh refinement during the iterative process are allowed, leading to a mesh with $640,000$ elements and $320,801$ nodes.

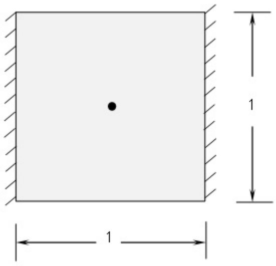
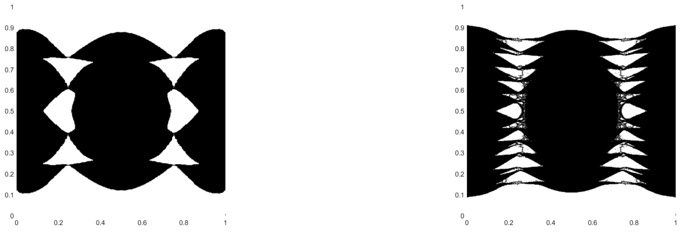


Figure 21: Initial domain for the Kirchhoff and Reissner-Mindlin plates

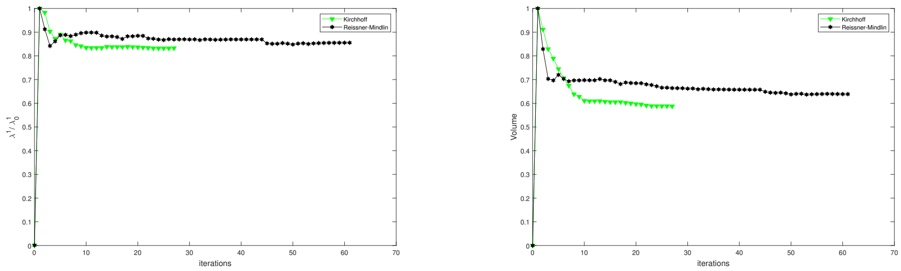
In Reissner-Mindlin case (Fig. 22(b)) it is observed the presence of small structures due to the numerical artefacts. The mesh refinement isn't enough to overcome this issue even if a higher mesh resolution is imposed. It is well-know that there is a lack of sufficient optimality conditions for such shape optimization problems, so that thin components like those pointed out may appear. In spite of the presence of small structures a local minimum has been reached up to a small numerical tolerance.



(a) Kirchhoff

(b) Reissner-Mindlin

Figure 22: Final Topologies



(a) Kirchhoff

(b) Reissner-Mindlin

Figure 23: Eigenvalue λ^1/λ^1_0 (a) and Volume Fraction history (b)

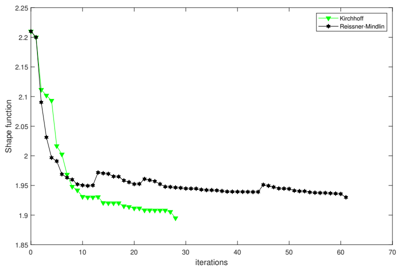


Figure 24: Shape Function history

5.6 Compliance Minimization

In the numerical experiment we consider for both problems (Kirchhoff and Reissner-Mindlin) a hold-all domain Ω given by a clamped square of dimensions $(0,1)\times(0,1)\text{m}^2$ submitted to concentrated forces, perpendicular to the plane of the plate, of values $f=-1\text{MN}$ located at the centre of plate. A circular elastic support of radius 0.2m and center at $(0.50,0.50)$ is also considered (see sketch in Fig. 25(a)-(b). The concentrated loads is represented by black dot whereas the support is represented by a hatched circular area in grey color. The Young modulus is $E=210\text{GPa}$, Poisson ratio $\nu=0.3$, the stiffness of the elastic support is $s=10^{-2}E$ and the plate thickness is $h=0.05\text{m}$. The contrasts are given by $\gamma_\alpha=\gamma_\rho=10^{-4}$ and the penalty parameter is set as $\mu=1.7$. The domain's sensitivity is computed using eq. 73 for Kirchoff plate and eq. 75 for Reissner-Mindlin plate. The experiments are labeled as Cases C1 and C3 for Reissner-Mindlin with and without support, respectively and Cases C2 and C4 for Kirchhoff with and without support, respectively. The final topologies are presented in Fig. 26(b)-(d) and Fig. 26(a)-(c) for Kirchhoff (Cases C2 and C4) and Reissner-Mindlin (Cases C1 and C3) plates, respectively. Finally, the history of the compliance, volume fraction and shape function obtained during the iterative process are presented in Fig. 27 to Fig. 29.

In addition, the domain is discretized by using linear triangular finite elements resulting in an initial uniform mesh with $14,400$ elements and $7,321$ nodes. In order to increase the accuracy as well as the topology smoothness 3 steps of uniform mesh refinement during the iterative process are allowed, leading to a mesh with $921,600$ elements and $461,761$ nodes.

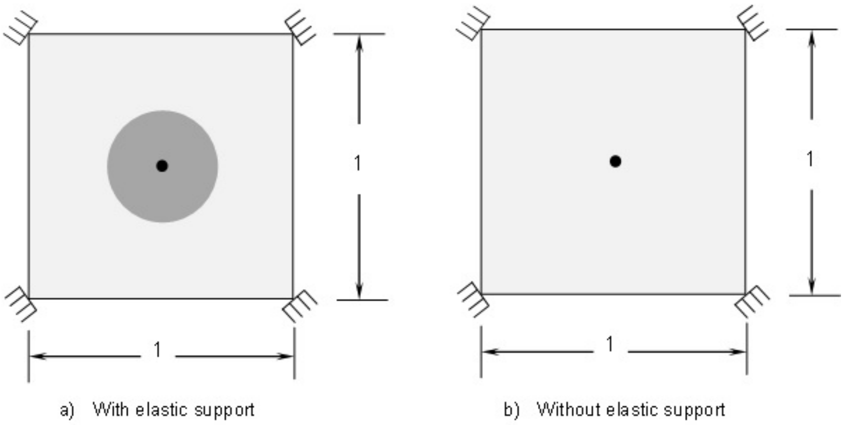
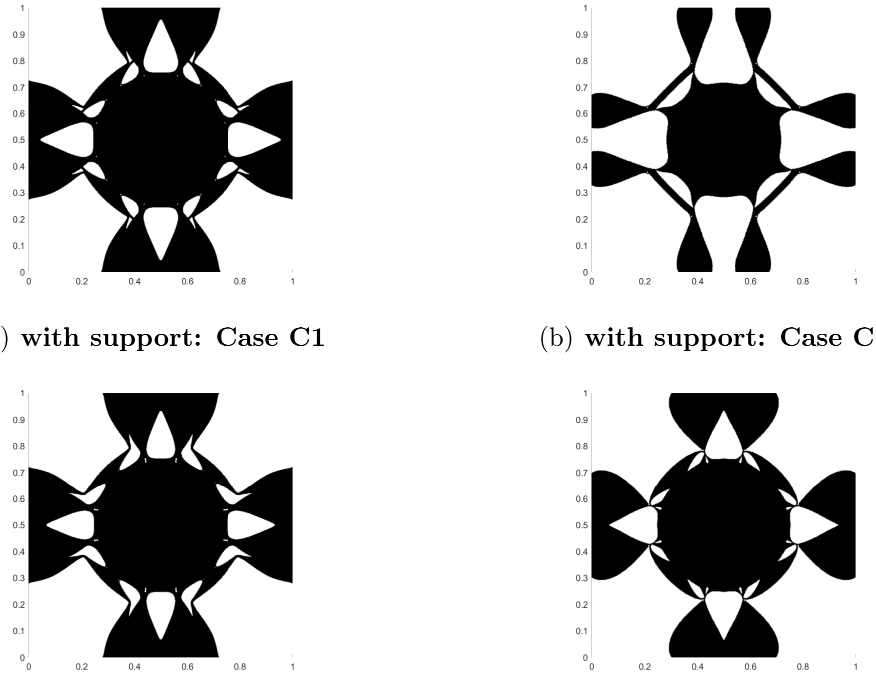


Figure 25: Initial domain with support (a) and without support (b). The concentrated loads are represented by black dots whereas the elastic support is represented by a hatched circular area in grey color



(a) **with support: Case C1**

(b) **with support: Case C2**

(c) **without support: Case C3**

(d) **without support: Case C4**

Figure 26: Final Topologies

6 Final remarks

In this chapter the D_T concept was introduced for classic problems of topology optimization.This methodology can be also extended to other applications as inverse problems and image processing. The D_T measures the sensitive of a domain when a singular perturbation is inserted inside the domain. According to this statement it is possible to glimpse that this methodology becomes suitable for detecting the presence of damage in structures. The presence of holes, cracks, or inclusions are

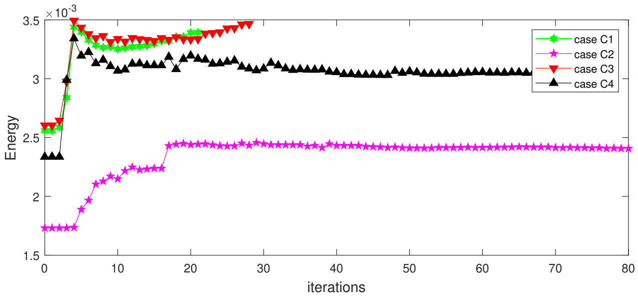


Figure 27: Compliance history

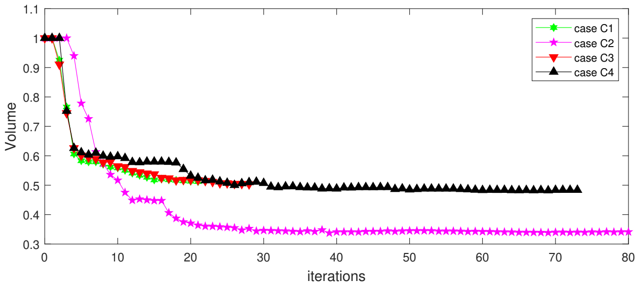


Figure 28: Volume Fraction history

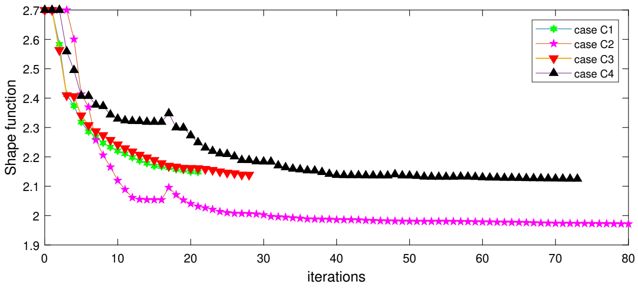


Figure 29: Shape Function history

typical examples of damages that can be evaluated by employing the appropriate D_T closed formulae to map those problematic regions. Based on this approach, the inverse problem can also be addressed by coupling the appropriate topological derivative to probabilistic optimizations methods. Furthermore, there are no constraint restrictions in the use of the present methodology with numerical methods for the direct problem, such as the finite element method, the boundary element method, or any other numerical method used for the discretization of the quantity of interest in the domain.

7 References

H. Ammari and A. Khelifi. Electromagnetic scattering by small dielectric inhomogeneities. *ournal de Mathématiques Pures et Appliquées*, 82:749–842, 2003.

S. Amstutz and H. Andrä. A new algorithm for topology optimization using a level-set method. *Journal of Computational Physics*, 216(2):573–588, 2006.

S. Amstutz and A. A. Novotny. Topological optimization of structures subject to von mises stress constraints. *Structural and Multidisciplinary Optimization*, 41 (3):407–420, 2010.

S. Amstutz and A. A. Novotny. Topological asymptotic analysis of the Kirchhoff plate bending problem. *ESAIM Control. Optim. Calc. Var.*, 17(3):705–721, 2011. doi: DOI:10.1051/cocv/2010010.

S. Amstutz, I. Horchani, and M. Masmoudi. Crack detection by the topological gradient method. *Control and Cybernetics*, 34(1):81–101, 2005.

C. Anflor, E. Albuquerque, and L. Wrobel. A topological optimization procedure applied to multiple region problems with embedded sources. *International Journal of Heat and Mass Transfer*, 78:121–129, 2014.

C. Anflor, K. Teotônio, and J. Goulart. Structural optimization using the boundary element method and topological derivative applied to a suspension trailing arm. *Engineering Optimization*, 50(10):1662–1680, 2018.

C. T. Anflor and R. J. Marczak. A boundary element approach for topology design in diffusive problems containing heat sources. *International Journal of Heat and Mass Transfer*, 52(19):4604–4611, 2009. ISSN 0017-9310. doi: https://doi.org/10.1016/j.ijheatmasstransfer.2009.02.048. URL `https://www.sciencedirect.com/science/article/pii/S0017931009002580`.

C. T. M. Anflor. *Otimização evolucionária e topológica em problemas governados pela equação de Poisson empregando o método dos elementos de contorno*. PhD Thesis - Federal University of Rio Grande do Sul, 2007.

C. Bertsch, A. P. Cisilino, S. Langer, and S. Reese. Topology Optimization of 3D Elastic Structures Using Boundary Elements. *PAMM*, 8(1):10771–10772, dec 2008. ISSN 16177061. doi: 10.1002/pamm.200810771. URL `http://doi.wiley.com/10.1002/pamm.200810771`.

A. Carpio and M. Rapún. Solving inhomogeneous inverse problems by topological derivative methods. *Inverse Problems*, 24(4):045014, 2008.

F. S. Carvalho. *Análise de sensibilidade topológica aplicada em problemas elípticos*. PhD Thesis - University of Brasilia, 2020.

F. S. Carvalho, D. Ruscheinsky, S. Giusti, C. Anflor, and A. Novotny. Topological derivative-based topology optimization of plate structures under bending effects. *Structural and Multidisciplinary Optimization*, pages 1–14, 2020.

J. Céa, S. Garreau, P. Guillaume, and M. Masmoudi. The shape and topological optimizations connection. *Computer methods in applied mechanics and engineering*, 188(4):713–726, 2000.

L. Comino, R. Gallego, and G. Rus. Combining topological sensitivity and genetic algorithms for identification inverse problems in anisotropic materials. *Computational Mechanics*, 41(2):231–242, 2008.

A. da Silva and A. Novotny. Damage identification in plate structures based on the topological derivative method. *Structural and Multidisciplinary Optimization*, 65 (1):1–12, 2022.

S. Garreau, P. Guillaume, and M. Masmoudi. The topological gradient. Technical report, Université Paul Sabatier - Toulouse 3, France, 1998.

S. Garreau, P. Guillaume, and M. Masmoudi. The topological asymptotic for PDE systems: the elasticity case. *SIAM journal on control and optimization*, 39(6): 1756–1778, 2001.

R. T. Haftka and Z. Gürdal. *Elements of structural optimization.* Kluwer, Dordrecht, third edition, 1992.

M. Hintermüller and A. Laurain. Multiphase image segmentation and modulation recovery based on shape and topological sensitivity. *Journal of Mathematical Imaging and Vision*, 35(1):1–22, 2009.

I. Larrabide, R. Feijóo, A. Novotny, and E. Taroco. Topological derivative: a tool for image processing. *Computers & Structures*, 86(13-14):1386–1403, 2008.

Marczak. Optimization of elastic structures using boundary elements and a topological-shape sensitivity formulation. *Latin American Journal of Solids and Structures*, pages 99–117, 2008.

A. Martinez Dominguez, A. Gümes Gordo, J. M. Perales Perales, and J. Vega de Prada. Topological derivative methods for damage detection. 2018.

S. Nazarov and J. Sokolowski. Shape sensitivity analysis of eigenvalues revisited. *Control and Cybernetics*, 37(4):999–1012, 2008.

A. Novotny, R. Feijóo, E. Taroco, and C. Padra. Topological sensitivity analysis for three-dimensional linear elasticity problem. *Computer Methods in Applied Mechanics and Engineering*, 196(41-44):4354–4364, 2007.

A. A. Novotny and J. Sokolowski. *Topological derivatives in shape optimization. Interaction of Mechanics and Mathematics.* Springer, Berlin, 2013.

A. A. Novotny, R. A. Feijóo, E. Taroco, and C. Padra. Topological sensitivity analysis. *Comput. Methods Appl. Mech. Eng.*, 2003. ISSN 00457825. doi: 10.1016/S0045-7825(02)00599-6.

S. Rocha and A. Novotny. Obstacles reconstruction from partial boundary measurements based on the topological derivative concept. *Structural and Multidisciplinary Optimization*, 55(6):2131–2141, 2017.

D. Ruscheinsky, F. Carvalho, C. Anflor, and A. A. Novotny. Topological asymptotic analysis of a diffusive–convective–reactive problem. *Engineering Computations*, 2020a.

D. Ruscheinsky, F. S. Carvalho, C. Anflor, and A. A. Novotny. Topological asymptotic analysis of a diffusiveâconvectiveâreactive problem. *Engineering Computations*, 2020b.

V. Sales, A. A. Novotny, and J. E. Muñoz Rivera. Energy change to insertion of inclusions associated with the Reissner-Mindlin plate bending model. *Int. J. Solids Struct.*, 59:132–139, may 2015. ISSN 00207683. doi: 10.1016/j.ijsolstr.2015.01.019.

A. Schumacher. Topologieoptimierung von bauteilstrukturen unter verwendung von lochpositionierungskriterien [phd thesis]. *Forschungszentrum für Multidisziplinäre Analysen und Angewandte Strukturoptimierung. Institut für Mechanik und Regelungstechnik*, 1996.

A. P. Seyranian, E. Lund, and N. Olhoff. Multiple eigenvalues in structural optimization problems. *Structural optimization*, 8(4):207–227, 1994.

J. Sokołowski and A. Żochowski. On topological derivative in shape optimization. Technical report, INRIA-Lorraine, France, 1997.

J. Sokolowski and A. Zochowski. Topological derivatives for elliptic problems. *Inverse Probl.*, 15(1):123–134, feb 1999. ISSN 0266-5611. doi: 10.1088/0266-5611/15/1/016. URL `http://stacks.iop.org/0266-5611/15/i=1/a=016?key=crossref.f22a4f973d8108972926691f51b74278`.

A. J. Torii and J. Rocha de Faria. Structural optimization considering smallest magnitude eigenvalues: a smooth approximation. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 39(5):1745–1754, 2017.

Z. Zhang, W. Chen, and X. Cheng. Sensitivity analysis and optimization of eigenmode localization in continuum systems. *Structural and Multidisciplinary Optimization*, 52:305–317, 2015.

Chapter 8: Ultrasound Obstacle Identification using the Boundary Element and Topological Derivative Methods

Chapter details

Chapter DOI:

https://doi.org/10.4322/978-65-86503-71-5.c08

Chapter suggested citation / reference style:

Cisilino, Adrián P., Anflor, Carla T. M. (2022). “Ultrasound Obstacle Identification using the Boundary Element and Topological Derivative Methods”. *In* Jorge, Ariosto B., et al. (Eds.) *Model-Based and Signal-Based Inverse Methods*, Vol. I, UnB, Brasilia, DF, Brazil, pp. 301–330. Book series in Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity.

P.S.: DOI may be included at the end of citation, for completeness.

Book details

Book: Model-based and Signal-Based Inverse Methods

Edited by: Jorge, Ariosto B., Anflor, Carla T. M., Gomes, Guilherme F., & Carneiro, Sergio H. S.

Volume I of Book Series in:

Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity

Published by: UnB City: Brasilia, DF, Brazil Year: 2022

DOI: https://doi.org/10.4322/978-65-86503-71-5

Ultrasound Obstacle Identification using the Boundary Element and Topological Derivative Methods

Adrián P. Cisilino[1*] and Carla Tatiana Mota Anflor[2]

[1*]Department of Mechanical Engineering & INTEMA - National University of Mar del Plata & CONICET, Mar del Plata, Argentina. E-mail: cisilino@fi.mdp.edu.ar

[2]Post-Graduate Program - Integrity of Engineering Materials, University of Brasilia, Brazil. E-mail: anflor@unb.br

[*]Corresponding author

Abstract

This chapter gathers together the key concepts of acoustic scattering and the inverse identification problems, the explicit expressions of the topological derivative for sound-hard and penetrable obstacles, and the key features of to solve the inverse identification problem numerically using the Boundary Element Method. Special attention is given to the implementation for the cases in which the search area has pre-existent obstacles, either sound-hard, penetrable or combination of both. Thus, a general BEM implementation framework is provided, which might be used by the reader to develop its own specific application. Several examples for sound-hard obstacles are used to demonstrate the performance of the method.

1 Introduction

Inverse scattering problems find practical applications in the detection and imaging of objects embedded in continuous media, such as the case of ultrasound in medicine, reflection of seismic waves in oil prospecting and crack detection in structural mechanics.

The standard procedure to solve inverse scattering problems consists in emitting waves that interact with the objects and measuring these waves at receptor locations. The total field, consisting of emitted, scattered and transmitted waves, solves a partial differential equation with boundary conditions at the interface between the obstacles and the medium. The inverse problem is stated as follows: knowing the emitted waves and the measured patterns, find the obstacles for which the solution of the corresponding boundary value problem agrees with the measured at the receptors. Detecting devices like radars, sonars, lidars and scanners follow this general procedure.

The main features of the inverse scattering problem are nonlinearity and ill-posedness. It is nonlinear since the solution to the scattering problem depends nonlinearly on the boundary of the obstacle, and it is ill-posed because given a number of arbitrary measurements at the receptors, a solution for the obstacle boundary may not exist, and if it exists, it may not depend continuously on the parameters that define the obstacle boundary.

Colton and Kress [2013, 2018] provide a review about the historical evolution and the methods to solve inverse scattering problems. The first attempts to solve the inverse obstacle problem dealt with the non-linearity issue by linearizing the problem with the aid of the physical optics approximation. Among others, this approach presents the drawback of being valid only for large wave numbers. Solutions of the full nonlinear inverse obstacle scattering problem can be obtained by means of the so called decomposition methods. These methods break up the problem into two parts: the first part deals with the ill-posedness by constructing the scattered wave from its far field pattern and the second part deals with the nonlinearity by determining the unknown boundary of the obstacle as the location where the boundary condition for the total field is satisfied. However, these methods face the difficulty that in the first step the domain of definition of the scattered wave is not known. Hence, mathematically satisfying formulations of decomposition methods need to combine both parts into an optimization reformulation of the inverse scattering problem.

More modern methods pose the inverse problem as a constrained optimization problem where the 'design variable' is the domain wherein the scattering problem is defined. A cost functional is constructed which quantifies the mismatch between the measured scattering pattern and the scattering pattern corresponding to the current approximation of the shape of the object. The problem is then solved using iterative shape optimization methods, which require, for each step, the computation of the cost function and its derivative with respect to the geometrical parameters that define the boundary of the obstacle, see for example Feijóo [2004] and Feijóo et al. [2004]. This approach is effective, but it does not allow for topological changes in the obstacle, *i.e.*, the number of obstacles has to be known from the beginning. This problem was solved by introducing deformations inspired by level-set methods, which allow creating and destroying boundaries during the iterative process. The paper by Dorn and Lesselier [2006] provides a review on the use of level-set methods for inverse scattering. Nevertheless, level-set based iterative methods may be rather slow unless a good initial guess of the obstacle shape and position is available.

Topological derivative methods have emerged as powerful tools for the numerical solution of inverse problems. The concept firstly appeared in Eschenauer et al. [1994] in connection with topological optimization of mechanical structures, allowing to implement algorithms where 'excess' material is iteratively removed until a satisfactory shape and topology are reached. The basic idea behind the topological derivative is the evaluation of the sensitivity of the cost function towards creating a hole within the problem domain. The topological derivative does not require pre-existent holes like in level-set methods, so it avoids the aforementioned limitations of shape optimization methods.

The topological derivative to for inverse scattering in acoustics can be traced back to the works by Feijóo [2004], Guzina and Bonnet [2006] and Carpio and Rapún [2008a]. Feijóo [2004] applied the approach introduced by Novotny et al. [2003] to derive topological derivative in a simple and constructive way by using shape sensitivity analysis concepts. On the other hand, Carpio and Rapún [2008a] further developed the topological derivative concept to compute the sensitivity to the insertion of sound-soft obstacles and to deal with non-empty domains.

The boundary element method (BEM) is a very effective numerical technique for solving acoustic problems, especially for simulations that involve infinite domains. One of its advantages is the mesh required. While other techniques like the finite element method (FEM) discretize the entire propagation medium, the discretization of the BEM is limited to the boundary of the objects only. This does not only speed up the model creation and mesh refinement, but it also results in models with less degrees of freedom compared to FEM. The drawback is that unlike FEM, BEM generates fully populated and non-symmetric matrices which limit the benefits of using iterative solvers and memory storage management schemes. Another important aspect of the BEM formulation is that the Sommerfeld's radiation condition(which implies that only outgoing waves are allowed) is implicitly satisfied. Additionally, the unknowns of BEM formulation are the pressure and its derivative, the flux, as such making the method very accurate for the representation of discontinuities Wrobel [2002]. The works by Bonnet [2006], Nemitz and Bonnet [2008], Abe et al. [2010] and Sisamón et al. [2014] are examples of BEM implementations of the topological derivative to inverse acoustic scattering.

This chapter gathers together the key concepts of acoustic scattering and the inverse identification problem (Sections 2, 3 and 4), the explicit expressions of the topological derivative for sound-hard and penetrable obstacles (Section 5), and the key features of to solve the inverse identification problem numerically (Section 6). Special attention is given in Section 6 to the BEM implementation for the cases in which the search area has pre-existent obstacles, either sound-hard, penetrable or combination of both. In this way, a general BEM implementation framework is provided, which might be used by the reader to develop its own specific application. Several examples for sound-hard obstacles are used to demonstrate the performance of the method. Finally, Section 7 is devoted to Comments and Conclusions, with same ideas about further extensions of the methodology to deal with three-dimensional and geometrically complex problems that might require of fast BEM solvers.

2 The Forward Scattering Problem

The general setting of the forward problem is schematized in Figure 1. It consists in an infinite exterior medium Ω_e with a buried obstacle Ω [1]. The illumination of

[1] The problem is posed in terms of a single obstacle to preventing the proliferation of subscripts. The extension to multiple obstacles is immediate and will be discussed later in the sections devoted

the medium results in the reflection of the incident radiation, u_{inc}, by the obstacle to produce scattered radiation, u_{sc}. The total radiation field,

$$u = u_{inc} + u_{sc}, \tag{1}$$

is measured along the boundary Γ_{meas} that is placed far enough from the obstacle.

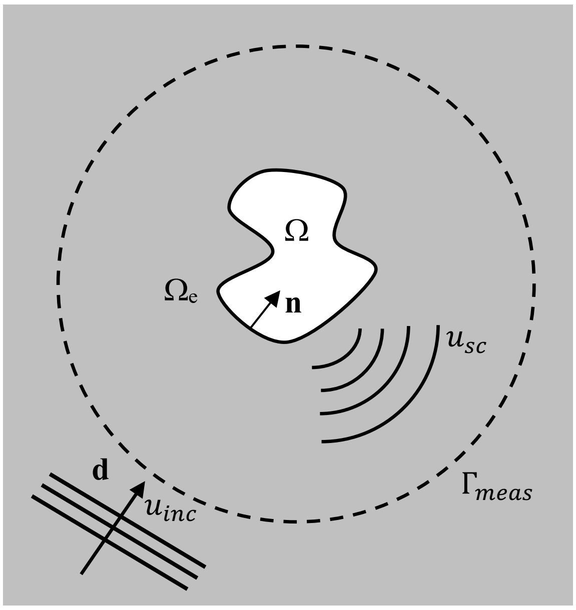


Figure 1: Set-up of the forward scattering problem.

Incident waves can be either planar

$$u_{inc}(\mathbf{x}, \mathbf{d}) = e^{i\lambda_e \mathbf{x}\cdot\mathbf{d}}, \tag{2}$$

where $\mathbf{d}$ is the propagation direction, or due to a point source,

$$u_{inc}(\mathbf{x}, \mathbf{x_0}) = \frac{i}{4} H_0^{(1)}(\lambda_e r), \tag{3}$$

where $\mathbf{x}$ is the evaluation point, $\mathbf{x_0}$ is the location of the point source, i is the imaginary unit, $r = |\mathbf{x} - \mathbf{x_0}|$ is the distance to the source point and $H_0^{(1)}$ is the Hankel function of the first kind. The symbol $\lambda_e = \omega/c$ is the wave number , *i.e.*, the relation of the angular frequency ω to the wave speed c.

The obstacle can be either opaque or penetrable to the incident radiation. An opaque obstacle completely reflects the incident radiation, while for a penetrable obstacle part of the radiation is reflected and the rest is transmitted inside. Thereby, the total wave field in the exterior domain Ω_e and in the interior of the obstacle Ω results after the solution of two coupled Helmholtz problems:

$$\begin{gathered} \nabla^2 u + \lambda_e^2 u = 0 \text{ in } \Omega_e \text{ and} \\ \alpha\nabla^2 u + \lambda^2 u = 0 \text{ in } \Omega, \end{gathered} \tag{4}$$

to numerical implementation.

where the coefficient α accounts for the radiation transmitted inside the obstacle –so that $\alpha = 0$ indicates an opaque obstacle– and λ is the wave number for the propagation in the interior of the obstacle. The coupling between the two problems are the transmission conditions on the interface Γ:

$$\begin{aligned} u^- - u^+ &= 0 \text{ and} \\ \alpha\partial_{\mathbf{n}} u^- - \partial_{\mathbf{n}} u^+ &= 0, \end{aligned} \tag{5}$$

where u^+ and u^- are the limits of u from the exterior and interior of Ω, respectively; $\mathbf{n}$ is the unit normal vector pointing outside Ω_e and $\partial_{\mathbf{n}}$ stands for the normal derivative along Γ.

In addition, it is imposed the standard Sommerfeld radiation condition on the propagation of the scattered field u_{sc} at infinity, which implies that only outgoing waves are allowed:

$$\lim_{r\to\infty} \sqrt{r}(\partial_{\mathbf{r}}(u - u_{inc}) - i\lambda_e(u - u_{inc})) = 0,\ r = |x|, \tag{6}$$

where $\partial_{\mathbf{r}}$ is the radial derivative and i the imaginary unit.

When the obstacle is opaque there is no transmitted wave inside Ω_e , what allows to reduce the problem to the solution of the external domain Ω_e with a null Neumann boundary condition on Γ. So, the problem in equations (4) to (6) reduces to

$$\begin{aligned} \nabla^2 u + \lambda_e^2 u &= 0 \text{ in } \Omega_e, \\ \partial_{\mathbf{n}} u &= 0 \text{ on } \Gamma \text{ and} \\ \lim_{r\to\infty} \sqrt{r}(\partial_{\mathbf{r}}(u - u_{inc}) - i\lambda_e(u - u_{inc})) &= 0, r = |\mathbf{x}|. \end{aligned} \tag{7}$$

3 The Inverse Problem

The inverse problem consists in finding the shape and location of the obstacle from the radiation measurements on Γ_{meas}. A variational approach to solve the inverse problem consists in looking for a domain Ω that minimizes the difference between the solution of the forward problem and the measurements on Γ_{meas}. This leads to the following constrained optimization problem: minimize

$$J(\Omega) = \frac{1}{2}\int_{\Gamma_{meas}} |u - u_{meas}|^2 dl. \tag{8}$$

where Ω is the design variable and the forward problem in Section 2 is the constraint on the admissible field u.

When measurements on Γ_{meas} for N different illuminations are available, the optimization problem is: minimize

$$J(\Omega) = \sum_{j=1}^{N} \frac{1}{2}\int_{\Gamma_{meas}} |u^j - u^j_{meas}|^2 dl. \tag{9}$$

The above problems are strongly ill-posed (Colton [1984]). Given arbitrary data u_{meas}, an associated scatterer Ω may not exist, and if it exists, it may not depend continuously on u_{meas}. The obstacle is uniquely determined by the far-field scattered wave for all incident directions and one fixed wave number (Kirsch and Kress [1993], Gerlach and Kress [1996]). Therefore, by an analyticity argument (see Colton and Kress [2013]), if Γ_{meas} is a circumference, then the values of the total wave on Γ_{meas} for all incident waves determine uniquely the obstacles. The question of uniqueness without any *a priori* knowledge about the location of the obstacles a finite number of incident plane waves is still an open problem.

4 The Adjoint Problem

As it will be shown in next section, the derivation of the expressions of the topological derivative requires the solution of the so-called adjoint problem. This auxiliary problem consists in propagating back the discrepancy $(u - u_{meas})$ from Γ_{meas} towards the region of analysis.

In the case the region of analysis is empty, the adjoint wave p solves (seee Carpio and Rapún [2008a])

$$\begin{aligned} \nabla^2 p + \lambda_e^2 p &= -(u - u_{meas})\delta_{\Gamma_{meas}} \text{ in } \mathbb{R}^2, \\ \lim_{r\to\infty} \sqrt{r}(\partial_{\mathbf{r}} p + i\lambda_e p) &= 0, \end{aligned} \tag{10}$$

where $\delta_{\Gamma_{meas}}$ is the Dirac Delta function on Γ_{meas}. For penetrable obstacles, the adjoint problem is

$$\begin{aligned} \nabla p + \lambda_e^2 p &= -(u - u_{meas})\delta_{\Gamma_{meas}} \text{ in } \Omega_e, \\ \alpha\nabla^2 p + \lambda^2 p &= 0 \text{ in } \Omega, \\ p^- + p^+ &= 0 \text{ on } \Gamma, \\ \partial_{\mathbf{n}} p^- + \partial_{\mathbf{n}} p^+ &= 0 \text{ on } \Gamma \text{ and} \\ \lim_{r\to\infty} \sqrt{r}(\partial_{\mathbf{r}} p + i\lambda_e p) &= 0, r = |\mathbf{x}|. \end{aligned} \tag{11}$$

Finally, for opaque obstacles, the adjoint the problem reduces to the Neumann problem

$$\begin{aligned} \nabla^2 p + \lambda_e^2 p &= -(u - u_{meas})\delta_{\Gamma_{meas}} \text{ in } \Omega_e, \\ \partial_{\mathbf{n}} p &= 0 \text{ on } \Gamma \text{ and} \\ \lim_{r\to\infty} \sqrt{r}(\partial_{\mathbf{r}} p + i\lambda_e p) &= 0, r = |\mathbf{x}|. \end{aligned} \tag{12}$$

As for the forward problem, the Sommerfeld radiation condition is imposed to the back-propagated field. In this sense, note the plus sign in the last of equations in (10) to (12) in contrast to the minus sign in the last of equations in (6).

5 The Topological Derivative for Inverse Scattering

The topological derivative measures the sensitivity of the objective function (8) to the inclusion of an infinitesimal obstacle into the problem domain, such that regions where it takes on large negative values are identified as the positions of the obstacles.

The derivation of the topological derivative formulas in this section follows Carpio and Rapún [2008a]. The setting of the problem is in Figure 2. Starting from a domain $\mathcal{R}$ with no obstacles, a small circular obstacle is introduced at a point $\mathbf{x}$, which modifies the domain by creating a 'hole' $B_\varepsilon(\mathbf{x})$ of radius ε. The topological derivative is defined as

$$D_T(\mathbf{x},\mathcal{R}) = \lim_{\varepsilon\to 0}\frac{J(\mathcal{R}_\varepsilon)-J(\mathcal{R})}{f(\varepsilon)}, \tag{13}$$

where $J(\mathcal{R})$ and $J(\mathcal{R}_\varepsilon)$ are the cost function evaluated for the reference and perturbated domains, $\mathcal{R}_\varepsilon = \mathcal{R} - B_\varepsilon(\mathbf{x})$, respectively; ε is the radius of the obstacle, and $f(\varepsilon)$ is a monotonically decreasing negative function that is chosen such that $f(\varepsilon)\to 0$ when $\varepsilon\to 0$. The function $f(\varepsilon)$ is usually related (although this is not mandatory) to the measure of area of the obstacle, $f(\varepsilon) = -\pi\varepsilon^2$, in this case.

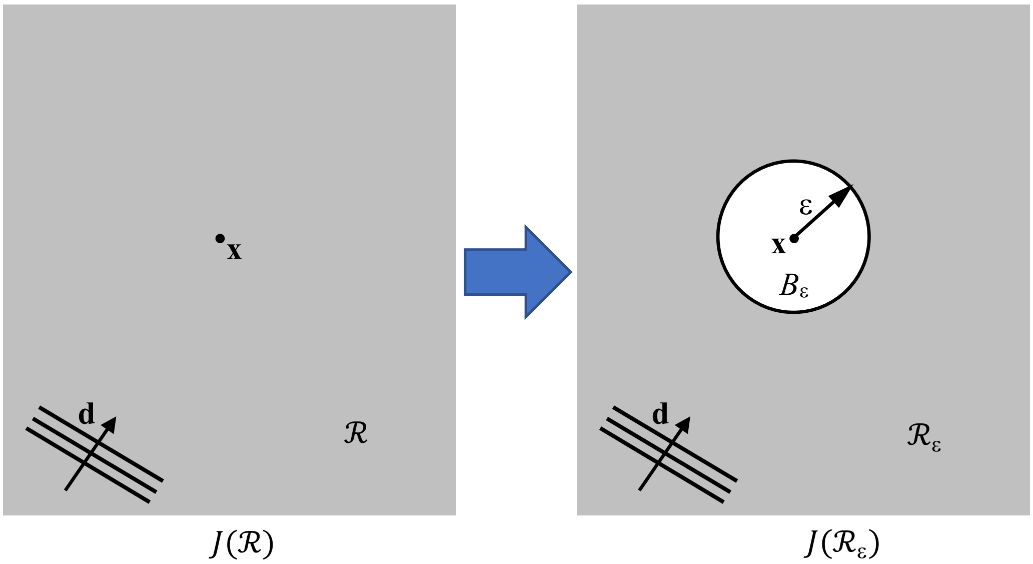


Figure 2: Definition o the topological derivative using the shape sensitivity approach.

Explicit expressions of the topological derivative can be obtained via the shape derivative (Feijóo [2004]). The link between the topological and shape derivatives is established by the relationship

$$J(\mathcal{R}_\varepsilon) = J(\mathcal{R}) + f(\varepsilon)D_T(x) + O(f(\varepsilon)). \tag{14}$$

The shape derivative of the functional (8) along a vector field $\mathbf{V}$ is defined as

$$DJ(\mathcal{R})\cdot\mathbf{V} = \frac{d}{d\varepsilon}J(\phi(\mathcal{R}))\Big|_{\varepsilon=0}, \tag{15}$$

where $\phi(\mathbf{x})$ is the mapping function between the reference and perturbated domains,

$$\phi(\mathbf{x}) = \mathbf{x} + \varepsilon\mathbf{V}(\mathbf{x}),\ \mathbf{x}\in\mathbb{R}^2, \tag{16}$$

in which the vector field points towards the obstacle boundary ∂B, *i.e.*, $\mathbf{V}(\mathbf{x}) = V_n\mathbf{n}(\mathbf{z})$ with $V_n < 0$ vanishes out of a neighborhood of ∂B. Then, the expression for the topological derivative results as

$$D_T(\mathbf{x}, \mathcal{R}) = \lim_{\varepsilon\to 0}\left(\frac{1}{f'(\varepsilon) \mid V_n \mid}\frac{d}{d\varepsilon}J(\phi(\mathcal{R}_\varepsilon))\Big|_{\varepsilon=0}\right), \tag{17}$$

where $f'(\varepsilon)$ if the derivative of $f(\varepsilon)$.

A family of explicit expressions of the topological derivative results after the solution of (17) for specific problem settings that consider the identification of opaque and penetrable obstacles. For the details about the derivations these solutions – which involve rather technical analyses– the reader is referred to Carpio and Rapún [2008a].

The expressions of the topological derivative for penetrable obstacles is

$$D_T(\mathbf{x}) = Re\left[\frac{2(1-\alpha)}{1+\alpha}\nabla u(\mathbf{x})\nabla\bar{p}(\mathbf{x}) + (\lambda^2 - \lambda_e^2)u(\mathbf{x})\bar{p}(\mathbf{x})\right], \tag{18}$$

where $\bar{p}$ stands for the conjugate of the adjoint problem solution. For the limiting case of perfect transmission across the interface, *i.e.* $\alpha = 1$, expression (18) reduces to

$$D_T(\mathbf{x}) = Re\left[(\lambda_i^2 - \lambda_e^2)u(\mathbf{x})\bar{p}(\mathbf{x})\right]. \tag{19}$$

Additionally, the topological derivative for opaque obstacles is

$$D_T(\mathbf{x}) = Re\left[2\nabla u(\mathbf{x})\nabla\bar{p}(\mathbf{x}) + \lambda_e^2 u(\mathbf{x})\bar{p}(\mathbf{x})\right]. \tag{20}$$

It is interesting to point out that the above three expressions for the D_T are independent of the structure of the incident wave, so they hold for plane waves and due to point sources or any other type of source.

Another cool feature of the above expressions is that, although they have been derived starting from an empty domain, they can be used for domains with pre-existent obstacles, say Ω_0. In such a case, the forward and adjoint problems are solved in $\Omega_e = \mathbb{R}^2 - \Omega_0$ with transmission or Neumann boundary conditions at the interface. The solution of such problems will be discussed thoroughly next in the section devoted to the numerical implementation.

Analogous formulae of the D_T have been deduced in three dimensions, see for example Guzina and Bonnet [2006] and Carpio and Rapún [2008a] .

6 Numerical implementation

6.1 Problem setting

For the sake of simplicity, but without loss of generality, the computation of the D_T is presented here in the context of sound scattering and planar incident waves.

The problem setting is depicted in Figure 3. Two cases may arise: either the search area is empty or it contains pre-existent obstacle Ω_0. In any cases the search area is illuminated from N different directions and the sound pressure is measured at $\mathbf{x}^k$ with $k = 1...M$ receptor points s that are placed on the circular boundary Γ_{meas} of radio R_{meas} that encloses the search area. As it will be shown later, the use of discrete point receptors to on Γ_{meas} simplifies the computations and allows for efficient BEM implementations.

The D_T is sampled at P regularly spaced points on the search area. These points are referred as the 'domain points' and they are as $\mathbf{x}^l$ with $l = 1...P$.

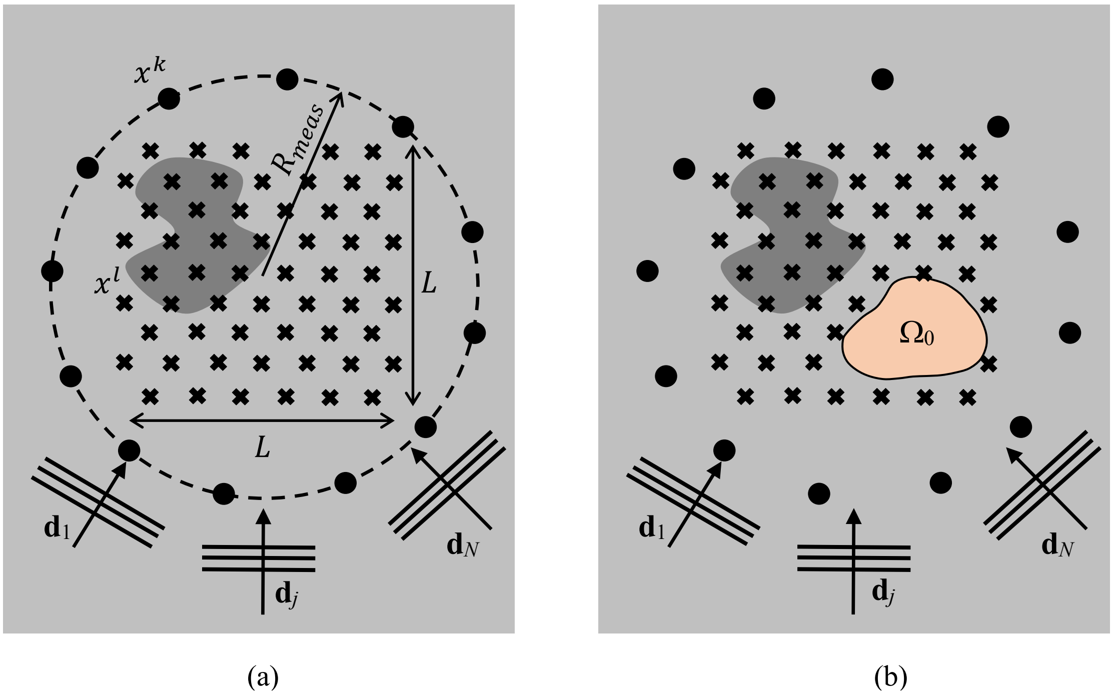


Figure 3: Topological derivative computation: (a) empty domain, (b) domain with a pre-existent obstacle. Symbols • and × indicate the positions of the receptors and the domain points, respectively.

The evaluation of D_T requires knowing the pressure and its gradient for the direct and adjoint problems (see equations (18), (19) and (20)) at the domains points. There are two implementations: the first for the case of the empty search area, in which all calculations can be performed analytically; and the second for the case of a pre-existing obstacle, which uses numerical solution by means of BEM models.

or it contains pre-existent obstacles.

6.2 Empty domains

The absence of obstacles makes the problem free of dispersion, which allows the pressures and their derivatives of both the direct problem and the adjoint problem to be calculated analytically.

6.2.1 Computation of the direct problem

The incident pressure field is computed using equation (2) for unit incident waves, see Figure 3(a). Due to the absence of scattering, the total pressure is $u = u_{inc}$ (see equation (1)).

On the other hand, the spatial derivatives of the planar wave are

$$u_{,m}(\mathbf{x}, \mathbf{d}) = \frac{\partial u(\mathbf{x}, \mathbf{d})}{\partial x_m} = (i\lambda_e \cdot \mathbf{d}_m)e^{i\lambda_e \mathbf{x}\cdot\mathbf{d}} \text{ with } m = 1, 2. \tag{21}$$

6.2.2 Computation of the adjoint problem

The adjoint problem consist in propagating back the pressure discrepancy a the receptors towards the search area. So, consistent with the point receptors used in this implementation, the radiation from the receptors are computed using equation (3) with the amplitudes given by $(u - u_{meas})$. Thus, the pressure field associated to a point source in the k^{th} receptor is

$$p(\mathbf{x}) = -[u(\mathbf{x}^k) - u_{meas}(\mathbf{x}^k)]\frac{i}{4}H_0^{(1)}(\lambda_e r), \tag{22}$$

and its spatial derivatives are

$$p_{,m}(\mathbf{x}) = -[u(\mathbf{x}^k) - u_{meas}(\mathbf{x}^k)]\frac{i\lambda_e}{4}H_1^{(1)}(\lambda_e r)\frac{\partial r}{\partial x_m} \text{ with } m = 1, 2, \tag{23}$$

where $r = |\mathbf{x} - \mathbf{x}^k|$ is the distance from the receptor.

6.2.3 Algorithm

i Set-up the problem as it depicted in Figure 3(a), and have available the data of the pressure measurements at the M receptors due to each of the N illumination directions. Store the pressure measurements in $\mathbf{u}^j_{\mathbf{meas}}(\mathbf{x}^k)$, which are $j = 1...N$ vectors with $k = 1...M$ elements each.

ii Use equations (2) and (21) to compute the pressures and pressure gradients due to each of the illumination directions at the domain points. These results are stored in $\mathbf{u}^j(\mathbf{x}^l)$, which are $j = 1...N$ vectors of length $l = 1...P$; and $\nabla\mathbf{u}^j(\mathbf{x}^l)$, which are $j = 1...N$ matrices with $l = 1...P$ rows and two columns, one for each component of the gradient, $m = 1, 2$.

iii Use equation (2) to compute $\mathbf{u}^j(\mathbf{x}^k)$, the pressures due to each of the $j = 1...N$ illumination directions at the $k = 1...M$ receptors.

iv Use equations (22) and (23) with the results in $\mathbf{u}^j_{\mathbf{meas}}(\mathbf{x}^k)$ and $\mathbf{u}^j(\mathbf{x}^k)$ to compute the adjoint pressure and pressure derivatives at the domain points. Note that since each illumination direction results in discrepancies at all the receptors, the solutions to the adjoint problem at each domain point is the sum of the contributions of the M receptors. Thus, the adjoint pressure field at the l^{th} domain point do the j^{th} incident wave is

$$\mathbf{p}^j(\mathbf{x}^l) = \sum_{k=1}^{M} -[\mathbf{u}^j(\mathbf{x}^k) - \mathbf{u}^j_{\mathbf{meas}}(\mathbf{x}^k)]\frac{i}{4}H_0^{(1)}(\lambda_e r^{(k)}), \tag{24}$$

and its spatial derivatives are

$$\mathbf{p}^j_{,m}(\mathbf{x}^l) = \sum_{k=1}^{M} -[\mathbf{u}^j(\mathbf{x}^k) - \mathbf{u}^j_{\mathbf{meas}}(\mathbf{x}^k)]\frac{i\lambda_e}{4}H_1^{(1)}(\lambda_e r^{(k)})\frac{\partial r^{(k)}}{\partial x_m} \text{ with } m = 1, 2. \tag{25}$$

Following the same scheme in step (ii), the results of the above computations are stored in the arrays $\mathbf{p}^j(\mathbf{x}^l)$ and $\nabla\mathbf{p}^j(\mathbf{x}^l)$, respectively.

v Select the adequate expression of D_T in Section 5 and use the results in $\mathbf{u}^j(\mathbf{x}^l)$, $\nabla\mathbf{u}^j(\mathbf{x}^l)$, $\mathbf{p}^j(\mathbf{x}^l)$ and $\nabla\mathbf{p}^j(\mathbf{x}^l)$, to compute the topological derivative at the domain points for each of the N illumination directions: $\mathbf{D}^j_{\mathbf{T}}(\mathbf{x}^l)$.

vi Compute the overall topological derivative at the domain points as the sum of the contributions of all the illumination directions:

$$D_T(\mathbf{x}^l) = \sum_{j=1}^{N} \mathbf{D}^j_{\mathbf{T}}(\mathbf{x}^l) \tag{26}$$

vii Draw a contour plot with the $D_T(\mathbf{x}^l)$ results over the search domain. The geometry of the hidden obstacle is that delimited by the largest negative values of the topological derivative.

6.2.4 Examples

Identification of a single obstacle. The hidden obstacle is that depicted in Figure 4a, which has a smooth boundary with concave and convex portions.

The problem set-up is similar to that depicted in Figure 3a. The search area is of dimensions $L \times L$ =10 m×10 m. The hidden obstacle is illuminated by N plane waves with their angles of incidence equally distributed over 2π. The number of receptors M, the distance R_{meas}, as well as the number of illumination waves N and their wavelengths λ are subjects of the analysis. The reference sound-pressure values at the receptors due to the dispersion of the N incident waves by the obstacle, $\mathbf{u}^j_{\mathbf{meas}}(\mathbf{x}^k)$, were produce synthetically using high-resolution BEM models with element sizes equal to one-tenth the incident wavelength (the details of the BEM models will be introduced later in Section 6.3).

The first set of results assess the effect of the wavelength on the quality of the identification. The analysis is performed for wavelengths $\lambda = c/f_i$ =3.4, 1.7, 0.85, 0.567 and 0.425 m, corresponding to frequencies f_i =100, 200, 400, 600 and 800 Hz for a speed of sound c =340 m/s. The number of incident waves, N = 100, the number of receptors, M = 100, and their distances, R_{meas}=10 m, are kept constant.

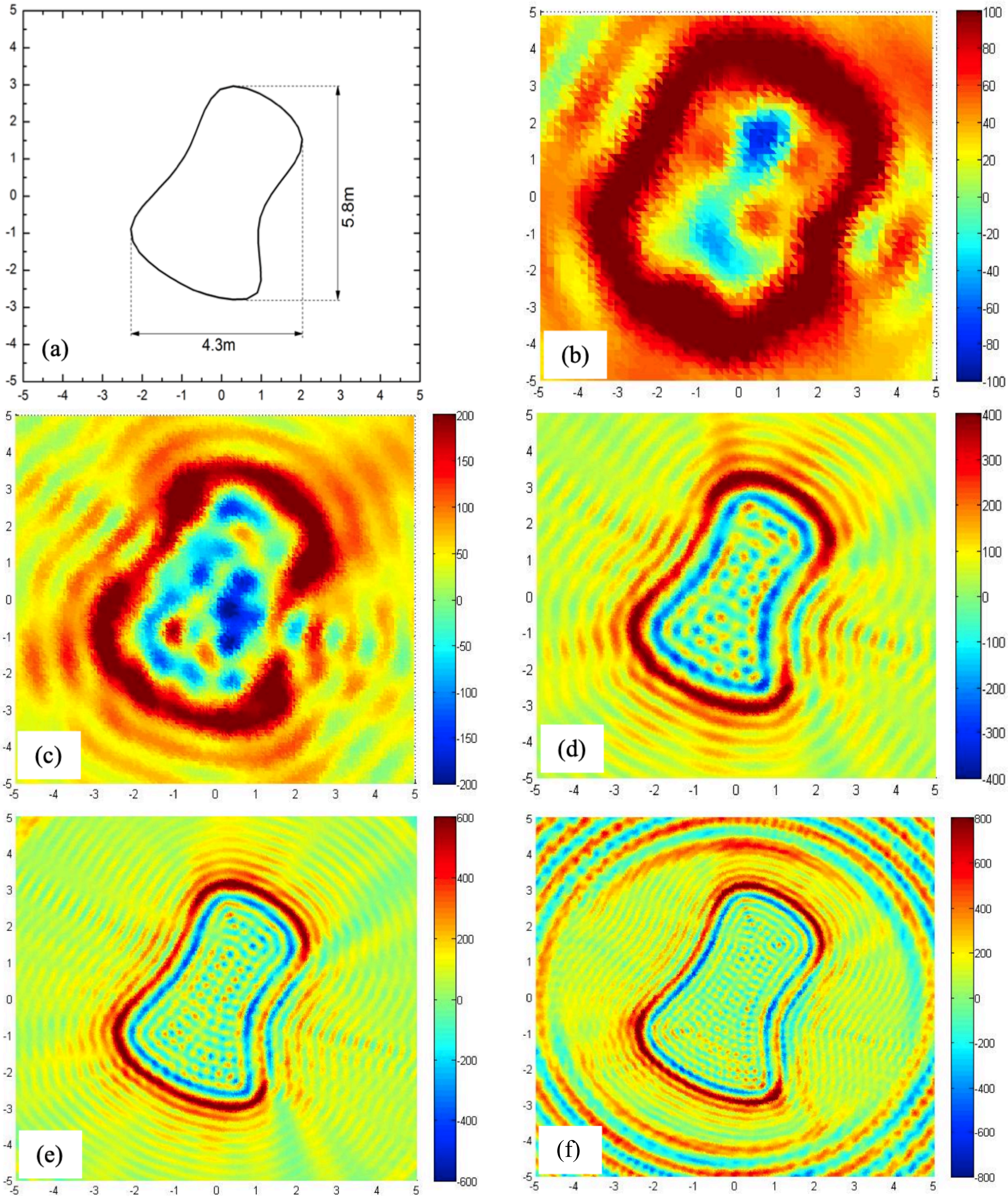


Figure 4: Identification of a single obstacle in an empty domain. Contour plots of D_T for different illumination wavelengths: (a) Geometry of the hidden obstacle and results for the obstacle shape for (b) $\lambda/D = 0.85$, (c) $\lambda/D = 0.425$, (d) $\lambda/D = 0.213$, (e)$\lambda/D = 0.142$ and (f) $\lambda/D = 0.106$.

Grid size of the domain point arrangement are one-tenth of the wavelength of the incident wave. Depending on the wavelength, the number of domain points ranged from 870 to 55360.

The results for the topological derivative D_T in terms of λ/D, the ratio between the wavelength and the characteristic length of the obstacle, are presented in Figures 4b to f. The shape of the scatterer is given by the locus of the most negative lowest D_T values. As it was expected, the quality of the identification improves as the wavelength decreases. The reconstruction provides a reasonable result of the obstacle shape for wavelengths $\lambda/D \leq 0.20$.

The second analysis is about the influence of the number of illumination directions. The problem is solved for $N_i = 25, 50, 75$ and 100. Based on the previous analysis, the source wavelength is set to $\lambda/D = 0.142$ ($f = 600Hz$). The number and distance of the receptors, M =100 and R_{meas} =10 m, are the same as in the previous analysis. The results are shown in Figure 5. It can be observed that a reduced number of sources as low as $N = 25$ provides a reasonable approximation of the obstacle shape.

The effect of the distance of the receptors is assessed in the third analysis. To this end, the problem is solved for the normalized distances R_{meas}/D =2.5, 5, 10 and 20. The number of illumination directions, their wavelengths and the number of receptors are set to $N = 100$, $\lambda/D = 0.142$ and $M = 100$, respectively. The results are shown in Figure 6. As it was expected, the quality of the identification deteriorates with the distance increases. However, this lost of performance is not significant for the range of distances considered.

Finally, the effect of the measurement error is investigated by adding noise to the pressure values at the receptors. Random noise is added to the amplitude and phase of $u^j_{meas_k}$. The effects of the noise amplitude are examined in the range from 5% to 50% the amplitude of the incident waves. Figure 7 illustrates some results for the example with $N = 100$, $\lambda/D = 0.142$, $M = 100$ and $R_{meas}/D = 10$. The effect of noise starts to be noticeable for noise amplitudes around 20% (compare Figures 6c and 7a). Noise amplitudes of 50% clearly deteriorates the performance of the method, see Figure 7b; however, it is still possible to distinguish the silhouette of the hidden obstacle.

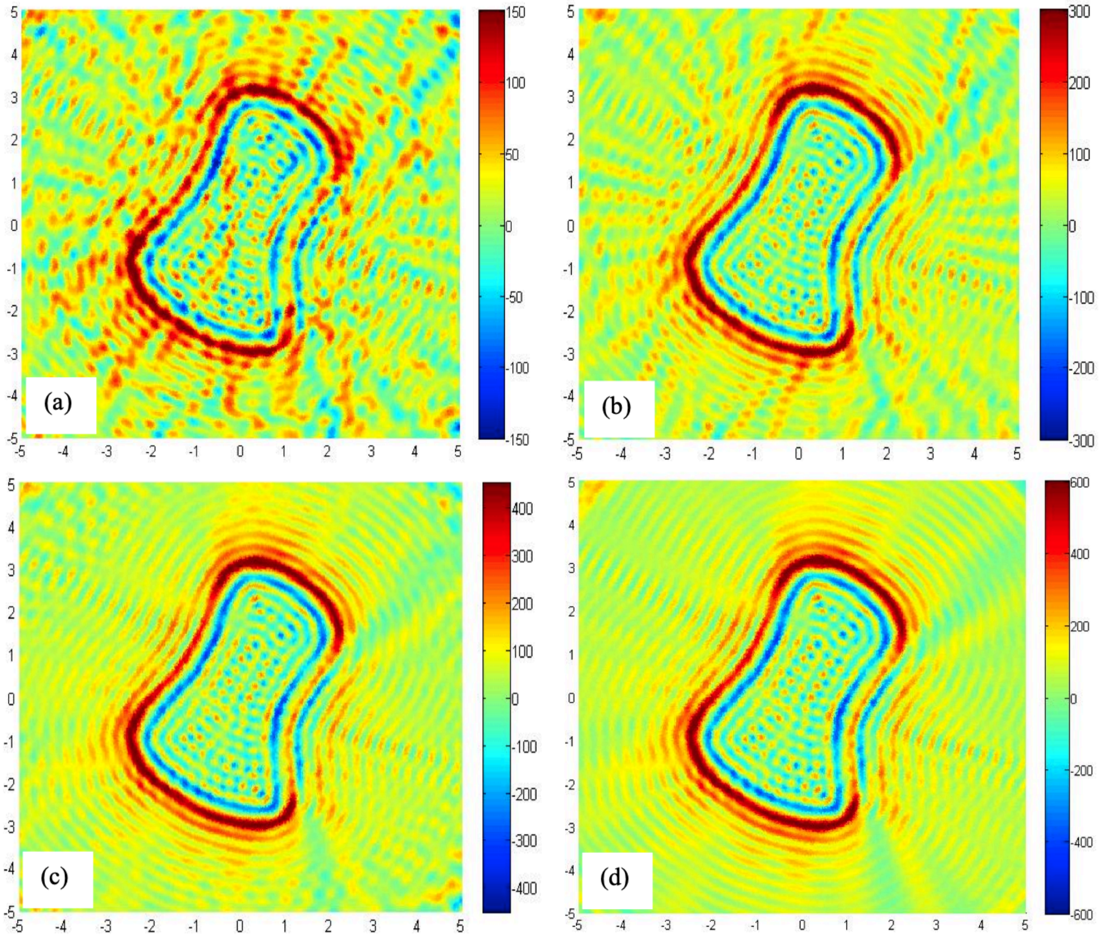


Figure 5: Identification of a single obstacle in an empty domain. Contour plots of D_T in terms of the number of sources: (a) $N = 25$, (b) $N = 50$, (c) $N = 75$ and (d) $N = 100$.

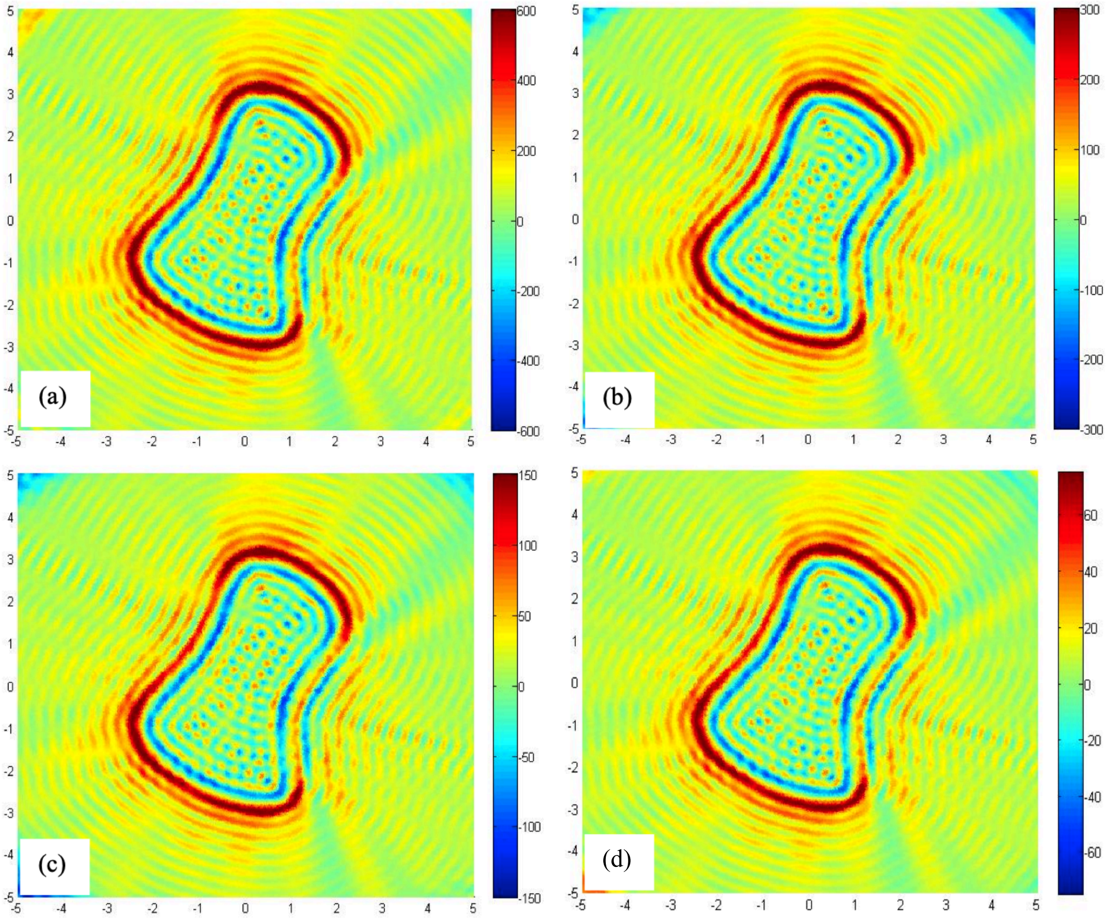


Figure 6: Identification of a single obstacle in an empty domain. Contour plots of D_T in terms of the distance to the receptors: (a) $R_{meas}/D = 2.5$, (b) R_{meas}/D =5, (c) R_{meas}/D =10 and (d) $R_{meas}/D = 20$.

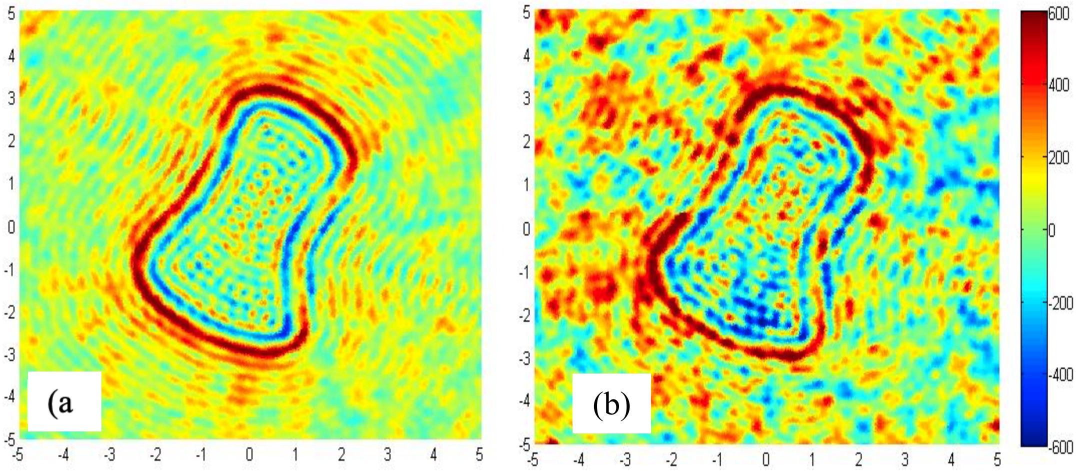


Figure 7: Identification of a single obstacle in an empty domain. Contour plots of D_T in terms of the number of sources: (a) $N = 25$, (b) $N = 50$, (c) $N = 75$ and (d) $N = 100$.

Identification of multiple obstables The problem geometry is illustrated in 8a. It consists in the identification of two hidden obstacles: a circle of radius r =2 m and a square with side length $s = 2r$ =4 m .

The performance of the method is demonstrated with regard to the distance between the objects, which is examined in the range $2\sqrt{2} \leq d/r \leq 6\sqrt{2}$, where d is the distance from the center of the circle to the center of the square. Based on the results of the previous example, $M = 100$ receptors are placed on the circular boundary Γ_{meas} of radius $R_{meas} = 10r = 20$ m. The search areas is illuminated by $N = 100$ planar waves with their angles of incidence equally distributed over 2π. The wavelength is set to λ =0.567 m, so that $\lambda/s = 0.142$. The size of the search area is set in accordance with the distance of the scatterers, from 12 $m\times$ 12 m for the closest positions to 20 $m\times$20 m for the furthest one. Accordingly, the number of domain points ranges from 58081 to 161336. Like in the previous example, reference sound-pressure values at the receptors due to the hidden scatterers, $u^j_{meas_k}$, were computed using high-resolution BEM models.

The D_T results in terms of distance between the obstacles s are plotted in Figure 8b to f. It can be observed that obstacles shapes are resolved with high precision. It is interesting to note that the D_T intensities along the portions of the obstacles boundaries facing each other are larger than along the portions of the boundaries facing the open space. This is because the obstacles shield each other, what hinders the illumination of the closest portions of the boundaries. Moreover, the highest values of D_T occur in the zone between the obstacles, what indicates that this is the most sensitive zone towards the minimization of the objective function. This is coherent with the behavior expected for the limiting case, in which the objects touch or overlap, and thus behave as a single obstacle.

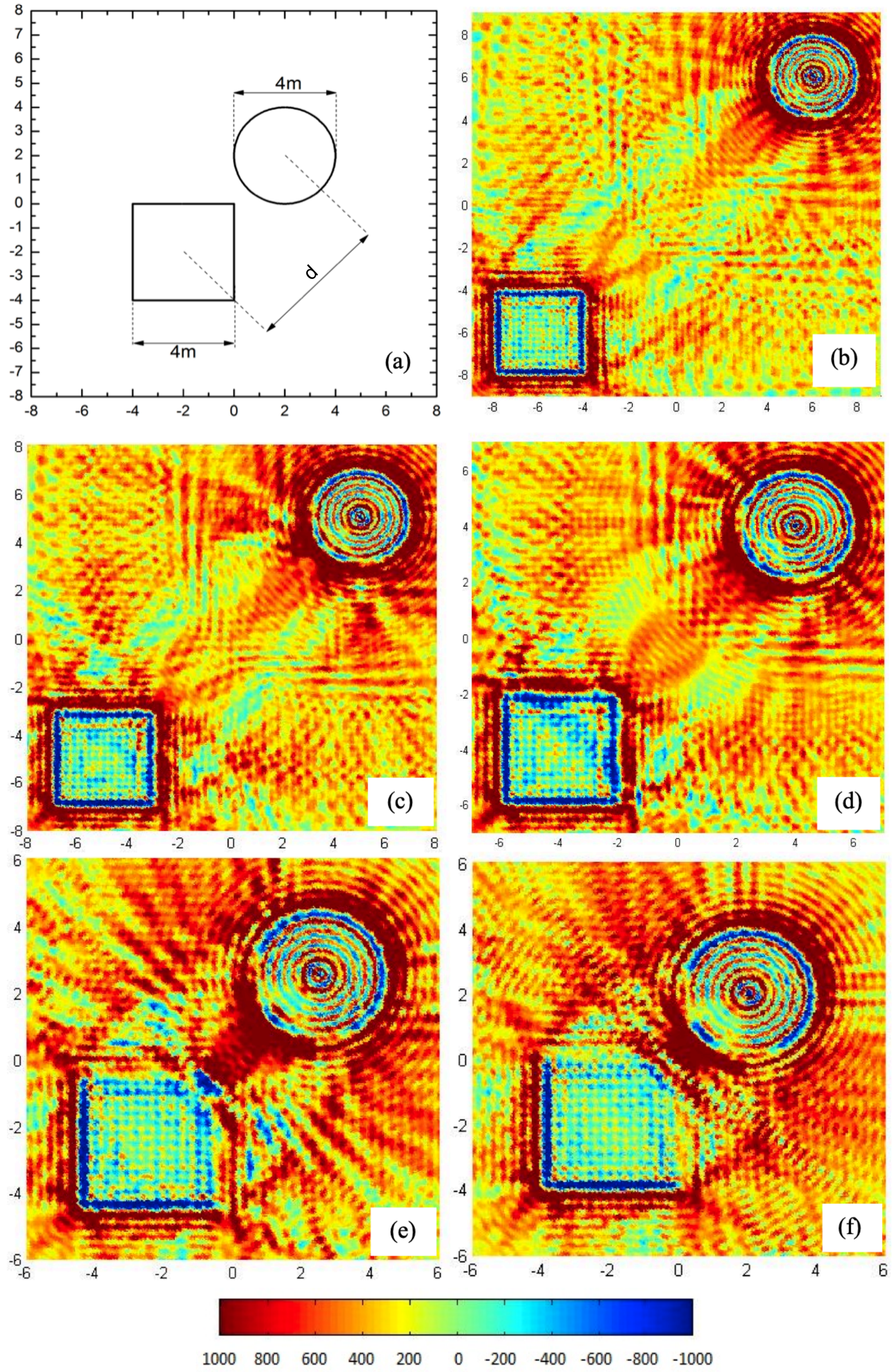


Figure 8: Identification of two obstacles in an empty domain. Contour plots of D_T in terms of the separation between the obstacles: (a) hidden obstacles, (b) $d/r = 6\sqrt{2}$, (c) $d/r = 5\sqrt{2}$, (d) $d/r = 4\sqrt{2}$, (e) $d/r = 2.5\sqrt{2}$ and (f) $d/r = 2\sqrt{2}$. The same color scale is used in all the figures.

6.3 Domains with pre-existent obstacles

In contrast to the empty-domain analyses, the evaluation of the D_T for domains with pre-existent obstacles requires of numerical models to solve the scattered fields of the forward and adjoint problems. Boundary element models are used for this purpose.

6.3.1 The BEM for the Helmholtz's equation

Main aspects of the Boundary Element Method (BEM) are described next, aiming to the specific task of computing the D_T. The BEM formulation is not introduced thoroughly; the reader is referred to classical books on BEM such as Wrobel [2002] or Brebbia et al. [1984] for full details.

Consider the problem in Figure 3(b) and assimilate Ω_0 to a hole, such that $\Omega_e = \mathcal{R} - \Omega_0$. The starting point to formulate the BEM for solving the scatter problem over Ω_e is the integral formulation of the Helmholtz equation:

$$c(\mathbf{x}')u_{sc}(\mathbf{x}') + \int_\Gamma q^*(\mathbf{x},\mathbf{x}')u_{sc}(\mathbf{x})d\Gamma(\mathbf{x}) = \int_\Gamma u^*(\mathbf{x},\mathbf{x}')q_{sc}(\mathbf{x})d\Gamma(\mathbf{x}) + u_{inc}(\mathbf{x}',\mathbf{d}), \quad (27)$$

which relates $u_{sc}(\mathbf{x}')$, the pressure at a position $\mathbf{x}'$ (the so-called collocation point), with the pressure and the pressure flux, $u_{sc}(\mathbf{x}')$ and $q_{sc}(\mathbf{x}) = \frac{\partial u_{sc}}{\partial \mathbf{n}}$, respectively, on the boundary $\Gamma(\mathbf{x})$. The term $c(\mathbf{x}')$ in (27) is a coefficient that depends on the boundary geometry at $\mathbf{x}'$; and $u^*(\mathbf{x},\mathbf{x}')$ and $q^*(\mathbf{x},\mathbf{x}')$ are the fundamental solutions for the pressure and flux fields, respectively.

The pressure fundamental solution u^* is the solution to the Helmholtz's equation for the pressure generated at a field point $\mathbf{x}$ by a concentrated unit source at $\mathbf{x}'$ in an unbounded space. Note that this solution is the same of the point source in equation (3). So,

$$u^*(\mathbf{x},\mathbf{x}') = \frac{i}{4}H_0^{(1)}(\lambda_e r), \quad (28)$$

with the distance $r = |\mathbf{x} - \mathbf{x}'|$ defined from the collocation to the field point. Similarly, the fundamental solution q^* is the flux in the normal direction at $\Gamma(\mathbf{x})$ generated by the unit source at $\mathbf{x}'$, this is

$$q^*(\mathbf{x},\mathbf{x}') = \frac{\partial u^*}{\partial \mathbf{n}} = -\frac{i\lambda_e}{4}H_0^{(1)}(\lambda_e r). \quad (29)$$

It is worth to note that the above formulation naturally satisfies the standard Sommerfeld radiation condition on the propagation of the scattered field at infinity (see Section 2).

The BEM consists in discretizing the boundary $\Gamma(\mathbf{x})$ into nel elements. Depending on the discretization strategy, $u(\mathbf{x})$ and $q(\mathbf{x})$ can be interpolated using constant, linear or quadratic elements. For the sake of simplicity but without the loss of generality, constant elements [2] are used in this work, such that $\Gamma = \sum_{j=1}^{nel}\Gamma_j$. The

[2] These elements approximate both the$u$ and q fields as constants and have a single node at their midpoints.

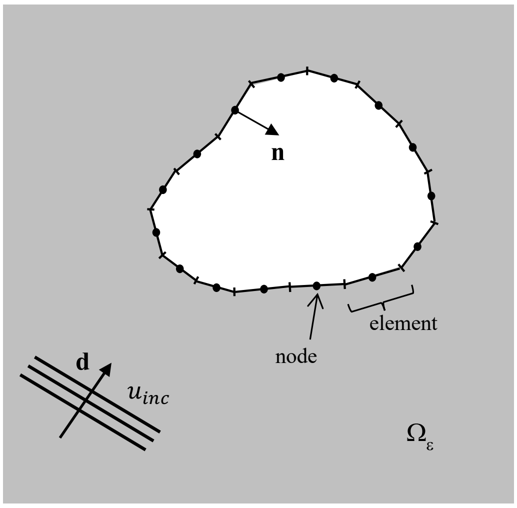


Figure 9: Boundary element discretization of the problem consisting in an open domain with an embedded hole.

discretized model is depicted in Figure 9. Notice the normal vector pointing out of the domain. The discretized version of equation (27) for the collocation point located at the i^{th} node is

$$c^i(\mathbf{x}^i)u^i_{sc}(\mathbf{x}^i)+\sum_{j=1}^{nel}\int_{\Gamma_j}u^j_{sc}(\mathbf{x})q^*(\mathbf{x},\mathbf{x}^i)d\Gamma(\mathbf{x})=\sum_{j=1}^{nel}\int_{\Gamma_j}q^j_{sc}(\mathbf{x})u^*(\mathbf{x},\mathbf{x}^i)d\Gamma(\mathbf{x})+u_{inc}(\mathbf{x}^i,\mathbf{d}). \tag{30}$$

Upon the evaluation the of the integrals, equation (30) can be written in matrix form as follows:

$$c^iu^i_{sc}+\sum_{j=1}^{nel}\hat{H}^{ij}u^j_{sc}=\sum_{j=1}^{nel}\int_{\Gamma_j}G^{ij}q^j_{sc}+u^i_{inc}, \tag{31}$$

where

$$\begin{aligned}\hat{H}^{ij}&=\int_{\Gamma_j}q^*(\mathbf{x},\mathbf{x}^i)d\Gamma(\mathbf{x})\text{ and}\\ G^{ij}&=\int_{\Gamma_j}u^*(\mathbf{x},\mathbf{x}^i)d\Gamma(\mathbf{x}).\end{aligned} \tag{32}$$

If the position of the collocation point i is made to vary from 1 to nel, that is, the fundamental solution is applied in each of the nodes successively, the evaluation of (31) to each node results in the system of equations

$$\mathbf{Hu_{sc}}=\mathbf{Gq_{sc}}+\mathbf{u_{inc}}, \tag{33}$$

where $\mathbf{H}$ collects the elements $\hat{H}^{ij}$ plus the terms $c^i(\mathbf{x^i})$.

Matrices $\mathbf{H}$ and $\mathbf{G}$ in equation (33) have dimensions $nel \times nel$, while $\mathbf{u_{sc}}$, $\mathbf{q_{sc}}$ and $\mathbf{u_{inc}}$ are vectors of length nel. This makes a system of nel equations with

$2 \cdot nel$ unknowns (u^i_{sc} and q^i_{sc}), which will not be solvable until boundary conditions are specified on Γ. In the next section, the formulation will be specialized for the cases of pre-existing sound-hard (opaque) and penetrable obstacles by specifying the appropriate boundary conditions on Γ.

6.3.2 Computation of the forward problem

Sound hard obstacles. When dealing with sound hard obstacles, the sound flux $q = \frac{\partial u_{sc}}{\partial \mathbf{n}}$ is zero along $\Gamma(\mathbf{x})$. So, the integrals in the right-hand side of (30) vanish to yield

$$c^i u^i_{sc} + \sum_{j=1}^{nel} \hat{H}^{ij} u^j_{sc} = u^i_{inc}, \tag{34}$$

and the system of equation (33) reduces to

$$\mathbf{H}\mathbf{u_{sc}} = \mathbf{u_{inc}}. \tag{35}$$

The system in (35) is solved to obtain the nodal pressures on the obstacle boundary, $\mathbf{u_{sc}}$. Notice that matrix $\mathbf{H}$ depends on the geometry of the obstacle only, so, it is computed once and then used repeatedly to solve the boundary pressures for each of the N incident waves. Conversely, vector $\mathbf{u_{inc}}$ depends on the direction of the incident wave, so it has to be computed repeatedly for each incident wave.

Once the boundary pressure is known, the total pressures and pressure gradients at the domain points are computed using equation (1) in combination with the pressure boundary integral equation. Thus, the discretized expressions for the total pressure and its gradient components are

$$u_{sc}(\mathbf{x}^l) = \sum_{j=1}^{nel} \int_{\Gamma_j} u^j_{sc}(\mathbf{x}) q^*(\mathbf{x}, \mathbf{x}^l) d\Gamma(\mathbf{x}) + u_{inc}(\mathbf{x}^l, \mathbf{d}) \tag{36}$$

and

$$u_{sc,m}(\mathbf{x}^l) = \frac{\partial u_{sc}(\mathbf{x}^l)}{\partial \mathbf{x}^l_m} = \sum_{j=1}^{nel} \int_{\Gamma_j} u_{sc}(\mathbf{x}) \frac{\partial q^*(\mathbf{x}, \mathbf{x}^l)}{\partial \mathbf{x}_m} d\Gamma(\mathbf{x}) + \frac{\partial u_{inc}(\mathbf{x}^l, \mathbf{d})}{\partial \mathbf{x}_m} \text{ with } m = 1, 2, \tag{37}$$

respectively. The discretized matrix forms of equations (36) and (37) are

$$\mathbf{u_{sc}}(\mathbf{x}^l) = \mathbf{A}\mathbf{u_{sc}} + \mathbf{u_{inc}} \tag{38}$$

and

$$\mathbf{u_{sc,m}}(\mathbf{x}^l) = \mathbf{A}'\mathbf{u_{sc}} + \mathbf{u_{inc,m}}. \tag{39}$$

Notice that matrices $\mathbf{A}$ and $\mathbf{A}'$ have to be computed repeatedly for each domain node, since they depend on the position $\mathbf{x}^l$. The computation of $\mathbf{A}$ and $\mathbf{A}'$ are the most expensive tasks of the algorithm, as they are typically computed hundreds or thousands of times.

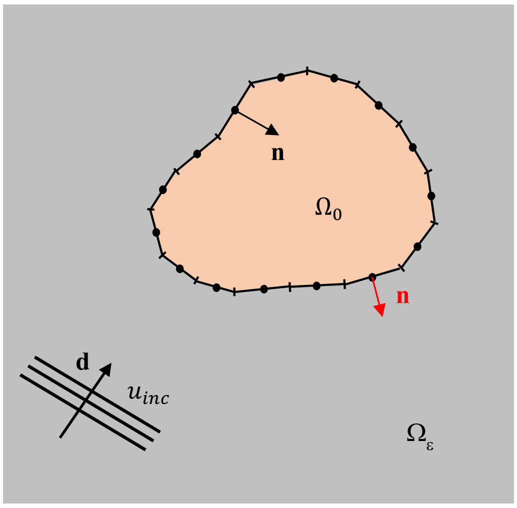


Figure 10: Boundary element discretization of the problem with a pre-existent obstacle.

It is also worth to notice that although the above formulation has been introduced for a single pre-existent obstacle, its extension to multiple obstacles is immediate. In such a case, it is only necessary to assimilate Γ to the union of boundaries of all obstacles.

In turn, expression (36) is also used to compute the sound pressures at the receptors, $\mathbf{u}(\mathbf{x}^k)$, which will later used to pose the adjoint problem.

Penetrable obstacles. The BEM setup for penetrable obstacles needs of multiple domains in order to account for the propagation within the obstacles. For the sake of simplicity, a formulation for a single pre-existent obstacle will be introduced next; its extension to multiple pre-existing obstacles is straightforward.

Two domains are considered: the external infinite domain, Ω_e, and that of an embedded pre-existent obstacle, Ω_0, as in Figure 3(b). A simple strategy for developing the general BEM model is to apply the boundary element procedure to each domain independently, and then combine their systems of equations using the compatibility and continuity conditions at the interface.

As shown in Figure 10, the two domains use the same boundary element mesh, but they have opposite normal vectors (this is because both are external normals of their corresponding domains). The integral formulation for the exterior problem is that in equation (27). Analogously, for the embedded domain is

$$c^0(\mathbf{x}')u^0_{sc}(\mathbf{x}') + \int_{\Gamma^0} q^*_0(\mathbf{x},\mathbf{x}')u^0_{sc}(\mathbf{x})d\Gamma(\mathbf{x}) = \int_{\Gamma^0} u^*_0(\mathbf{x},\mathbf{x}')q^0_{sc}(\mathbf{x})d\Gamma(\mathbf{x}), \qquad (40)$$

where fundamental solutions u^*_0 and q^*_0 are the same in (28) and (29), respectively, but with λ_e replaced by λ_0.

The discretized matrix form of equation (40) is

$$\mathbf{H_0 u^0_{sc}} = \mathbf{G_0 q^0_{sc}}. \qquad (41)$$

If one adopts the pressure and the flux of Ω_e as reference values (which is equivalent to say that the normal on the interface is the is the normal to Ω_e), the compatibility and equilibrium conditions in equations (5) results in

$$\mathbf{u_{sc}} = \mathbf{u^0_{sc}} \tag{42}$$

and

$$\mathbf{q_{sc}} = -\alpha \mathbf{q^0_{sc}}. \tag{43}$$

These conditions can be introduced in (33) and (41), which can now be written together as follows:

$$\begin{bmatrix} \mathbf{H} \\ \mathbf{H_0}. \end{bmatrix} \mathbf{u_{sc}} = \begin{bmatrix} \mathbf{G} \\ -\alpha \mathbf{G_0} \end{bmatrix} \mathbf{q_{sc}} + \begin{bmatrix} \mathbf{u_{inc}} \\ \mathbf{0} \end{bmatrix} \tag{44}$$

The system above can be rewritten as

$$\begin{bmatrix} \mathbf{H} & -\mathbf{G} \\ \mathbf{H_0} & \alpha \mathbf{G_0} \end{bmatrix} \begin{bmatrix} \mathbf{u_{sc}} \\ \mathbf{q_{sc}} \end{bmatrix} = \begin{bmatrix} \mathbf{u_{inc}} \\ \mathbf{0} \end{bmatrix}, \tag{45}$$

where the matrix on the left-hand side has dimensions $2 \cdot nel \times 2 \cdot nel$ and the vector containing $\mathbf{u_{sc}}$ and $\mathbf{q_{sc}}$ is of length $2 \cdot nel$. The system is then solved to obtain the pressures and fluxes on the obstacle boundary.

The above formulation can be easily extended to handle multiple obstacles, which could have different penetration coefficients α. Moreover, sound-hard and penetrable obstacles can be combined into a single BEM scheme. It is also interesting to point out that when more obstacles are included, the system of equations (45) tend to have large number of zero submatrices, which improves computational efficiency.

Once pressures and fluxes on the obstacle boundary are known, total pressures and pressure gradients at the domain points in Ω_e are computed using their boundary integral representations. Their discretized expressions are

$$u_{sc}(\mathbf{x}^l) = \sum_{j=1}^{nel} \int_{\Gamma_j} u^j_{sc}(\mathbf{x}) q^*(\mathbf{x}, \mathbf{x}^l) d\Gamma(\mathbf{x}) + \sum_{j=1}^{nel} \int_{\Gamma_j} q^j_{sc}(\mathbf{x}) u^*(\mathbf{x}, \mathbf{x}^l) d\Gamma(\mathbf{x}) + u_{inc}(\mathbf{x}^l, \mathbf{d}) \tag{46}$$

and

$$u_{sc,m}(\mathbf{x}^l) = \frac{\partial u_{sc}(\mathbf{x}^l)}{\partial \mathbf{x}^l_m} = \sum_{j=1}^{nel} \int_{\Gamma_j} u_{sc}(\mathbf{x}) \frac{\partial q^*(\mathbf{x}, \mathbf{x}^l)}{\partial \mathbf{x}_m} d\Gamma(\mathbf{x}) + \sum_{j=1}^{nel} \int_{\Gamma_j} q_{sc}(\mathbf{x}) \frac{\partial u^*(\mathbf{x}, \mathbf{x}^l)}{\partial \mathbf{x}_m} d\Gamma(\mathbf{x}) + \\ + \frac{\partial u_{inc}(\mathbf{x}^l, \mathbf{d})}{\partial \mathbf{x}_m} \text{ with } m = 1, 2. \tag{47}$$

Notice that in comparison to the sound-hard case, the above equations incorporate an extra term that accounts for the boundary flux. The discretized matrix forms of equations (46) and (47) are

$$\mathbf{u_{sc}}(\mathbf{x}^l) = \mathbf{A} \mathbf{u_{sc}} + \mathbf{B} \mathbf{q_{sc}} + \mathbf{u_{inc}} \tag{48}$$

and

$$\mathbf{u_{sc,m}}(\mathbf{x}^l) = \mathbf{A}'\mathbf{u_{sc}} + \mathbf{B}'\mathbf{q_{sc}}\mathbf{u_{inc,m}}. \tag{49}$$

The comments for the reusability of $\mathbf{A}$, $\mathbf{A}'$, $\mathbf{B}$ and $\mathbf{B}'$ and the most efficient implementation strategy are the same as for the sound-hard case.

6.3.3 Computation of the adjoint problem

The solution of the adjoint problem in Section 4 is decomposed –as it is done for the forward problem– into the incident and scattered parts, *i.e.*, $p = p_{inc} + p_{sc}$.

- The solutions for the incident pressure and the pressure gradient are the same of forward problem in Section 6.2.2. So, p_{inc} and $p_{inc,m}$ fields due to the point source in the k_{th} receptor are computed using equations (22) and (23), respectively.

- Te solution of the scattered field is analogous to the forward problem in Section 6.3.2, so p_{sc} and $p_{sc,m}$ can be computed using the same BEM schemes. For this, it is only necessary to specify the incident waves as point sources of amplitude $-[u(\mathbf{x}^k) - u_{meas}(\mathbf{x}^k)]$ at the receptors. Note that all BEM matrices (those to solve the boundary problem and to compute the solutions at the internal points) are the same of the forward problem, so they do not need to be recomputed.

6.3.4 Algorithm

The algorithm for the evaluation of the D_T is basically the same introduced in Section 6.2.3 for the empty space. However, it needs to be adapted to replace the analytical solutions by BEM ones.

i Set-up the problem as it depicted n Figure 3(b), and have available the data of the pressure measurements at the M receptors due to each of the N illumination directions. Store the pressure measurements in $\mathbf{u}^j_{\mathbf{meas}}(\mathbf{x}^k)$.

ii Use the BEM to solve the forward problem for each of the illumination directions and compute the corresponding pressures, $\mathbf{u}^j(\mathbf{x}^l)$, and pressure gradients, $\nabla\mathbf{u}^j(\mathbf{x}^l)$, at the P domain points. For this, choose the appropriate BEM formulation from Section 6.3.2, depending on whether the pre-existent obstacles are sound-hard or penetrable.

iii Use the BEM model to compute the pressures $\mathbf{u}^j(\mathbf{x}^k)$ at the M receptors for each of the illumination directions.

iv Follow the strategy in Section 6.3.3 to use the BEM model and the discrepancies at the receptors, $[u(\mathbf{x}^k) - u_{meas}(\mathbf{x}^k)]$, to calculate the adjoint pressure and pressure derivatives at the domain points for each illumination direction. Then, add up the contributions of the M receptors to compute $\mathbf{p}^j(\mathbf{x}^l)$ and $\nabla\mathbf{p}^j(\mathbf{x}^l)$, the adjoint fields at the domain points for each illumination direction.

v Select the adequate expression of D_T in Section 5 and use the results in $\mathbf{u}^j(\mathbf{x}^l)$, $\nabla\mathbf{u}^j(\mathbf{x}^l)$, $\mathbf{p}^j(\mathbf{x}^l)$ and $\nabla\mathbf{p}^j(\mathbf{x}^l)$, to compute the topological derivative at the domain points for each illumination directions: $\mathbf{D}^j_\mathbf{T}(\mathbf{x}^l)$.

vi Compute $D_T(\mathbf{x}^l)$, the overall topological derivative at the domain points as the sum of the contributions of all the illumination directions.

vii Draw a contour plot with the $D_T(\mathbf{x}^l)$ results over the search domain. The geometry of the hidden obstacle is that delimited by the largest negative values of the topological derivative.

6.3.5 Example

This example revisits the problem of the multiple obstacle identification in Section 6.2.4, but with the variation that square obstacle is visible while the circular one is hidden.

The problem is solved using the same wavelength, number and position of the receptors and number of incident waves as in the previous analysis. The BEM models for the computation of the forward and adjoint problems are discretized with elements of size equal to one-tenth of the wavelength.

The D_T results in terms of distance between the obstacles are plotted in Figure 11. Plots show the effectiveness of the procedure to identify the circular obstacle. It is interesting to see that the quality of the identification of the circle, especially the portion of the boundary that faces the square, is better than in the previous analysis. This result because, being the presence of the square known *a priori*, the identification does not suffer the shielding effects when the obstacles are close to each other.

The robustness of the method to measurement error is investigated using the same strategy as for the previous example. The case $d/r = 5\sqrt{2}$ in Figure 11b, is selected for the analysis. Figure 12 illustrates some results. It is found that the degradation of the results starts to be noticeable for noise amplitudes around 10 %, see Figure 10a. However, and like in the previous example, it is still possible to identify the hidden object with measurement errors of up to around 50%, , see Figure 12b.

7 Comments and Conclusions

A framework for the inverse analysis of acoustic scattering problems in open domains using the Boundary Element and Topological Derivative Methods has been presented in the previous sections. The modeling strategy takes advantage of the inherent characteristics of the BEM to effectively deal with problems with infinite domains. The tool has the capability to identify obstacles in initially empty domains and as well as in domains with pre-existent obstacles.

The performance of the tool is demonstrated for sound-hard obstacles. Such problems are solved using single-domain BEM models. An efficient BEM imple-

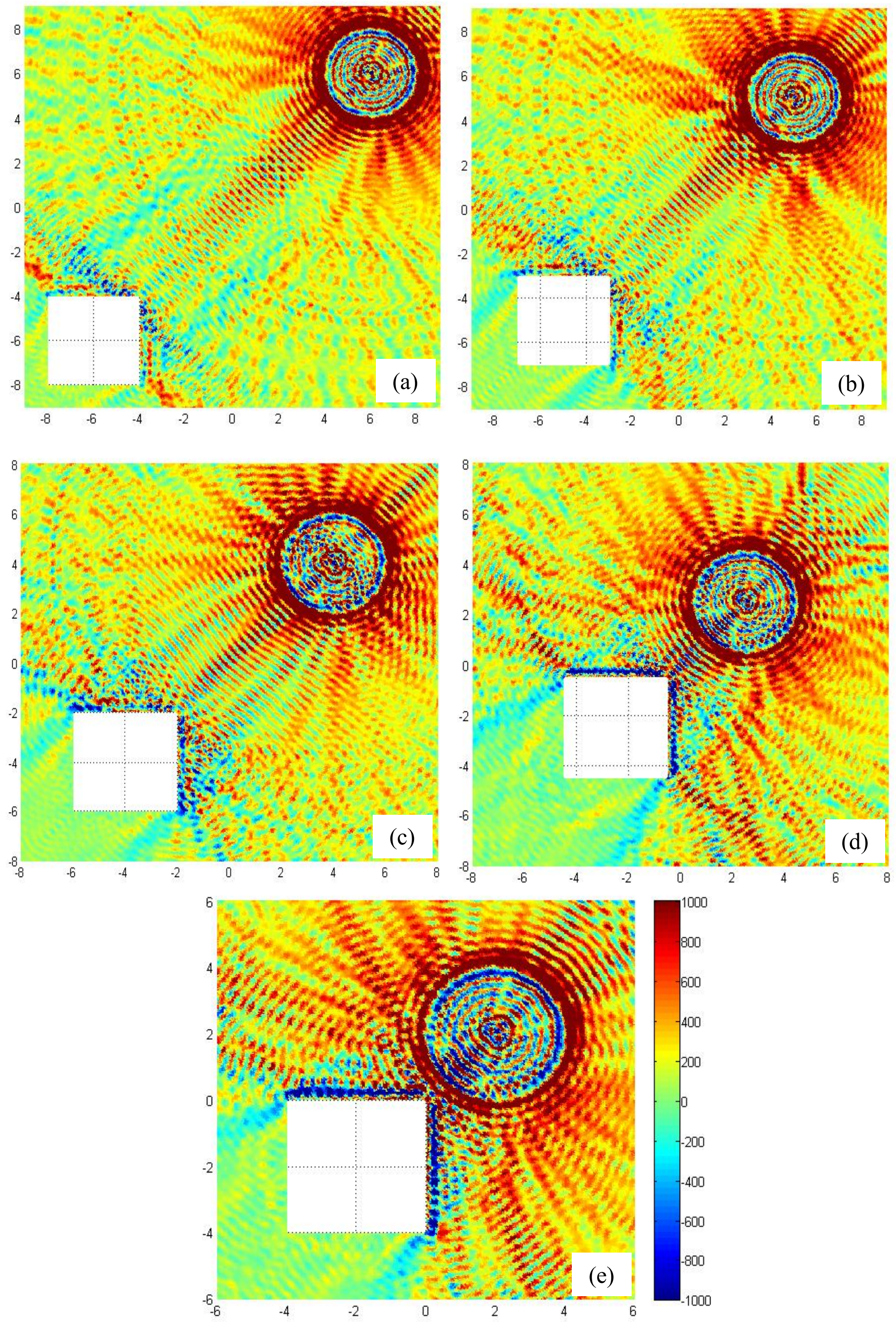


Figure 11: Identification of the square obstacle in the presence of a preexistent circular obstacle. Contour plots for the topological derivative results as function of the distance between obstables: (a) $d/r = 6\sqrt{2}$, (b) $d/r = 5\sqrt{2}$, (c) $d/r = 4\sqrt{2}$, (d) $d/r = 2.5\sqrt{2}$, (e) $d/r = 2\sqrt{2}$. The same color scale is used in all subfigures.

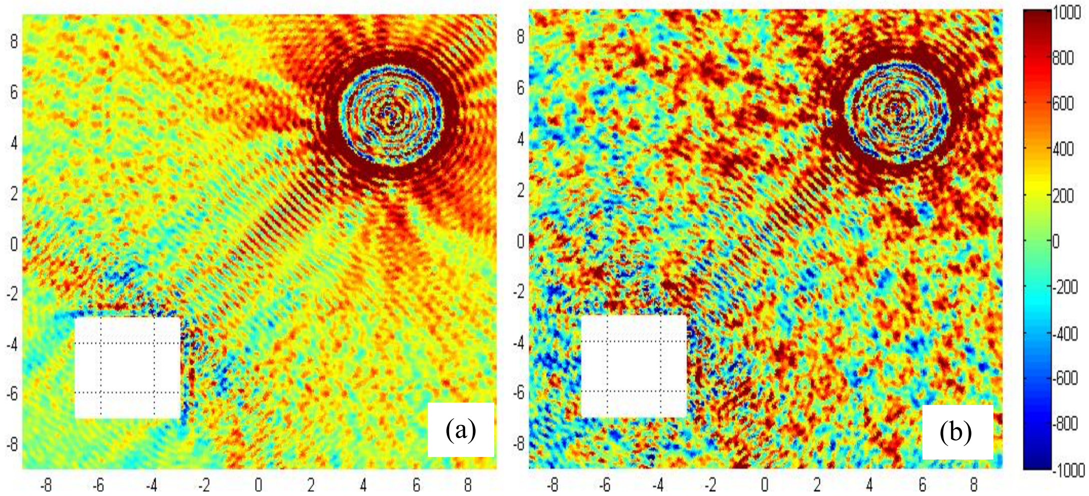


Figure 12: Assessment of the measuring error. Contour plots for the topological derivative results with (a) 10% and (b) 50% noise in the pressure values at the receptors.

mentation is proposed, which re-uses the matrices assembled for the solution of the forward problem for the computation of the adjoint problem.

Good quality results are obtained provided the lengths of the incident waves are smaller than one fifth of the characteristic size of the obstacles. An initial set-up with 50 sources and 100 receptors is recommended for practical identification problems. These settings can be adjusted later depending on the characteristics of each particular problem. The method shows low sensitivity to measurement error. Measurements with a noise of 50% the amplitude of the incident wave allow for positive identifications of the hidden obstacles. On the other hand, the flexibility of the method in the problem set-up makes it attractive for the implementation of adaptive strategies for the optimum placement of the sources and/or the receptors.

Although they have been practically shown for sound-hard obstacles, the formulations introduced for the inverse scattering problem and the BEM are general. They can handle problems involving penetrable obstacles (both, pre-existent and hidden ones), and even combinations of obstacles with different penetration coefficients. The solution of such problems will require the implementation of multi-domain BEM.

The extension of the analysis to three-dimensional problems leads to analogous formulae. See the works by Carpio and Rapún [2008b], Guzina and Bonnet [2006] and Nemitz and Bonnet [2008] for the details.

The method exhibits the potential to solve problems larger and more geometrically complex than those presented in this work. However, solving such problems would require faster algorithms to speed up not only the solution of limits but, more importantly, the post-processing on the BEM results at internal points for the computation of the D_T. In the BEM used in this work, the computational cost of solving direct and adjoint boundary problems is modest compared to post-processing at internal points.

The solution of the direct and adjoint boundary problems can be easily adapted to benefit from fast BEMs for acoustics, like the fast multipole in Nemitz and Bonnet [2008] and hierarchical matrices in Brancati et al. [2012]. On the other hand, iterative methods are an alternative to avoid the exhaustive sampling of the optimization for the computation of the D_T . A simply approach based on that introduced by Carpio and Rapún [2008a]. The idea is simple. Perform a first solution to the problem for a coarse array of internal points and use the points in which the D_T falls below a certain negative threshold to define a first guess the obstacle geometry. Then solve the problem for the new geometry and compute the T_D on a finer array of internal points. Such array of internal points only need to extend around the pre-existent obstacles and on zones with lowest values of the T_D in the previous step. Find the points with the lowest values of T_D for the new solution: those points which are close to any of the existing obstacles are included into that obstacle, whereas the points that far enough from the existing obstacles are used to create a new obstacle. Update the configuration and repeat the process until a stopping criterion (for instance, a threshold value of the D_T or a limit to the rate of change to the obstacle area) is fulfilled.

References

K. Abe, T. Fujiu, and K. Koro. A BE-based shape optimization method enhanced by topological derivative for sound scattering problems. *Engineering Analysis with Boundary Elements*, 34(12):1082–1091, dec 2010. ISSN 09557997. doi: 10.1016/j.enganabound.2010.06.017. URL `http://linkinghub.elsevier.com/retrieve/pii/S0955799710001682`.

M. Bonnet. Topological sensitivity for 3D elastodynamic and acoustic inverse scattering in the time domain. *Computer Methods in Applied Mechanics and Engineering*, 195(37-40):5239–5254, jul 2006. ISSN 00457825. doi: 10.1016/j.cma.2005.10.026. URL `http://linkinghub.elsevier.com/retrieve/pii/S004578250500544X`.

A. Brancati, M. H. Aliabadi, and V. Mallardo. A BEM sensitivity formulation for three-dimensional active noise control. *International Journal for Numerical Methods in Engineering*, 2012. doi: 10.1002/nme.

C. A. Brebbia, J. C. F. Telles, and L. C. Wrobel. *Boundary Element Techniques*. Springer-Verlag, Berlin, 1984.

A. Carpio and M. Rapún. Topological derivatives for shape reconstruction. *Inverse Problems and Imaging*, 1943:85–133, 2008a. URL `http://www.springerlink.com/index/q47k22718j526414.pdfhttp://link.springer.com/chapter/10.1007/978-3-540-78547-7_5`.

A. Carpio and M.-L. Rapún. Solving inhomogeneous inverse problems by topological derivative methods. *Inverse Problems*, 24(4):045014, jul 2008b. doi:

10.1088/0266-5611/24/4/045014. URL https://doi.org/10.1088/0266-5611/24/4/045014.

D. Colton. The inverse scattering problem for time-harmonic acoustic waves. *SIAM Review*, 26:323–350, 1984. URL https://doi.org/10.1137/1026072.

D. Colton and R. Kress. *Inverse Acoustic and Electromagnetic Scattering Theory*. Springer-Verlag, New York, 2013.

D. Colton and R. Kress. Looking back on inverse scattering theory. *SIAM Review*, 60:779–807, 01 2018. doi: 10.1137/17M1144763.

O. Dorn and D. Lesselier. Level set methods for inverse scattering. *Inverse Problems*, 22(4):R67–R131, aug 2006. ISSN 0266-5611. doi: 10.1088/0266-5611/22/4/R01. URL http://stacks.iop.org/0266-5611/22/i=4/a=R01?key=crossref.7850dcf318fd34e2a14ed6ff787ecfd2.

H. A. Eschenauer, V. V. Kobelev, and A. Schumacher. Bubble method for topology and shape optimization of structures. *Structural optimization*, 8(1):42–51, 1994. doi: 10.1007/BF01742933. URL https://doi.org/10.1007/BF01742933.

G. R. Feijóo. A new method in inverse scattering based on the topological derivative. *Inverse Problems*, 20(6):1819–1840, dec 2004. ISSN 0266-5611. doi: 10.1088/0266-5611/20/6/008. URL http://stacks.iop.org/0266-5611/20/i=6/a=008?key=crossref.5cbe06cbe4450026468cf3c8a96c16d7.

G. R. Feijóo, A. A. Oberai, and P. M. Pinsky. An application of shape optimization in the solution of inverse acoustic scattering problems. *Inverse Problems*, 20(1):199–228, feb 2004. ISSN 0266-5611. doi: 10.1088/0266-5611/20/1/012. URL http://stacks.iop.org/0266-5611/20/i=1/a=012?key=crossref.978136e24e80b02a4c17a91eb54e459e.

T. Gerlach and R. Kress. Uniqueness in inverse obstacle scattering with conductive boundary condition. *Inverse Problems*, 12(5):619–625, oct 1996. URL https://doi.org/10.1088/0266-5611/12/5/006.

B. B. Guzina and M. Bonnet. Small-inclusion asymptotic of misfit functionals for inverse problems in acoustics. *Inverse Problems*, 22(5):1761–1785, sep 2006. doi: 10.1088/0266-5611/22/5/014. URL https://doi.org/10.1088/0266-5611/22/5/014.

A. Kirsch and R. Kress. Uniqueness in inverse obstacle scattering (acoustics). *Inverse Problems*, 9(2):285–299, apr 1993. URL https://doi.org/10.1088/0266-5611/9/2/009.

N. Nemitz and M. Bonnet. Topological sensitivity and FMM-accelerated BEM applied to 3D acoustic inverse scattering. *Engineering Analysis with Boundary Elements*, 32(11):957–970, nov 2008. ISSN 09557997. doi: 10.1016/j.enganabound.2007.02.006. URL http://linkinghub.elsevier.com/retrieve/pii/S095579970800043X.

A. A. Novotny, R. A. Feijóo, E. Taroco, and C. Padra. Topological sensitivity analysis. *Computer Methods in Applied Mechanics and Engineering*, 192(7-8): 803–829, feb 2003. ISSN 00457825. doi: 10.1016/S0045-7825(02)00599-6. URL `http://linkinghub.elsevier.com/retrieve/pii/S0045782502005996`.

A. E. Sisamón, S. C. Beck, S. C. Langer, and A. P. Cisilino. Inverse scattering analysis in acoustics via the BEM and the topological-shape sensitivity method. *Computational Mechanics*, jul 2014. ISSN 0178-7675. doi: 10.1007/s00466-014-1051-z. URL `http://link.springer.com/10.1007/s00466-014-1051-z`.

L. C. Wrobel. *The Boundary Element Method, Volume 1: Applications in Thermo-Fluids and Acoustics.* John Wiley and Sons, Chichister, UK, 2002.

Chapter 9: Fundamental Concepts on Wavelet Transforms

Chapter details

Chapter DOI:

https://doi.org/10.4322/978-65-86503-71-5.c09

Chapter suggested citation / reference style:

Palechor, Erwin U. L., et al. (2022). “Fundamental Concepts on Wavelet Transforms”. *In* Jorge, Ariosto B., et al. (Eds.) *Model-Based and Signal-Based Inverse Methods*, Vol. I, UnB, Brasilia, DF, Brazil, pp. 331–356. Book series in Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity.

P.S.: DOI may be included at the end of citation, for completeness.

Book details

Book: Model-based and Signal-Based Inverse Methods

Edited by: Jorge, Ariosto B., Anflor, Carla T. M., Gomes, Guilherme F., & Carneiro, Sergio H. S.

Volume I of Book Series in:

Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity

Published by: UnB City: Brasilia, DF, Brazil Year: 2022

DOI: https://doi.org/10.4322/978-65-86503-71-5

Fundamental Concepts on Wavelet Transforms

Erwin Ulises Lopes Palechor[1*], Ramon Saleno Yure Costa Silva[2], Marcus Vinicius Girão de Morais[2,4], Luciano Mendes Bezerra[2,3] and Ariosto Bretanha Jorge[4]

[1] Science and Technology Center, Federal University of Cariri, Brazil. e-mail: erwin.lopez@ufca.edu.br
[2] Faculty of Technology, University of Brasilia, Brazil. e-mail: ramon.silva@unb.br; mvmorais@unb.br
[3] Post-Graduate Program in Structures and Civil Construction, University of Brasilia, Brazil. e-mail: lmbz@unb.br
[4] Post-Graduate Program in Integrity of Engineering Materials, University of Brasilia, Brazil. e-mail: ariosto.b.jorge@gmail.com

*Corresponding author

Abstract

This chapter presents some fundamental concepts on wavelet transforms, both discrete and continuous, each are useful to understand the ability of the wavelet transform to capture sudden changes in material properties or parameters along the span of structural components being assessed. The chapter also discusses some families of wavelet transforms, some wavelet properties and regularization aspects.

Keywords: Wavelet Transform, Continuous Wavelet, Discrete Wavelet, Wavelet Family Function, Wavelet Packet.

1 Introduction of Wavelet Transform

Wavelet comes from the French word “ondalette”, which means small wave. Wavelets were first mentioned in the appendix of Haar's thesis (Haar, 1910). Haar wavelets remained anonymous for several years until in the 1930s (Reis, 2018). Working independently, various groups made research using the wavelet functions using a base and varying scales. On that occasion, using the Haar wavelets as a basis, Paul Levy investigated the Brownian motion (Reis, 2018). He showed that Haar-based functions were better than Fourier-based functions for studying the small Brownian motion with intricate details.

For a long period, Haar wavelets continued to be the only known orthonormal wavelet basis. In 1985, Mallat gave wavelets a big boost through his/her work in digital image processing. Meyer (1989), inspired by Mallat's results, constructed the first non-trivial (smooth) wavelet. Unlike the Haar wavelet, Meyer wavelet is continuously differentiable, but do not have compact support. In 1990, Ingrid Daubechies used Mallat's work to build a set of smooth wavelet orthonormal bases with compact supports. Daubechies' works are the foundation of current Wavelets applications (Palechor *et al.*, 2019; Silva *et al.*, 2019).

For a good understanding of wavelet analysis, it is necessary to start with the analysis of simple techniques. In practice, many signals can be represented in the time domain and in the frequency domain.

The representation of the signal in the frequency domain is obtained by applying the Fourier transform (FT) to the original signal expressed in the time domain. The result of this transformation is a set of frequencies that characterize the original signal. But the question arises: why do we need frequency information? Often the information that is needed cannot be seen in the time domain, but in the frequency domain. In other cases, the most important part of the signal information is “hidden” in their frequencies. This transformation can be applied to non-stationary signals; that is, signals that change their parameters over time.

In many applications, periodic oscillatory behavior is intermittent. For these analyses, the FT is no longer the most suitable as the series must be stationary (its parameters remain constant over time). As an alternative to solve this problem, there is the Short-Time Fourier Transform (STFT) (a generalization of the Windowed Gabor Transform) (Michel Misiti *et al.*, 1997). Its application allows the signal information achievement in time and frequency. The STFT methodology has some difficulties in determining the ideal time domain window width (e.g., interval of time that is slid along time series).

Wavelet analysis takes a different approach as it is based on the idea that any signal can be broken down into a series of basic functions called “wave”. This technique also allows the use of basic functions with variable size, and the use of long and short time and space intervals to capture the necessary information (Figure 1).

A great advantage provided by wavelets is the ability to perform local analysis. Consider a sinusoidal signal with a small break, so small that it is barely visible as in Figure 2.

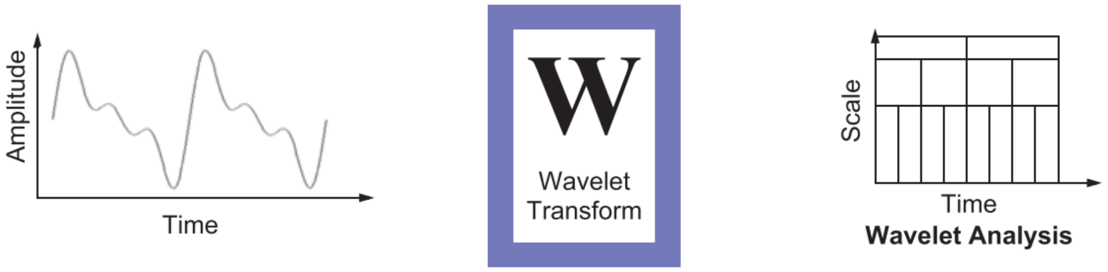


Figure 1 – Wavelet Transform (adapted from Misiti et al. 1997).

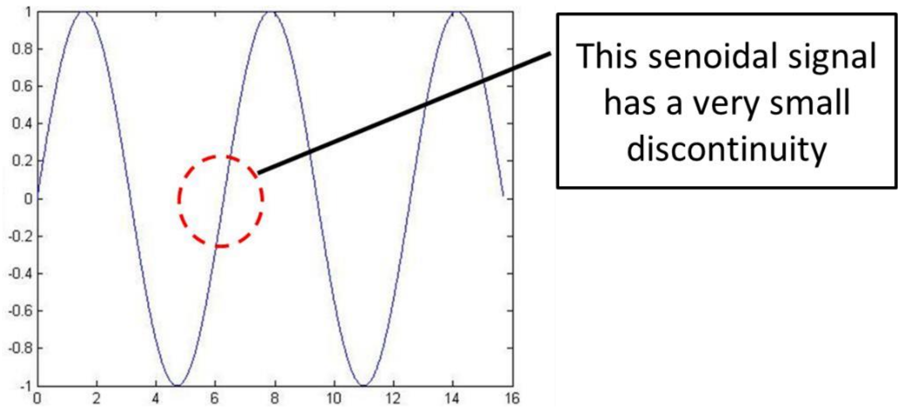
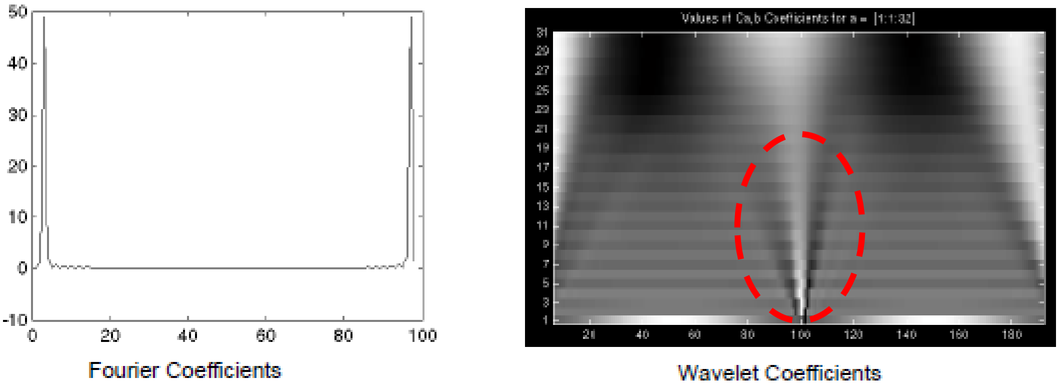


Figure 2 – Discontinuity on signal (adapted from M Misiti *et al.*, 1997)

A graph of Fourier amplitudes (Fourier coefficients) as a function of frequency, obtained by the FT of this sinusoidal signal, does not show anything, except the peaks due to the characteristic frequencies of sinusoidal signal, Figure 3(a). However, a portion of wavelet coefficients clearly shows the exact location in time of the discontinuity generated by the signal disturbance, Figure 3(b).



(a) (b)

Figure 3 – Comparation between Fourier coefficients (Weeks, 2007)**(a) and Wavelet coefficients** (M Misiti *et al.*, 1997) **(b)**

A wavelet is an effectively time-limited waveform with a zero-value averaging. One of the main advantages provided by the Wavelet Transform is the ability to perform local analysis; that is, it can analyze a restricted data series of a larger signal. This analysis can reveal aspects that other signal processing techniques cannot obtain, aspects such as: trends, points of degradation, discontinuities. In the case of damage identification, these discontinuities may be caused, for example, by cracks (Ovanesova, 2000; Ovanesova and Suárez, 2004).

The wavelet decomposition consists of calculating a “resemblance index” between the signal and the wavelet function. If the index is large then the similarity is strong, otherwise the similarity is weak. The Wavelet transform of a signal $\boldsymbol{f(x)}$, is the family $\boldsymbol{C(a,b)}$, which depends on two indices $\boldsymbol{a}$ and $\boldsymbol{b}$, the values $\boldsymbol{C(a,b)}$ are called coefficients, that will be described section 3.

Wavelets have been widely used to analyze time-domain signals. For the Wavelet analysis of spatial-domain signals, we can simply replace time with a spatial coordinate f(x), corresponding to vibration modes or displacements due to static load (Wu and Wang, 2011).

Similar to the windowed Fourier transform, the unidimensional Wavelet Transform projects a signal into two-dimensional space. The Wavelet Transform of the signal $f(x)$ is defined as:

$$W_{\psi}^{f}(a,b) = |a|^{-1/2} \int_{-\infty}^{\infty} f(x)\psi^{*}\left(\frac{x-b}{a}\right) dx \tag{1}$$

where $\psi^{*}(.)$ indicates the complex conjugate of $\psi(.)$ it is assumed that the mean value of the function $\psi(x)$ is null:

$$\int_{-\infty}^{\infty} \psi(x)\, dx = 0 \tag{2}$$

In both Short-Time Fourier Transform and the Wavelet Transform, the signal $f(x)$ is multiplied by a function of two variables. In the case of Short-Time Fourier transform variables, the function is as follows:

$$w^{w,\tau}(x) = \frac{1}{2\pi} w(x-\tau)e^{-iwx} \tag{3}$$

The respective function for the wavelet transform is given by:

$$\psi^{a,b}(x) = |a|^{-1/2}\psi^{*}\left(\frac{x-b}{a}\right) \tag{4}$$

The $\psi^{a,b}$ functions are called wavelets or mother wavelet functions. Short-time Fourier transform functions usually fluctuate and decay rapidly. In contrast to the $\psi^{a,b}(x)$ functions, the number of oscillations remains constant with window changes. This means that a wavelet is “scaled” along the axis of time (or space). For short-time Fourier transform, the size of the window remains constant while the number of oscillations changes. This principle is illustrated in Figure 4.

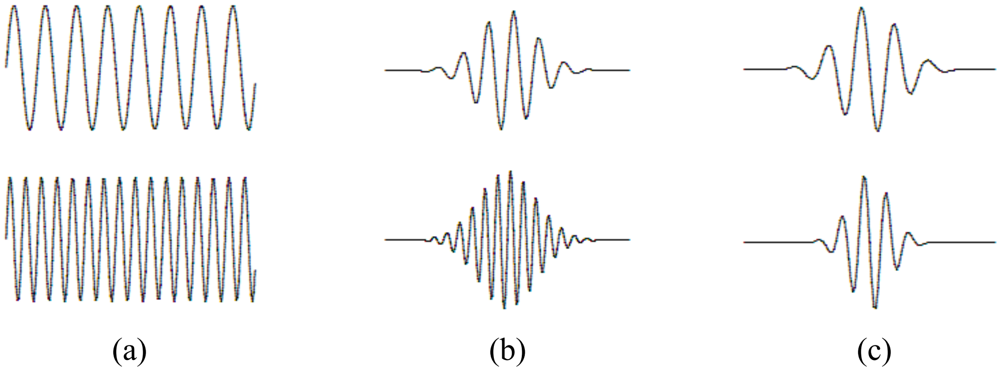


Figure 4 – Comparation between (a) Fourier transform, (b) short-time Fourier transform, (c) Wavelet transform (Michel Misiti *et al.*, 1997)**.**

For the wavelet transform analysis, $\psi(x)$ is the complex function of values located in the spatial domain x. The function $\psi(x)$ is the mother wavelet that generates the wavelet coefficients by shifting and scaling (Wu and Wang, 2011). The shifting from mother wavelet can be defined as:

$$\psi_{a,b}(x) = 2^{\frac{a}{2}}\psi(2^a x - b) \tag{5}$$

where a and b are scaling and shifting, respectively.

For a spatial signal $f(x)$ in the interval $[a, b]$, its wavelet transform is given by:

$$C_{a,b} = \int_{-\infty}^{\infty} f(x)\,\psi_{a,b}(x)dx \tag{6}$$

where $C_{a,b}$ is wavelet coefficient for mother wavelet $\psi_{a,b}\ (x)$ with scale a, and position b (Wu and Wang, 2011).

Wavelet Transform includes Continuous Wavelet Transform (TCW) and Discrete Wavelet Transform (TDW). The main advantage of TCW is its ability to provide time and scale information (Li *et al.*, 2009). The difference between the two transformations is in the form of scale representation (Table 1):

- In continuous analysis, the scale varies almost continuously between 2^1 and 2^5, for example. When a scale is small, only small details are analyzed. This is why continuous analysis is often easier to interpret (Ovanesova, 2000; Ovanesova and Suárez, 2004);
- In discrete analysis, the scale is dyadic, for example, 2^1, 2^2, 2^3, 2^4, and 2^5. Each level coefficient k is repeated 2^k times. This is why discrete analysis guarantees saving of coding space and is sufficient for synthesis (Ovanesova, 2000; Ovanesova and Suárez, 2004).

Table 1 – TCW and TDW differences (Ovanesova, 2000; Ovanesova and Suárez, 2004)

Continuous Time	**Continuous Time**	**Discrete Time $(\Delta = 1)$**
Continuous Analysis	**Continuous Analysis**	**Discrete Analysis**
$C_{a,b} = \int_R S(t)\frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right)dt$ $a \,\epsilon\, R^+, b \,\epsilon\, R$	$C_{a,b} = \int_R S(t)\frac{1}{\sqrt{a}}\psi\left(\frac{t-b}{a}\right)dt$ $a = \Delta 2^j, b = \Delta k 2^j$ $j,k \,\epsilon\, Z^2$	$C_{j,k} = \sum_{n \epsilon Z} s(n) g_{j,k}(n)$ $a = 2^j, b = k2^j$ $j \,\epsilon\, N, k \,\epsilon\, Z$

2 Wavelets Properties

Wavelet functions have different properties suitable for certain purpose. According to Estrada (2008), the most relevant properties of the wavelet function to enable damage detection are:

Propriety I – **Orthogonality and Biorthogonality:** Two functions $u(x)$ and $g(x)$ are orthogonal if their scalar product is null:

$$\langle u(x), g(x)\rangle = \int_a^b u(x)g^*(x)\,dx = 0 \tag{7}$$

where $g^*(x)$ is the complex conjugate of function $g(x)$. The term "biorthogonal" refers to two different bases orthogonal to each other, but the bases do not form an orthogonal set of functions.

These properties ensure fast determination of wavelet coefficients. Unfortunately, not all wavelet functions have the two following properties.

Propriety II – **Compact Support:** support of a function is a set of points where the function is non-zero. A function has compact support if the adherence of the set, of non-null points, forms a closed and delimited set. This property means that the wavelet function does not assume a zero value for finite intervals, allowing for a more efficient representation of signals that have localized characteristics.

Propriety III – **Vanishing Moment:** More precisely, if the mean value of $x^k\psi(x)$ is equal to zero, for $k = 0,1,\dots,n$, where $\psi(x)$ is the corresponding scaling wavelet function. Then the wavelet function has $n+1$ vanishing moments and polynomials of degree n are suppressed by this wavelet function. This property determines the maximum degree of the polynomial that can be approximated. This property is used to select the most suitable mother wavelet function for damage detection.

PROPRIETY IV – **Regularity:** is the number of times a function is differentiable at point x_0. Singularities in a function can be detected by this property of regularity. $\boldsymbol{s}$ is the regularity of the function f; if the derivative of order m of f , at *x₀*, approaches $|x - x_0|^r$ locally around x_0, then $\boldsymbol{s} = m + r$, with $0 < r < 1$.

Table 2 – Proprieties of the mother wavelet functions.

Propriety	**morl**	**mexh**	**meyr**	**haar**	**dbN**	**symN**	**coifN**	**BiorNr.Nd**
Regular Infinitely	x	x	x					
Compact Orthogonal Support				x	x	x	x	
Compact Biorthogonal Support								x
Orthogonal			x	x	x	x	x	
Biorthogonal			x	x	x	x	x	x
Number of Arbitrary Null-Momentum					x	x	x	x
Continuous Transform	x	x	x	x	x	x	x	x
Discrete Transform			x	x	x	x	x	x

According to these properties, the most known mother wavelets are classified into (Ovanesova and Suárez, 2004):

- The wavelets functions Haar, Daubechies of *n*-th order, Meyer, Symlets of *n*-th order and Coiflets of *n*-th order are examples of orthogonal mother wavelets.
- The wavelets functions Haar, Daubechies of *n*-th order, Symlets of *n*-th order, and Coiflets are mother wavelets with compact support.
- The wavelets functions Daubechies of *n*-th order, Symlets of *n*-th order and Coiflets of *n*-th order are examples of mother wavelets with a number arbitrary of vanishing moments.
- The wavelets functions Morlet, Meyes and Gaussian are regular. Alternatively, the functions Daubechies of nth order, Symlets of *n*-th order and Coiflets of *n*-th order are mother wavelets with weak regularity.

3 Continuous Wavelets Transform (CWT)

The Fourier Transform is defined as the projection of the signal of the product of $f(t)$ and the exponential complex (orthogonal) function at time, as described in expression:

$$F(w)=\int_{-\infty}^{+\infty} f(t)e^{-jwt}dt \tag{8}$$

The TF results are Fourier coefficients, for each frequency ω. The reconstitution of the original signal $f(t)$ is obtained by multiplication of Fourier coefficients by harmonic function $\exp(\jmath\omega t)$ (Figure 5).

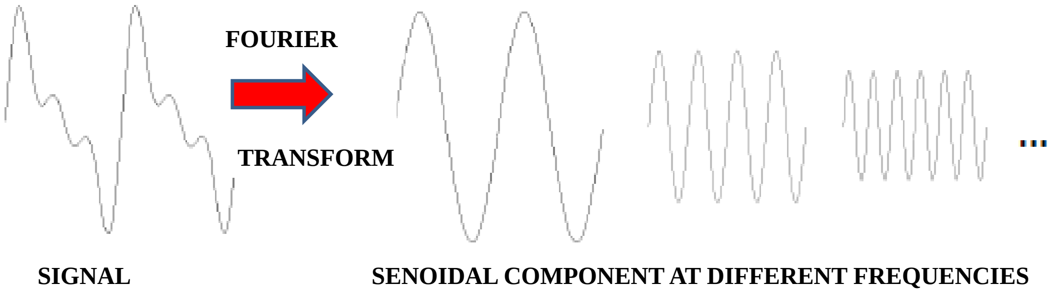


Figure 5 – Decomposition process of Fourier Transform (adapted from Misiti *et al.*, 1997)

Likewise, Continuous Wavelet Transform (CWT) is defined as the sum over entire time (or entire space) of the signal multiplied by mother wavelet function for a specific scale and position.

$$C(scale, position)=\int_{-\infty}^{+\infty} f(x)\psi(scale, position)dx \tag{9}$$

Results of CWT is the wavelet coefficient C. Such coefficients depend on a specific scale and position. Multiplying each wavelet coefficient by appropriately scaled mother wavelet produces the original signal (Figure 6).

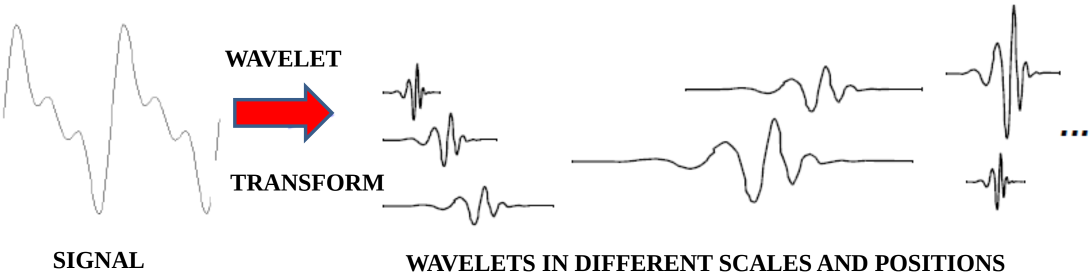


Figure 6 – Wavelet Transform Process (adapted from M Misiti *et al.*, 1997)**.**

3.1 Scale

Scaling a wavelet means stretching or compressing a function. Figure 7 present the effect of scale factor for a function$f(t)$, if the signal was a sinusoids.

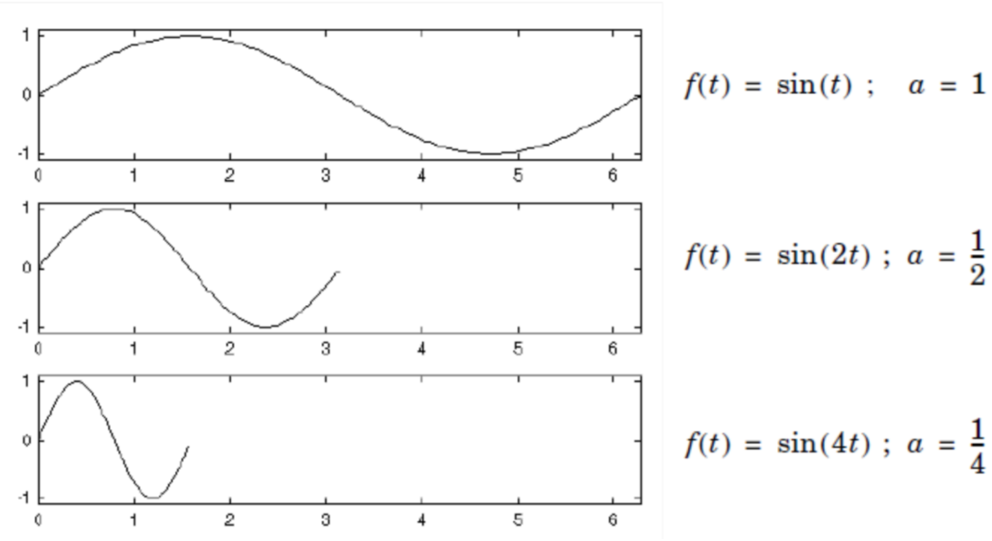


Figure 7 – Scale of Wavelet Transform (Weeks, 2007)**.**

3.2 Shifting

Shifting wavelets is simply delaying its onset along of signal. Mathematically, delaying $\boldsymbol{k}$ times, a function $\boldsymbol{f(t)}$ is represented by $\boldsymbol{f(t-k)}$ (Figure 8).

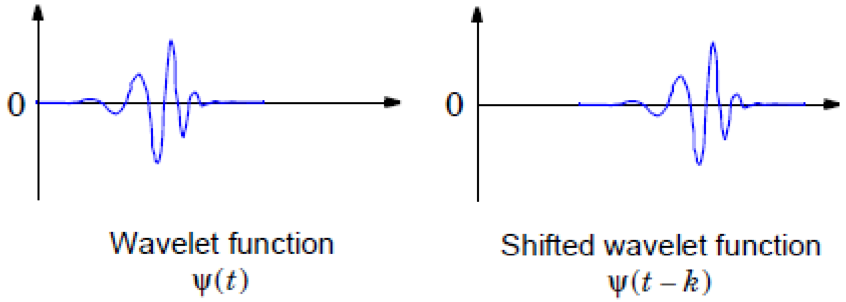


Figure 8- Wavelet function (Weeks, 2007)**.**

The signal CWT is the integration over signal multiplied by scale a and offset b. This process produces wavelet coefficients function of dimension a and position b. The option of CTW coefficients is performed by an algorithm in five steps:

1. Choose a mother wavelet and compare it to signal interval at the beginning of original signal.
2. Calculate a coefficient C representing the similarity ("resemblance index") between mother wavelet and original signal at analyzed interval. Note that the resulting coefficient depend on chosen wavelet shape (Figure 9).

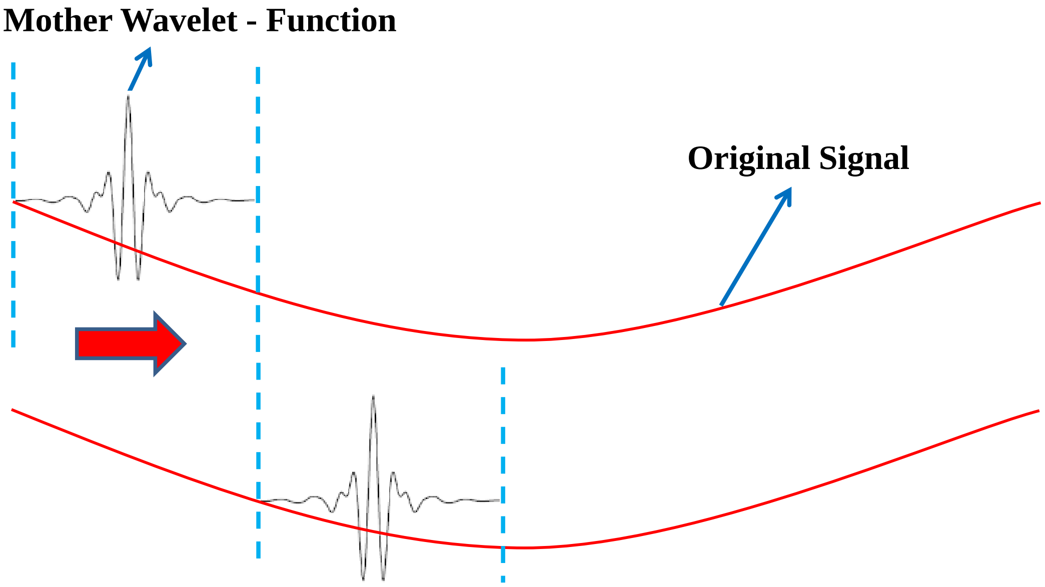


Figure 9 – Schematic calculus of wavelet coefficients (adapted from M Misiti *et al.*, 1997)**.**

3. Shift mother wavelet to the right and repeat steps 1 and 2 until it covered the entire signal (Figure 10).
4. Scale mother wavelet on analyzed stretch and repeat steps 1 to 3.

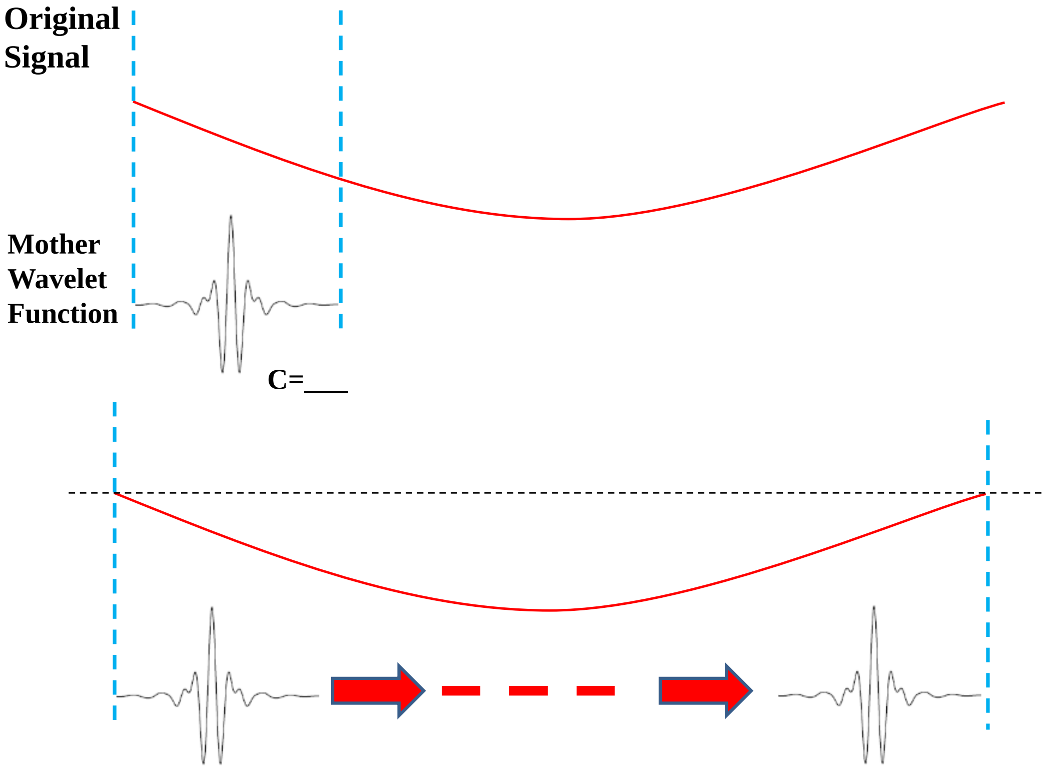


Figure 10 – Illustration of Wavelet scale (adapted from M Misiti *et al.*, 1997)**.**

5. Repeat steps 1 to 4 for all scales.

As a result, CWT produces wavelet coefficients for different scales. The x-axis represents position along the signal (time or space) and the y-axis represents the scale a. The color at each point (x, y) in space signal-scale represents the magnitude of wavelet coefficients C (Figure 11). Figure 12 shows the wavelet coefficients´ map generated by TCW.

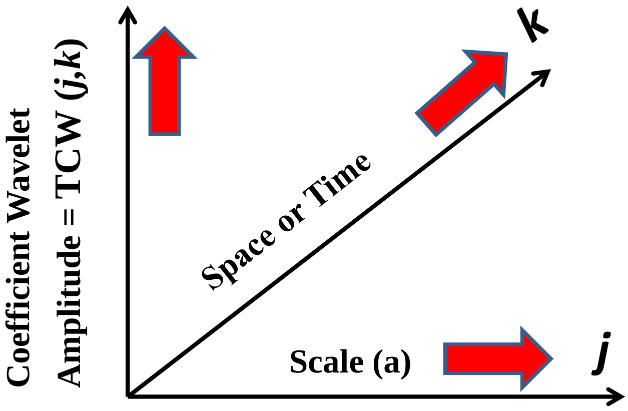
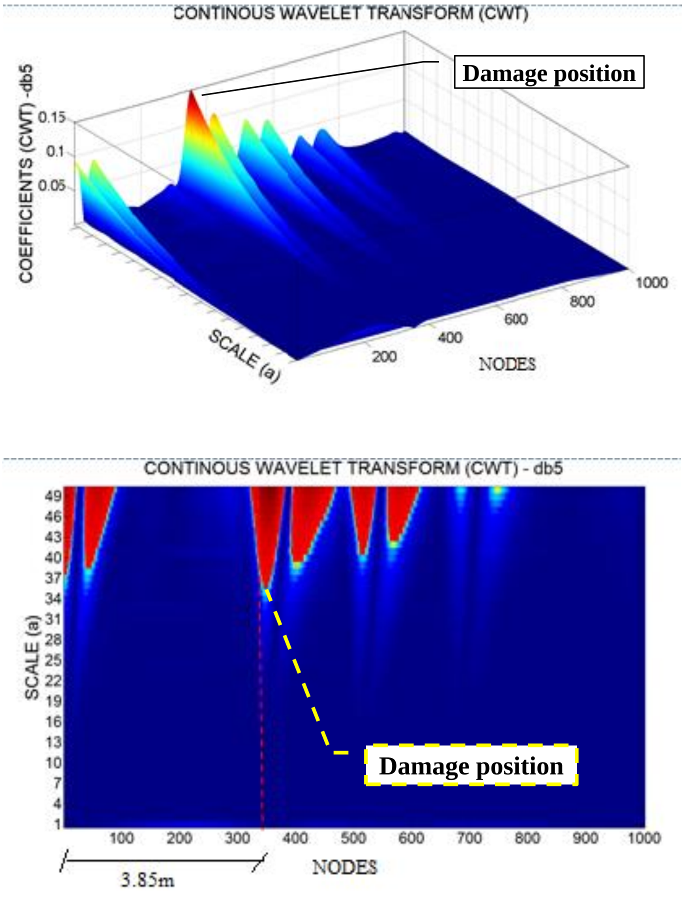


Figure 11 – Axis explanation of TCW graphs (modified by Gutierrez, 2002)**.**



(a) 3D

(b) 2D

Figure 12 – Examples of TCW graphs (Silva *et al.*, 2019)**.**

For CWT, the coefficient wavelet $\psi(a, b)$ surface can be described as an analytical function, function of parameters a (scale) and b (shifting) changing continuously over all space $\mathbb{R}^2$ (excluding $a = 0$). The CWT is defined by the following equation:

$$W_\psi^f(a, b) = |a|^{-1/2} \int_{-\infty}^{\infty} f(x)\psi\left(\frac{x-b}{a}\right) dx \tag{10}$$

3.3 CWT Analytical Example

Calculate the CWT of the following function

$$f(x) = e^{\frac{x^2}{2}} \tag{11}$$

using the Mexican hat wavelet (Ricker wavelet):

$$\begin{aligned} W_\psi^f(a, b) &= \frac{1}{\sqrt{a}} \int f(x)\psi^*\left(\frac{x-b}{a}\right) dx \\ &= \frac{1}{\sqrt{a}} \int e^{\frac{x^2}{2}} \left(1 - \left(\frac{x-b}{a}\right)^2\right) e^{-\frac{\left(\frac{x-b}{a}\right)^2}{2}} dx \end{aligned} \tag{12}$$

Scaling for $a = 1$ and shifting for $b = 0$, the coefficients can be obtained by the following expressions:

$$W(1{,}0) = \frac{1}{\sqrt{1}} \int e^{\frac{x^2}{2}} \left(1 - \left(\frac{x-0}{1}\right)^2\right) e^{-\frac{\left(\frac{x-0}{1}\right)^2}{2}} dx \tag{13}$$

In other terms,

$$\begin{aligned} W(1{,}0) &= \int e^{\frac{x^2}{2}} (1 - x^2) e^{-\frac{x^2}{2}} dx \\ &= \int (1 - x^2) e^0 \, dx \\ &= x - \frac{x^3}{3} + C \end{aligned} \tag{14}$$

The constant C appears once we have an indefinite integral. If we estimate $W(1{,}0)$ we compute $x = 10, -323{,}33$ plus the constant.

4 Discrete Wavelet Transform

The wavelet coefficients calculus on a continuum scale is a complicated task due to the generation of a fair amount of data. To minimize this task, only a discontinuous subset of scales and positions is chosen. This subset of scales and positions chosen are based on powers of 2 (dyadic scales) which is more efficiently computed. This approximate kind of wavelet analysis is called Discrete Wavelet Transform (DWT) (Ovanesova, 2000; Ovanesova and Suárez, 2004).

For this purpose, we define the scale $a = 2^j$ and the shifting $b = k(2^j)$, where $(j, k) \in Z$ and Z is integer set. Using these discrete parameters, DWT is given as:

$$TDW_{j,k} = 2^{-j/2} \int_{-\infty}^{\infty} f(x)\psi(2^{-j}x - k)dx = \int_{-\infty}^{\infty} f(x)\psi_{j,k}(x)dx \tag{15}$$

The following three-step algorithm describes the stages for damage detection on a structure using DWT:

1. Obtain a signal or structure response associated with the complete structure or exam just a specific area of the structure.
2. Calculate the wavelet coefficients, performing signal DWT to different scales. Wavelet coefficients $C_{j,k}$ are obtained by:

$$C_{j,k} = \int_Z f(x)\,\psi_{j,k}(x)dx \tag{16}$$

where the analyzed signal $f(x)$ is described by a scale $j \in N$ and position $k \in Z$; N and Z are the set of all positive scale integers and all position integers, respectively; and $\psi_{j,k}(x)$ is a mother wavelet and can be expressed by:

$$\psi_{j,k}(x) = 2^{0.5j}\,\psi_{j,k}(2^j x - k) \tag{17}$$

3. Plot the wavelet coefficient for each scale level.

The examination of the distribution of the wavelet coefficients for each level is done so that suddenly changes can be observed (*i.e.*, a peak) meaning local disturbance. If a peak is not caused by a known source, such as geometric or material discontinuity, the detected disturbance means that there is damage near to disturbance peak location.

5 Wavelet Family

Mathematically, a function $\psi(x)$, to be considered a mother wavelet, if and only if, it belongs to $L^2(R)$ space (Daubechies, 1992; Mehra, 2018) and satisfy admissibility conditions. Without much mathematical rigor, a mother wavelet is a function that oscillates, has finite energy, and has an average value of zero. The different families of wavelet functions are enumerated below, presenting the wavelet families used to damage detection in next chapters. One note exhaustive list of wavelet families are presented in Mehra (2018).

5.1 Haar Wavelet Family

Haar wavelet is the first, and the simplest of all wavelets. Haar wavelet resembles a step function. It is a special case of Daubechies wavelet, the db1 wavelet itself.

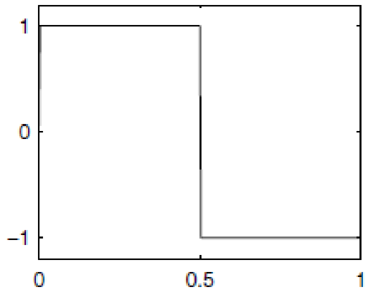
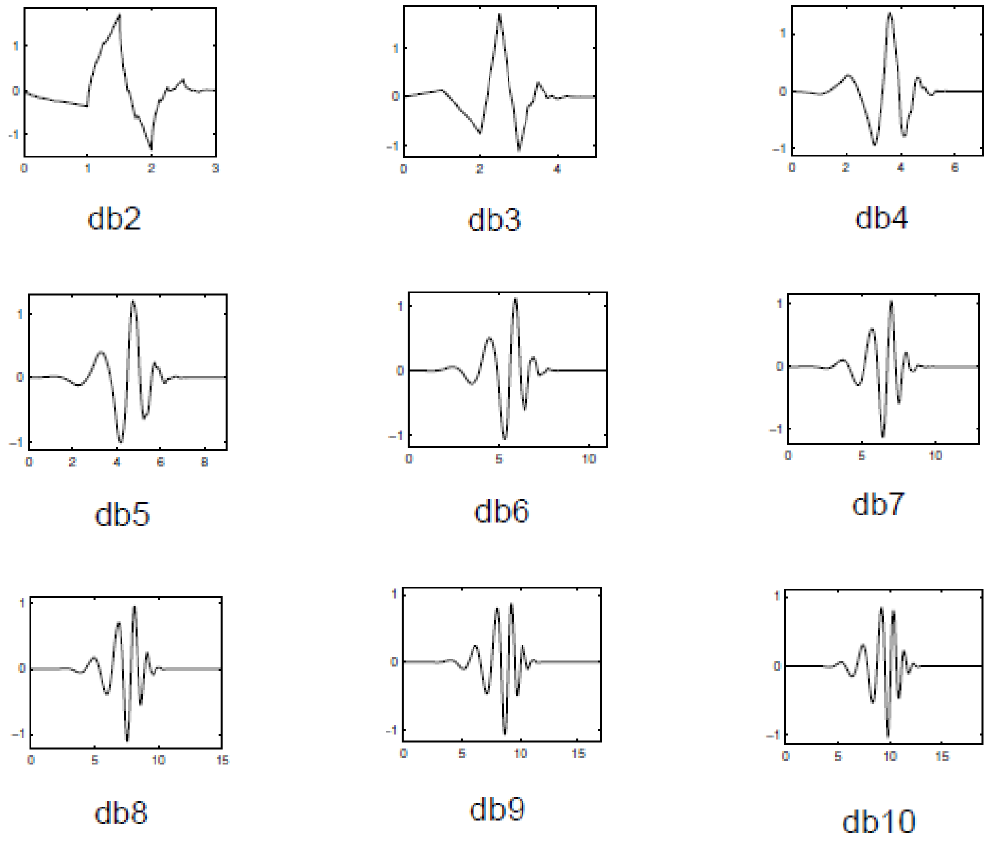


Figure 13- Harr Wavelet Family (Modified from M Misiti *et al.*, 1997)**.**

5.2 Daubechies Wavelet Family.

Ingrid Daubechies, one of the brightest stars in wavelet research world, invented the known orthonormal wavelets. The Daubechies wavelets names are also written as "dbN", where N is the order. As mentioned above, db1 wavelet is the Haar wavelet. Figure 14 show the next nine mother wavelet functions of Daubechies family.



db2 db3 db4

db5 db6 db7

db8 db9 db10

Figure 14- Daubechies Wavelets Family (Weeks, 2007)**.**

Orthogonal Daubechies wavelets, “dbN” are perfectly time compact. But in the frequency domain, they have a high degree of spectral superposition between the scales. The orthogonality is their main advantage. An error in the input signal does not increase after the transformation. This property ensures computational and numerical stability.

5.2.1 Coiflets Wavelet Family

Concepted by Daubechies by Coffman's request, the mother wavelet has $2N$ and scale function has $2N - 1$ vanishing moments, respectively. The two functions have a support length of $6N - 1$ (Figure 15).

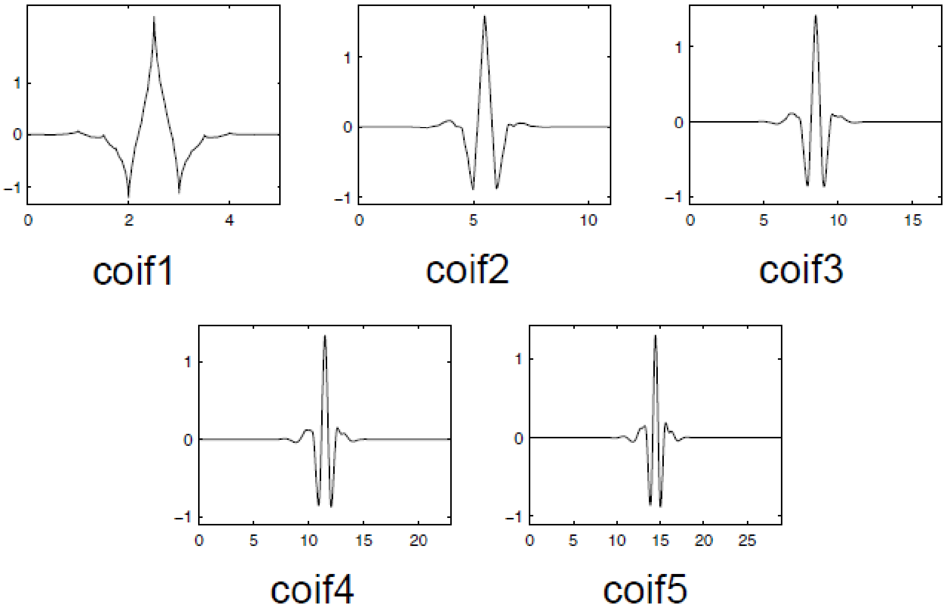


Figure 15- Coiflet Wavelets Family (Daubechies, 1992)**.**

5.3 Biorthogonal Wavelet Family

Biorthogonal wavelet bases (Daubechies, 1992) were conceived to obtain a symmetric wavelets family with compact support (De Souza *et al.*, 2007). Figure 16 shows some examples of biorthogonal wavelets.

5.4 Symlets Wavelet Family.

Proposed as dbN family modification by Daubechies, Symlet wavelets are almost symmetrical. The properties of dbN and Sym family are similar. But Symlet functions tend to be symmetric. Figure 17 presents examples of Symlet wavelet functions.

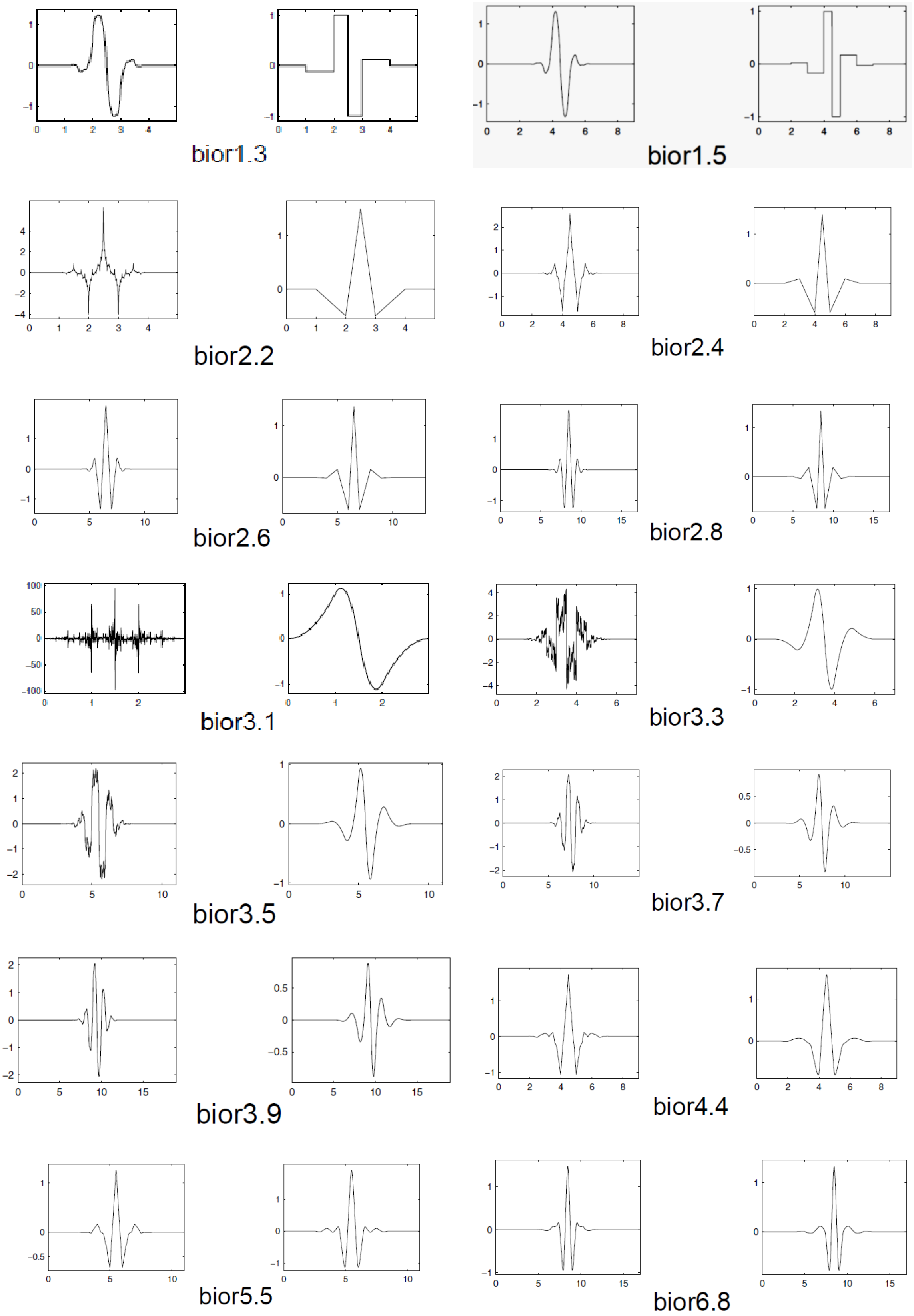


Figure 16- Daubechies Wavelets Family (Daubechies, 1992)**.**

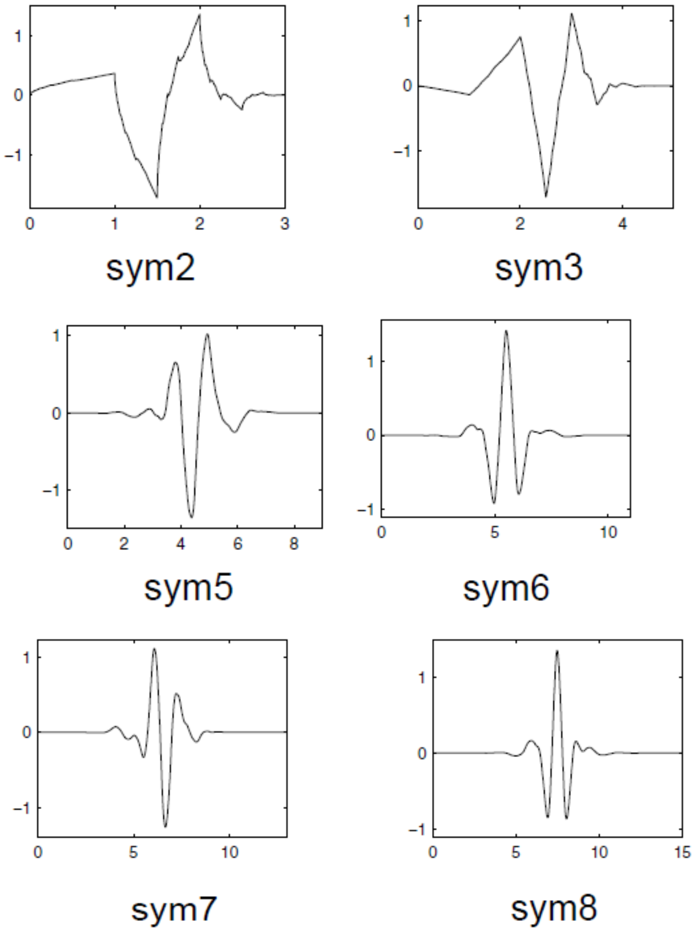


Figure 17- Symlet Wavelets Family.

5.5 Recommendations to choose a Wavelet family

One of the main criticisms directed towards the wavelet transform is the criteria to choose the better mother wavelet function $\psi^{a,b}$. To aid in this choice, there are a series of criteria and recommendations in the literature to be considered. After choosing a wavelet family, Torrence and Compo (1998) enumerate some important considerations, as presented below:

- ✓ **Orthogonal or non-orthogonal criteria**
  The wavelet transform, using families of orthogonal wavelets (Meyer, 1989), provides a more compact representation of the analyzed signal. Conversely, the Wavelet transform obtained by non-orthogonal wavelet families is highly redundant at larger scales, in which the Wavelet Spectrum at adjacent times is highly correlated (Meyer, 1989). The non-orthogonal Wavelet Transform is useful in analyzing time series (also valid for spatial series) where smooth and continuous variations in amplitude are expected.

- ✓ **Complex or real criteria**:
  A complex wavelet function, providing amplitude and phase information, is better suited to capture time-series oscillatory behavior. Instead, a real wavelet

function only provides information about component amplitude. This wavelet function can only be used to locate peaks and discontinuities.

- ✓ **Criteria support:**
  The wavelet function resolution is determined by the balance between its support in real space and frequency space. A function with more compact (narrower) support will have good time-domain resolution and poorer frequency-domain resolution. While a function with wider (broader) support will have a poorer resolution in time-domain resolution, and a good resolution in frequency-domain (a consequent characteristic which is due to the Heisenberg uncertainty principle).

- ✓ **Format Criteria:**
  The wavelet function chosen must reflect the characteristics of the analyzed signal. For series with peaks or discontinuities, a good choice would be Haar wavelet. While, for smoother series with more subtle variations, a wavelet function such Morlet wavelet family should be chosen. If the main interest is to obtain the Wavelet Energy Spectrum, then the choice of the wavelet function is not critical and any one of them will provide the same qualitative result.

6 Wavelet Packet Transform

The Wavelet Packet Transform (WPT) is a technique that decompose repeatedly a signal into successive low and high frequency components using a recursive decimation filter operation. This technique was first introduced by Coifman, Meyer and Wickerhauser (Mallat, 1999; Peng, Hao and Li, 2012). Wavelet packages consist of a usual linearly combined wavelet functions family. Figure 18 shows binary tree of a temporal signal $f(t)$ up to the 3rd level of WPT.

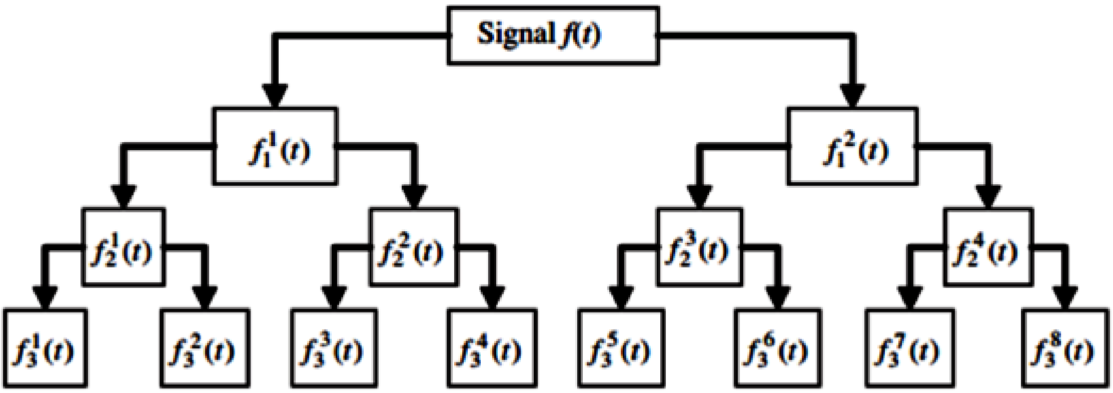


Figure 18- Wavelet Packet Decomposition (WPD) (adapted from Peng, Hao and Li, 2012).

After the j-th level of decomposition, the original signal $f(t)$ can be constructed by the sum of $2j$ components as follows:

$$f(t) = \sum_{i=1}^{2j} f_j^i(t) \tag{18}$$

where, $f_j^i(t)$ is the Wavelet Packet signal component that can be expressed by a linear combination of wavelet functions:

$$f_j^i(t) = \sum_{k=-\infty}^{\infty} c_{j,k}^i\,(t)\psi_{j,k}^i(t) \tag{19}$$

where the indexes i, j and k are integers defined as modulation, scale and shifting parameter, respectively. The coefficient $c_{j,k}^i$ and the function $\psi_{i,k}^i(t)$ are, respectively, Wavelet Packet coefficient and function. The Wavelet Packet coefficient $c_{j,k}^i$ is obtained by orthogonal projection of the signal $f(t)$ by report to Wavelet Packet family $\psi_{j,k}^j(t)$, as described as follows:

$$c_{j,k}^i\ = \int_{-\infty}^{\infty} f(t)\psi_{j,k}^i(t)dt \tag{20}$$

And Wavelet Packet function is defined as:

$$\psi_{j,k}^i(x) = 2^{j/2}\psi^i\left(2^j t - k\right) \tag{21}$$

where $\psi^1(t) = \psi\quad(t)$ is defined as mother wavelet function. And wavelet functions ψ^i $(i\ \geq\ 1)$ are function of recursive relationships:

$$\psi^{2i} = \sqrt{2}\sum_{k=-\infty}^{\infty} h(k)\,\psi^i(2t-k) \tag{22}$$

$$\psi^{2i+1} = \sqrt{2}\sum_{k=-\infty}^{\infty} g(k)\,\psi^i(2t-k) \tag{23}$$

where $h(k)$ and $g(k)$ are Quadrature Mirror Filters (QMF) associated to scale function and the mother wavelet function.

Each component in the Wavelet Packet Decomposition (WPD) tree can be seen as output from a filter tuned to a particular base function (Figure 18). At WPD top (root node), where decomposition level is low, we have a good time-domain resolution but a poor frequency-domain resolution. At WPD bottom (leaf node), where decomposition level is relatively high, we have a good frequency-domain resolution but a poor time-domain resolution. For structural integrity monitoring purposes, good frequency domain resolution is more important and thus, a high level of DWP is often needed to detect minor changes in signals (Sun and Chang, 2002, 2004).

6.1 Wavelet Packet Energy

Wavelet Packet Energy has demonstrated to be a more robust tool for damage identification compared to Wavelet Packet coefficients (Sun and Chang, 2002; Peng, Hao and Li, 2012).

The energy of the wavelet packet signal is given by:

$$E_f = \int_{-\infty}^{\infty} f^2(t)dt = \sum_{m=1}^{2^j} \sum_{n=1}^{2^j} \int_{-\infty}^{\infty} f_j{}^m(t) f_j{}^n\,(t)dt \tag{24}$$

where $f_j{}^m$ and $f_j{}^n$ are decomposed wavelet components. The total energy of the signal is expressed as the energy summation of wavelet packet components when mother wavelet is orthogonal:

$$E_f = \sum_{n=1}^{2^j} E_{f_j^i} = \sum_{i=1}^{2^j} \int_{-\infty}^{\infty} f_j{}^i(t)^2\,(t)dt \tag{25}$$

From Eqs. (24) and (25), the signal component $f_j{}^i(t)$ is a superposition of wavelet functions $\psi_{j,k}^i\,(t)$ of the same scale j, but translated in the time domain $(-\infty < k < +\infty)$. The energy component $E_{f_j^i}$ is the energy stored in a frequency band determined by wavelet functions $\psi_{j,k}^i(t)$.

7 Regularization Methods

The Wavelets coefficient disturbance is generated by several causes (e.g., signal noisy). To reduce this disturbance, Tikhonov regularization method can be applied (Tikhonov and Arsenin, 1977).

In general, inverse problems are ill-posed and their solutions are very sensible to noise. Small errors, due to experimental measured data, can result in a significant difference in ill-posed problems. Regularization methods seek to reduce oscillations in numerical solution by modification of objective function (Tikhonov and Arsenin, 1977; Schnur and Zabaras, 1990; Bezerra and Saigal, 1993; Bezerra, 1995; Tikhonov *et al.*, 1995; Silva Neto and Moura Neto, 2005). The most used "regularization" terms are zero-order, first-order, and second-order terms (Beck, Blackwell and Clair Jr, 1985; Silva Neto and Moura Neto, 2005). The zero-order term controls change in the magnitude of vector u, the first-order term controls change in the amplitude of the rate of change of vector u, and second-order terms can be expressed in integral form as (Schnur and Zabaras, 1990):

$$\rho = \beta_0 \int (u^2)\,\mathrm{ds} + \beta_1 \int \left(\frac{\partial u}{\partial s}\right)^2 \mathrm{ds} + \beta_2 \int \left(\frac{\partial^2 u}{\partial s^2}\right)^2 \mathrm{d}s \tag{26}$$

In finite differences, an analogous regularization equation is written as:

$$\rho = \beta_0 \sum_{\mathrm{i}=1}^{\mathrm{p}} \left(u_{\mathrm{i}}^{(n)}\right)^2 + \beta_1 \sum_{\mathrm{i}=1}^{\mathrm{p}} \left(u_{\mathrm{i}}^{(n)} - u_i^{(n-1)}\right)^2 + \beta_2 \sum_{\mathrm{i}=1}^{\mathrm{p}} \left(u_i^{(n)} - 2u_{\mathrm{i}}^{(n-1)} + u_{\mathrm{i}}^{(n-2)}\right)^2 \tag{27}$$

where, to an iteration number n, β_j is regularization parameters, s spatial parameter, and u_i the components of u. According to Beck *et al* (1985), Equation (27) is analogous to Equation (26). The finite difference regularization expression (27) is simple to implement.

With large values of β_j, variations of the vector u are obtained and tend to delay convergence. While small values of β_j can result in large oscillations of solution (Bezerra and Saigal, 1993; Bezerra, 1994, 1995).

Example 01: Wavelet Signal Regularization

To increase the changes caused by damage at response signal, Tikhonov regularization method was applied. Figure 19 shows difference between the wavelet coefficients for a smoothed and an unregulated signal.

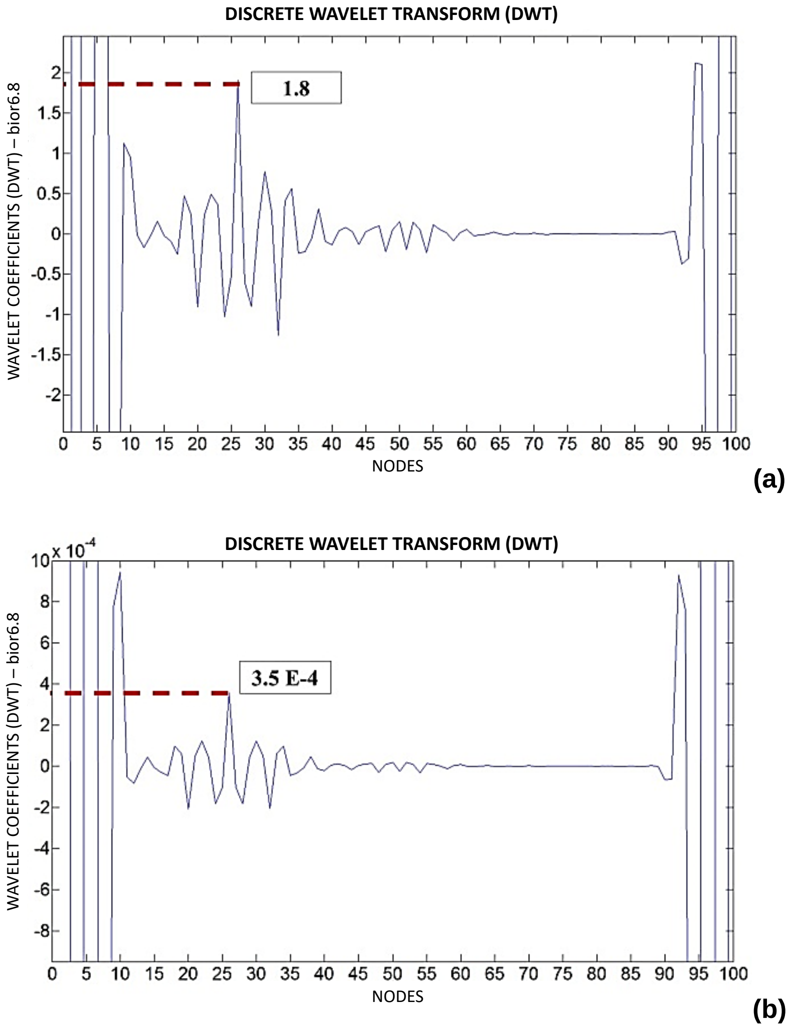


Figure 19 – Wavelet coefficients (a) with and (b) without Tikhonov Regularization.

In the Annex, Algorithm 3 is the Tikhonov Regularization coded in MATLAB.

8 Concluding Remarks

In this chapter some fundamental concepts on wavelet transforms were presented, both for discrete and continuous cases. The wavelet transforms are useful to understand the ability of wavelet transform to capture sudden changes in material properties or parameters along the span of structural component being assessed. Several families of wavelet transforms were discussed, including some wavelet properties and regularization aspects.

9 References

Beck, J. V, Blackwell, B. and Clair Jr, C. R. S. (1985) *Inverse heat conduction: Ill-posed problems*. Edited by J. Wiley & Sons. James Beck.

Bezerra, L. M. (1994) *Inverse elastostatics solutions with boundary elements.* Carnegie Mellon University.

Bezerra, L. M. (1995) 'The solution of ill-posed inverse problems in solid mechanics and the nuclear industry', in UFRGS (ed.) *Transactions of the 13th International Conference on Structural Mechanics in Reactor Technology (SMiRT 13)*. Porto Alegre: SMiRT, pp. 748–757. Available at: http://repositorio.ipen.br/handle/123456789/19080.

Bezerra, L. M. and Saigal, S. (1993) 'A boundary element formulation for the inverse elastostatics problem (IESP) of flaw detection', *International Journal for Numerical Methods in Engineering*, 36(July 2015), pp. 2189–2202. doi: 10.1002/nme.1620361304.

Daubechies, I. (1992) *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics. doi: 10.1137/1.9781611970104.

Estrada, R. S. (2008) *Damage Detection Methods in Bridges through Vibration Monitoring : Evaluation and Application*. University of Minho. Available at: http://hdl.handle.net/1822/9023.

Gutierrez, C. E. C. (2002) *Uma aplicação à série de Preço Spot de Energia Elétrica do Brasil Dissertação de Mestrado Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Sistemas de Energia Elétrica do Mecânica e Elétrica*.

Haar, A. (1910) 'Zur Theorie der orthogonalen Funktionensysteme - Erste Mitteilung', *Mathematische Annalen*. Springer-Verlag, 69(3), pp. 331–371. doi: 10.1007/BF01456326.

Li, H. *et al.* (2009) 'Evaluation of earthquake-induced structural damages by wavelet transform', *Progress in Natural Science*. Science Press, 19(4), pp. 461–470. doi: 10.1016/j.pnsc.2008.09.002.

Mallat, S. (1999) *A Wavelet Tour of Signal Processing*, *A Wavelet Tour of Signal Processing*. Elsevier. doi: 10.1016/B978-0-12-466606-1.X5000-4.

Mehra, M. (2018) *Wavelets Theory and Its Applications*, *Wavelet Theory and Its Applications*. Edited by Springer Nature Singapore Pte Ltd. Singapore: Springer Singapore (Forum for Interdisciplinary Mathematics). doi: 10.1007/978-981-13-2595-3.

Meyer, Y. (1989) 'Orthonormal Wavelets', in Combes, J.-M., Grossmann, A., and Tchamitchian, P. (eds) *Wavelets*. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 21–37.

Misiti, M *et al.* (1997) *Wavelet Toolbox for Use with MATLAB*, *The Mathworks Inc., Natick, MA, USA*. Natick, Massachusetts. Available at: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Wavelet+Toolbox+For+Use+with+MATLAB#3%5Cnhttp://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Wavelet+Toolbox+for+Use+with+MATLAB,+1996#3.

Misiti, Michel *et al.* (1997) *Wavelet Toolbox^{TM} 4 User's Guide Product enhancement suggestions Wavelet Toolbox^{TM} User's Guide*. Available at: www.mathworks.com%0Awww.mathworks.com/contact_TS.html.

Ovanesova, A. V. (2000) 'Applications of Wavelets to Crack Detection in Frame Structures.', p. 235.

Ovanesova, A. V. and Suárez, L. E. (2004) 'Applications of wavelet transforms to damage detection in frame structures', *Engineering Structures*. Elsevier BV, 26(1), pp. 39–49. doi: 10.1016/j.engstruct.2003.08.009.

Palechor, E. U. L. *et al.* (2019) 'Damage identification in beams using additional rove mass and wavelet transform', *Frattura ed Integrità Strutturale*, 13(49), pp. 614–629. doi: 10.3221/IGF-ESIS.49.56.

Peng, X.-L., Hao, H. and Li, Z.-X. (2012) 'Application of wavelet packet transform in subsea pipeline bedding condition assessment', *Engineering Structures*, 39, pp. 50–65. doi: 10.1016/j.engstruct.2012.01.017.

Reis, N. R. (2018) *Teoria geral das wavelets e aplicações*. Universidade Federal de Minas Gerais. Available at: http://hdl.handle.net/1843/EABA-B4YKHV.

Schnur, D. S. and Zabaras, N. (1990) 'Finite element solution of two-dimensional inverse elastic problems using spatial smoothing', *International Journal for Numerical Methods in Engineering*. John Wiley & Sons, Ltd, 30(1), pp. 57–75. doi: 10.1002/NME.1620300105.

Silva Neto, A. J. da and Moura Neto, F. D. (2005) *Problemas Inversos: Conceitos Fundamentais e Aplicações*. Edited by UERJ. Rio de Janeiro, RJ, Brazil: EdUERJ.

Silva, R. S. Y. R. C. *et al.* (2019) 'Damage detection in a reinforced concrete bridge applying wavelet transform in experimental and numerical data', *Frattura ed Integrità Strutturale*, 13(48), pp. 693–705. doi: 10.3221/IGF-ESIS.48.65.

De Souza, F. M. *et al.* (2007) 'Comparação das Bases de Wavelets Ortonormais e Biortogonais: Implementação, Vantagens e Desvantagens no Posicionamento com GPS', *TEMA - Tendências em Matemática Aplicada e Computacional*, 8(1). doi: 10.5540/tema.2007.08.01.0149.

Sun, Z. and Chang, C. C. (2002) 'Structural Damage Assessment Based on Wavelet Packet Transform', *Journal of Structural Engineering*, 128(10). doi: 10.1061/(asce)0733-9445(2002)128:10(1354).

Sun, Z. and Chang, C. C. (2004) 'Statistical Wavelet-Based Method for Structural Health Monitoring', *Journal of Structural Engineering*, 130(7), pp. 1055–1062. doi: 10.1061/(ASCE)0733-9445(2004)130:7(1055).

Tikhonov, A. N. *et al.* (1995) *Numerical Methods for the Solution of Ill-Posed Problems*, *Numerical Methods for the Solution of Ill-Posed Problems*. Dordrecht: Springer Netherlands. doi: 10.1007/978-94-015-8480-7.

Tikhonov, A. N. and Arsenin, V. J. (1977) *Solutions of Ill-posed Problems*. Winston (Halsted Press book). Available at: https://books.google.com.br/books?id=ECrvAAAAMAAJ.

Torrence, C. and Compo, G. P. (1998) 'A Practical Guide to Wavelet Analysis', *Bulletin of the American Meteorological Society*, 79(1), pp. 61–78. doi: 10.1175/1520-0477(1998)079<0061:APGTWA>2.0.CO;2.

Weeks, M. (2007) *DIGITAL SIGNAL PROCESSING Using MATLAB and Wavelets*, *Book.*

Wu, N. and Wang, Q. (2011) 'Experimental studies on damage detection of beam structures with wavelet transform', *International Journal of Engineering Science*. Pergamon, 49(3), pp. 253–261. doi: 10.1016/j.ijengsci.2010.12.004.

Annex

Algorithm 1 – Signal Discontinuity

Algorithm 1 – Signal discontinuity (MATLAB implementation).

```
t1= 0:0.1:6; x1= sin(2*pi*t1);
t2= 6:0.1:6; x2= sin(2*pi*t2+0.1);
t = [t1 t2]; x = [x1 x2]; plot(t,x)
```

Algorithm 2 – Cubic spline interpolation

Algorithm 2 – Cubic spline interpolation (MATLAB implementation).

```
x = [1 2 3 4 5];
y = [0 1 0 1 0];
xx = 1:.05:5;
yy = spline(x,y,xx);
x_interp = 1.5;
y_interp = spline(x,y,x_interp)
y_interp = 1.1250
plot(x,y,'o',xx,yy,'r-',x_interp,y_interp,'*')
```

Algorithm 3 – Tikhonov Regularization

Algorithm 3 – Tikhonov regularization (MATLAB implementation).

```
B0=100;
B1=100;
B2=100;
```

```
n=length(u);
Part1=0;
Part2=0;
Part3=0;
for i=1:n
     if i==1
          a=(u(i,2))^2;
          Part1=a;
          b=((u(i,2)))^2;
          Part2=b;
          c=((u(i,2)))^2;
          Part3=c;
          u_reg(i)=(B0*(Part1))+(B1*(Part2))+(B2*(Part3));
      elseif i==2
          a=(u(i,2))^2;
          Part1=a;
          b=((u(i,2))-(u(i-1,2)))^2;
          Part2=b;
          c=((u(i,2))-(2*(u(i-1,2))))^2;
          Part3=c;
          u_reg(i)=(B0*(Part1))+(B1*(Part2))+(B2*(Part3));
      else
          a=(u(i,2))^2;
          Part1=a;
          b=((u(i,2))-(u(i-1,2)))^2;
          Part2=b;
          c=((u(i,2))-(2*(u(i-1,2)))+u(i-2,2))^2;
          Part3=c;
          u_reg(i)=(B0*(Part1))+(B1*(Part2))+(B2*(Part3));
      end
end
figure
hold on
grid on
plot(u(:,1),u(:,2),'b')
plot(u(:,1), u_reg(:),'r')
title('TIKHONOV Regulatization')
```

where, `u_reg(i)` is the regularization vector of signal `u(:,2)`.

Chapter 10: Application of Wavelet Transforms to Structural Damage Monitoring and Detection

Chapter details

Chapter DOI:

https://doi.org/10.4322/978-65-86503-71-5.c10

Chapter suggested citation / reference style:

Palechor, Erwin U. L., et al. (2022). "Application of Wavelet Transforms to Structural Damage Monitoring and Detection". *In* Jorge, Ariosto B., et al. (Eds.) *Model-Based and Signal-Based Inverse Methods*, Vol. I, UnB, Brasilia, DF, Brazil, pp. 357–381. Book series in Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity.

P.S.: DOI may be included at the end of citation, for completeness.

Book details

Book: Model-based and Signal-Based Inverse Methods

Edited by: Jorge, Ariosto B., Anflor, Carla T. M., Gomes, Guilherme F., & Carneiro, Sergio H. S.

Volume I of Book Series in:

Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity

Published by: UnB City: Brasilia, DF, Brazil Year: 2022

DOI: https://doi.org/10.4322/978-65-86503-71-5

Application of Wavelet Transforms to Structural Damage Monitoring and Detection

Erwin Ulises Lopez Palechor[1*], Ramon Saleno Yure Costa Silva[2], Gilberto Gomes[2], Marcus V. Girão de Morais[2,4], Luciano Mendes Bezerra[3] and Ariosto Bretanha Jorge[4]

[1] Science and Technology Center, Federal University of Cariri, Brazil. e-mail: erwin.lopez@ufca.edu.br
[2] Faculty of Technology, University of Brasilia, Brazil. e-mail: ggomes@unb.br; ramon.silva@unb.br; mvmorais@unb.br
[3] Post-Graduate Program in Structures and Civil Construction, University of Brasilia, Brazil. e-mail: lmbz@unb.br
[4]Post-Graduate Program in Integrity of Engineering Materials, University of Brasilia, Brazil. e-mail: ariosto.b.jorge@gmail.com

*Corresponding author

Abstract

This chapter presents applications of Wavelet Transforms in damage identification, in 1D 2D and 3D structures, using numerical and experimental data, the research works developed at the Graduate Program of Structures and Civil Construction of University of Brasília.

Keywords: Wavelet application, Structural Health Monitoring, Wavelet Transform, Continuous Wavelet, Discrete Wavelet, Wavelet Family Function.

1 Wavelet Applications Review on Damage Detection of Structures

Damage detection and localization inside structures has great importance due to being critical for the serviceability and safety of structural elements. Only ultrasonic and x-ray are reliable techniques for nondestructive examination of structures. They may be applied in practice for the precise location of subsurface damages. Both of them require special procedures and are time consuming and expensive. Current research point toward the possible use of numerical-computational methods in assisting the detection of damages inside structures.

Vibration-based methods of damage detection rely on the fact that dynamic characteristics such as natural frequencies, mode shapes and damping are influenced by stiffness (Friswell, 2007; Janeliukstis *et al.*, 2017). The most serious limitation of those methods is the need for a structural response of the healthy structure. But, in the past few years, wavelet transform analysis have attracted attention for vibration-based damage localization techniques (Kim and Melhem, 2004; Yang and Oyadiji, 2017). The wavelet transform analysis has the advantage of need only damage structure response.

Palechor (2013) presents experimental results using Wavelet Transform to determine the damage localization in a laminated steel beam (I-profile), using the static response (displacements) of the damaged beam.

Silva, Palechor et al. (2019) presents a damage localization analysis using experimental data of Dogna Bridge, Italy. In this case, this methodology use CWT of experimental modal shape, with cubic spline interpolation and regularization techniques, to localize damage in beam grid structures. The paper uses experimental results and calibrate numerical model to localize induced damage. The experimental/numerical methodology do not need comparation to intact structural model.

Silva, Bezerra et al (2019) associate boundary element method with wavelet transform to detect and localize damage in structures subjected to static loads. The applications cases concern the localization of subsurface cracks in structures two-dimensional with different support conditions. This methodology investigates two-dimensional structures with single and multiple cracks with different orientations inside. The effectiveness of this methodology is discussed through the resumé of principal numerical results obtained.

All presented applications of damage localization using Wavelet Transforms, in 1D 2D and 3D structures, are research results developed by Graduate Program of Structures and Civil Construction of University of Brasília.

2 Basics Concepts of Wavelet Transform

The Wavelet Transform, widely used in several engineering domain, was applied to damage detection in structures. Damage is a local phenomenon is not sensible apparent in vibration response data of structure. Wavelet transform can identify even slight changes in global response of vibrational signal. Wavelet transform make possible this using base functions $\psi(x)$, called wavelets, to analysis any signal in several scale and position (Palechor *et al.*, 2022). Wavelet Transform projects a signal into two-dimensional space, function of scale and position.

The Continuous Wavelet Transform (CWT) of the signal $f(x)$ is defined as:

$$W_{\psi}^{f}(a,b) = \int_{-\infty}^{\infty} f(x)\psi_{a,b}(x)dx \tag{1}$$

where, $\psi_{a,b}(x) = |a|^{-1/2}\psi'(x - b/a)$ is the mother wavelet function, and coefficients a and b are scale and position, respectively.

The Discrete Wavelet Transform (DWT) (Ovanesova, 2000; Ovanesova and Suárez, 2004; Palechor *et al.*, 2022) of the signal $f(x)$ is defined as:

$$W_{j,k} = \int_{-\infty}^{\infty} f(x)\psi_{j,k}(x)dx \tag{2}$$

where, $\psi_{j,k}(x) = \psi\left(2^{-j}x - k\right)$ is the discrete mother wavelet; the scale $a = 2^j$ and the shifting $b = k(2^j)$, where $(j,k) \in Z$ (integer set).

The wavelet coefficients $W_{\psi}^{f}(a,b)$ (for CWT) and $W_{j,k}$ (for DWT) has a property to be sensitive to local singularities in signal. This property is particularly useful to identify a beam discontinuity due to a loss of stiffness (stiffness damage).

3 Damage Detection of Steel Beams using Static Displacement

Using experimental results, Wavelet transform is applied to figure out the open crack position in damage laminated steel beam.

The metallic beams evaluated is simply supported beam in MR-250 steel, with $6m$ of length, subjected to various levels of load applied at middle-span. The geometric and material characteristics of essayed beams are shown in Table 1. It is worth mentioning that the values, the yield stress (fy), modulus of elasticity (E) and Poisson's ratio (ν), were chosen from the catalog provided by the manufacturer.

Figure 1 show the discretization of beams span into sixteen longitudinal segments, with a length $\Delta L = 375mm$, totaling 17 nodal points.

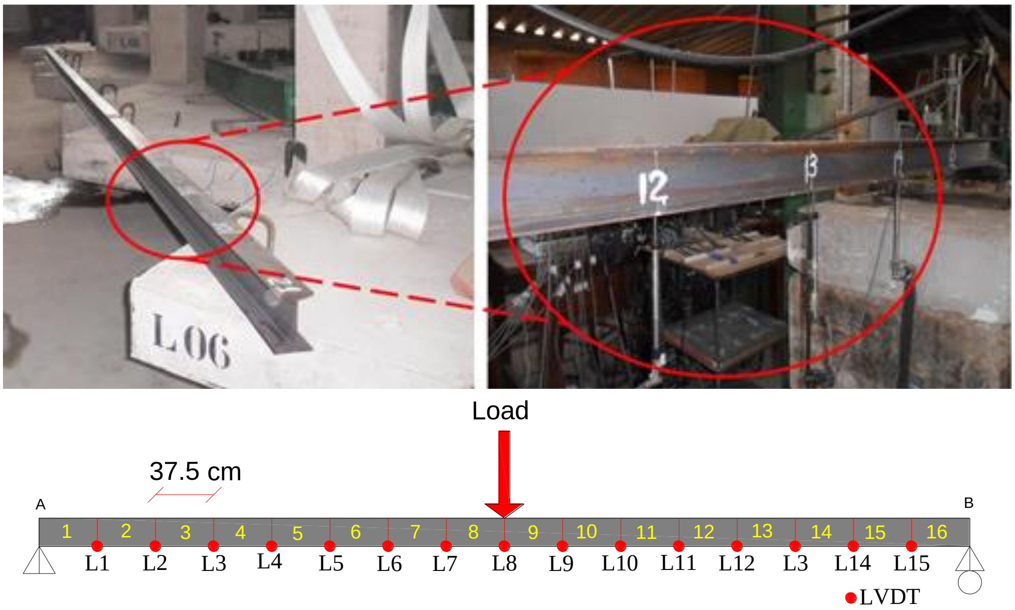
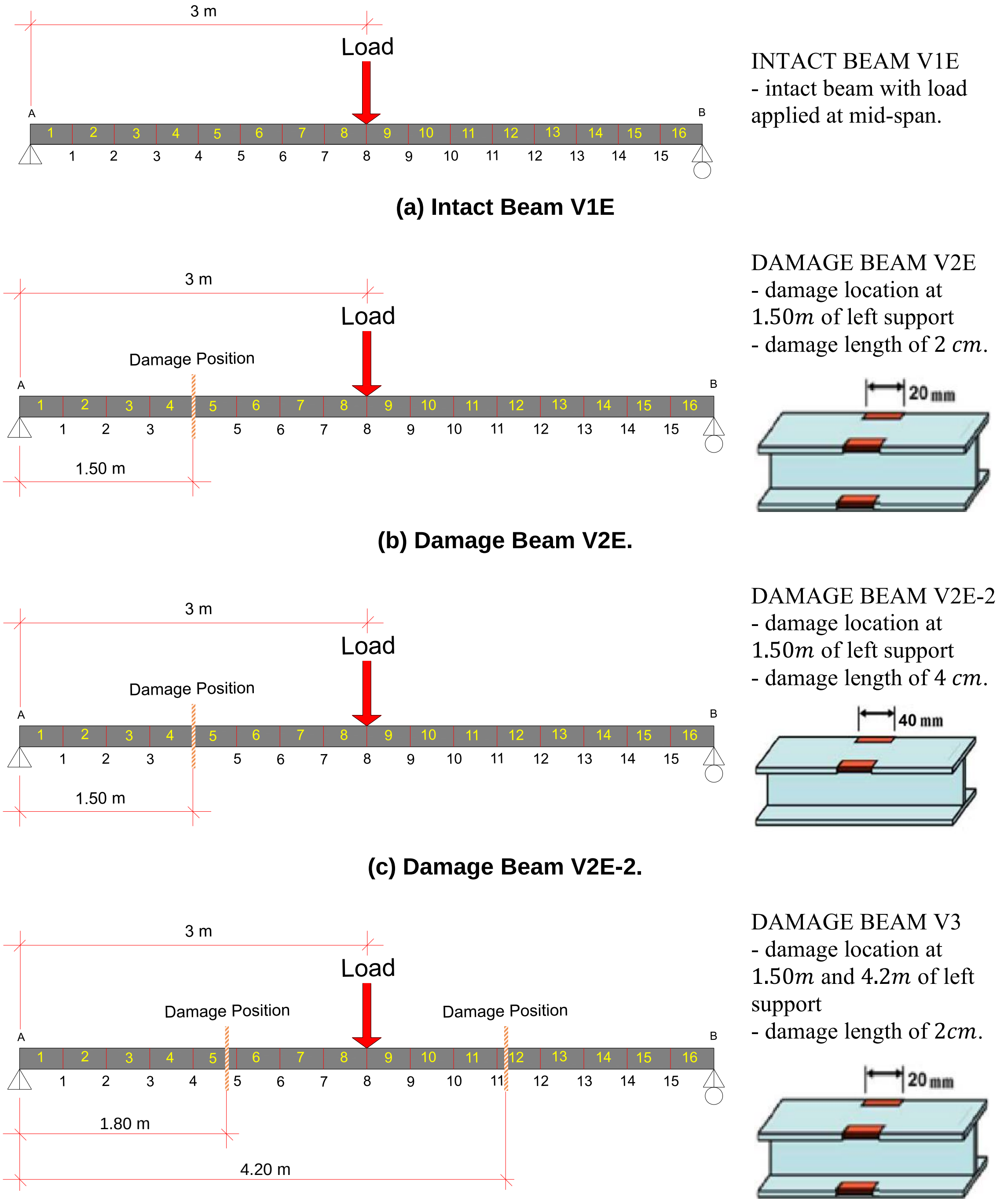


Figure 1 – Schematic representation of steel beam discretization into 16 segments.

Figure 2 show schematic representation for location and characteristics of induced damage on essayed beams. The load application in mid-span was the same for all essays, at node 8.



INTACT BEAM V1E
- intact beam with load applied at mid-span.

(a) Intact Beam V1E

DAMAGE BEAM V2E
- damage location at 1.50m of left support
- damage length of 2 cm.

(b) Damage Beam V2E.

DAMAGE BEAM V2E-2
- damage location at 1.50m of left support
- damage length of 4 cm.

(c) Damage Beam V2E-2.

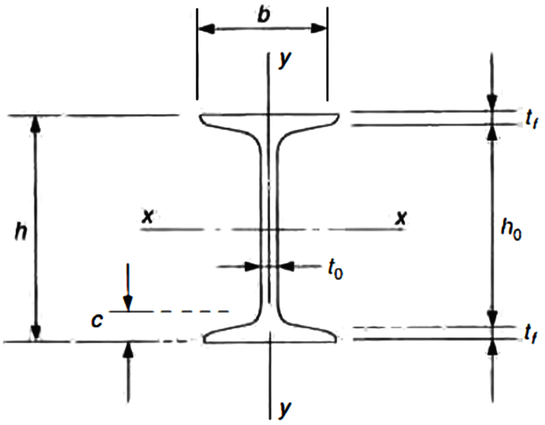
DAMAGE BEAM V3
- damage location at 1.50m and 4.2m of left support
- damage length of 2cm.

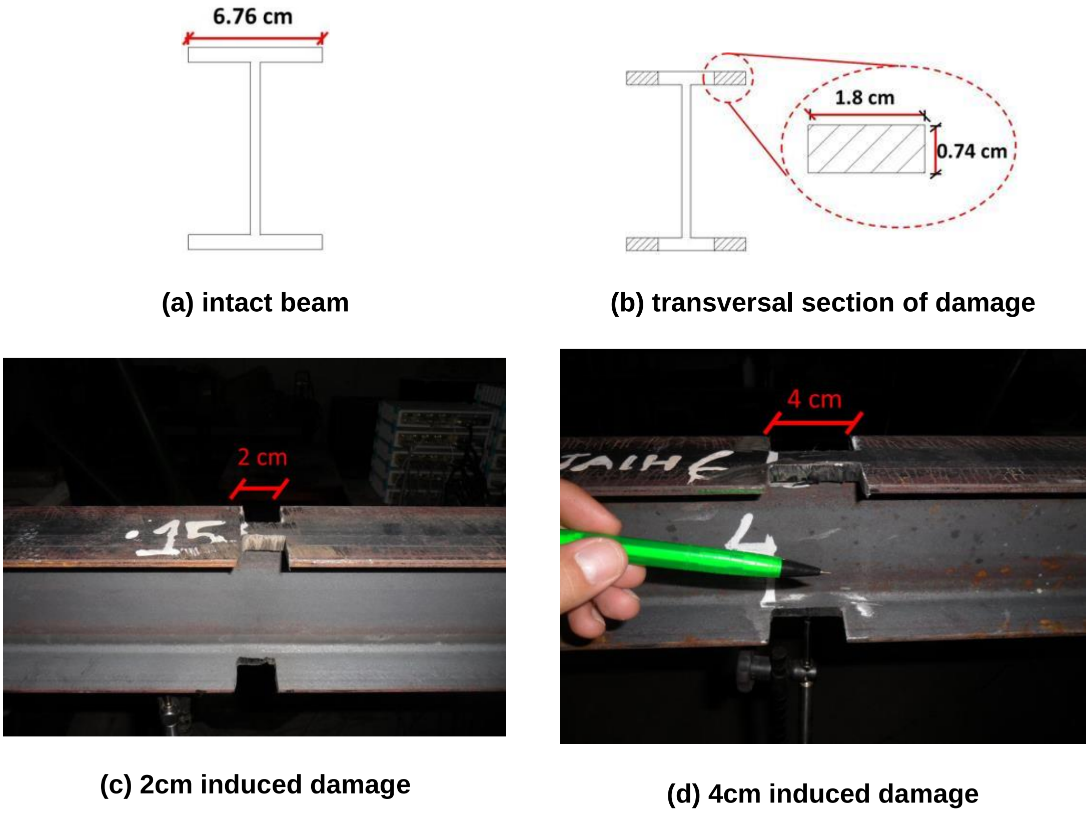
Figure 2 – Schematic description of induced damage for each experimental static essays

The induced damage was carried out using a circular saw with cuts with 2cm or 4cm length in longitudinal direction. Both open cracks induce a decrease of second moment of in cross section (Figure 3).

Table 2 shows area moment of inertia for intact and damage section.

Table 1 – Material and geometric characteristics of essayed beam.

Steel I-shape 102 X 11,4		
h(cm)	10,16	 b y t_f h x x h_0 t_0 c t_f y
h_0(cm)	8,68	
t_f(cm)	0,74	
t_0(cm)	0,483	
c(cm)	1,59	
b(cm)	6,76	
Surface (cm^2)	14,5	
I_x (cm^4)	252	
W_x (cm^3)	49,7	
i_x (cm)	4,17	
I_y (cm^4)	31,7	
W_y (cm^3)	9,37	
i_y(cm)	1,48	
Z_x(cm^3)	56,220	
Z_y (cm^3)	17,414	
f_y(kN/cm^2)	25	
E (kN/cm^2)	20000	
Length L (m)	6	



(a) intact beam

(b) transversal section of damage

(c) 2cm induced damage

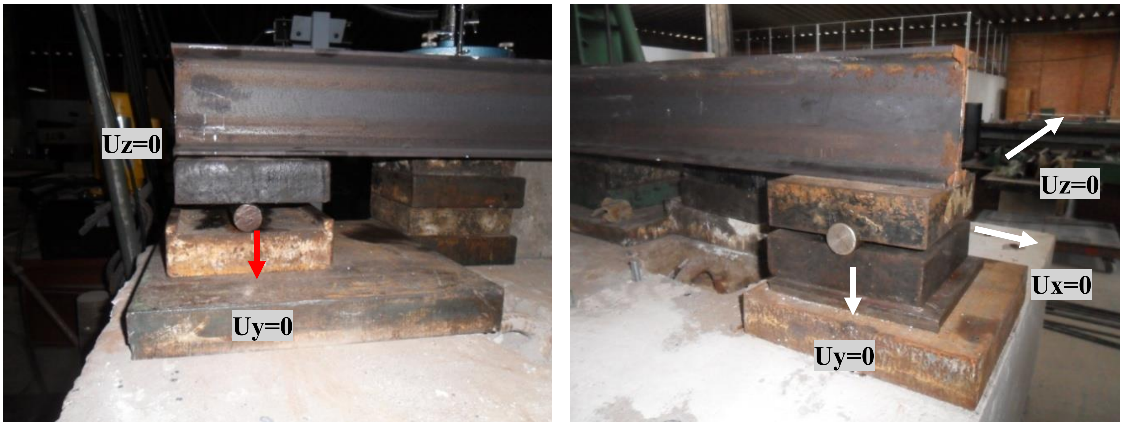
(d) 4cm induced damage

Figure 3 – Geometry of induced damage in steel beam

Table 2 – Decreased moment of inertia of the cross section induced by open crack

	$I_x\,(cm^4)$	$I_y\,(cm^4)$	$r_x{}^{1}\,(cm)$	$r_y\,(cm)$
Intact Section	252	31,7	4,17	1,48
Damage Section	130,71	4,0215	3,83	0,67

To ensure simple support (Figure 4a), two plane plates and a roller were used to allow displacement only in the x direction. For the hinged support (Figure 4b), two grooved plates with a roller were designed to restrict the translation in all direction, permitting only rotation.



(a) Roller Support. **(b) Pinned Support**

Figure 4 – Structural support description of steel span.

Along of steel beam, seventeen LVDTs collected the experimental data corresponding to nodal vertical displacements for essayed beams. Exported to Matlab, experimental data was interpolated, using cubic-spline tool (`spline` command), to smooth data with more registers. The Tikhonov regularization method was applied to the interpolation results. As a last procedure, the damage identification was done using TDW and TCW.

For each essayed beam, Table 3 describe the correspondence of damage location for spatial position and wavelet node (signal position) to make easy the visualization of wavelet coefficients.

Table 3 – Correspondence between spatial distance and wavelet node to each damage beam.

Beam	Position(m) from left support	Node (#) TDW	Node (#) TCW
V2E	damage of 2cm at 1,5 m	25	250
V2E-2	damage of 4cm at 1,5 m	25	250
V3E	damage of 2cm at 1,8 m and 4,2 m	30 e 70	300 e 700

[1] Radius of gyration $r = \sqrt{I/A}\ (r_x, r_y)$

Figure 5 show experimental results of vertical displacements U_y for each tested beam statically essayed. Each steel beam is subjected for several static load levels. But wavelet transform was obtained for an intermediated load (V2E – 3330N, V2E-2 – 3990N and V3E – 3120N), inferior of maximum resistance load of intact beam (4373N).

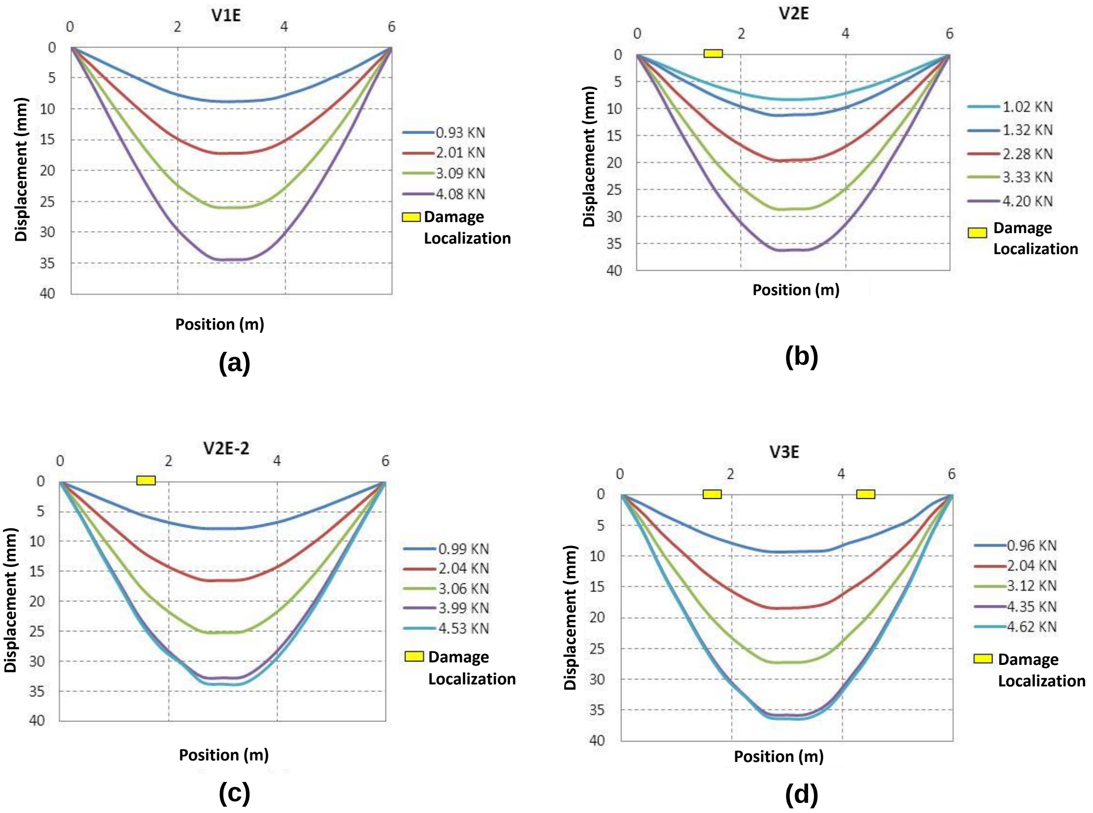


Figure 5 – Horizontal displacement of structural span for study conditions: (a) intact beam, and damage beam V2E (b), V2E-2 (c) and V3E (d).

The damage location was performed using only the static experimental responses of damaged beams V2E, V2E-2 and V3E. The obtained results with DWT and CWT are presented below.

3.1 Discrete wavelet transform results

From all mother wavelet functions in Matlab to calculate TDW coefficients, Palechor et al. (2014) analyses four mother wavelets (bior6.8, rbio2.6, sym6, coif3 and db5). For this chapter, Daubechies mother wavelet (db5) are presents due the fact of others mother wavelets analyzed is similar.

Figure 6 show discrete wavelets coefficients of experimental static data of V2E beam using Daubechies mother wavelet (db5). The damage is located at node 25 corresponding to 1.5 m from left support. At node 25, the DWT coefficients observe a small peak around the damaged region surrounding by others smaller ones due to signal noise. In addition, DWT generated disturbances at the extremity of DWT coefficient data, due to the geometric discontinuities of the supports.

Figure 7 show discrete wavelets coefficients of experimental static data of V2E-E beam with damage location at node 25. Compared to V2E (Figure 6), the DWT coefficients observe a highest pick due to the increase of open crack, now with 4cm. The noise pollution surrounding the damage is smaller too.

The V3E beam have two open cracks located at node 30 (1.8m from left support) and at node 70 (4.2 m). Figure 8 show discrete wavelets coefficients of experimental static data of V3E-E beam. The discrete wavelets coefficients were able to detect clearly the open crack at 70. The damage locates at node 30 have smaller spikes.

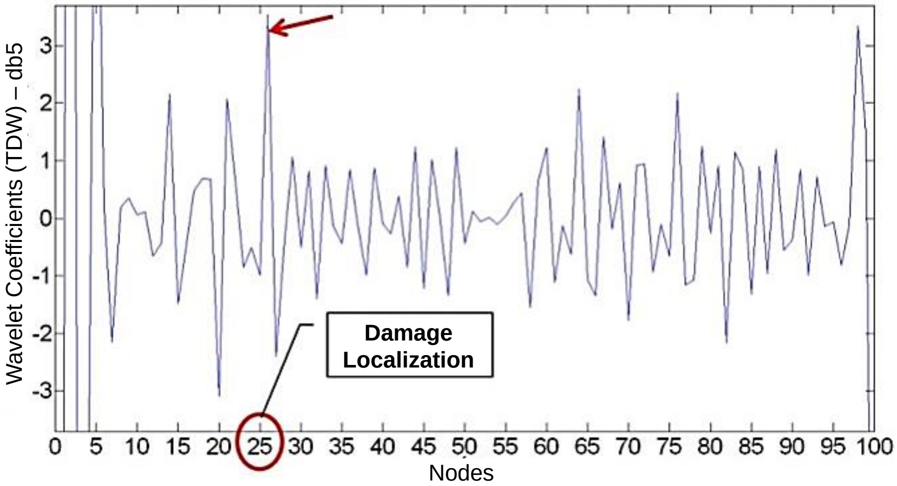


Figure 6 – DWT coefficients for V2E beam using db5.

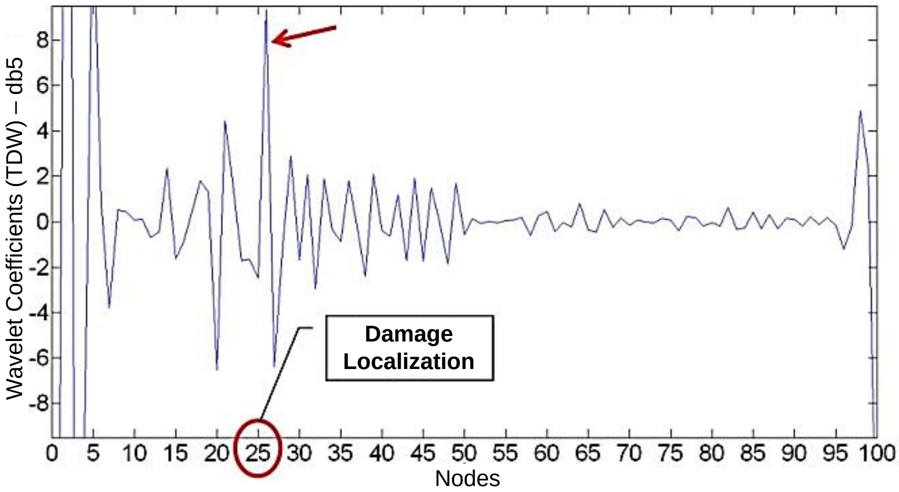


Figure 7 – DWT coefficients for V2E-2 beam using db5.

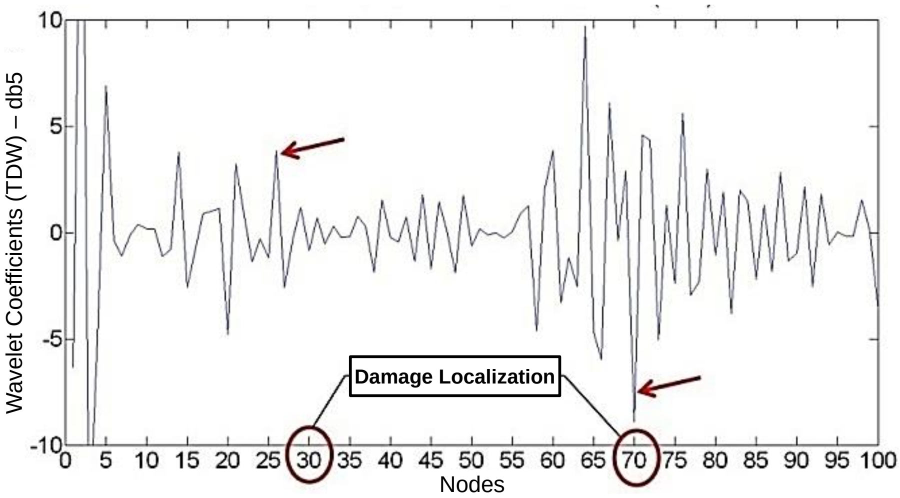


Figure 8 – DWT coefficients for V3E beam using db5.

3.2 Continuous wavelet transform results

Similar to DWT, Pachelor et al. (2014) select the same few set of mothers wavelets functions to calculate TDW coefficients of experimental data. Daubechies mother wavelet (db5) were choose due to others mother wavelets analyzed is similar. CWT presents the wavelet coefficients as function of position and scale which can be represented as 3D and 2D graph.

Figure 9 has the continuous wavelets coefficients of experimental static data of V2E beam using Daubechies mother wavelet (db5). Figure 10 and Figure 11 represent the same result for beam V2E-2 and V3E, respectively. The damage location using CWT is similar to DWT results. The damage is located at node 25 corresponding to 1.5 m from left support.

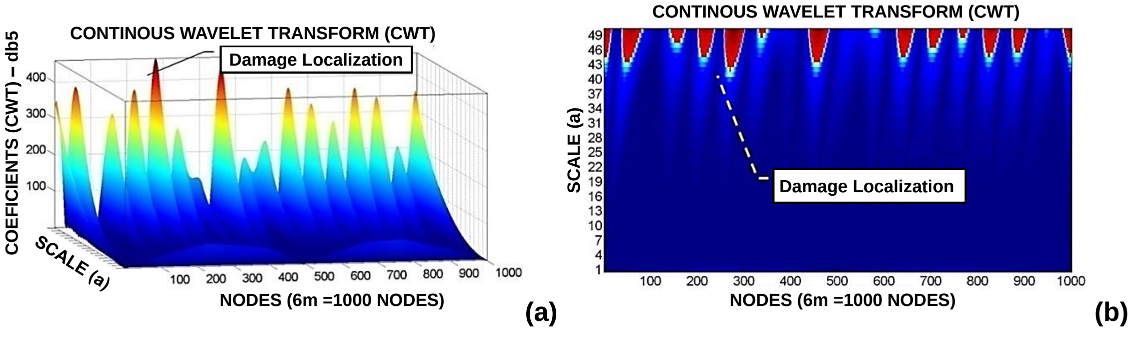


Figure 9 – CWT coefficients for beam V2E using db5: (a) 3D view and (b) 2D plan.

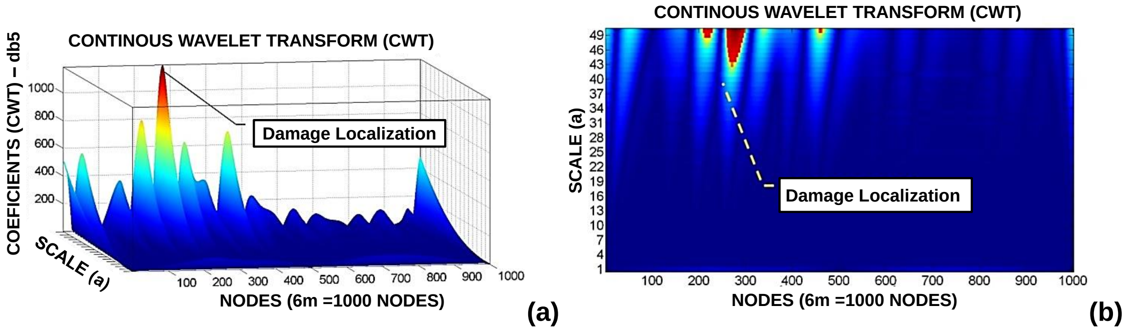


Figure 10 – CWT coefficients for beam V2E-2 using db5: (a) 3D view and (b) 2D plan.

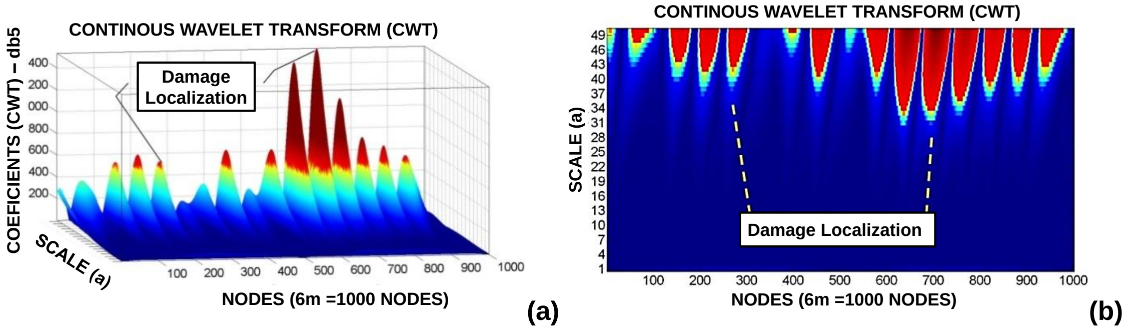


Figure 11 – CWT coefficients for beam V3E using db5: (a) 3D view and (b) 2D plan.

3.3 Important Remarks

We select some highlight:

- Continuous and discrete wavelet transform present similar results respect to damage localization
- The mother wavelet bior6.8, rbio2.6, sym6, coif3 and db5 are efficient to damage localization in experimented beams (Palechor *et al.*, 2014).
- Wavelet coefficients are sensitive to geometric discontinuities of structural supports. The wavelet coefficients graphs shown pics similar (or superior) to damage discontinuities.
- Wavelet coefficients are sensible to damage intensity. Intensity of damage results in wavelet coefficients picks. The open cracks with 4 cm cut present betters results than 2cm cut.

4 Damage Detection in Bridge using Experimental Modal Shape

Silva et al. (2019) present a damage localization methodology using CWT associated with spline interpolation and regularization techniques applied to identified mode shapes. . As application test, the authors present the analysis of Dogna Bridge (Italy) using only damage response.

The villages of Crivela and Valdogna, located the Italian north east region, are connect by Dogna bridge crossing river Fella (Artemis Modal, 2014). This a reinforced concrete bridge has four-span with single-lane structure with 64m long (16m each) and 4m wide

(Figure 12ab). Three longitudinal beams, with rectangular cross-sections 120×35 cm^2, support an 18cm-thick slab. Figure 12c shows transversal section of three longitudinal beams and slab. Progressive damage introduction is detailed in the next sections.

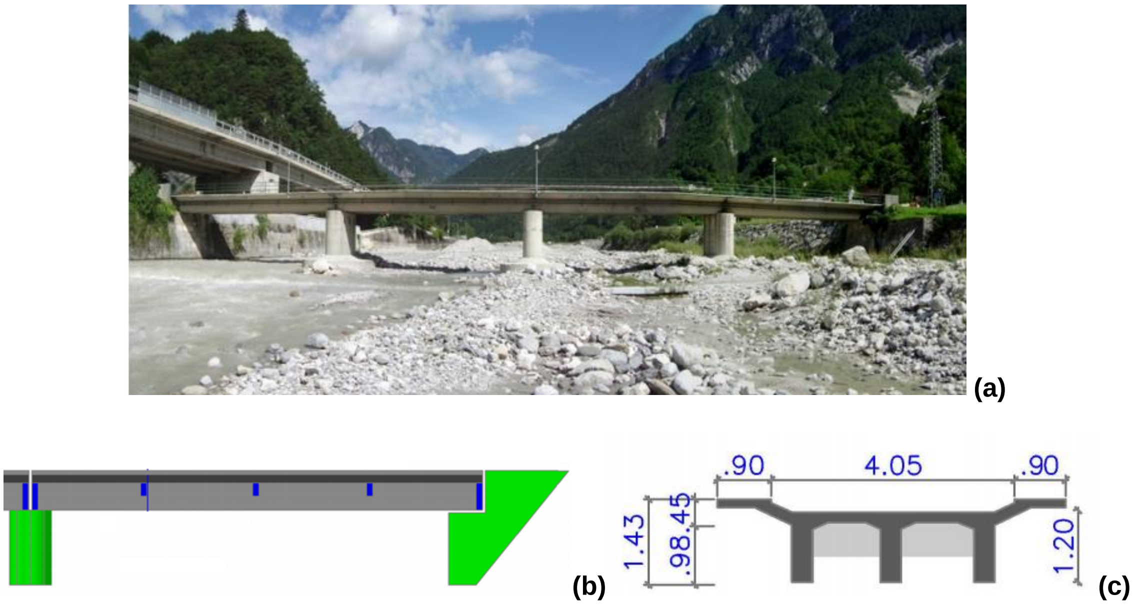


Figure 12 – Picture with overview of Dogna Bridge (a) and its schematic representation of longitudinal view (b) and transversal section (c).

4.1 Proposed Methodology

The proposed methodology is based on the use of three well-known techniques: cubic spline interpolation, Tikhonov regularization (Friswell, 2007; Palechor *et al.*, 2022) and continuous wavelet Transform. Silva et al (2019) applied the proposed methodology for damage localization using experimental data or numerical results.

Firstly, the data of damage structure need be obtained by experimental test or numerical simulation. And it is necessary extract the mode shapes from damage structure data. Secondly, the cubic spline interpolation determines intermediated spatial data due to experimental data are compose of few measurement points (Palechor, 2013). Thirdly, Tikhonov regularization technique reduce numerical oscillations of interpolated mode shape signal (Palechor *et al.*, 2022). Lastly, it is determined the continuous wavelet coefficients of regularized mode shape. It is search for discontinuities in wavelet coefficients plots that correspond to damaged region. Figure 13 summarize the flowchart of proposed methodology.

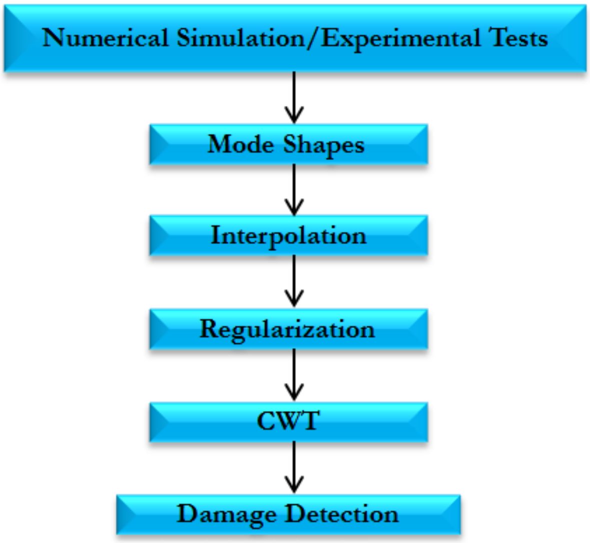
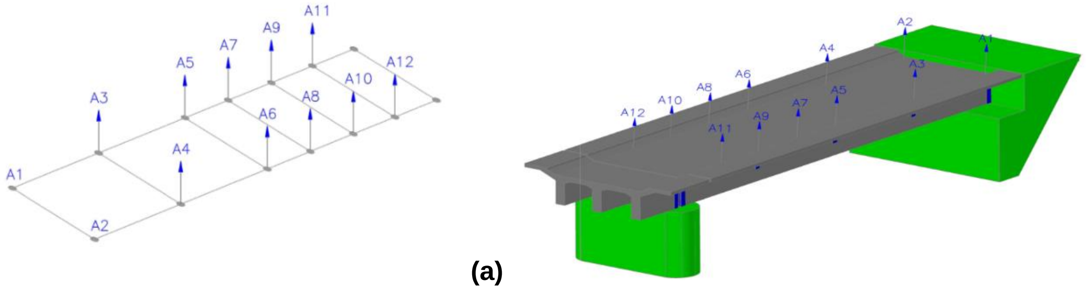


Figure 13 – Proposed methodology flowchart.

4.2 Dynamic test of intact bridge

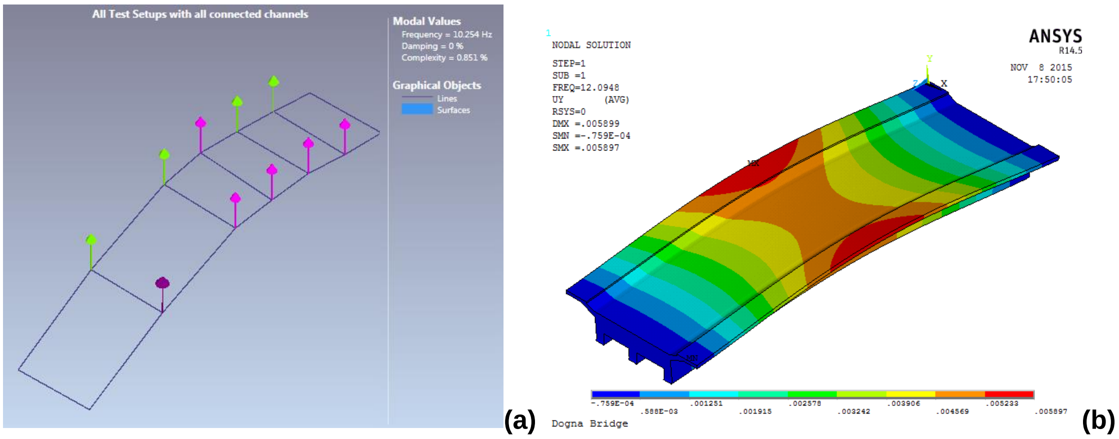
The Dogna bridge was essayed experimentally to obtain the dynamic proprieties of modal frequency and modal shape. The dynamic test use ambient vibration, as excitation. Figure 14 show the position of ten accelerometers. All the experimental signal accelerations of intact and damaged bridge cases were provided by Structural Vibration and Solutions (SVIBS) company, developer of ARTeMIS software (Artemis Modal, 2022).



(a) **(b)**

Figure 14 – Instrumentation in Dogna bridge:
(a) instrumental layout and (b) perspective view.

The frequencies and mode shapes identification was performed using the frequency domain decomposition method available in ARTeMIS software (Artemis Modal, 2022). Numerical model of reinforced concrete bridge was done in Ansys Mechanical using element SOLID65 (3-D Reinforced Concrete Solid). The equivalent reinforced concrete material characteristics was modal elasticity $E = 32GPa$, Poisson ratio $\nu = 0.3$ and concrete density $\rho = 2500\, kg/m^3$. The degrees of freedom on the supports were modeled by imposing nodal displacement constraint at ends of longitudinal beams. Figure 15 to Figure 17 compare experimental and numerical of first three mode shape of undamaged bridge structure. Figure 18 show the average normalized singular values of spectral density functions of all experimental signal used by FDD technique. Figure 19 shows auto-MAC correlation matrix of experimental mode shapes with a good agreement of first three modal shapes.



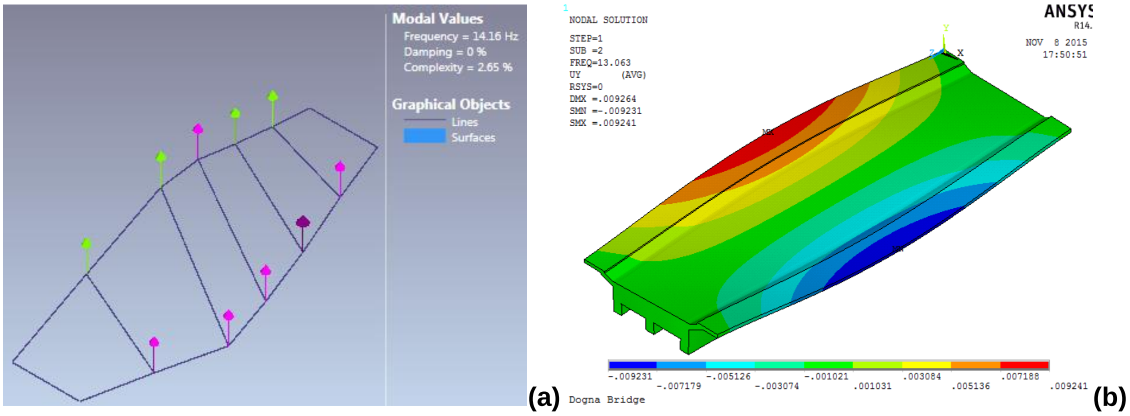
All Test Setups with all connected channels

Modal Values
Frequency = 10.254 Hz
Damping = 0 %
Complexity = 0.851 %

```
NODAL SOLUTION

STEP=1
SUB =1
FREQ=12.0948
UY        (AVG)
RSYS=0
DMX =.005899
SMN =-.759E-04
SMX =.005897
```

Figure 15 – First frequency mode shapes of intact beam: (a) experimental (10.25Hz), (b) numerical (12.09Hz).

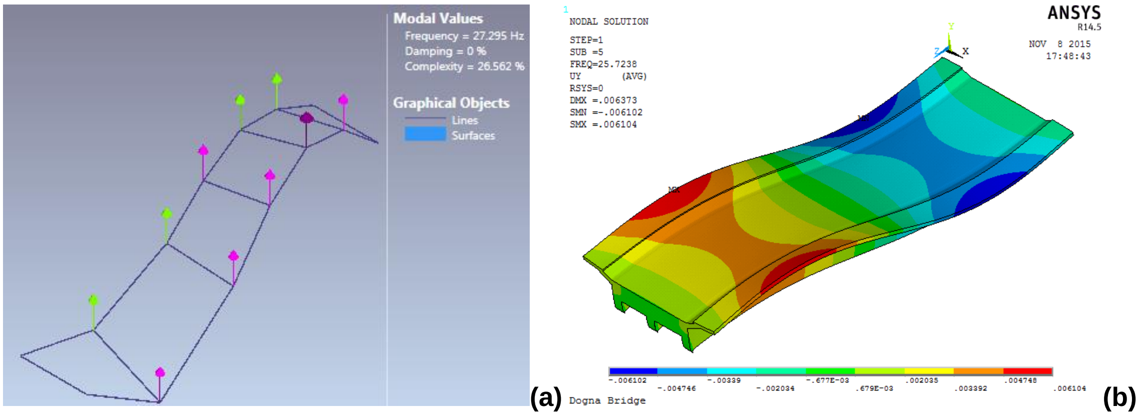


Modal Values
Frequency = 14.16 Hz
Damping = 0 %
Complexity = 2.65 %

```
NODAL SOLUTION

STEP=1
SUB =2
FREQ=13.063
UY        (AVG)
RSYS=0
DMX =.009264
SMN =-.009231
SMX =.009241
```

Figure 16 – Second frequency mode shapes of intact beam: (a) experimental (14.16Hz), (b) numerical (13.06Hz).



Modal Values
Frequency = 27.295 Hz
Damping = 0 %
Complexity = 26.562 %

```
NODAL SOLUTION

STEP=1
SUB =5
FREQ=25.7238
UY        (AVG)
RSYS=0
DMX =.006373
SMN =-.006102
SMX =.006104
```

Figure 17 – Third frequency mode shapes of intact beam: (a) experimental (27.29Hz), (b) numerical (25.72Hz).

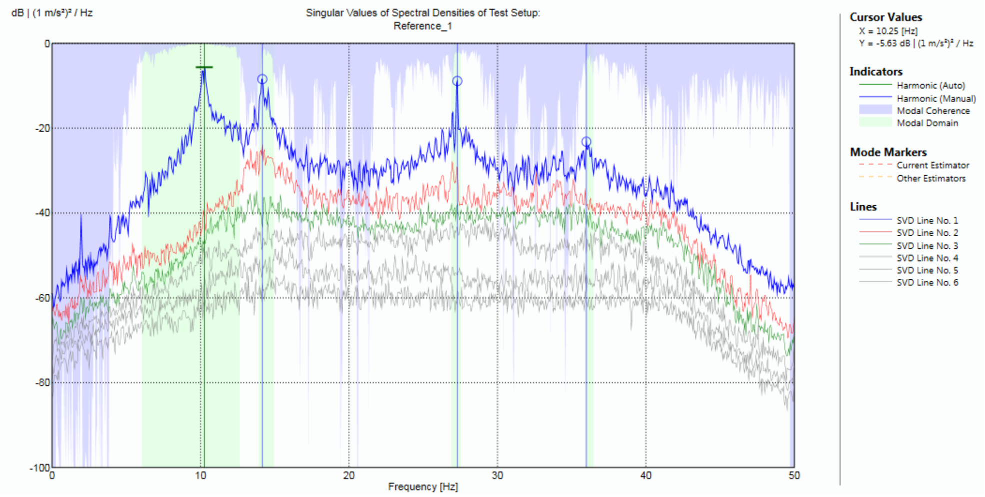


Figure 18 – Average normalized singular values of spectral density function using FDD.

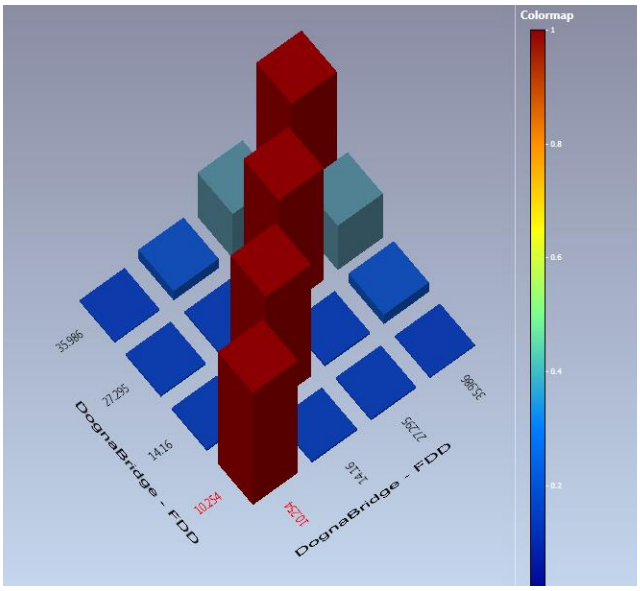


Figure 19 – AutoMAC correlation matrix of experimental mode shapes.

4.3 Damage localization

The damage case D1 consists of a half cut (45cm) of external beam. Damage case D1 represent reduction of 8% of total cross section of Dogna Bridge (Figure 20). A concrete cutter saw machine introduce the damage in external longitudinal beam (Figure 21). The experimental campaign for damage case D1 carries out a similar experimental procedure of intact case. Numerical modelling introduces damage deleting finite elements at the same position as the experimental test.

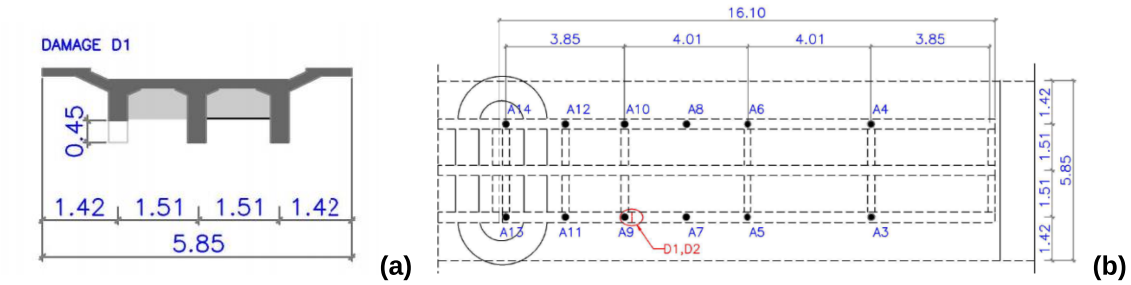


Figure 20 – Transversal cut view of damage D1(a) and localization of damage position by plan view(b).

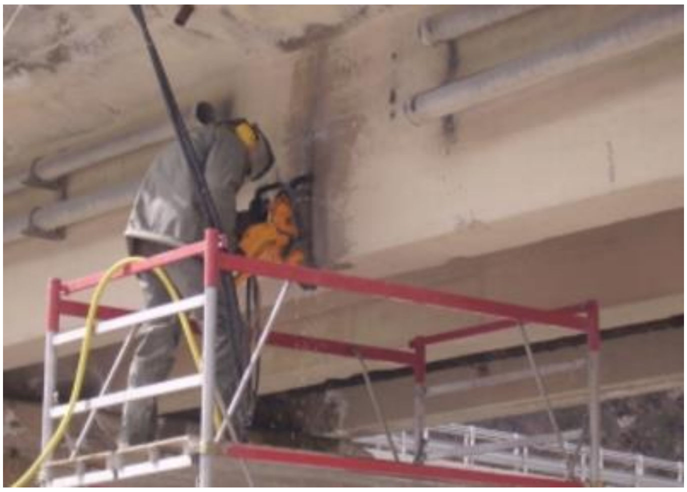


Figure 21 – View of artificial introduction of damage in longitudinal beam.

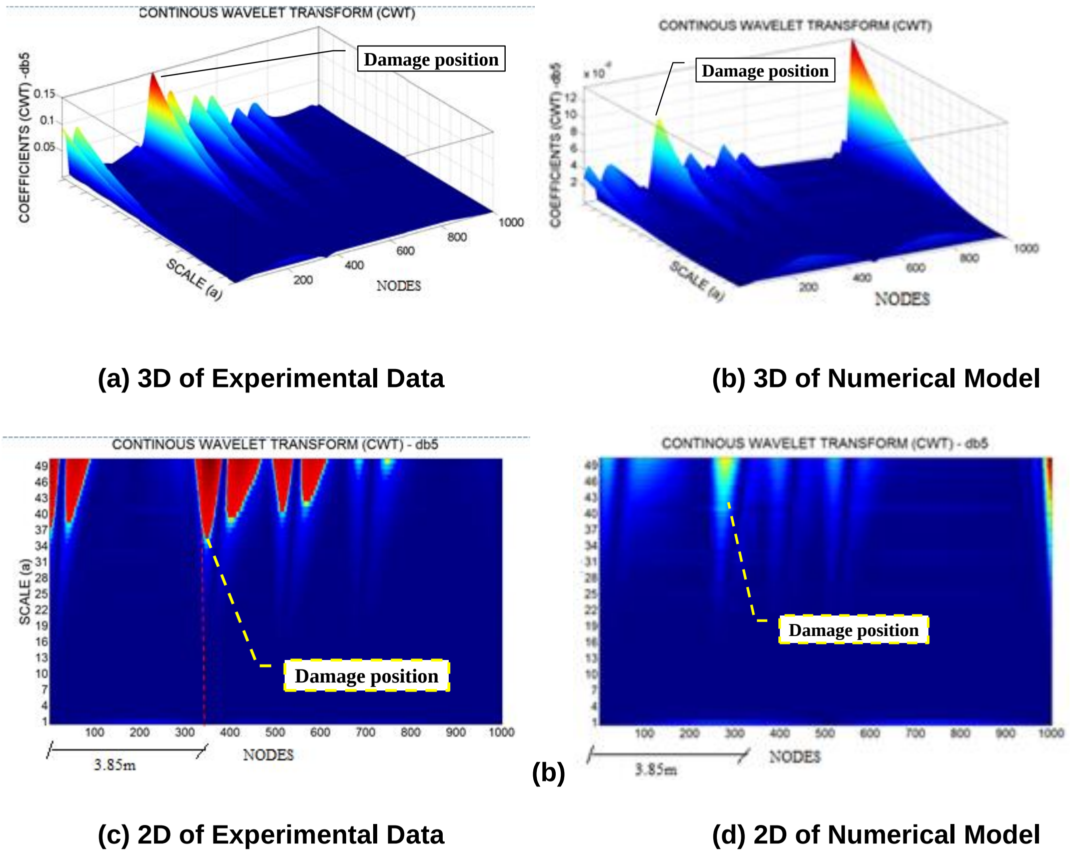
Table 4 compare modal frequency of first four modes for intact and damage D1 cases. presents the values of numerical natural frequencies of the intact and the damaged bridge. First frequency of experimental intact and damage results a sensible difference. And the numerical damage model presents a minor reduction compared to intact ones.

Table 4 – Comparation of modal frequencies (Hz) for intact and damage case D1.

Mode	Intact		Damage Case D1	
	Experimental	Numerical	Experimental	Numerical
1º	10.25	12.09	9.96	12.08
2º	14.16	13.06	14.06	13.05
3º	27.29	25.72	27.63	25.72
4º	35.99	38.29	35.32	38.28

Applying the proposed methodology, it's used the first mode shape of damaged D1 to determine damage localization. The data set of five points, corresponding to a line of first mode shape is exported to Matlab. Cubic spline interpolate this data set transforming 5 points in 1000 nodes. The Tikhonov regularization technique regularize this interpolated data set. Finaly, CWT determine wavelet coefficients of regularized data set.

Figure 22 show the experimental and numerical comparation of CWT for damage D1 case. The determination of wavelet coefficients uses Daubechies5 wavelet mother function (Db5). We observe important discontinuity of wavelet coefficients near the damage position for experimental results and numerical model.



(a) 3D of Experimental Data

(b) 3D of Numerical Model

(c) 2D of Experimental Data

(d) 2D of Numerical Model

Figure 22 – Experimental and numerical comparation of CWT for damage D1 case.

5 Damage Detection using 2D Boundary Element Method

Silva, Bezerra et al (2019) localize damage in 2D structure by wavelet transform of static boundary element solution. After presenting boundary element formulation, we show some examples concerning the localization of subsurface cracks using static loads in two dimensional structures. Single and multi-cracks with different orientations inside two-dimensional structures with different support conditions are investigated. The results show the proposed numerical procedure is effective in indicating the location of damages.

5.1 Numerical examples

There are three cases of two-dimension deep beam: (a) cantilever beam single cracked, (b) cantilever beam single cracked with simulated noise, and (c) fixed panel multi-cracked. These three cases are simulated with same material with Young modulus $E = 200GPa$ and Poisson ratio $\nu = 0.30$. The static solution, obtained by boundary element method, was carried out by internal points in reference line. DWT obtain wavelet coefficients using mother wavelet Biorthognal 3.7 (bior3.7).

5.1.1 Cantilever beam

Figure 23 present a schematic representation of cantilever deep beam case with boundary conditions clamped-free, i.e., fixed in left side and free and right side. The dimensions are length $L = 500mm$ and height $H = 100mm$. Static load is a concentrated force $F = 500kN$ at the top of free right end. An embedded vertical crack with length $25mm$, are induced at a distance $d = 330mm$. Boundary element model discretized with 100 quadratic elements for beam domain and 25 quadratic elements for crack (Figure 23). Schematic representation illustrates reference line at $25mm$ from bottom line, without intersect crack, with 480 discretized points equally distributed of $1mm$. The reference line starts at $10mm$ away from left side and finish at $10mm$ before right side. But previous analysis show that others reference line present similar results for wavelet transformation.

After simulate described BEM model using Elast_qua (Silva, Bezerra, *et al.*, 2019), Figure 24 show beam deformed shape. Figure 25 show DWT wavelet coefficients of vertical displacement $u(x)$, from internal points of reference line, performed by MATLAB/Wavetoolbox Biorthogonal 3.7 (bior3.7).

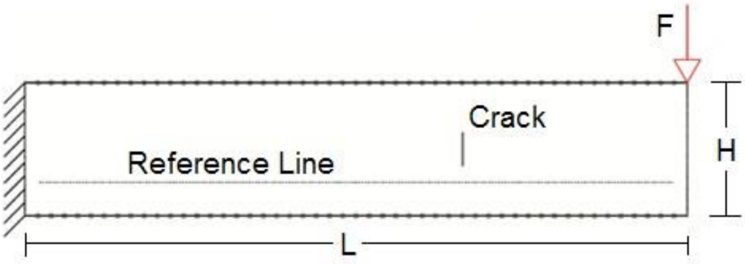


Figure 23 – BE mesh of cantilever beam case and internal nodes.

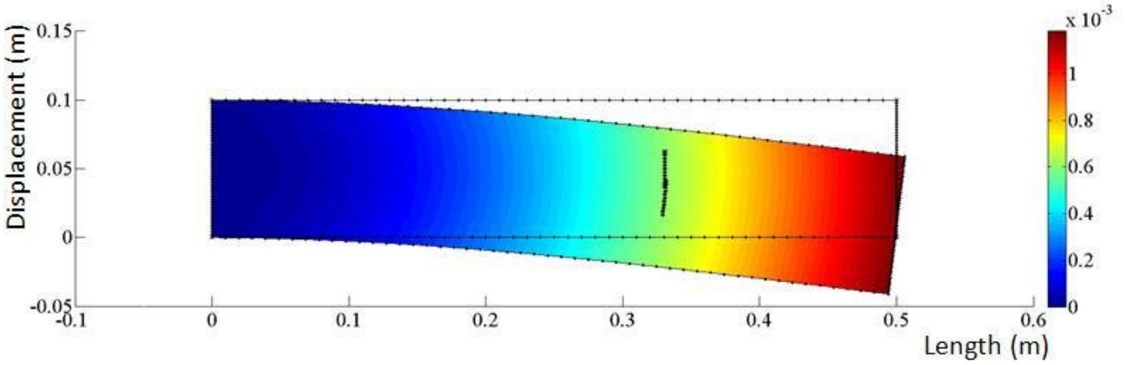


Figure 24 – Deformed shape of cantilever beam case.

Figure 25 shows wavelet coefficients results of DWT of $u(t)$ (reference line). The position at 320 nodes correspond approximately to internal crack, i.e., estimated crack distance $d_{est} = (320 - 1) \cdot 1mm + 10mm \simeq 330mm$ is well estimated by present wavelet transformation. And similar to others results, it can be observed similar anomalous perturbation at extremity of wavelet coefficients DWT.

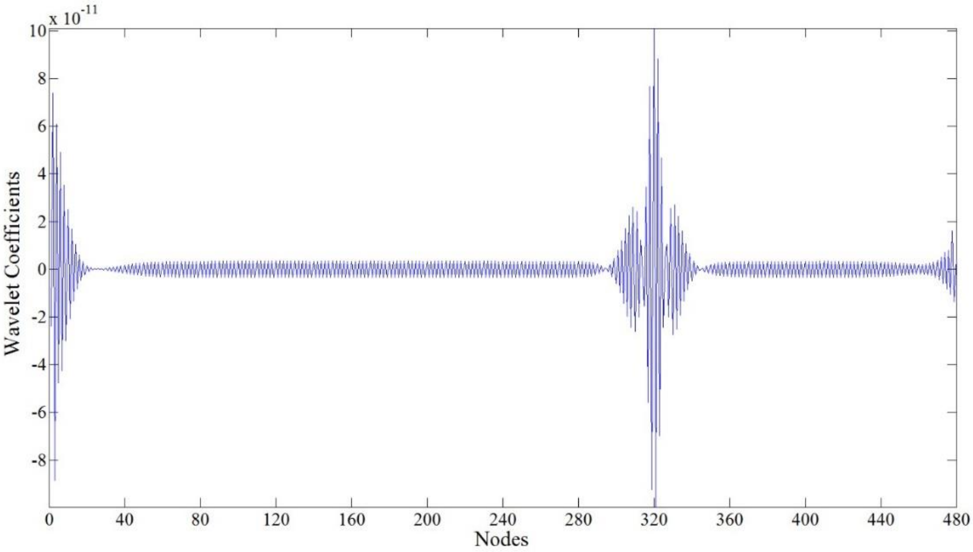


Figure 25 – Wavelet coefficients DWT using bior3.7 for cantilever beam case.

5.1.2 Cantilever beam with noise

During experimental tests, there are noticeable levels of noise in sensor readings signal. To simulate experimental conditions, white Gaussian noise has been introduced into static signal $u(t)$ at reference line. The noise levels were 0.5%, 1% and 2% of noise of the maximum measured displacements.

As describe in Figure 26, the dimensions are length $L = 500mm$ and height $H = 100mm$. And the concentrated load $F = 500kN$ at top of free right side. An embedded vertical crack with length $25mm$, are induced at a distance $d = 125mm$. The reference line starts at $10mm$ away from left side and finish at $10mm$ before right side.

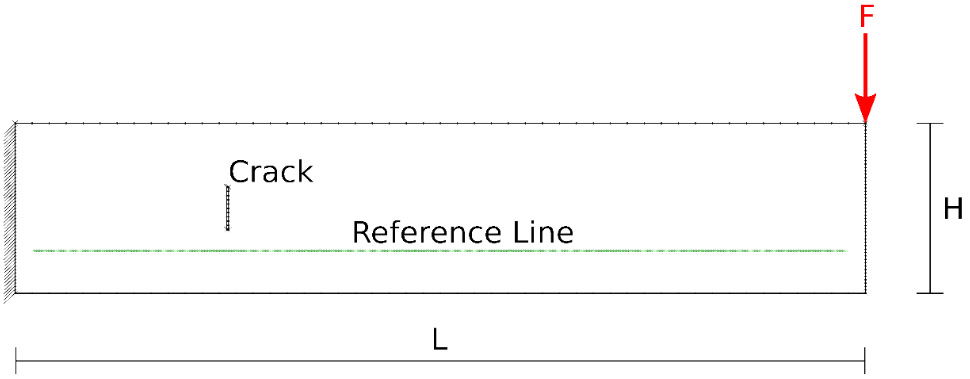


Figure 26 – BE mesh of cantilever beam case with noise.

Figure 27 show beam deformed shape. Figure 28 show DWT wavelet coefficients for the present cantilever beam without noise. Figure 29 to Figure 31 show DWT wavelet coefficients of static displacement $u(t)$ added with 0.5%, 1.0% and 2.0% of white Gaussian noise, respectively. The noise/signal ratio is important to present methodology of damage localization. For low levels of noise, the disturbance in DWT wavelet coefficients do not difficult the localization of crack. But, with 2.0% of noise, the noise pollution difficult to detect clearly the crack position.

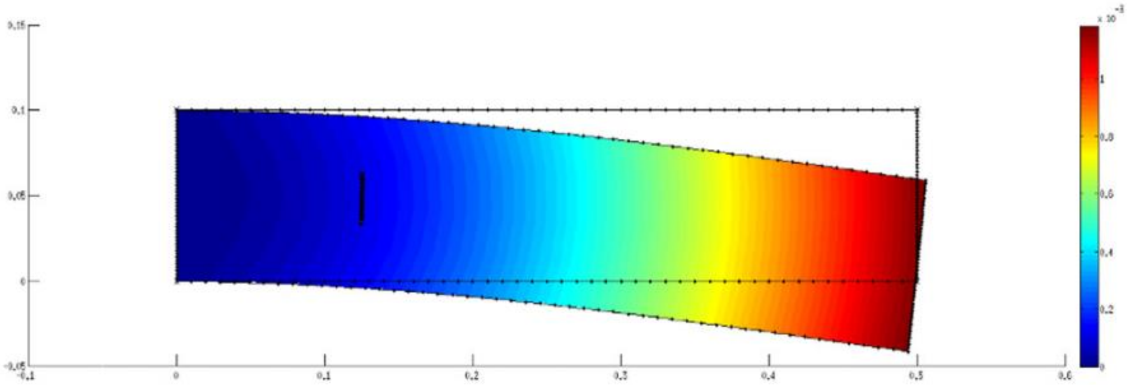


Figure 27 – Deformed shape of cantilever beam case with noise.

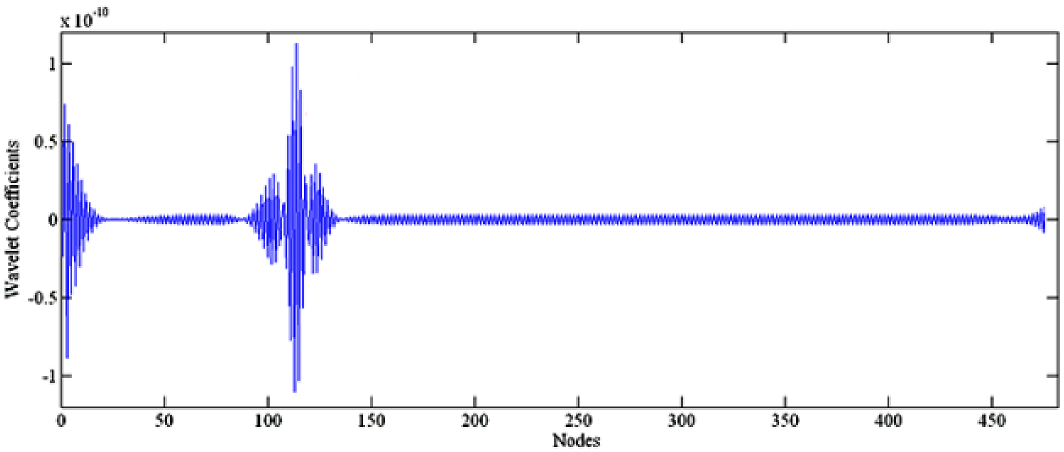


Figure 28 – Wavelet coefficients using bior3.7 of cantilever beam case without noise.

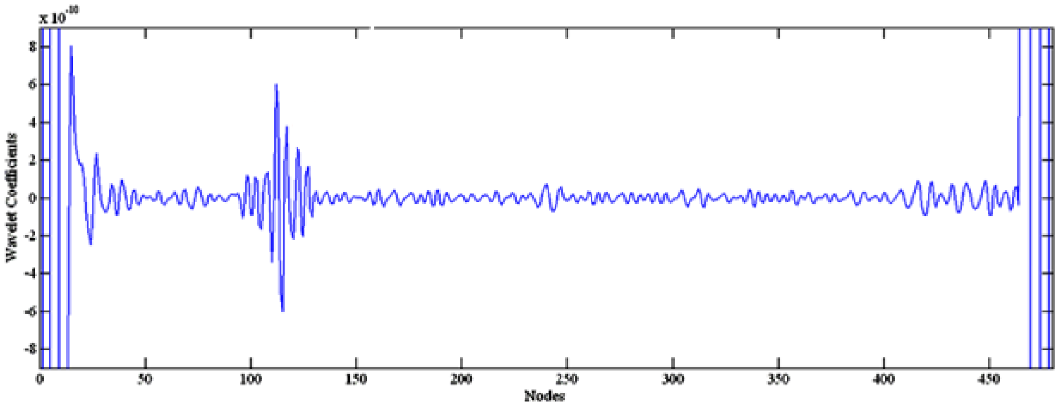


Figure 29 – Wavelet coefficients using bior3.7 of cantilever beam case with 0.5% of noise.

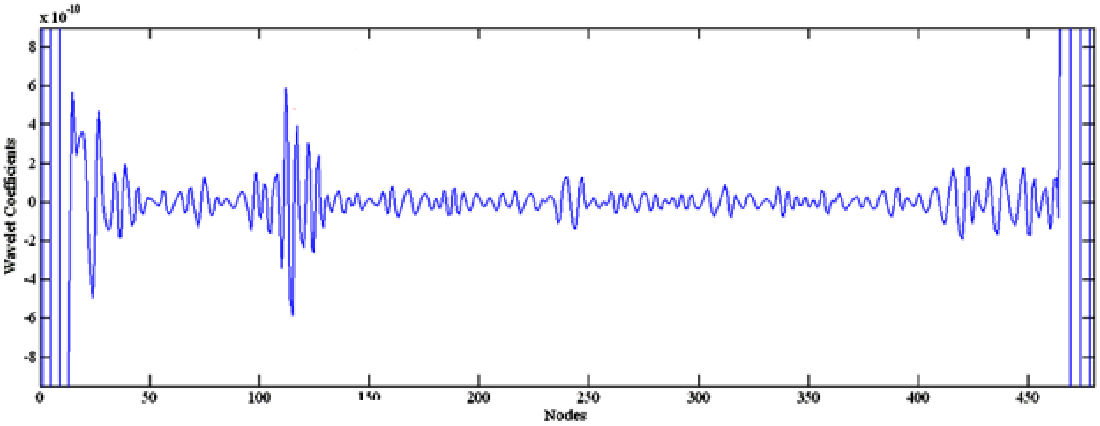


Figure 30 – Wavelet coefficients using bior3.7 of cantilever beam case with 1.0% of noise.

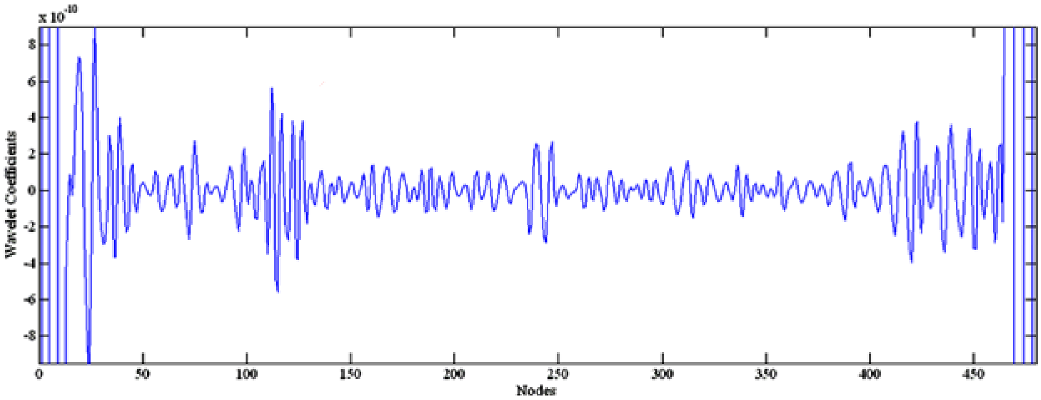


Figure 31 – Wavelet coefficients using bior3.7 of cantilever beam case with 2.0% of noise.

5.1.3 Beam with two fixed ends

Figure 32 show an example with coupled bending and tension forces. The panel fixed on left end. And a uniform distributed load $q = 0.5kN/mm$ is applied vertically at panel top and horizontally at panel right end. The dimensions are length $L = 2000mm$ and height $H = 1000mm$. Multiple cracks are positioned vertically ($d = 333mm$ and $1667mm$ at middle) and horizontally ($d = 1000mm$) with respect left end side, as describe in Figure 32. Embedded cracks have length $25mm$. Reference line, positioned $167mm$ from bottom line, is discretized with 480 nodes equally distributed. The reference line starts at $44mm$ away from left side and finish at $44mm$ before right side. Figure 33 show beam deformed shape.

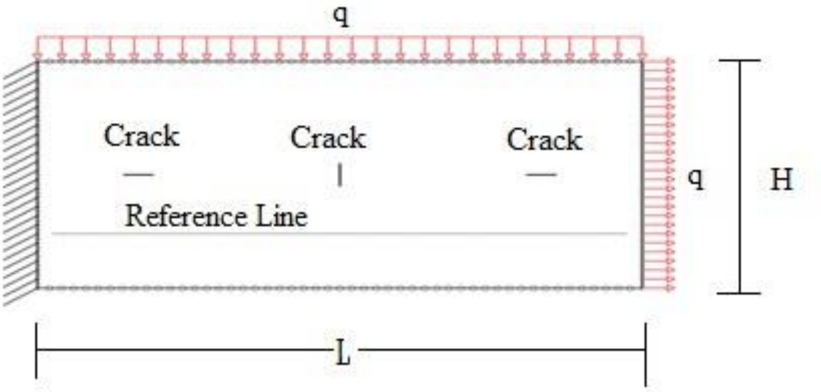


Figure 32 – BE mesh of panel with one side fixed.

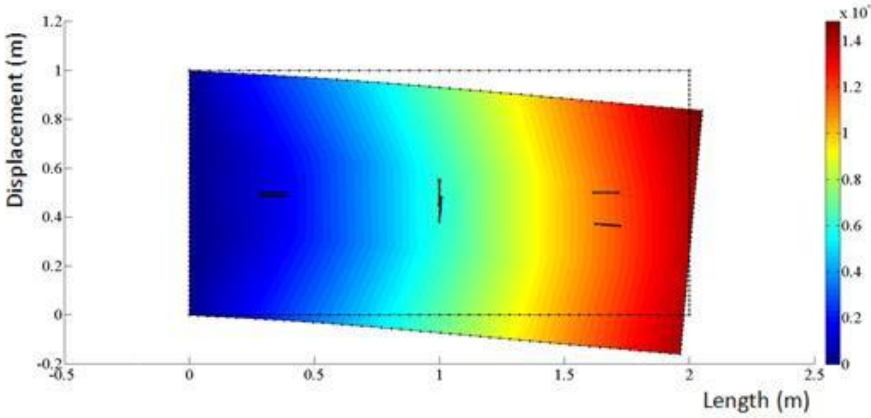


Figure 33 – Deformed shape of panel with one side fixed.

Figure 34 show DWT wavelet coefficients of vertical displacement $u(x)$, from internal points of reference line, performed by MATLAB/Wavetoolbox Biorthogonal 3.7 (bior3.7). The wavelet coefficients peaks correspond nodes 74, 240 and 408. These peaks correspond to damages locates with centers near to 339, 1000, and $1669mm$ measured from panel left end. Under coupled bending and tension forces, the bending deformation is more intense close to free end. The wavelet coefficients have narrow peaks near to vertical crack while the other horizontal cracks spread peaks in DWT. In any case, the numerical methodology was able to indicate the damage localization.

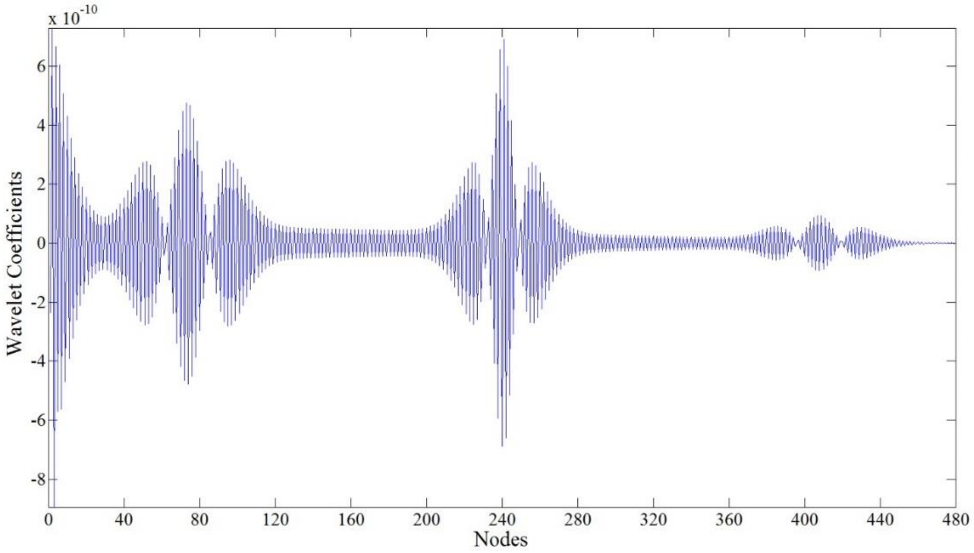


Figure 34 – Wavelet coefficients DWT using bior3.7 of panel with one side fixed.

6 Concluding remarks

Wavelet transform is an interesting tool to damage detection and localization. In this chapter, three research application of wavelet transform are shown in 1D, 2D and 3D structures using experimental and/or numerical data. The wavelet transform only requires structure damage response. It's a advantage in comparation to traditional damage identification methods.

Firstly, the detection of open crack damage on laminated steel beams are presented using experimental static displacement. Discrete and continuous wavelet transform with Daubechies mother wavelets was applied to obtain wavelet coeficients of experimental data. Wavelet coefficients are sensible to damage intensity. The 40mm-depth cracks results in more intensive wavelet coefficients picks than 20mm-depth ones.

The second aplication presents the damage localization on Dogna bridge (Italy). Using a proposed methodology, a damage localization using continous wavelet transform associated with spline interpolation and regularization techniques applied to experimentaly identified mode shapes is presented. Numerical 3D FE model was calibrated with experimental data. The damage localization compare the wavelet coefficients of experimental data and numerical 3D FE results with similar results. As application test, the authors present the analysis of Dogna Bridge (Italy) using only damage response.

At last, the wavelet coeficient was applied to localize damange in 3D structure modeled by boundary element method. Static response of elastic linear structures with single and multi-cracks was analysed using discrete wavelet transform. The damage localization using DWT with bi-ortoghonal mother wavelets was effective. And noise/signal ratio is an important influence to methodology successfully.

7 References

Artemis Modal (2014) *Damage Detection of the Dogna Bridge Italy, Internal Report*. Available at: https://svibs.com/cases/damage-detection-of-the-dogna-bridge-italy/ (Accessed: 4 February 2022).

Artemis Modal (2022) 'Overview of ARTeMIS Modal Versions and Features ARTeMIS Modal is a powerful and versatile tool designed for the following analysis types : Operational Modal Analysis (OMA) Experimental Modal Analysis (EMA) Operating Deflection Shapes (ODS) Structura', p. 2.

Friswell, M. I. (2007) 'Damage identification using inverse methods', *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 365(1851), pp. 393–410. doi: 10.1098/rsta.2006.1930.

Janeliukstis, R. *et al.* (2017) 'Experimental structural damage localization in beam structure using spatial continuous wavelet transform and mode shape curvature methods', *Measurement*. Elsevier, 102, pp. 253–270. doi: 10.1016/j.measurement.2017.02.005.

Kim, H. and Melhem, H. (2004) 'Damage detection of structures by wavelet analysis', *Engineering Structures*, 26(3), pp. 347–362. doi: 10.1016/j.engstruct.2003.10.008.

Ovanesova, A. V. (2000) 'Applications of Wavelets to Crack Detection in Frame Structures.', p. 235.

Ovanesova, A. V. and Suárez, L. E. (2004) 'Applications of wavelet transforms to damage detection in frame structures', *Engineering Structures*. Elsevier BV, 26(1), pp. 39–49. doi: 10.1016/j.engstruct.2003.08.009.

Palechor, E. U. L. (2013) *Damage Identification in Steel Beams by Wavelets and Numerical/Experimental Signal (por. Identificação de Danos em Vigas Metálicas Utilizando Wavelets e Dados Numéricos e Experimentais).* Universidade de Brasília. Available at: https://repositorio.unb.br/handle/10482/14814.

Palechor, E. U. L. *et al.* (2014) 'Damage Identification in Beams Using Experimental Data', *Key Engineering Materials*, 607, pp. 21–29. doi: 10.4028/www.scientific.net/KEM.607.21.

Palechor, E. U. L. *et al.* (2022) 'Fundamental Concepts on Wavelet Transforms', in.

Silva, R. S. Y. R. C., Bezerra, L. M., *et al.* (2019) 'Boundary element and wavelet transform methods for damage detection in 2D structures', *International Journal for Computational Methods in Engineering Science and Mechanics*. Taylor & Francis, 20(3), pp. 242–255. doi: 10.1080/15502287.2019.1631407.

Silva, R. S. Y. R. C., Palechor, E. U. L., *et al.* (2019) 'Damage detection in a reinforced concrete bridge applying wavelet transform in experimental and numerical data', *Frattura ed Integrità Strutturale*, 13(48), pp. 693–705. doi: 10.3221/IGF-ESIS.48.65.

Yang, C. and Oyadiji, S. O. (2017) 'Damage detection using modal frequency curve and squared residual wavelet coefficients-based damage indicator', *Mechanical Systems and Signal Processing*. Academic Press, 83, pp. 385–405. doi: 10.1016/j.ymssp.2016.06.021.

Chapter 11: Inverse Methods using KF, EKF, EIF, PF, and LS Techniques for Detection, Localization, and Parameter Estimation

Chapter details

Chapter DOI:

https://doi.org/10.4322/978-65-86503-71-5.c11

Chapter suggested citation / reference style:

Myers, Michael R., Jorge, Ariosto B. (2022). "Inverse Methods using KF, EKF, EIF, PF, and LS Techniques for Detection, Localization, and Parameter Estimation". *In* Jorge, Ariosto B., et al. (Eds.) *Model-Based and Signal-Based Inverse Methods*, Vol. I, UnB, Brasilia, DF, Brazil, pp. 382–442. Book series in Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity.

P.S.: DOI may be included at the end of citation, for completeness.

Book details

Book: Model-based and Signal-Based Inverse Methods

Edited by: Jorge, Ariosto B., Anflor, Carla T. M., Gomes, Guilherme F., & Carneiro, Sergio H. S.

Volume I of Book Series in:

Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity

Published by: UnB City: Brasilia, DF, Brazil Year: 2022

DOI: https://doi.org/10.4322/978-65-86503-71-5

Inverse Methods using KF, EKF, EIF, PF, and LS Techniques for Detection, Localization, and Parameter Estimation

Michael Richard Myers[1*], Ariosto Bretanha Jorge[2]

[1*]Senior Systems Modeling & Simulation Engineer, Axiom Space, Inc., Houston, Texas, USA.
E-mail: mike.myers@funkandmyers.com
[2]Post-Graduate Program - Integrity of Engineering Materials, University of Brasilia, Brazil.
E-mail: ariosto.b.jorge@gmail.com

*Corresponding author.

Abstract

This chapter presents some fundamental concepts for detection, localization, and parameter estimation using Kalman filter, extended Kalman filter, extended information filter, particle filter, and least squares. These techniques can be used to characterize singularities, discontinuities, material properties, boundary conditions, loading, damage, structural changes, etc. Unfortunately, directly characterizing parameters of interest such as structural damage and concentrated heating sources is often difficult or not possible with current sensors. Other quantities can be measured and the parameters of interest can be estimated using an inverse method. Several types of static and dynamic loads and the structural deterioration process can cause different types of structural damage. Parameter estimation addresses the problem of estimating quantities that are not directly observable and can be inferred from sensor data. Sensors carry only partial information and their measurements are corrupted by noise. Parameter estimation seeks to recover state variables from the sensor data. Probabilistic parameter estimation algorithms compute belief distributions over possible world states.

Keywords

detection, localization, paramter estimation, characterization, structural damage, heating source, material properties, Kalman filter, extended Kalman filter, particle filter, least squares, information filter.

1 Introduction

It is increasingly essential that a structure or equipment not fail. When considering the particular case of an aircraft wing, where failure can be catastrophic, it is possible to see the importance of maintaining structural integrity. All aircraft receive maintenance during its life, where the frequency and type of maintenance depend on the type of aircraft and the cycle (takeoff - landing) or flight hours. But some points of the aircraft are difficult to access, if not impossible, without damaging the structure. In order to monitor these points, it was thought the inclusion of sensors during manufacture, that during the same maintenance in a hangar, a window would have access to the communication of sensors, with some equipment for data collection. These data provided by sensors, with the analysis made by the present method in a station of data collection, indicate the presence or absence of damage to the monitored region of the structure. Structural damage estimates could be compared between two independent objective functions such as mean stress and octahedral stress. A pareto front could be used to find the best match.

Characterizing concentrated heating sources is another engineering challenge. For example, knowledge of where air flowing across a body transitions from laminar flow to turbulent flow (Figure 1) can provide numerous

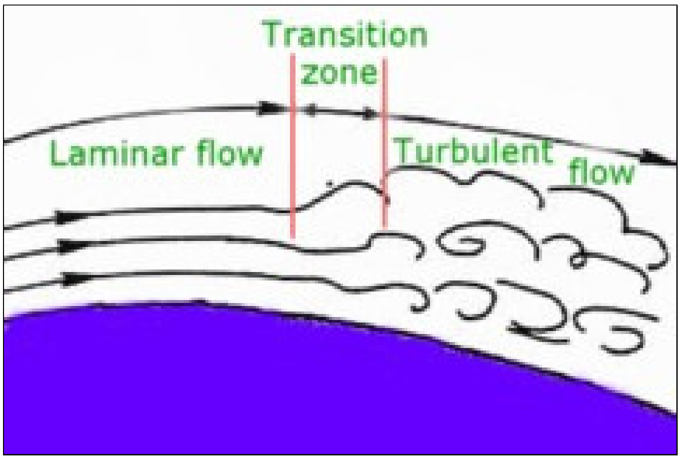


Figure 1: Flow regimes Recreational Aviation Australia [2010].

benefits to air vehicle design, thermal protection system design, and air vehicle in-flight control Reed et al. [1997]. Of particular interest is the transition region for hypersonic vehicles (Mach 5+). At the transition between laminar and turbulent flow, a change in body-surface temperature has been measured for hypersonic conditions Horvath et al. [2002], Schneider [1999, 2004], Berger et al. [2009] and is illustrated in Figure 2. Thus, a measurement system is envisioned that leverages the hypersonic body-surface heating profile to locate the boundary layer transition region.

Unfortunately, directly characterizing structural damage and concentrated heating sources is often difficult or not possible with current sensors. Other quantities can be measured and the parameters of interest can be estimated using an inverse method. Several types of static and dynamic loads and the structural deterioration process can cause different types of structural damage. The damage can be characterized by a change in the structure, such as the presence of holes and cracks. The knowledge of the change in the material properties corresponding to the damage depends on the type of material and structural configurations. The proper assessment of the damage in a structure can be useful to infer its remaining service life. The assessment of the structural damage can be performed through a comparison between measured and simulated data. To provide the simulated data, a numerical code is required, in which a direct model of the problem is consistently used by an inverse problem algorithm. For the direct problem, a model is required to obtain the information on the distribution of the quantity of interest throughout the structure, given the boundary conditions and the presence of the damage. For the inverse problem, a model is required for the procedure of locating the damage in the structure given some (partial) information on the quantity of interest at some particular locations (e.g. where some sensors are placed).

The presence of damage may induce rapid changes in the field variable of the problem and even discontinuities in the governing equation in the domain. Classical calculus-based optimization methods require evaluation of derivatives of the objective function, which may not be possible to be obtained, or may be numerically obtained, with unacceptable inaccuracy. These problems can also have several local minima (multiple solutions), and thus a global optimization method is a better choice for the numerical solution Stavroulakis and Antes [1998], Engelhardt et al. [2006].

Numerical methods, such as the boundary element method or the finite element method can be used for modelling the direct problem. The damage detection problem can be considered as a problem of system identification or an inverse problem. The inverse problem of identifying the presence, location and size of damage, such as cracks and holes, in a plate structure can be modelled using optimization and parameter identification techniques. The remainder of this chapter focuses on inverse methods using finite element method and the annex focuses on inverse methods using boundary element method Vieira et al. [2011].

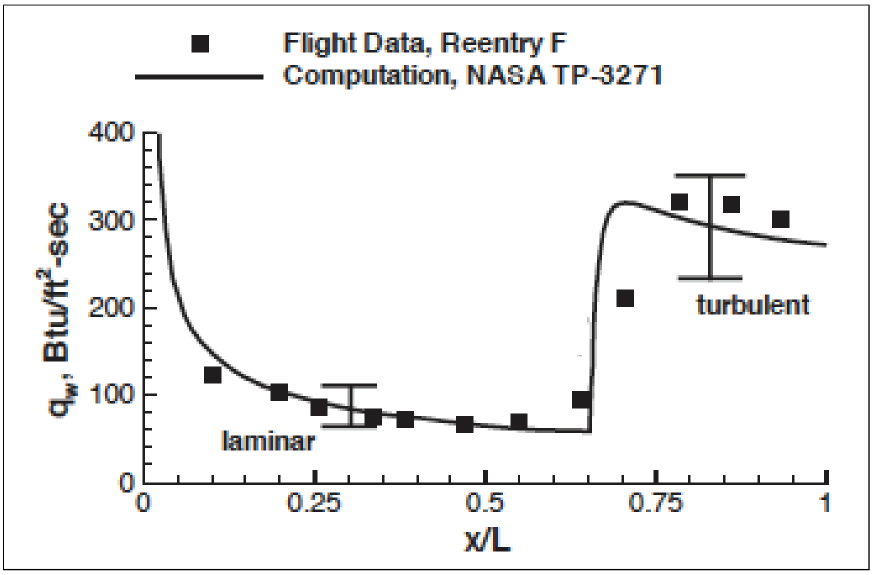


Figure 2: Heating profile on a ballistic RV, peak Mach=20 Schneider [2004].

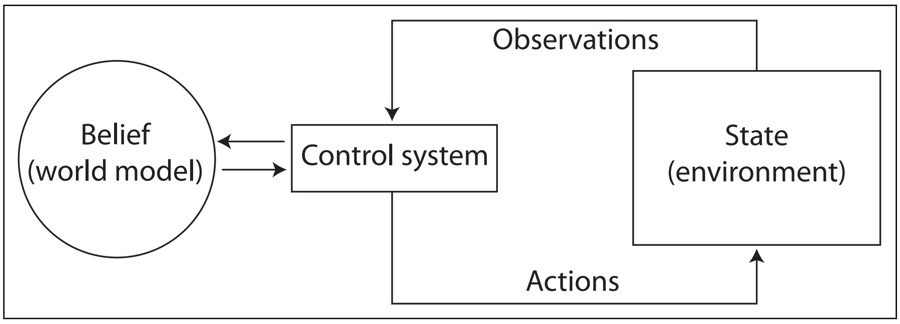


Figure 3: Object-environment interaction.

2 Detection, localization, and parameter estimation

Figure 3 illustrates the conceptual interaction of an object with its environment. The environment is a dynamic system that possesses an internal state. It is convenient to think of the state as the collection of all the object's aspects and its environment that can affect the future. Certain state variables can change over time such as the object's speed, acceleration, temperature, and location. Other state variables tend to remain static, such as the location of the ground or the earth. There are endless possibilities for potential state variables. The first challenge is to determine which of the potential state variables are important and need to be included in the problem. Throughout this chapter, state will be denoted X and the specific variables included in the state will depend upon the context. The state at time t will be denoted X_t. Time is considered discrete, that is, all interesting events take place at discrete time steps $t = 0, 1, 2, \ldots$.

There are two fundamental types of interactions between an object and its environment: The object can influence the state of its environment through its control system, and it can gather information about the state through its sensors. Control actions include moving control surfaces and applying force to accelerate or decelerate the object. Sensors are noisy and many parameters cannot be measured directly. As a consequence, the object maintains an internal belief with regards to its state. The distinction between measurement and control is crucial. While measurements over time tend to increase the object's knowledge, motion and control inputs tend to induce a loss of knowledge due to the inherent noise in control actuation and the stochasticity of the environment.

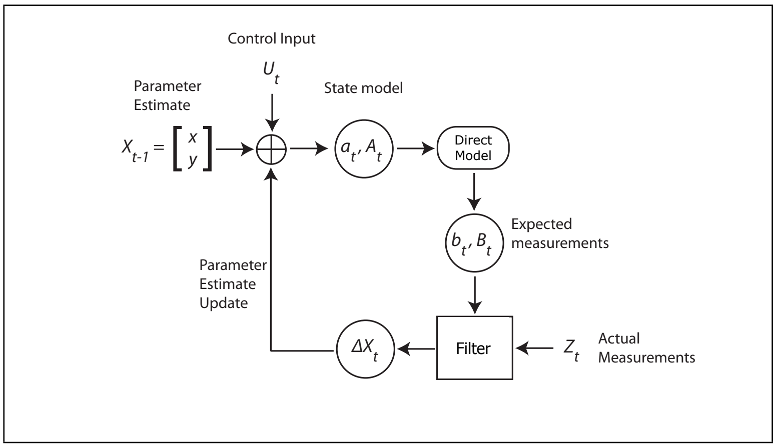


Figure 4: Parameter estimate update process using an inverse method and a direct model.

Parameter estimation addresses the problem of estimating quantities that are not directly observable but that can be inferred from sensor data. Sensors carry only partial information and their measurements are corrupted by noise. Parameter estimation seeks to recover state variables from the sensor data. Probabilistic parameter estimation algorithms compute belief distributions over possible world states. A state X_t is called complete if it is the best predictor of the future. Completeness entails that knowledge of past states, measurements, and control inputs carry no additional information that would help us predict the future more accurately. In practice, it is impossible to specify a complete state for any realistic singularity or discontinuity situation. A complete state includes not only aspects of the object itself and of the environment immediately surrounding the object being studied but also the environment away from the object that may affect its future. Some of these elements are hard to obtain and therefore practical implementations single out a small subset of all state variables.

The Markov assumption postulates that past and future data are independent if one knows the current state X_t. Unmodeled environment dynamics, inaccuracies in probabilistic models, and approximation errors induce violations of the Markov assumption. In principle, many of these variables can be included in the state, however, incomplete state representations are often preferable to more complete ones to reduce the computational complexity of the filter algorithm. It is advisable to exercise care when defining the state X_t so that the effect of unmodeled state variables has close to random effects.

Figure 4 illustrates the parameter estimate update process for an inverse method and a direct model. The estimated state is $X_t = [x_s, y_s]^T$ where x_s and y_s represent the estimated parameters of interest at time t. The parameters of interest (state) may be the location and size of structural damage or concentrated heating source.

3 Methods

This section details some inverse methods that have proven useful for estimating state variables that cannot be directly observed.

3.1 Kalman Filter

The Kalman filter was developed by R. E. Kalman in 1960 to solve the model of space state of Wierner, recursively. Combining the filter with the advances in digital computing, the technique had a strong application in automation and assisted navigation. The Kalman filter, was originally developed for discrete linear systems and have white Gaussian noise. What does not occur in real systems, but may, in some regions, be approximated to linear, extending its application. But for the Kalman filter is able to estimate the state, it is necessary that the process is observable. This filter is the original Kalman filter and provided the basis for the development of derivatives, the extended Kalman filter, and many hybrid filters.

The Kalman filter is a member of a family of recursive state estimators collectively called Gaussian filters.

Table 1: Kalman filter algorithm.

Step	Operation
1	$\overline{X}_t = A_t X_{t-1} + B_t X_t$
2	$\overline{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + Q_t$
3	$K_t = \overline{\Sigma}_t C_t^T (C_t \overline{\Sigma}_t C_t^T + R_t)^{-1}$
4	$X_t = \overline{X}_t + K_t (Z_t - C_t \overline{X}_t)$
5	$\Sigma_t = (I - K_t C_t) \overline{\Sigma}_t$
6	Return to step 1 for next time step

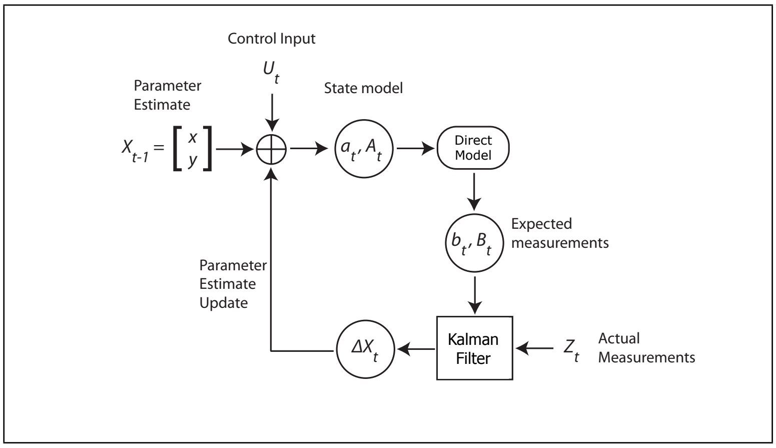


Figure 5: Parameter estimate update process using the Kalman filter and a direct model.

Historically, Gaussian filters constitute the earliest tractable implementations of the Bayes filter for continuous spaces Thrun et al. [2006]. Kalman filters construct a framework of predicting the state based on an input to the system and correcting the predicted state based on sensor observations. Kalman filters were invented by Swerling (1958) and Kalman (1960) as a technique for filtering and prediction in linear Gaussian systems Thrun et al. [2006]. Kalman filters assume that all continuous random variables possess probability density functions (PDFs). A common density function is that of the one-dimensional normal distribution with mean μ and variance σ^2. The PDF of a normal distribution is given by the following Gaussian function:

$$p(x) = (2\pi\sigma^2)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right\} \tag{1}$$

Normal distributions play a major role in Kalman filters and are abbreviated as $N(\mu, \sigma^2)$ which specifies mean of the random variable and its variance. The normal distribution in equation 1 assumes that x is a scalar value. All Gaussian filters share the basic premise that beliefs are multivariate normal distributions. The PDF of a multivariate normal distribution is given by the following Gaussian function:

$$p(X) = \det(2\pi\Sigma)^{-\frac{1}{2}} \exp -\frac{1}{2}(X-\mu)^T \Sigma^{-1}(X-\mu) \tag{2}$$

Where X is a multivariate vector, μ is the mean vector, Σ is a positive semidefinite and symmetric matrix called the covariance matrix.

The Kalman filter consists of two phases, the phase prediction and phase correction. After the initial estimate, the Kalman filter begins one update cycle of the two phases until convergence of the state, ie, the state predicted corresponds to the state that generated the measurement. Figure 5 illustrates the process cycle of Kalman filter and Table 1 contains the full Kalman filter algorithm.

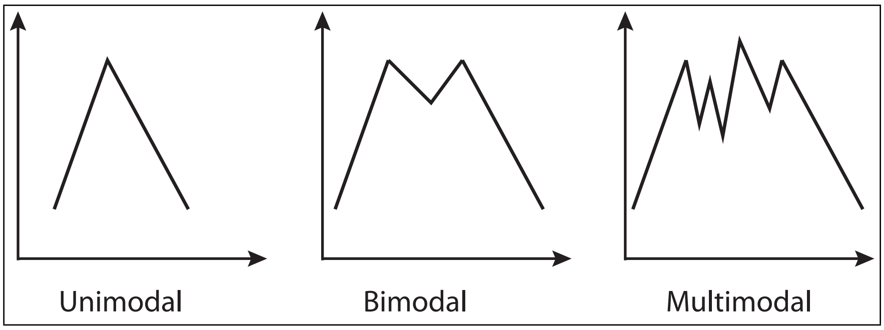


Figure 6: Unimodal illustration.

3.2 Extended Kalman Filter

The extended Kalman filter linearizes nonlinear Gaussian systems. Kalman filters implement belief computation for continuous states with all disturbances additive and Gaussian with zero mean.

$$X_t = a(U_t, X_{t-1}) + \epsilon_t, \quad \epsilon_t \sim N(0, Q_t) \tag{3}$$

$$Z_t = b(X_t) + \delta_t, \quad \delta_t \sim N(0, R_t) \tag{4}$$

Where a and b are nonlinear functions, U_t is the input, X_t is the state, Z_t is the observation, and $\epsilon_t \sim N(0, Q_t)$ and $\delta_t \sim N(0, R_t)$ represent Gaussian random disturbances with zero mean and the specified covariance. a represents the state transition function and its purpose is to predict the current state $\overline{X}_t$, based on the previous state X_{t-1} and the current control input U_t. For a hypersonic vehicle, the control input to the extended Kalman filter could be the pilot's flight control commands (e.g., throttle, attitude controls, etc.) and sensor data (e.g., angle of attack, altitude, etc.). b represents the measurement transition function and are the expected measurements based on the current state X_t. ϵ_t is a Gaussian random variable that models the uncertainty introduced by the state transition function and δ_t describes the measurement noise.

Kalman filters assume a unimodal approximation to the true belief. A function is unimodal if for some value m (the mode), it is monotonically increasing for $x \leq m$ and monotonically decreasing for $x \geq m$. In that case, the maximum value of $f(x)$ is $f(m)$ and there are no other local maxima (Figure 6).

The Kalman filter assumes linear state and measurement models. Unfortunately, state transitions and measurements are rarely linear in practice. The extended Kalman filter relaxes this linear assumption by approximating the nonlinear state and measurement models with a first order Taylor expansion linear model (Figure 7). Instead of passing the Gaussian through the nonlinear function g, it is passed through a linear approximation of g. The linear function is tangent to g at the mean of the original Gaussian. The resulting Gaussian is shown as the dashed line in the upper left graph. The linearization incurs an approximation error, as indicated by the mismatch between the linearized Gaussian (dashed) and the Gaussian computed from the highly accurate but expensive Monte-Carlo estimate (solid).

The extended Kalman filter is computationally quite efficient (Table 2). It is polynomial in measurement dimensionality k and state dimensionality $n : O(k^{2.4}+n^2)$ Thrun et al. [2006]. The input to the extended Kalman filter is the belief at time $t-1$ represented by X_{t-1} and Σ_{t-1}. In step 1, the predicted state $\overline{X}_t$ is computed using the state transition function $a(U_t, X_{t-1})$ and the control input. The uncertainty estimate $\overline{\Sigma}_t$ grows in step 2 by incorporating the state model Jacobian A_t, the state model covariance Q_t, and the uncertainty from the previous time step Σ_{t-1}. The Kalman gain is computed in step 3 by leveraging the predicted covariance $\overline{\Sigma}_t$, the measurement transition Jacobian B_t which contain the derivatives of the measurements with respect to the state variables, and the measurement covariance matrix R_t. The Kalman gain specifies the degree that the measurement update Z_t is incorporated into the new state estimate X_t. The output is the belief at time t, represented by X_t and Σ_t, which are computed in steps 4 and 5 where the Kalman gain is incorporated. The measurement update corrects the predicted state $\overline{X}_t$ and shrinks the uncertainty. The filter represents the belief at time t by the state X_t and the covariance Σ_t.

Figure 8 illustrates the parameter estimate update process for an extended Kalman filter and a direct model. The estimated state is $X_t = [x_s, y_s]^T$ where x_s and y_s represent the estimated parameters of interest at time t.

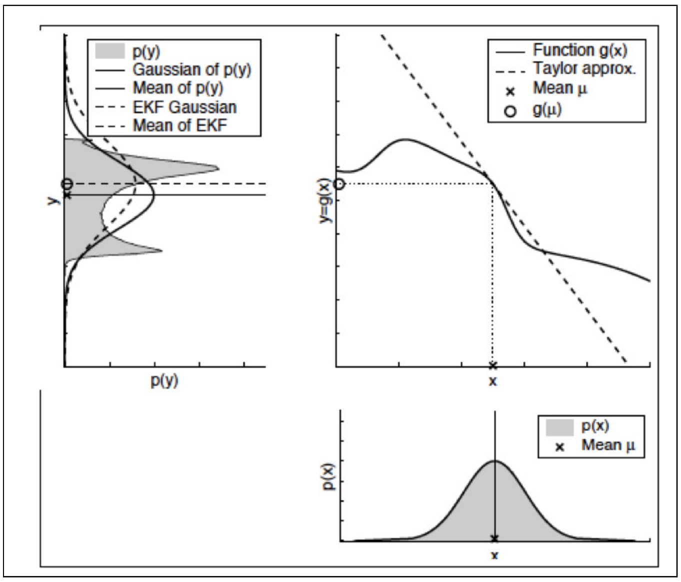


Figure 7: Illustration of linearization applied by the extended Kalman filter Thrun et al. [2006].

Table 2: Extended Kalman filter algorithm.

Step	Operation
1	$\overline{X}_t = a(U_t, X_{t-1})$
2	$\overline{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + Q_t$
3	$K_t = \overline{\Sigma}_t B_t^T (B_t \overline{\Sigma}_t B_t^T + R_t)^{-1}$
4	$X_t = \overline{X}_t + K_t(Z_t - b(\overline{X}_t))$
5	$\Sigma_t = (I - K_t B_t)\overline{\Sigma}_t$
6	Return to step 1 for next time step

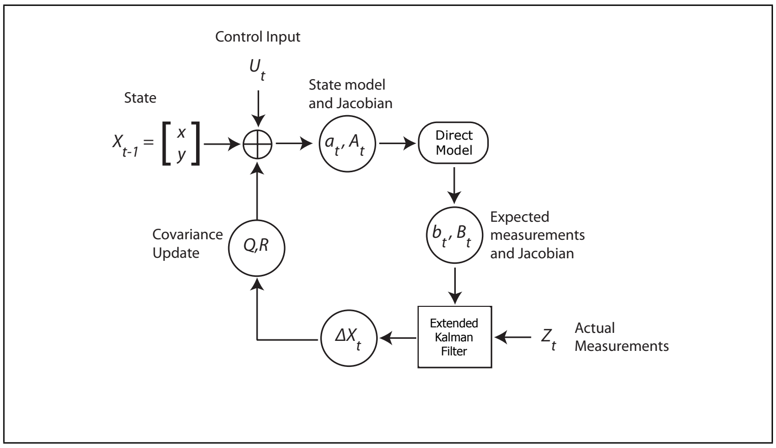


Figure 8: Parameter estimate update process using a the extended Kalman filter and a direct model.

Table 3: Adaptive extended Kalman filter algorithm.

Step	Operation
1	$\overline{X}_t = a(U_t, X_{t-1})$
2	$\overline{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + Q_t$
3	$K_t = \overline{\Sigma}_t B_t^T (B_t \overline{\Sigma}_t B_t^T + R_t)^{-1}$
4	$X_t = \overline{X}_t + K_t(Z_t - b(\overline{X}_t))$
5	$\Sigma_t = (I - K_t B_t)\overline{\Sigma}_t$
6	$Q_{t+1} = \begin{cases} Q_t * M_t & \text{if } \lvert\Delta\Sigma_t\rvert > \Sigma_{tolerance} \text{ and } \Delta X_t < \Delta X_{limit} \\ Q_t / M_t & \text{if } \lvert\Delta\Sigma_t\rvert > \Sigma_{tolerance} \text{ and } \Delta X_t \geq \Delta X_{limit} \\ Q_t & \text{if } \lvert\Delta\Sigma_t\rvert \leq \Sigma_{tolerance} \end{cases}$
7	Return to Step 1 for next time step

3.3 Adaptive extended Kalman filter

Examining the extended Kalman filter algorithm (Table 2), if the measurement covariance is known with a small uncertainty, the state model covariance is unknown, and the state model covariance and measurement covariance are correlated, changes can be made during each iteration to the state model covariance to improve convergence. A value is needed from the extended Kalman filter to drive changes to the state model covariance. Two possible sources exist in the EFK algorithm: the Kalman gain (K_t) and the state covariance (Σ_t). The Kalman gain specifies the degree that the measurement update Z_t is incorporated into the new state estimate X_t while the state covariance Σ_t represents the uncertainty in the new state estimate X_t.

An adaptive extended Kalman filter was developed Myers et al. [2012a] and is presented in Table 3 and illustrated in Figure 9. Step 6 is the only change from the extended Kalman filter. The state model covariance matrix Q is modified at the end of each iteration based on the state covariance Σ and the rate of change in the estimated state ΔX_t. Three conditions are possible when modifying Q. First, if the covariance is increasing at a rate greater than a predefined tolerance value and if the estimated state is changing less than a predefined limit, Q is multiplied by an predefined adaptive gain M. This adaptive gain will have a value greater than 1, which, in this first condition, has the effect of increasing the magnitude of Q and increasing the rate of convergence. From the analysis above, an increasing state covariance Σ indicates the solution is not converged and if the change in the estimate state is below a threshold, convergence time can be reduced by increasing the magnitude of Q. Second, if the covariance is increasing at a rate greater than a predefined tolerance value and if the estimated state is changing more than a predefined limit, Q is divided by the predefined gain M. In this second condition, increasing the magnitude of Q might cause erratic convergence or a failure to reach a solution. Thus, by reducing the magnitude of Q, convergence is dampened. Third, if the covariance is increasing at a rate less than a predefined tolerance value, no change is needed to Q.

3.4 Extended Information Filter

The extended information filter is also called the information form of the Kalman filter. The key difference between the extended Kalman filter and the extended information filter is the way the Gaussian belief is represented. In the extended Kalman filter, the Gaussians are represented by their mean and covariance moments. In the extended information filter, the Gaussians are represented by their canonical parameterization where

$$\Omega = \Sigma^{-1} \qquad \phi = \Sigma^{-1}\mu \tag{5}$$

Ω is called the information matrix and ϕ is the information vector. The mean and covariance of the Gaussian can be obtained from the canonical parameterization by

$$\Sigma = \Omega^{-1} \qquad X = \Omega^{-1}\phi \tag{6}$$

The algorithm is presented in Table 4. U_t, $a(U_t, X_{t-1})$, A_t, $b(\overline{X}_t)$, B_t, Q_t, and R_t are identical to those in the extended Kalman filter. The prediction is implemented in steps 1 through 3 while the correction is implemented in steps 4 through 6.

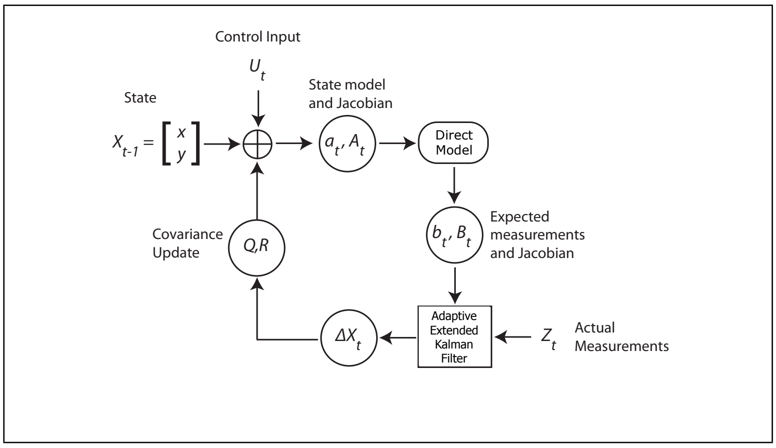


Figure 9: Parameter estimate update process for the adaptive extended Kalman filter and a direct model.

Table 4: Extended information filter algorithm.

Step	Operation
1	$X_{t-1} = \Omega_{t-1}^{-1}\phi_{t-1}$
2	$\overline{\Omega}_t = (A\Omega_{t-1}^{-1}A^T + Q)^{-1}$
3	$\overline{\phi}_t = \overline{\Omega}_t a(U_t, X_{t-1})$
4	$\overline{X}_t = a(U_t, X_{t-1})$
5	$\Omega_t = \overline{\Omega}_t + B^T R^{-1} B$
6	$\phi_t = \overline{\phi}_t + BR^{-1}[Z_t - b(\overline{X}_t) + B\overline{X}_t]$
7	Return to step 1 for next time step

The extended information filter is polynomial in measurement dimensionality k and state dimensionality $n : O(k^2 + n^{2.4})$. Comparison with the extended Kalman filter reveals the duality the these two filters. The measurement update is the difficult step in the extended Kalman filters because it requires a matrix inversion of a large matrix for every iteration. The extended information filter possesses an advantage of allowing R^{-1} to be computed once and reused for all iterations.

Figure 10 illustrates the parameter estimate update process for an extended information filter and a direct model. The estimated state is $X_t = [x_s, y_s]^T$ where x_s and y_s represent the estimated parameters of interest at time t.

3.5 Particle Filter

The particle filter is an alternative nonparametric implementation of the Bayes filter and is a Monte Carlo technique used for the solution of state estimation problems. The main idea is to represent the required posterior density function by a set of random samples with associated weights and to compute the estimates based on these samples and weights Vianna et al. [2010]. Figure 11 illustrates how particles are used to represent the belief. The lower right graph in Figure 11 shows samples drawn from a Gaussian random variable, x. The samples are passed through the nonlinear function $y = g(x)$ shown in the upper right graph. The resulting samples are distributed according to the random variable y which is the actual posterior density function. Figure 11 can be compared to the illustration of extended Kalman filter in Figure 7 where the posterior density function is approximated with a Gaussian. Because it is nonparametric, the particle filter can represent a much broader space of distributions than Gaussians and has the ability to model nonlinear transformations of random

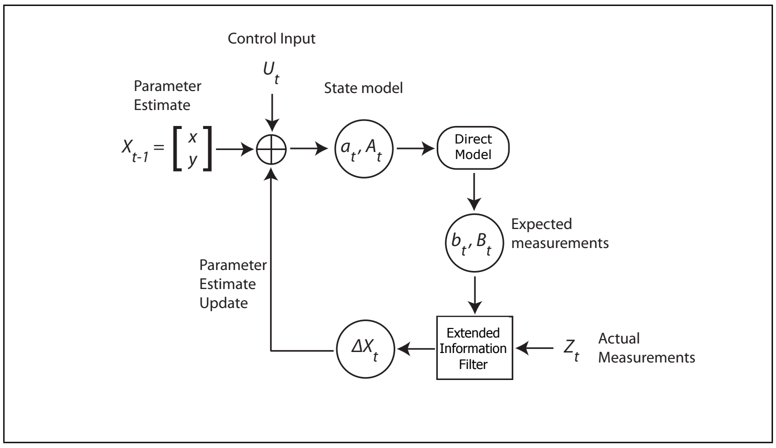


Figure 10: Parameter estimate update process using the extended information filter and a direct model.

Table 5: Particle filter algorithm.

Step	Operation
1	Generate m random possible parameters of interest (particles)
2	Obtain expected measurements for each particle i for the current time t ($Z_{i,t}$)
3	Obtain actual measurements for the current time ($Ztrue_t$)
4	Weight each particle i according to $N(Ztrue_t, I)$
5	Normalize weights for each particle i into bins from 0 to 1
6	Resample best particles using normal distribution
7	Add position noise to each particle
8	Return to Step 2 for next time step

variables Thrun et al. [2006]. The particle filter algorithm to locate the source can be found in Table 5.

Figure 12 illustrates the parameter estimate update process for a particle filter and a direct model. The estimated state is $X_t = [x_s, y_s]^T$ where x_s and y_s represent the estimated parameters of interest at time t.

3.6 Least Squares

Ordinary least squares is applied to approximate solutions of overdetermined systems, i.e. systems of equations in which there are more equations than unknowns. Ordinary least squares is often applied in statistical contexts, particularly regression analysis. Ordinary least squares may be interpreted as a method of fitting data. The best fit, between modeled data and observed data, in its least-squares sense, is an instance of the model for which the sum of squared residuals has its least value, where a residual is the difference between an observed value and the value provided by the model. The method was first described by Carl Friedrich Gauss around 1794 Bretscher [1995]. Ordinary least squares corresponds to the maximum likelihood criterion if the experimental errors have a normal distribution and can also be derived as a method of moments estimator Woodbury [2003a].

The ordinary least squares method is sometimes called the ÃGauss method of minimizationÃ Woodbury [2003a]. For a given state X, the value of T at $X = X + \Delta X$ is obtained through the truncated Taylor's series as

$$T|_{X+\Delta X} \approx T|_X + \left.\frac{\partial T}{\partial X}\right|_X \Delta X. \tag{7}$$

The ordinary least squares objective function is

$$S = (Y - T|_X - B\Delta X)^T (Y - T|_X - B\Delta X). \tag{8}$$

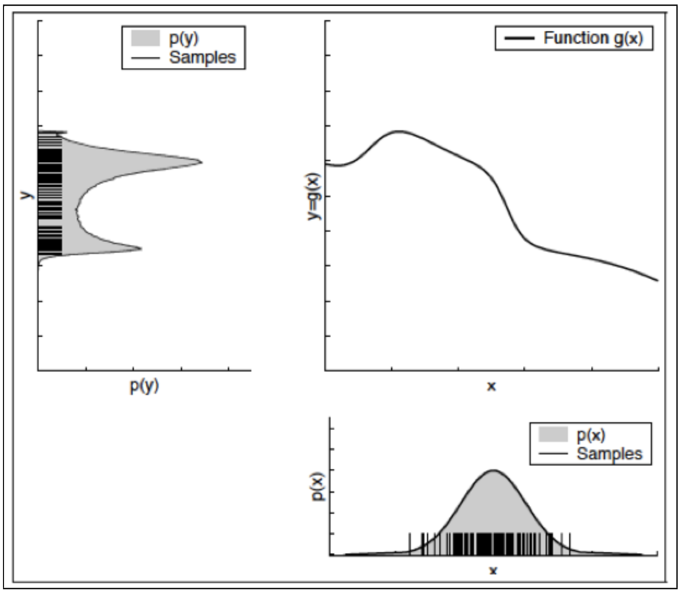


Figure 11: Illustration of how the particle filter represents the posterior density function Thrun et al. [2006].

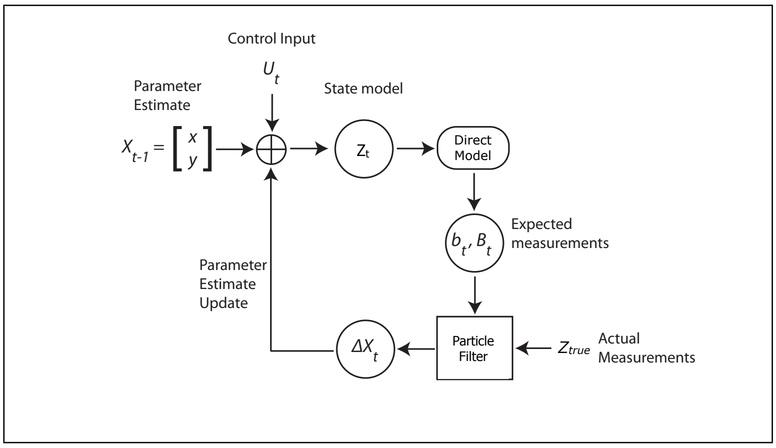


Figure 12: Parameter estimate update process using the particle filter and a direct model.

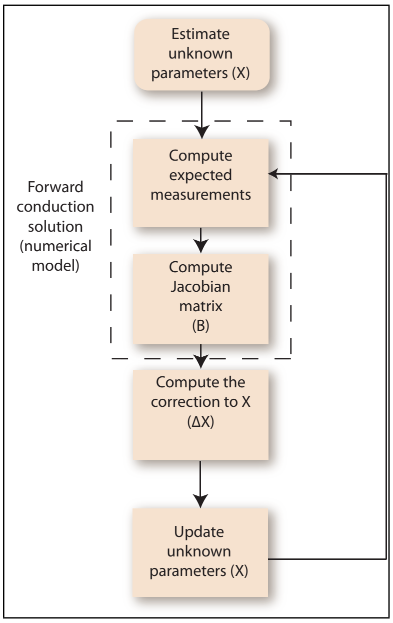

Estimate unknown parameters (X)
Forward conduction solution (numerical model)
Compute expected measurements
Compute Jacobian matrix (B)
Compute the correction to X (ΔX)
Update unknown parameters (X)

Figure 13: Ordinary least squares algorithm.

The minimizer of equation 8 is found by forcing to zero the derivative with respect to ΔX resulting in the estimator

$$\Delta X = (B^T B)^{-1} B^T (Y - T|_X). \tag{9}$$

The sensitivity matrix B can be normalized to produce a better conditioned matrix for the inversion in equation 9. As illustrated in Figure 13, X is updated using ΔX and a new ΔX is computed. Once each component in ΔX is small, the solution is converged yielding our estimate for the unknown parameters.

Figure 13 illustrates the parameter estimate update process for a least squares and a direct model. The estimated state is $X_t = [x_s, y_s]^T$ where x_s and y_s represent the estimated parameters of interest at time t.

4 Measurement Models

Different measurement methods and sensors are available when estimating a quantity of interest such as movement, speed, acceleration, vibration, temperature, heat flux, ultrasonic signal propagation time, and x-ray profile. This section details heating source experiments to demonstrate and develop measurement models for use with inverse methods in detection, localization, and parameter estimation.

Consider a $61cm$ x $30.5cm$ x $0.635cm$ stainless steel 316L plate (Figure 14) with constant properties (Table 6). Four K-type thermocouples are attached on one side and four on the other. With plate center being the origin and the x-axis being the length (Figure 14), thermocouples were attached at (x, y) locations of ($1cm$, $1cm$), ($2cm$, $2cm$), ($3cm$, $3cm$), and ($-1cm$, $-1cm$) on the heated side ($z = 0$) and on the non-heated side (z=0.635cm). The desire is to have thermocouple pairs in exactly the same position on either side of the plate allowing measurement of the temperature difference between the two sides. The thermocouples are secured to the plate with thermal grease and Kapton tape to ensure good thermal contact. Flat black paint is applied to a $1.5cm$ diameter area at the plate center to maximize energy absorption from the heater. The plate is oriented vertically with the positive y-axis pointing up. A Research, Inc. SpotIR® 4150 heater with focusing cone is positioned approximately $2mm$ from the plate surface such that its beam strikes the plate center. Experiments are conducted with the heater running at full-power which, according to manufacturer's specifications, produces $1.7MW/m^2$ of heat flux on the plate in a circular area $0.635cm$ in diameter. Consequently, approximately $54Watts$ of energy are being absorbed by the plate when the heater is on.

During the experiment, the heater is turned on at $t = 300s$ and turned off and removed at $t = 600s$. Data acquisition equipment is used to record thermocouple temperature readings every second during the experiment.

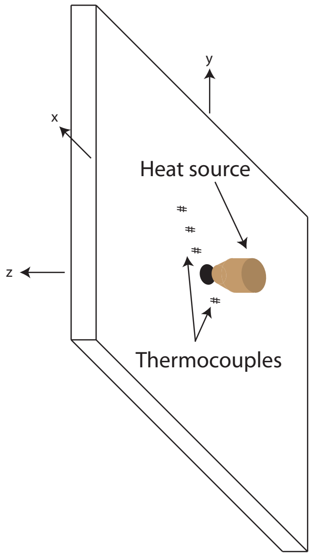


Figure 14: Illustration of flat plate with heat source and sensors (not drawn to scale).

Table 6: Material properties for the stainless steel 316L test sample used in the conduction experiments.

Property	Value
density (ρ)	$8,000 kg/m^3$
thermal conductivity (k)	$14.6 W/m\,K$
specific heat (c_p)	$500 J/kg\,K$
sound speed (v_0)	$5,100 m/s$ @ $293\,K$
ultrasonic TOF temperature factor (ξ)	$110{-}61/\,K$
sample length	61cm
sample width	30.5cm
sample height	0.635cm

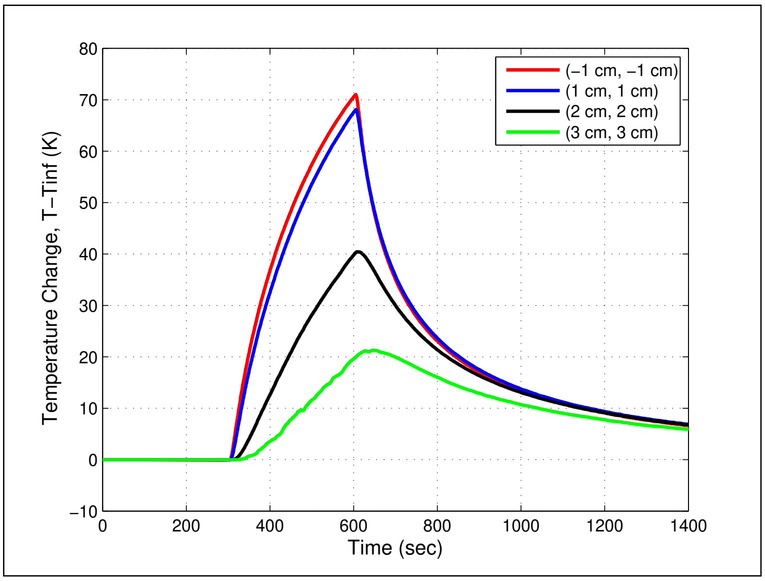


Figure 15: Temperature response on non-heated side of the plate at four sensor locations.

A MIKRON Thermo Scan TS7302 infrared camera is used to collect thermal images of the plate and heater. Coupled with a laptop computer, this system records thermal images every five seconds during the experiment. Benefits gained from the thermal images include visualization of the temperature distribution throughout the experiment and the need to model secondary convection and radiation heating in addition to modeling the primary high heat flux coming from the heater's beam. Figure 15 illustrates the thermocouple temperature data recorded during the experiment. Analysis of the data indicates that a spatial temperature gradient of $6°C/mm$ exists during heating in the area of the thermocouple sets closest to the source $[(1cm,\ 1cm)$ and $(-1cm,\ -1cm)]$. Positioning the heating source and the sensors within this degree of precision proved difficult. Therefore, sensor and heating source placement error is the most likely cause of the discrepancies between the two thermocouple sets closest to the source.

The forward conduction solution leverages COMSOL Multiphysics® by the COMSOL Group and MATLAB® by The Mathworks, Inc. The COMSOL® model uses a finite element mesh with smaller elements near the heat source and larger elements near the plate edges to conserve computing resources.

For the flat plate detailed above, the governing equation for the subdomain (conduction in the plate) is

$$\rho C_p \frac{\partial T}{\partial t} - \nabla \cdot (k \nabla T) = Q \tag{10}$$

where ∇ is the Laplacian and Q is an internal heat source (0 in this case). For the flat plate, k is assumed constant. Thus, the subdomain governing equation is

$$\nabla^2 T = \frac{\rho C_p}{k} \frac{\partial T}{\partial t} \tag{11}$$

or

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\partial^2 T}{\partial z^2} = \frac{\rho C_p}{k} \frac{\partial T}{\partial t} \tag{12}$$

The boundary condition is

$$n \cdot (k \nabla T) = q_0 + h(T_{inf} - T) \tag{13}$$

where n is the surface normal vector, q_0 is the inward heat flux. Radiation effects are assumed negligible for the plate.

The first meshing method analyzed in this section is the 3D free mesh using tetrahedral elements. A $0.635cm$ diameter cylindrical subdomain in the plate's center is used to create a boundary for applying the heating source. This technique also creates small elements near the heating source and large elements far

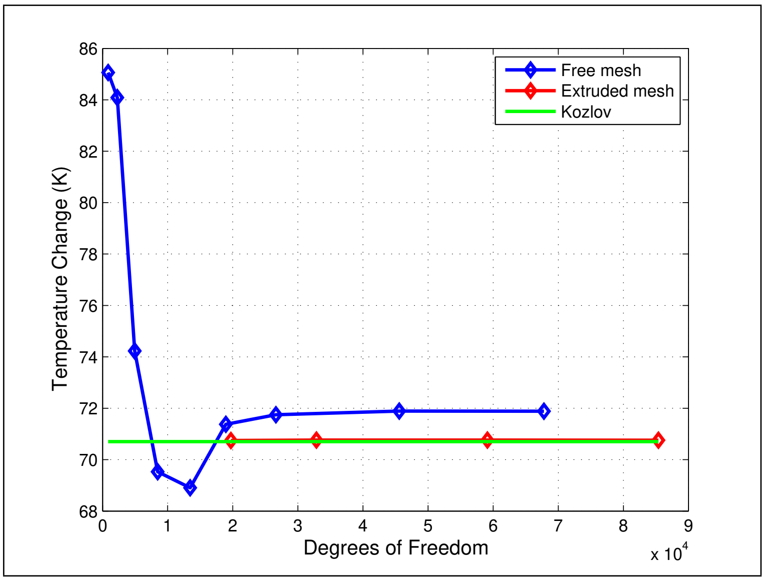


Figure 16: Grid independence results for the heated side at 1.4cm from the source and $t = 400s$.

away from the source where temperature gradients are small thereby conserving computing resources. Mesh refinement is accomplished using all of the predefined free mesh sizes available in the direct model starting with the coarsest mesh and proceeding to the finest mesh. Grid convergence is achieved with 13,256 elements and 26,628 degrees of freedom, however the solution does not agree with an analytical solution of heating through a circular domain without convection Kozlov et al. [1989] as illustrated in Figure 16. Element sizes from the converged 3D free mesh were used to create an extruded mesh which does agree the analytic solution (Figure 16). The extruded mesh is generated by first creating 2D triangle elements in the plate's $x - y$ plane and then extruding the 2D mesh in the z-direction to create prism elements. Two subdomains consisting of a $0.635cm$ diameter circle with a maximum element size of $13m$ and a $6cm$ diameter circle with a maximum element size of $5{-}3m$ were used. The 2D mesh is created with the predefined normal free mesh setting in the direct model. The mesh extrusion process incorporates an option to create multiple mesh layers, therefore grid independence is contingent upon the number of layers through the thickness of the plate. The worst case is where the highest temperature gradients through the plate's thickness exist which is located at plate center. Figure 17 illustrates the computed temperature profile through the plate at plate center with one, two, four, and six mesh layers. The grid convergence study led to the selection of three mesh layers through the plate's thickness dimension, 9,780 total elements, and 45,983 degrees of freedom. Agreement between the final the direct model solution and the closed-form solution Kozlov et al. [1989] is acceptable with mean absolute error less than $0.5K$.

Even with manufacturer specifications, the heat transfer between the radiative heater and the plate is not known with much certainty. Further complicating matters, the heater's proximity to the plate implies an unknown amount of secondary radiation and convection heating on the plate. The focusing cone reaches temperatures in excess of $200°C$ and the lamp is cooled with forced air that exits the heater through the focusing cone pointed at the plate. Based on the focusing cone temperature and a focusing cone area of $20cm^2$, approximately $2.5W$ of radiation and convection energy are absorbed by the plate outside of area illuminated by the lamp. If we assume the area affected by this secondary heating is a circle with a radius of $9cm$, we can approximate the secondary heating with a Gaussian profile of $q_g'' = 100W/m^2$ and $\sigma_g^2 = 0.0009m^2$. Figure 18 illustrates the boundary conditions used in modeling the plate. For this initial analysis, the main heat flux and convection coefficient are estimated. The convection coefficient being estimated is the average value over the duration of the experiment. While areas of the heated side of the plate may have a different convection coefficient value during heating, once the heater is removed, the average convection coefficient is identical on both sides of the plate. Thus, the heat transfer coefficient h is assumed constant and identical on both sides of the plate. Estimating h using free convection correlations Incropera and DeWitt [2002] produces an expected range of $2W/m^2K \leq h \leq 5W/m^2K$. Since the plate edges do not contribute significantly to the thermal load,

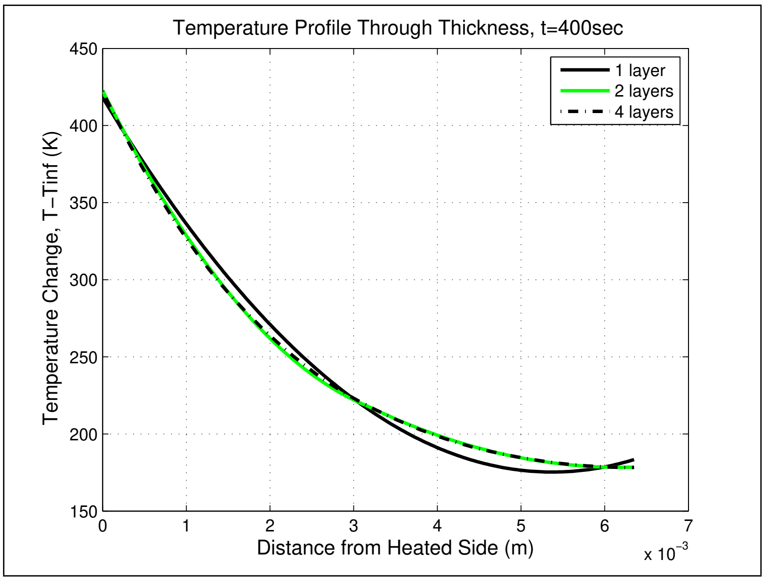


Figure 17: Number of mesh layers for best accuracy.

$h = 3W/m^2K$ is assumed on all four plate edges.

Three inverse methods are compared to quantify the heat flux (q'') and convection coefficient (h) on the plate: least squares, extended Kalman filter, and extended information filter. For the inversion, the entire experiment is treated as one event and temperature measurements are combined together. The experiment covers 1,400 seconds and data are recorded at one second intervals. Not all of the data is needed for the inverse and longer time steps can be used during periods of little thermal activity. Accordingly, one measurement at $t = 0s$, one measurement per second from $t = 290$ to $800s$ (the heater is on from $t = 300$ to $600s$), and one measurement per five seconds from $t = 805$ to $1,400s$ is used. The 5,056 temperature measurements therefore are effectively 5,056 separate sensors. All three methods start with an initial guess of the state $x_0 = [q''\ h]^{\mathrm{T}} = [1.7MW/m^2\ 5.0W/m^2K]^{\mathrm{T}}$ and are processed recursively to convergence.

For least squares, the estimated temperatures T for each sensor location and for each time t, a $5,056 \times 1$ matrix, depend on a vector of two unknowns in the state X and the value of T at $X = X + \Delta X$ is obtained through the truncated Taylor's series as Woodbury [2003b]

$$T|_{X+\Delta X} \approx T|_X + \left.\frac{\partial T}{\partial X}\right|_X \Delta X. \tag{14}$$

The gradient coefficient in equation 14 is the $5,056 \times 2$ sensitivity matrix

$$B = \frac{\partial T}{\partial X} = \begin{bmatrix} \frac{\partial T_1}{\partial q''}q'' & \frac{\partial T_1}{\partial h}h \\ \frac{\partial T_2}{\partial q''}q'' & \frac{\partial T_2}{\partial h}h \\ \vdots & \vdots \\ \frac{\partial T_{5,056}}{\partial q''}q'' & \frac{\partial T_{5,056}}{\partial h}h \end{bmatrix} \tag{15}$$

which has been normalized to have units of temperature producing a better conditioned matrix for the inversion in the least squares estimator $\Delta X = (B^TB)^{-1}B^T(Y - T|_X)$ Woodbury [2003b].

The direct numerical model produces values of T based on current estimates for the state X. The sensitivity matrix B is obtained using finite differences (a $5,056 \times 2$ matrix) by independently varying the state parameters 0.1%. We designate our experimentally obtained temperature measurements as Y, a $5,056 \times 1$ matrix. Our goal is to improve the estimate for the state X based on the observations Y.

The algorithm for the extended Kalman filter is listed in Table 2 where $\overline{X}_t$ is the predicted state, $a(U_t, X_{t-1})$ is the state model based on the input U_t and the previous state X_{t-1}, A is the state model Jacobian, $\overline{\Sigma}$ is the uncertainty estimate, Q_t is the state model covariance, K_t is the Kalman gain, B_t is the measurement Jacobian,

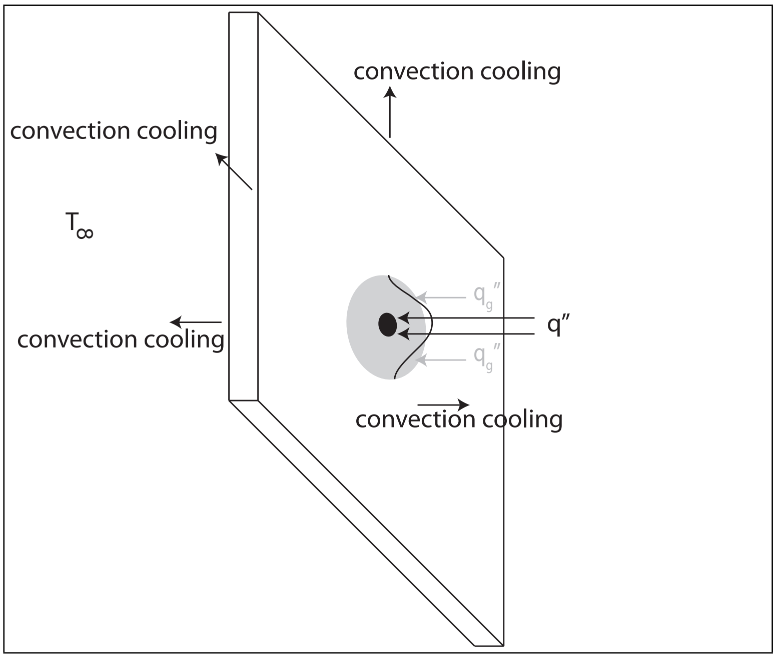


Figure 18: Illustration of boundary conditions on the flat plate.

R_t is the measurement covariance, $b(\overline{X}_t)$ is the measurement transition function and represents the predicted measurements from the forward conduction solution based on the predicted state, and Z_t represents the actual measurements. The filter represents the belief at time t by the state X_t and the covariance Σ_t. For the flat plate considered here, there is no input to the state thus the state model is $a = I_2$ and the state model Jacobian is $A = I_2$, where I_2 is a 2×2 identity matrix. The measurement transition function b is a $5,056 \times 1$ matrix of the predicted temperatures from the forward conduction solution, and the measurement Jacobian B is obtained using finite differences (a $5,056 \times 2$ matrix) by independently varying the state parameters 0.1%. The state model covariance matrix Q is a 2×2 diagonal matrix using $\sigma_q^2 = 0.1MW^2/m^4$ and $\sigma_h^2 = 0.1W^2/m^4K^2$. These values were chosen through a parameter sweep to achieve smooth convergence behavior since small values for the state model covariance matrix cause the Gaussian filters to diverge while arbitrarily large values for the state model covariance matrix render the Gaussian filters essentially identical to the least squares method. The thermocouples have a measurement accuracy of $\pm 1.5°C$, which translates to a measurement variance of $\sigma_T^2 = 0.25°C^2$. This value is used for the diagonal elements of the measurement covariance matrix R, a $5,056 \times 5,056$ matrix. The filter is initialized with the initial state x_0 (stated above) and covariance $\Sigma_0 = 0$. For the extended information filter (Table 4), a, A, b, B, R, and Q are identical to those in the extended Kalman filter. The extended information filter possesses an advantage of allowing the inverse of the measurement covariance matrix Q^{-1} to be computed once and reused for all iterations. Because the initial state covariance matrix Σ_0 is inverted in the extended information filter, the filter is initialized with $\Sigma_0 = R$ instead of the zero matrix used to initialize the extended Kalman filter.

Figures 19 and 20 illustrate the convergence behavior for all three methods. The extended Kalman filter and extended information filter converge identically and are presented together. The Gaussian filters converge a bit slower than the least squares method, however the convergence is smoother. Once convergence is achieved, statistical moments are computed from the last three iterations (Figure 21). Results are similar for all three methods. The least squares method uses the least amount of wall time and memory of the three methods. Wall time for each iteration, independent of recomputing the direct model for the updated parameters, is approximately two orders of magnitude longer for extended information filter and four orders of magnitude longer for extended Kalman filter than the wall time for least squares. Memory usage is approximately 1.5 times more for extended information filter and 2 times more for extended Kalman filter than the memory required by least squares. When considering convergence behavior and computational cost, least squares outperforms the other methods for this type of parameter estimation.

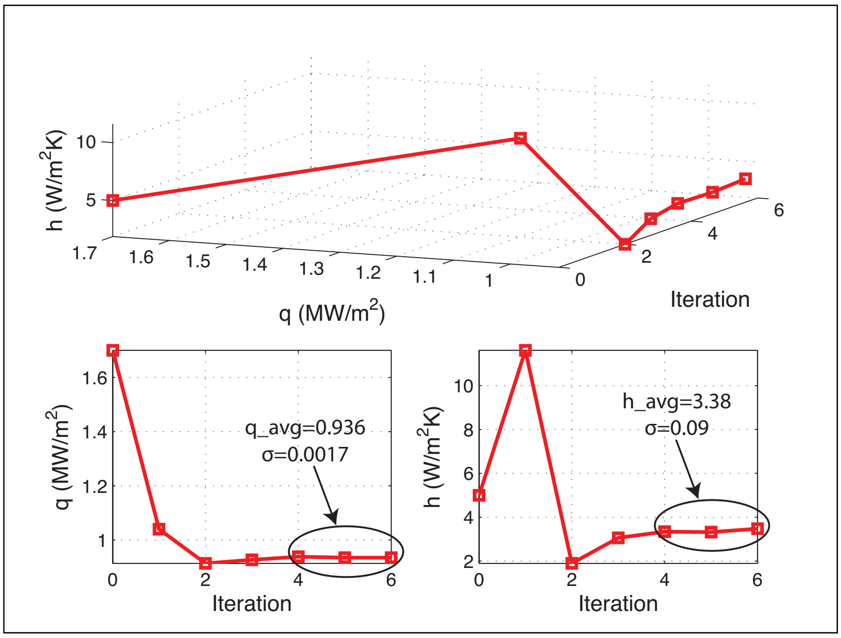


Figure 19: Least squares convergence.

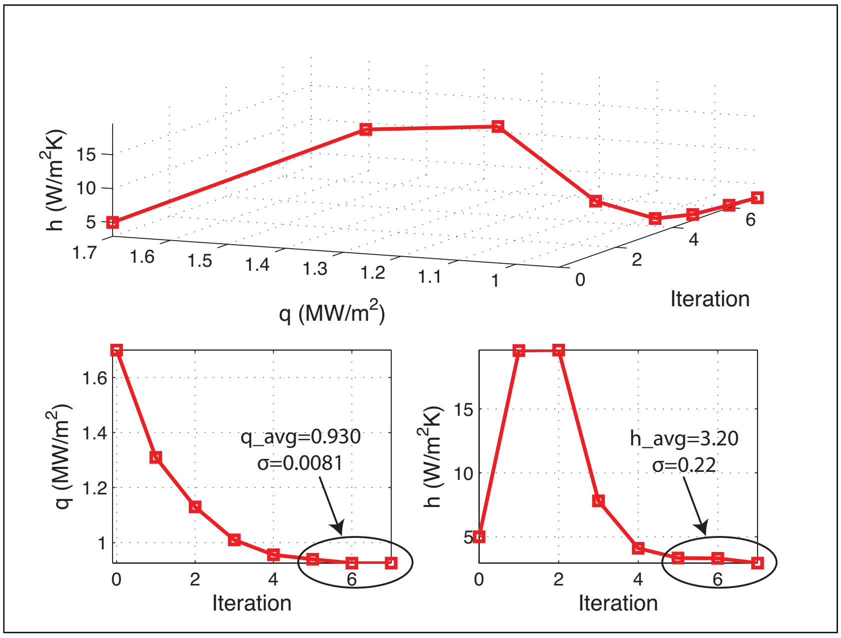


Figure 20: Extended Kalman filter and extended information filter convergence. The filters produce identical results and are presented together.

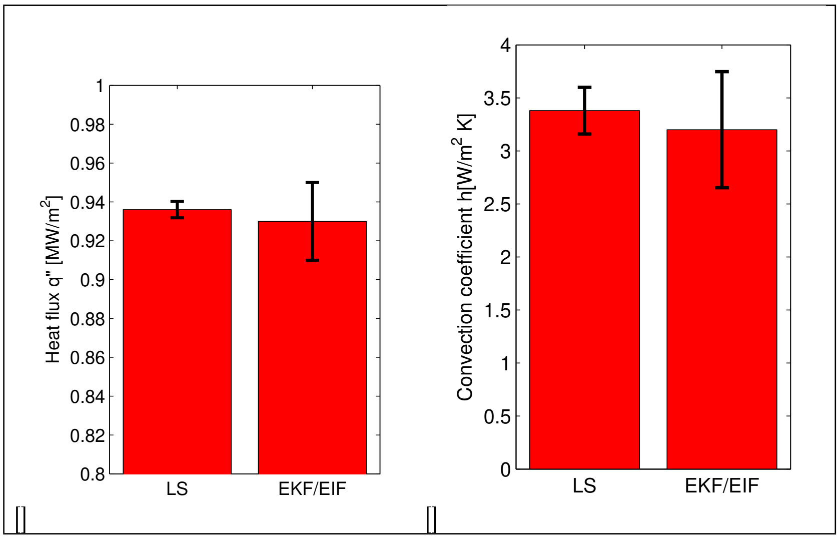


Figure 21: Statistical moments from parameter identification for (a) heat flux q'' and (b) convection coefficient h comparing least squares, extended Kalman filter, and extended information filter.

Figure 22 compares the temperature response measured during the experiment with the temperature response of the model using the results of the estimation (i.e., $q'' = 0.930MW/m^2$ and $h = 3.20W/m^2\,K$). The residuals Beck and Woodbury [1998], Dowding and Blackwell [2001] are illustrated in Figure 23. Agreement between the model and the experiment is acceptable, however improvement could be achieved through modifications to the heating profile (e.g., secondary heating). Agreement with the experiment is better when simultaneously estimating q'' and h than when estimating q'' with h arbitrarily fixed.

A check of the boundary effect errors is conducted to ensure the plate is sized sufficiently large (Figure 24). Of particular interest is in the region of (±4cm,±4cm) where the errors remain well below 0.5% for the entire experiment. Even at (±10cm,±10cm), the errors are below 1% for much of the experiment and stay below 3% for the entire experiment.

4.1 Measurement model selection

One important question in this analysis is determining if an ultrasound-based solution can outperform a thermocouple solution. This section seeks to answer this question by examining six measurement models, two incorporating thermocouples and four incorporating ultrasonic transducers. The following six measurement models have been identified for analysis and will be detailed in the following sections:

1. Temperature measurement model
2. Radius from temperature measurement model
3. Ultrasonic pulse-echo time of flight measurement model
4. Radius from ultrasonic pulse-echo time of flight measurement model
5. Ultrasonic pulse one-way time of flight measurement model
6. Ellipse from ultrasonic one-way pulse time of flight measurement model

These measurement models represent different ways to collect measurements (sensors) and different ways to process the data. Comparison of the six measurement models is performed using the extended Kalman filter (algorithm in Table 2) to locate the source (x_q, y_q). For all six measurement models, the estimated state is $X_t = [x_s, y_s]^T$ where x_s and y_s represent the estimated location of the source at time t. There is no input to the

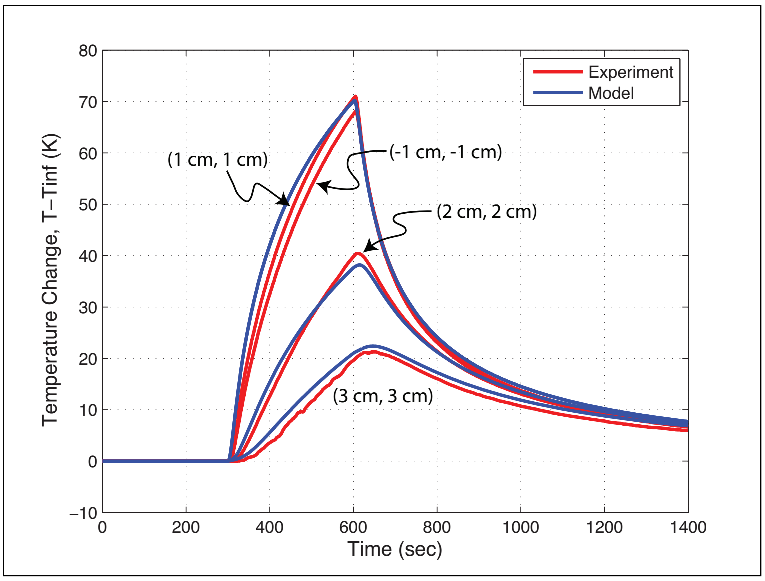


Figure 22: Comparison of the temperature response on non-heated side of the plate at four sensor locations.

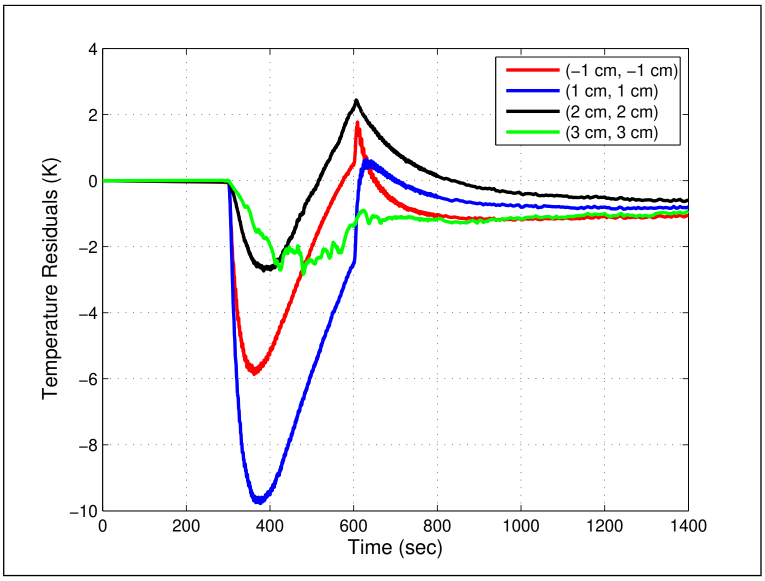


Figure 23: Residuals of the model when compared to the experiment measurements on the non-heated side.

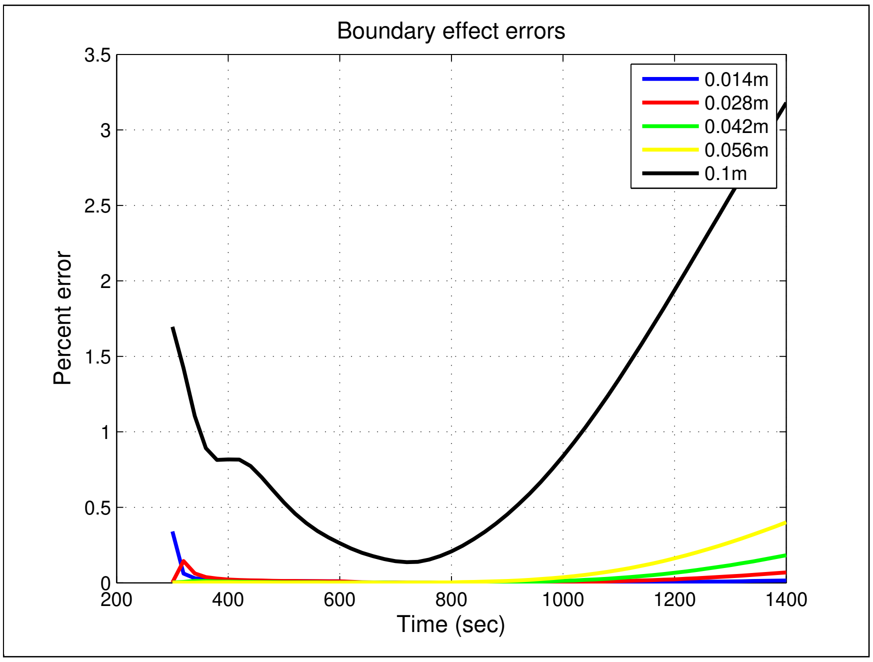


Figure 24: Illustration of when and where the plate edges introduce errors in the temperature distribution.

state thus the state model is $a = I_2$ and the state model Jacobian is $A = I_2$. Sensitivity of the state variance is compared for values from $\sigma^2 = 0.01m^2$ to $0.000001m^2$ with the lower values providing a damping effect. A state variance of $\sigma^2 = 0.0001m^2$ provides smooth, fast convergence without producing erratic convergence behavior exhibited by the higher state variance values and will be used for all measurement model comparisons in this section. Thus, the state model covariance matrix is $Q_t = 0.0001m^2 * I_2$, where I_2 is a 2×2 identity matrix.

Locating and characterizing a heating source depends upon many factors such as heating source movements in time, heating source magnitude changes in time, and other transient behaviors. Fairly restrictive assumptions can be imposed that simplify the problem. Analysis and algorithm development can proceed using these restrictive assumptions and then assumptions can be relaxed in stages to achieve the end result of source localization and characterization. The assumptions for this section are:

1. Source in fixed position (location unknown)
2. Source applied at time $t = 300s$ and removed at $t = 600s$
3. $q'' = 0.930MW/m^2$ over $0.00635m$ diameter circular area while source applied (value obtained in parameter estimation above)
4. Secondary heating is characterized by a Gaussian with magnitude $q''_g = 100W/m^2$ and variance $\sigma_g^2 = 0.0009m^2$ while source applied
5. Convection coefficient $h = 3.20W/m^2K$ on both sides of the plate (value obtained in parameter estimation above)
6. Convection coefficient $h = 3W/m^2K$ on the plate edges
7. Thermal conductivity $k = 15W/mK$
8. Specific heat $C_p = 500J/kgK$ and density $\rho = 8,000kg/m^3$
9. Positions of sensors are $(\pm 4cm, \pm 4cm)$ on the non-heated side

4.2 Temperature measurement model

In this direct model, temperatures are measured using four thermocouples on the non-heated side of the plate. Expected temperatures and the partial derivatives are obtained directly from the direct model to form the measurement transition function $b(\overline{X}_t)$ and the Jacobian B_t.

$$b(\overline{X}_t) = \begin{bmatrix} \bar{\theta}_1 \\ \bar{\theta}_2 \\ \bar{\theta}_3 \\ \bar{\theta}_4 \end{bmatrix} \tag{16}$$

$$B_t = \begin{bmatrix} -\frac{\partial\bar{\theta}_1}{\partial x_1} & -\frac{\partial\bar{\theta}_1}{\partial y_1} \\ -\frac{\partial\bar{\theta}_2}{\partial x_2} & -\frac{\partial\bar{\theta}_2}{\partial y_2} \\ -\frac{\partial\bar{\theta}_3}{\partial x_3} & -\frac{\partial\bar{\theta}_3}{\partial y_3} \\ -\frac{\partial\bar{\theta}_4}{\partial x_4} & -\frac{\partial\bar{\theta}_4}{\partial y_4} \end{bmatrix} \tag{17}$$

where t is time in seconds with a time step of $1s$, $\bar{\theta}$ is the expected change in temperature relative to a reference, obtained from the direct model, if the heating source is located at (x_s, y_s), and (x_i, y_i) with $i = 1, 2, 3, 4$ indicating the locations of the four thermocouples. The Jacobian B_t is constructed using the derivatives with respect to sensor position for convenience since this information can be obtained with one direct model simulation. The derivatives are obtained directly from the direct model. Based on the flat plate experiment above, sensor noise is assumed be $\pm 0.045K$ and is normally distributed ($\sigma^2 = (0.045/3)^2 = 2.225{-}4K^2$). The measurement covariance matrix is $R = 2.225{-}4K^2 * I_4$.

4.3 Radius from temperature measurement model

This indirect model is similar to the previous model in that temperatures are measured using thermocouples, but in this model, the direct model is used as a lookup table to convert measured temperatures to a radius from each sensor to the source. Knowledge of the heating start time, one of the assumptions in this section, enables a simple direct model lookup of expected temperatures for a range of radius values from the heating source. Linear interpolation is used with this lookup table to obtain an expected radius for each temperature measurement.

$$r_i = \sqrt{(x_i - x_s)^2 + (y_i - y_s)^2} \tag{18}$$

where (x_i, y_i) is the location of sensor i for $i = 1, 2, 3, 4$ and (x_s, y_s) the heating source location. The Jacobian is based solely on geometry, which may reduce errors.

$$\begin{aligned} \frac{\partial r_i}{\partial x_s} =& \frac{1}{2}\left((x_i - x_s)^2 + (y_i - y_s)^2\right)^{-\frac{1}{2}} \\ & \left(\frac{\partial}{\partial x_s}(x_i^2 - 2x_i x_s + x_s^2)\right) \end{aligned} \tag{19}$$

$$\frac{\partial r_i}{\partial x_s} = \frac{x_s - x_i}{r_i}, \frac{\partial r_i}{\partial y_s} = \frac{y_s - y_i}{r_i} \tag{20}$$

The measurement transition function $b(\overline{X}_t)$ and the Jacobian B_t are then

$$b(\overline{X}_t) = \begin{bmatrix} \sqrt{(x_1 - x_s)^2 + (y_1 - y_s)^2} \\ \sqrt{(x_2 - x_s)^2 + (y_2 - y_s)^2} \\ \sqrt{(x_3 - x_s)^2 + (y_3 - y_s)^2} \\ \sqrt{(x_4 - x_s)^2 + (y_4 - y_s)^2} \end{bmatrix} \tag{21}$$

$$B_t = \begin{bmatrix} \frac{x_s - x_1}{r_1} & \frac{y_s - y_1}{r_1} \\ \frac{x_s - x_2}{r_2} & \frac{y_s - y_2}{r_2} \\ \frac{x_s - x_3}{r_3} & \frac{y_s - y_3}{r_3} \\ \frac{x_s - x_4}{r_4} & \frac{y_s - y_4}{r_4} \end{bmatrix} \tag{22}$$

where t is time in seconds with a time step of $1s$, $\overline{r}_i$ with $i = 1, 2, 3, 4$ is the radius from the sensor to the source, obtained from the direct model, if the source is located at (x_s, y_s), and (x_i, y_i) with $i = 1, 2, 3, 4$ indicating the locations of the four thermocouples. Based on the flat plate experiment above, sensor noise is assumed be $\pm 0.045K$ and is normally distributed ($\sigma^2 = (0.045/3)^2 = 0.000225K^2$). Since measured temperature is being related to radius, sensor noise must be converted into radius noise. The complication in this conversion arises from the fact that radius is a non-linear function of temperature and time. Based on insights gained from the forward conduction model and analysis of the temperature response in the plate, a value of $0.015m/K$ is used resulting in a radius noise of $\pm 0.000675m$ with normal distribution ($\sigma^2 = 5.06{-}8m^2$). The measurement covariance matrix, therefore, is $R = 5.06{-}8m^2 * I_4$.

4.4 Ultrasonic pulse-echo time of flight measurement model

This direct model uses ultrasonic pulses to measure the average temperature through the material thickness at each sensor location. In the pulse-echo method, the ultrasonic pulse travels through the material thickness, reflects off the boundary, and returns to the transducer. The time of flight is Myers et al. [2008, in review, 2010]

$$G_{ii} = \frac{2L}{v_o}\left(1 + \xi \theta_{avg}\big|_0^L\right) \tag{23}$$

where L represents the material thickness, v_0 is the speed of sound in the material at a reference temperature, ξ is the ultrasonic time of flight factor which is material dependent, and θ is the change in temperature from the reference temperature. The ultrasonic pulse time of flight measurement model consists of obtaining expected temperatures from the direct model, computing the average temperature between the transducer and the boundary, and then computing an expected time of flight using equation 23 to form the measurement transition function $b(\overline{X}_t)$ (equation 24). The Jacobian partial derivatives are obtained using time of flight difference when moving the source in the x and y directions independently (equation 25).

$$b(\overline{X}_t) = \begin{bmatrix} \overline{G}_1 \\ \overline{G}_2 \\ \overline{G}_3 \\ \overline{G}_4 \end{bmatrix} \tag{24}$$

$$B_t = \begin{bmatrix} -\frac{\partial \overline{G}_1}{\partial x_1} & -\frac{\partial \overline{G}_1}{\partial y_1} \\ -\frac{\partial \overline{G}_2}{\partial x_2} & -\frac{\partial \overline{G}_2}{\partial y_2} \\ -\frac{\partial \overline{G}_3}{\partial x_3} & -\frac{\partial \overline{G}_3}{\partial y_3} \\ -\frac{\partial \overline{G}_4}{\partial x_4} & -\frac{\partial \overline{G}_4}{\partial y_4} \end{bmatrix} \tag{25}$$

where t is time in seconds with a time step of 1 second, $\overline{G}_i$ with $i = 1, 2, 3, 4$ is the expected ultrasonic pulse time of flight, obtained from the direct model, with the heating source at location (x_s, y_s), and (x_i, y_i) with $i = 1, 2, 3, 4$ indicating the locations of the four transducers. The Jacobian B_t is constructed using the derivatives with respect to sensor position for convenience since this information can be obtained with one direct model

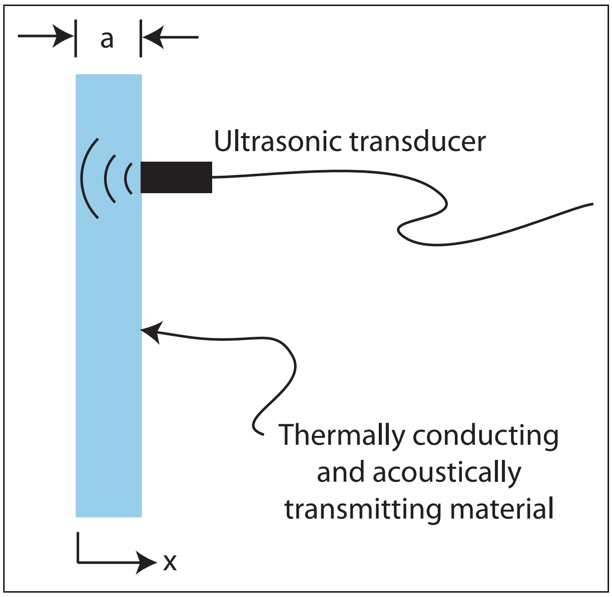


Figure 25: Ultrasonic pulse-echo technique.

simulation. The derivatives are obtained from the direct model using finite differences by independently varying the x and y positions of all sensors by $0.0001m$. Based on the flat plate experiment above, sensor noise is assumed be $\pm 2.3{-}10s$ and is normally distributed ($\sigma^2 = 5.88{-}21sec^2$). The measurement covariance matrix, therefore, is $R = 5.88{-}21sec^2 * I_4$.

4.5 Radius from ultrasonic pulse-echo time of flight measurement model

In this indirect model, ultrasonic pulse-echo time of flight is measured using four transducers on the non-heated side of the plate. Similar to radius from temperature method, this method converts the measured time of flight to a radius using the direct model as a lookup table. Knowledge of the heating start time, one of the assumptions in this section, enables a simple direct model lookup of expected temperatures for a range of radius values from the heating source. Temperatures in the plate are related to time of flight through equation 23. Linear interpolation is used with this lookup table to obtain an expected radius for each time of flight measurement. Equations 18 to 20 develop the geometry behind the measurement transition function $b(\overline{X}_t)$ and the Jacobian B_t which are

$$b(\overline{X}_t) = \begin{bmatrix} \sqrt{(x_1 - x_s)^2 + (y_1 - y_s)^2} \\ \sqrt{(x_2 - x_s)^2 + (y_2 - y_s)^2} \\ \sqrt{(x_3 - x_s)^2 + (y_3 - y_s)^2} \\ \sqrt{(x_4 - x_s)^2 + (y_4 - y_s)^2} \end{bmatrix} \tag{26}$$

$$B_t = \begin{bmatrix} \frac{x_s - x_1}{r_1} & \frac{y_s - y_1}{r_1} \\ \frac{x_s - x_2}{r_2} & \frac{y_s - y_2}{r_2} \\ \frac{x_s - x_3}{r_3} & \frac{y_s - y_3}{r_3} \\ \frac{x_s - x_4}{r_4} & \frac{y_s - y_4}{r_4} \end{bmatrix} \tag{27}$$

where t is time in seconds with a time step of 1 second, $\overline{r}_i$ with $i = 1, 2, 3, 4$ is the radius from the sensor to the source, obtained from the direct model, if the source is located at (x_s, y_s), and (x_i, y_i) with $i = 1, 2, 3, 4$ indicating the locations of the four thermocouples. Based on the flat plate experiment above, sensor noise is assumed be $\pm 2.3{-}10s$ and is normally distributed ($\sigma^2 = 5.88{-}21sec^2$). Sensor noise in terms of temperature

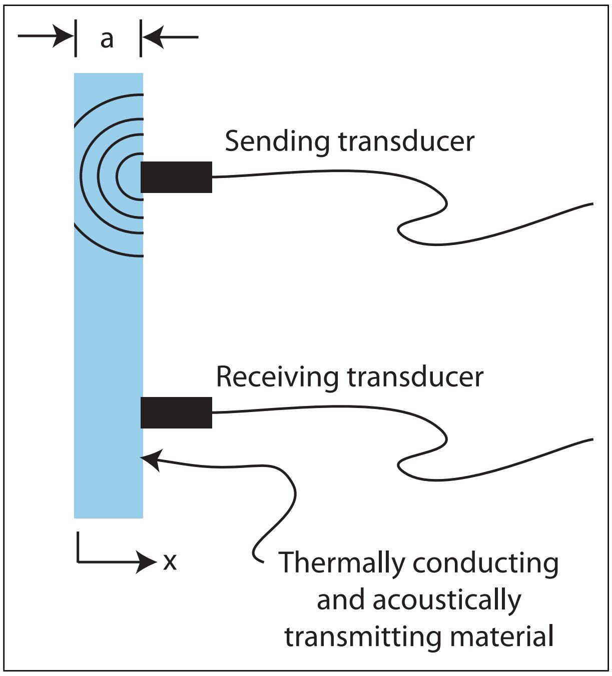


Figure 26: One-way ultrasonic pulse technique.

can be expressed as

$$\theta_{noise} = \frac{G_{noise}v_0}{2L\xi} = 0.84K \tag{28}$$

Since measured time of flight is being related to radius, sensor noise must be converted into radius noise. The complication in this conversion arises from the fact that radius is a non-linear function of time of flight and time. Based on insights gained from the forward conduction model and analysis of the temperature response in the plate, a value of $0.015m/K$ is used resulting in a radius noise of $\pm 0.0126m$ with normal distribution ($\sigma^2 = 1.76{-}5m^2$). The measurement covariance matrix, therefore, is $R_t = 1.76{-}5m^2 * I_4$.

4.6 Ultrasonic pulse one-way time of flight measurement model

Instead of sending an ultrasonic pulse through to a boundary and receiving the echo at the original transducer, one transducer can transmit the pulse and another transducer can receive the pulse. The time of flight is

$$G_{ij} = \frac{R_{ij}}{v_o}\left(1 + \xi\theta_{avg}|_i^j\right) \tag{29}$$

where R_{ij} is the distance between transducers (m). This direct measurement model consists of obtaining expected temperatures from the direct model, computing the average temperature between the transducers, and then computing an expected time of flight to form $a(U_t, X_{t-1})$ (equation 30). For the current analysis, the average temperature is based on the line on the plate surface between the two sensors. The Jacobian partial derivatives are obtained using time of flight difference when moving the source in the x and y directions

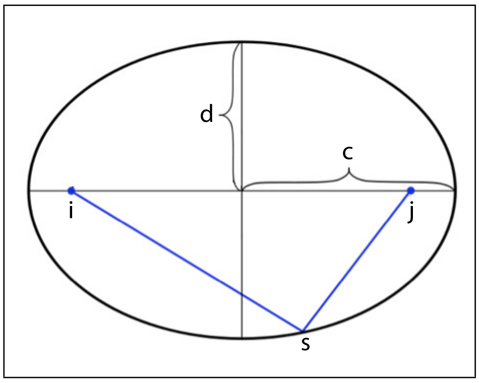


Figure 27: Ellipse properties.

independently (equation 31).

$$b(\overline{X}_t) = \begin{bmatrix} \overline{G}_1 \\ \overline{G}_2 \\ \overline{G}_3 \\ \overline{G}_4 \end{bmatrix} \tag{30}$$

$$B_t = \begin{bmatrix} -\frac{\partial \overline{G}_1}{\partial x_1} & -\frac{\partial \overline{G}_1}{\partial y_1} \\ -\frac{\partial \overline{G}_2}{\partial x_2} & -\frac{\partial \overline{G}_2}{\partial y_2} \\ -\frac{\partial \overline{G}_3}{\partial x_3} & -\frac{\partial \overline{G}_3}{\partial y_3} \\ -\frac{\partial \overline{G}_4}{\partial x_4} & -\frac{\partial \overline{G}_4}{\partial y_4} \end{bmatrix} \tag{31}$$

where t is time in seconds with a time step of $1s$, $\overline{G}_i$ with $i = 1, 2, 3, 4$ is the ultrasonic pulse time of flight, obtained from the direct model, with the heating source located at (x_s, y_s), and (x_i, y_i) with $i = 1, 2, 3, 4$ indicating the locations of the four transducers. The Jacobian B_t is constructed using the derivatives with respect to sensor position for convenience since this information can be obtained with one direct model simulation. The derivatives are obtained from the direct model using finite differences by independently varying the x and y positions of all sensors by $0.0001m$. Based on the flat plate experiment above, sensor noise is assumed be $\pm 1.05{-}8s$ and is normally distributed ($\sigma^2 = ((1.05{-}8)/3)^2 = 1.225{-}17sec^2$). The measurement covariance matrix, therefore, is $R = 1.225{-}17sec^2 * I_4$.

4.7 Ellipse from ultrasonic pulse one-way time of flight measurement model

In this indirect model, a particular ultrasonic pulse time of flight at a particular time after the heater is turned on means that the source could be anywhere on an assumed elliptical shape around the sensors. Figure 27 illustrates the geometry of an ellipse. The two sensors are assumed to be the focus points for the ellipse. Since the distance between sensors is known, ellipse parameters c and d can be related to each other and the ellipse can be represented with just one parameter c.

$$r_{is} + r_{js} = 2c = \sqrt{r_{ij}^2 + 4d^2} \tag{32}$$

where i and j are sensors and s is heat source.

$$c = \frac{1}{2}\sqrt{r_{ij}^2 + 4d^2} = \frac{r_{is} + r_{js}}{2} \tag{33}$$

$$r_{is} = \sqrt{(x_i - x_s)^2 + (y_i - y_s)^2} \tag{34}$$

$$r_{js} = \sqrt{(x_j - x_s)^2 + (y_j - y_s)^2} \tag{35}$$

$$\frac{\partial c_i}{\partial x_s} = \frac{1}{2}\left[\frac{x_s - x_i}{r_{is}} + \frac{x_s - x_j}{r_{js}}\right] \tag{36}$$

$$\frac{\partial c_i}{\partial y_s} = \frac{1}{2}\left[\frac{y_s - y_i}{r_{is}} + \frac{y_s - y_j}{r_{js}}\right] \tag{37}$$

The parameter c is measured indirectly by first measuring the one-way ultrasonic pulse time of flight. The forward conduction solution is used to get time of flight for a range of c values and interpolated using the spline method to obtain c for the measured time of flight. The measurement transition function $b(\overline{X}_t)$ and the Jacobian B_t are then

$$b(\overline{X}_t) = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix} \tag{38}$$

$$B_t = \begin{bmatrix} \frac{\partial c_1}{\partial x_s} & \frac{\partial c_1}{\partial y_s} \\ \frac{\partial c_2}{\partial x_s} & \frac{\partial c_2}{\partial y_s} \\ \frac{\partial c_3}{\partial x_s} & \frac{\partial c_3}{\partial y_s} \\ \frac{\partial c_4}{\partial x_s} & \frac{\partial c_4}{\partial y_s} \end{bmatrix} \tag{39}$$

where t is time in seconds with a time step of $1s$, c_i with $i = 1, 2, 3, 4$ is the ellipse parameter if the source is located at (x_s, y_s). Based on the flat plate experiment above, sensor noise is assumed be $\pm 1.05{-}8s$ and is normally distributed ($\sigma^2 = 1.22{-}17s^2$). The sensor noise in terms of temperature can be expressed as

$$\theta_{noise} = \frac{G_{noise} v_0}{L\xi} = 6.09K \tag{40}$$

Since measured time of flight is being related to the ellipse parameter c, sensor noise must be converted into ellipse parameter noise. The complication in this conversion arises from the fact that c is a non-linear function of time of flight and time. Based on insights gained from the forward conduction model and analysis of the temperature response in the plate, a value of $0.015m/K$ is used resulting in an ellipse noise of $\pm 2.044m$ for the c parameter with normal distribution ($\sigma^2 = 4.62{-}9m^2$).

4.8 Extended Kalman filter convergence behavior

Extended Kalman filter convergence behavior for all six measurement models are compared in Figures 28 through 31. With the heating source located inside the sensor grid (Figure 28), all measurement models converge to the correct location, however both temperature measurement models exhibit rather noisy convergence. The ellipse from ultrasonic pulse one-way time of flight measurement model produces the best results with the heating source located inside the sensor grid. With the heating source located at the edge of the sensor grid (Figure 29), all measurement models once again converge to the correct location and both temperature measurement models and the radius from ultrasonic pulse-echo time of flight measurement model exhibit undesirable convergence behavior. The ellipse from ultrasonic pulse one-way time of flight measurement model produces the best results with the heating source located at the edge the sensor grid. With the heating source located outside of the sensor grid (Figure 30), none of the measurement models converge to the correct location, however the ellipse from ultrasonic pulse one-way time of flight and radius from ultrasonic pulse-echo time of flight measurement models converge to within $1cm$ of the actual location. These examples started with an initial guess of ($0cm$, $0cm$) for the heating source location. Figure 31 illustrates the convergence behavior for all six models using an

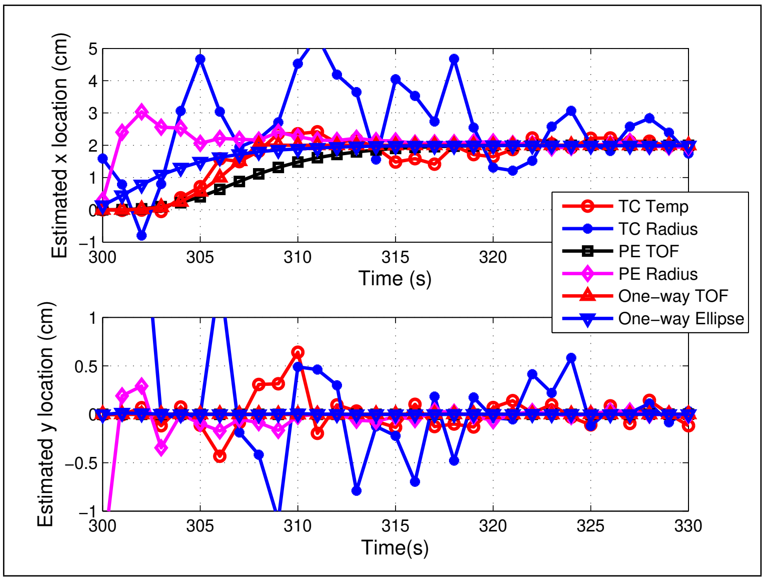


Figure 28: Extended Kalman filter convergence for all six measurement models with source at $(2cm,\ 0cm)$ and initial guess of $(0cm,\ 0cm)$.

initial guess of $(8cm,\ 8cm)$ for the heating source located inside the sensor grid. Interestingly, all direct models fail to converge to the correct location in this scenario. Overall, the ellipse from ultrasonic pulse one-way time of flight measurement model produces the best results when considering accuracy of converged solution, ability to converge to the correct solution given different initial guesses, and smoothness of convergence behavior.

Because we are using numerical tools to solve the governing equations, we lack a set of state equations and cannot determine the observability index in the standard fashion. We can, however, examine sensitivity to heating source location relative to sensor location as well as sensitivity to other parameters including heating source magnitude, plate thermal conductivity, and plate surface convection coefficient. Sensitivity analysis to address observability is included in Section 5.

5 Sensitivity

Sensitivity is important to understand and characterize when measuring and estimating parameters of interest. This section details sensitivity to source position, boundary conditions, and thermal conductivity from the flat plate experiments discussed in the previous section.

5.1 Sensitivity to source position

To determine the one-way pulse method's sensitivity to source location, it is necessary to examine how the average temperature between the two sensors is affected by the relative position of the heat source. Note that this discussion assumes the source is static and therefore not moving with time. Figure 32 illustrates the temperature profile on the plate's non-heated side at $t = 320s$ with the source located at $(x = 0cm,\ y = 0cm)$. The average temperature difference between the two sensors is $13.5K$. If the source is located at $(x = 2cm,\ y = 0cm)$ as in Figure 33, the average temperature difference is nearly identical at $13.2K$. We conclude, then, that even with knowledge that the source is between the sensors, its x-location cannot be determined very accurately. If the source were further to the right, say at $(x = 3cm,\ y = 0cm)$, the average temperature difference would be $11.8K$ indicating that sensitivity to x-location is greater close to the transducers. However, the observations do not reveal if the heat source is closer to the pulse generator or the receiver. If the source is instead offset in the y-direction at $(x = 0cm,\ y = 1cm)$ as in Figure 34, the average temperature difference is $5.3K$. Thus, the sensitivity to source position in the y-direction is greater than for the x-direction for this

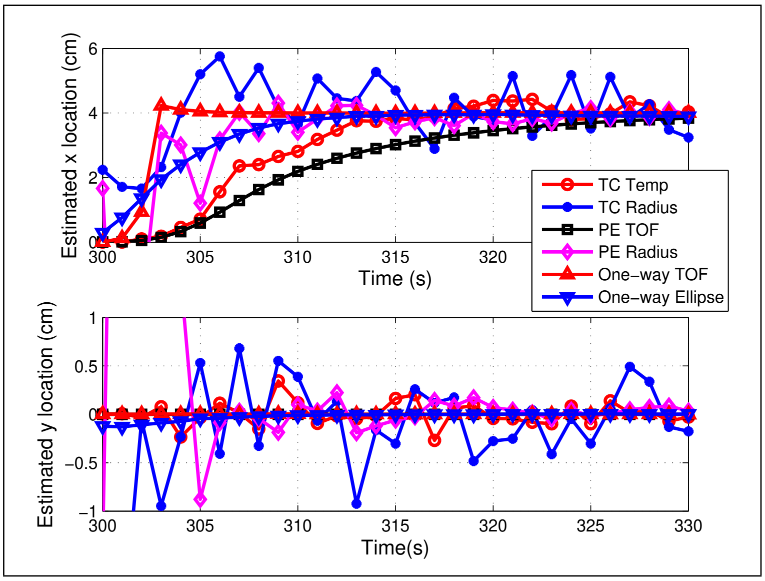


Figure 29: Extended Kalman filter convergence for all six measurement models with source at $(4cm,\ 0cm)$ and initial guess of $(0cm,\ 0cm)$.

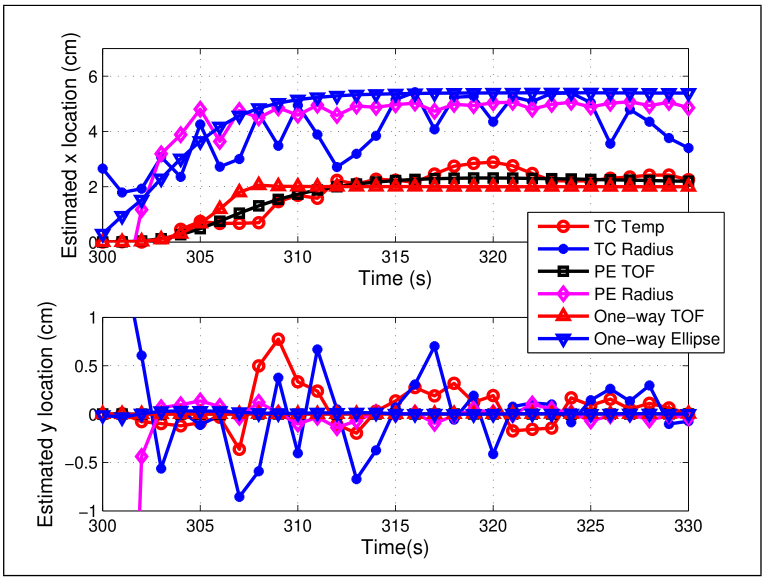


Figure 30: Extended Kalman filter convergence for all six measurement models with source at $(6cm,\ 0cm)$ and initial guess of $(0cm,\ 0cm)$.

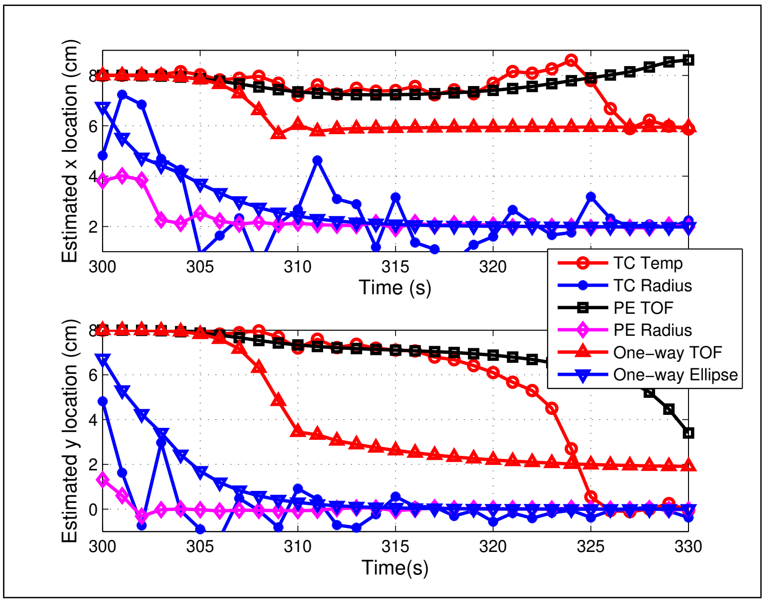


Figure 31: Extended Kalman filter convergence for all six measurement models with source at $(2cm,\ 0cm)$ and initial guess of $(8cm,\ 8cm)$.

sensor pair. More generally, the sensitivity is greater in the direction normal to the ultrasonic pulse propagation path.

The sensitivity to heating source location is expressed as

$$S_{xy} = \sqrt{\left(\frac{\partial\theta_{avg}}{\partial x}\right)^2 + \left(\frac{\partial\theta_{avg}}{\partial y}\right)^2}. \tag{41}$$

Figures 35 and 36 illustrate S_{xy} at two different times during the heating. These figures highlight the high sensitivity regions around the sensor path and the drastic drop-off near the path. These figures support the observation above that sensitivity is greater perpendicular to the ultrasonic propagation path. One should notice the rapid decrease in sensitivity as the source location nears the path between sensors.

5.2 Sensitivity to boundary conditions and thermal conductivity

Sensitivity to boundary conditions (Figure 18) and thermal conductivity are analyzed for an ultrasonic sensor configuration of four sensors in an $8cm$ square configuration (Figure 37). This configuration is based on the sensitivity analysis in Section 5 and represents a starting point for analysis. Sensitivity for the primary heat flux (q''), secondary heating magnitude and spread (q''_g and σ_g^2), convection coefficients for the plate (h_{sides} and h_{edges}), and thermal conductivity of the plate (k) are illustrated and analyzed for source locations inside the sensor grid and up to $2cm$ outside the grid (i.e., a $12cm$ by $12cm$ area).

Assumptions for the sensitivity to boundary conditions and thermal conductivity analysis are:

1. Source at known, fixed position
2. Source applied at time $t = 300s$ and removed at $t = 600s$
3. Main heat flux $q'' = 0.930MW/m^2$ over $0.00635m$ diameter circular area during heating (value obtained in previous study Myers et al. [2010a,b, 2012c])
4. Secondary heating is characterized by a Gaussian with magnitude $q''_g = 100W/m^2$ and variance $\sigma_g^2 = 0.0009m^2$ during heating

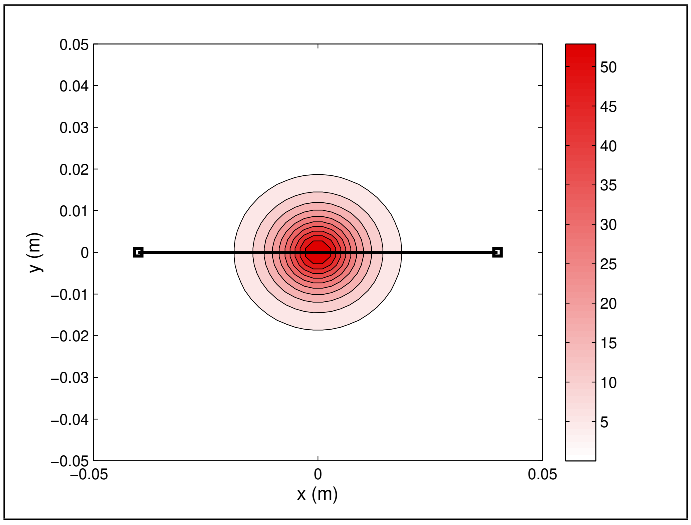


Figure 32: Temperature response at $t = 320s$ with source located at $(x = 0cm,\ y = 0cm)$. $\theta_{avg} = 13.5K$ between sensors.

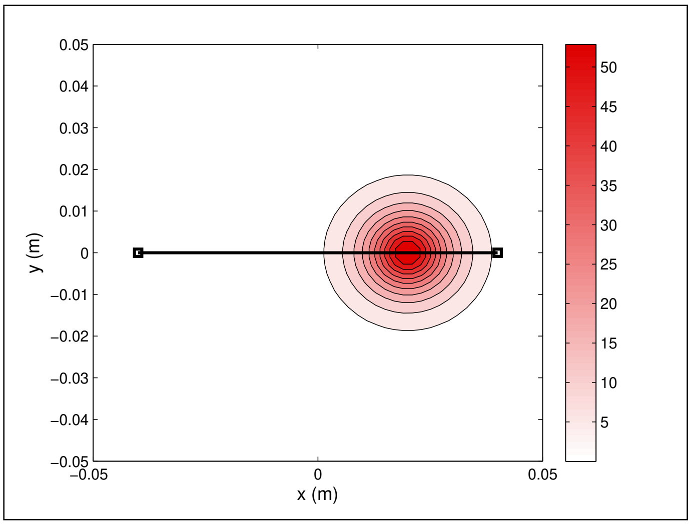


Figure 33: Temperature response at $t = 320s$ with source located at $(x = 2cm,\ y = 0cm)$. $\theta_{avg} = 13.2K$ between sensors.

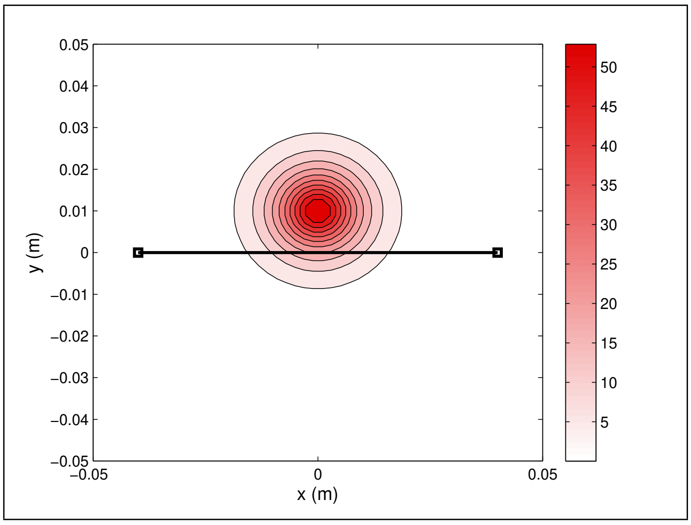


Figure 34: Temperature response at $t = 320s$ with source located at $(x = 0cm, y = 1cm)$. $\theta_{avg} = 5.3K$ between sensors.

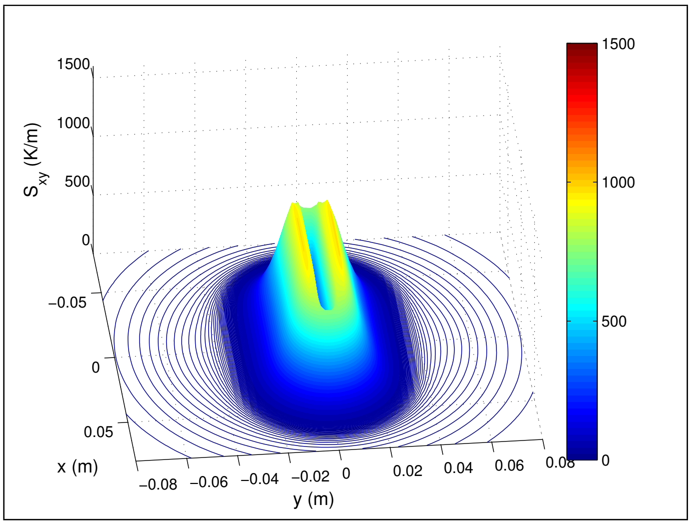


Figure 35: Heating source location sensitivity for one-way pulse sensor configuration and all possible heating source locations at t=320s.

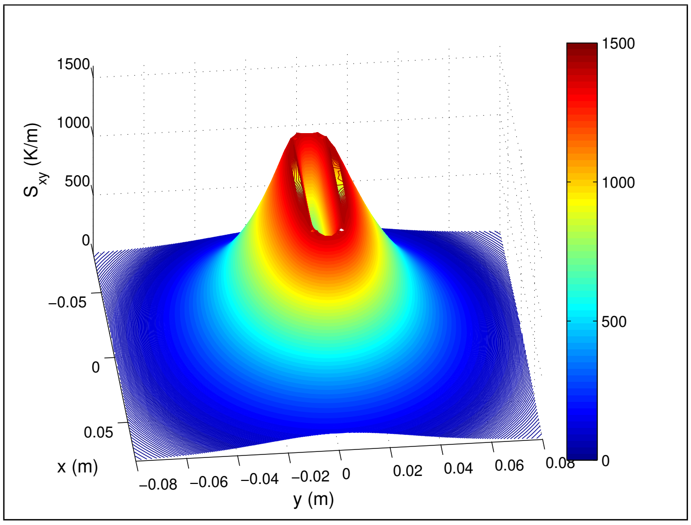


Figure 36: Heating source location sensitivity for one-way pulse sensor configuration and all possible source locations at $t = 450s$.

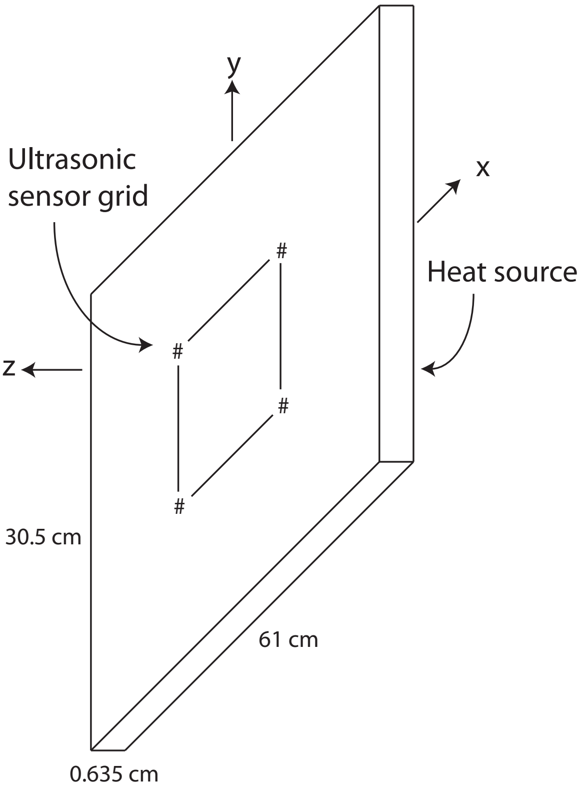


Figure 37: Ultrasonic sensor grid on the non-heated side of the plate with # symbols representing sensors and lines representing the ultrasonic pulse propagation paths between sensors (not drawn to scale).

5. Convection coefficient $h = 3.20W/m^2 - K$ on both sides of the plate (value obtained in previous study Myers et al. [2010a,b, 2012c])

6. Convection coefficient $h = 3W/m^2 - K$ on the plate edges

7. Thermal conductivity $k = 14.6W/m - K$

8. Specific heat $C_p = 500J/kgK$ and density $\rho = 8,000kg/m^3$

9. Positions of sensors are $(\pm 4cm, \pm 4cm)$ on the non-heated side

Sensitivity is computed using finite differences where baseline ultrasonic pulse time of flight values are computed using the assumptions above and then new ultrasonic pulse time of flight values are computed with the parameter being investigated multiplied by $1+\delta$. Sensitivities are presented scaled according to the relationBeck and Arnold [1977]

$$S_{\beta_i} = \beta_i \frac{\partial G}{\partial \beta_i} \tag{42}$$

$$S_{\beta_i} \approx \beta_i \frac{G(x,y,z,t,\beta_1,\cdots,\beta_i(1+\delta),\cdots,\beta_p) - G(x,y,z,t,\beta_1,\cdots,\beta_p)}{\beta_i(1+\delta) - \beta_i} \tag{43}$$

$$S_{\beta_i} \approx \frac{G(x,y,z,t,\beta_1,\cdots,\beta_i(1+\delta),\cdots,\beta_p) - G(x,y,z,t,\beta_1,\cdots,\beta_p)}{\delta} \tag{44}$$

where β_i is the parameter being investigated and G is ultrasonic pulse time of flight. The δ parameter used in this section is $\delta = 0.001$. By normalizing the sensitivities, direct comparison between all investigated parameters can be performed.

6 Comparison

The flat plate experiments detailed above are used in this section for comparing filter performance. This section examines heating source localization using four ultrasonic transducers in an $8cm$ square pattern (Figure 37).

Locating and characterizing a heating source depends upon many factors such as heating source movements in time, heating source magnitude changes in time, and other transient behaviors. Fairly restrictive assumptions can be imposed that simplify the problem. Analysis and algorithm development can proceed using these restrictive assumptions and then assumptions can be relaxed in stages to achieve the end result of source localization and characterization. The assumptions for this work are:

1. Source in fixed position (location unknown)

2. Source applied at time $t = 300s$ and removed at $t = 600s$

3. Main heat flux $q'' = 0.930MW/m^2$ over $0.00635m$ diameter circular area while source applied (value obtained in previous study Myers et al. [2010a,b, 2012c])

4. Secondary heating is characterized by a Gaussian with magnitude $q''_g = 100W/m^2$ and spread $\sigma_g^2 = 0.0009m^2$ while source applied

5. Convection coefficient $h = 3.20W/m^2K$ on both sides of the plate (value obtained in previous study Myers et al. [2010a,b, 2012c])

6. Convection coefficient $h = 3W/m^2K$ on the plate edges

7. Thermal conductivity $k = 14.6W/mK$

8. Specific heat $C_p = 500J/kgK$ and density $\rho = 8,000kg/m^3$

9. Positions of sensors are $(\pm 4cm, \pm 4cm)$ on the non-heated side

The three inverse methods compared are: extended Kalman filter, particle filter, and least squares. The two measurement models studied are: ultrasonic pulse one-way time of flight measurement model, and ellipse from ultrasonic pulse one-way time of flight measurement model. With the particle filter, only the ultrasonic pulse one-way time of flight measurement model is considered because, as will become clear later, the particle filter does not depend upon a Jacobian, and the ellipse model is an alternative way of expressing the Jacobian for those methods that require a Jacobian. Comparison of the five methods is performed in locating the heating source on the plate in the $x - y$ plane (x_q, y_q). For all methods, the state therefore is $X_t = [x_q, y_q]^T$. In all methods considered, the ultrasonic time of flight is normalized by the time of flight before the heating source is applied to the plate (G_{ij}/G_0).

6.1 Extended Kalman filter with ultrasonic pulse one-way time of flight measurement model

The extended Kalman filter algorithm to locate the source can be found in Table 2 . The state is $X_t = [x_q, y_q]^T$, and there is no input (U_t) to the state; thus the extended Kalman filter state model is $a = I_2$ and the state model Jacobian is $A = I_2$, where I_2 is a 2×2 identity matrix. A parameter sweep is conducted for the state model variance from $\sigma^2 = 0.01m^2$ to $\sigma^2 = 0.000001m^2$. A small value for the state model variance $(\sigma^2 = 0.000001m^2)$ produces a damping effect on the convergence whereas a large value $(\sigma^2 = 0.01m^2)$ produces fast but erratic convergence. From this parameter sweep, it is determined that a state variance of $\sigma^2 = 0.0001m^2$ provides a good compromise between damping and stability and this value is used. Thus, the state model covariance matrix is $Q_t = 0.0001m^2 \times I_2$.

This measurement model consists of obtaining expected temperatures from the direct model using the predicted state $\overline{X}_t$, computing the average temperature between the transducers, and then computing an expected time of flight from 29 to form $b(\overline{X}_t)$ (equation 45). For the current analysis, the average temperature is computed along the path on the non-heated plate surface between the two sensors. The Jacobian partial derivatives are obtained using finite difference when moving the source in the x and y directions independently (equation 46).

$$b(\overline{X}_t) = \begin{bmatrix} \overline{G}_1 \\ \overline{G}_2 \\ \overline{G}_3 \\ \overline{G}_4 \end{bmatrix}; \tag{45}$$

$$B_t = \begin{bmatrix} -\frac{\partial \overline{G}_1}{\partial x_1} & -\frac{\partial \overline{G}_1}{\partial y_1} \\ -\frac{\partial \overline{G}_2}{\partial x_2} & -\frac{\partial \overline{G}_2}{\partial y_2} \\ -\frac{\partial \overline{G}_3}{\partial x_3} & -\frac{\partial \overline{G}_3}{\partial y_3} \\ -\frac{\partial \overline{G}_4}{\partial x_4} & -\frac{\partial \overline{G}_4}{\partial y_4} \end{bmatrix}, \tag{46}$$

where t is time in seconds with a time step of $1s$, $\overline{G}_i$ with $i = 1, 2, 3, 4$ is the ultrasonic pulse time of flight with the heating source located at (x_s, y_s), and (x_i, y_i) with $i = 1, 2, 3, 4$ are the locations of four transducers. The Jacobian B_t is constructed using the derivatives with respect to sensor position for convenience because this information can be obtained with one direct model simulation. The derivatives are obtained from the direct model using finite differences by independently varying the x and y positions of all sensors by $0.0001m$. Based on the flat plate experiment above, the sensor noise is assumed be $\pm 6{-}5$ (a non-dimensional number based on G_{ij}/G_0) and is normally distributed $(\sigma^2 = ((6{-}5)/3)^2 = 4{-}10)$. Solution instabilities were present when using this variance, which were reduced by increasing the variance to $4{-}7$. This larger variance effectively dampens the solution and prevents large changes from one iteration to the next. The measurement covariance matrix, therefore, is $R = 4{-}7 \times I_4$.

6.2 Extended Kalman filter with ellipse from ultrasonic pulse one-way time of flight measurement model

In an attempt to simplify the sensitivity calculation, the lines of constant time of flight around the sensor pairs form approximate ellipses. With this approximation, the sensitivities can be calculated algebraically. Figure 27 illustrates the geometry of an ellipse, where the two sensors are assumed to be the foci for the ellipse. Since the distance between sensors is known, ellipse parameters c and d can be related to each other and the ellipse can be represented with just one parameter c.

$$r_{is} + r_{js} = 2c = \sqrt{r_{ij}^2 + 4d^2} \tag{47}$$

where i and j are sensors and s is heat source.

$$c = \frac{1}{2}\sqrt{r_{ij}^2 + 4d^2} = \frac{r_{is} + r_{js}}{2} \tag{48}$$

$$r_{is} = \sqrt{(x_i - x_s)^2 + (y_i - y_s)^2} \tag{49}$$

$$r_{js} = \sqrt{(x_j - x_s)^2 + (y_j - y_s)^2} \tag{50}$$

$$\frac{\partial c_i}{\partial x_s} = \frac{1}{2}\left[\frac{x_s - x_i}{r_{is}} + \frac{x_s - x_j}{r_{js}}\right] \tag{51}$$

$$\frac{\partial c_i}{\partial y_s} = \frac{1}{2}\left[\frac{y_s - y_i}{r_{is}} + \frac{y_s - y_j}{r_{js}}\right] \tag{52}$$

This measurement model consists of measuring the one-way ultrasonic pulse time of flight, using the direct model to obtain the average temperature between the transducers for a range of c values, using equation 29 to compute time of flight for the range of c values, and then interpolating the time of flight results using the spline method to obtain c for the measured time of flight. The measurement transition function $b(\overline{X}_t)$ and the Jacobian B_t are then

$$b(\overline{X}_t) = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{bmatrix}; \tag{53}$$

$$B_t = \begin{bmatrix} \frac{\partial c_1}{\partial x_s} & \frac{\partial c_1}{\partial y_s} \\ \frac{\partial c_2}{\partial x_s} & \frac{\partial c_2}{\partial y_s} \\ \frac{\partial c_3}{\partial x_s} & \frac{\partial c_3}{\partial y_s} \\ \frac{\partial c_4}{\partial x_s} & \frac{\partial c_4}{\partial y_s} \end{bmatrix}, \tag{54}$$

where t is time in seconds with a time step of 1 second, c_i with $i = 1, 2, 3, 4$ is the ellipse parameter if the source is located at (x_s, y_s). Based on the flat plate experiment above, sensor noise is assumed be $\pm 1.05{-}8s$ and is normally distributed $(\sigma^2 = 1.22{-}17sec^2)$. The sensor noise in terms of temperature can be expressed as

$$\theta_{noise} = \frac{G_{noise}v_0}{L\xi} = 6.09K \tag{55}$$

Using the average slope of $0.015m/K$ determined in Section 4 and documented in previous work Myers et al. [2010a,b, 2012c], ellipse noise for the c parameter from ultrasonic pulse time of flight measurement model is $\pm 0.0914m$ and is normally distributed $(\sigma^2 = 9.28{-}4m^2)$. Since the measurement covariance matrix R represents the measurement noise of the c parameter, the measurement covariance is $3.19{-}10m^2$ ($[1.05{-}8s/3 \times 5, 100m/sec$ sound speed$]^2$).

6.3 Particle filter with ultrasonic pulse one-way time of flight measurement model

The particle filter is an alternative nonparametric implementation of the Bayes filter and is a Monte Carlo technique used for the solution of state estimation problems. The main idea is to represent the required posterior density function by a set of random samples with associated weights and to compute the estimates based on these samples and weights Vianna et al. [2010]. Because it is nonparametric, the particle filter can represent a much broader space of distributions than Gaussians and has the ability to model nonlinear transformations of random variables Thrun et al. [2006]. The particle filter algorithm to locate the source can be found in Table 5.

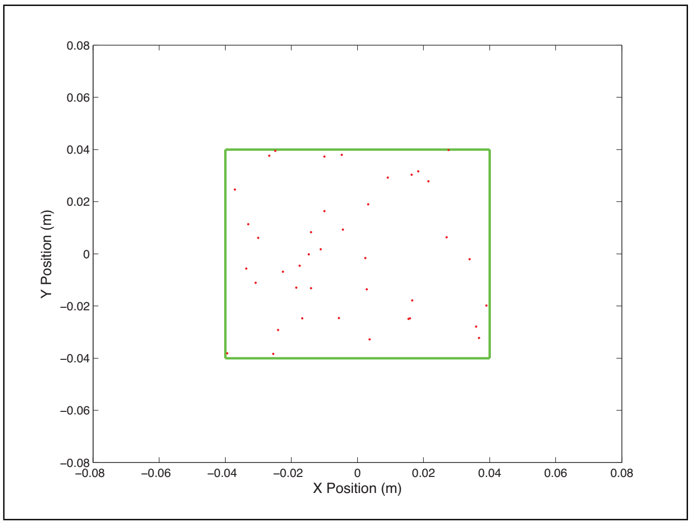


Figure 38: Particle locations ($m = 40$) at $300s$.

Implementation of the particle filter for heating source localization starts with defining the area of possible source locations on the plate. The number of particles m to use in the algorithm must also be defined. A large number of particles yields a higher probability that one or more particles will be located near the actual source but the downside is higher computational cost. For this study, the area is defined as the $8cm \times 8cm$ sensor grid and the number of particles is $m = 40$. The algorithm starts with the generation of m random particles within the defined area of possible source locations (Figure 38). Step 2 involves obtaining expected temperatures from the direct model with the heating source at each particle location, computing the average temperature between the transducers, and then computing an expected time of flight to form $Z_{i,t}$, a 4×1 matrix for each particle i at the current time t. For the current analysis, the average temperature is computed along the path on the non-heated plate surface between the two sensors. Step 3 entails obtaining actual ultrasonic time of flight measurements at the current time t for all four sensor pairs to form $Ztrue_t$, a 4×1 matrix. The particle filter relies on an importance factor, or weight $w_{i,t}$, to incorporate the measurement $Ztrue_t$ into the particle set. The weight $w_{i,t}$ for each particle is computed in Step 4 using

$$\begin{aligned} w_{i,t} &= N(Ztrue_t, I) \\ &= det(2\pi I)^{-\frac{1}{2}} exp\left\{-\frac{1}{2}((Z_{i,t} - Ztrue_t) \times gain)^T I^{-1}((Z_{i,t} - Ztrue_t) \times gain)\right\}. \end{aligned} \tag{56}$$

Gain is discussed in detail later in this section. A number of resampling techniques have been devised Thrun et al. [2006], Vianna et al. [2009], Arulampalam et al. [2002]. This work employs the sampling importance resampling technique because this technique requires fewer particles than some of the other methods and focuses the computational resources to regions in the state space with high posterior probability Thrun et al. [2006]. In Step 5, the particle weights are normalized into bins from 0 to 1, which gives the particles with the highest weight the largest bins and then in Step 6, the particles are resampled using a normal distribution. By resampling across the bins from 0 to 1 using a normal distribution, a higher probability exists that the best particles will be chosen but some particles that are not the best will be chosen too. It is important to note that the number of particles m remains constant through the resampling process, thus some particles will have identical locations on the plate after resampling. Degeneracy is common with particle filters, a situation where the solution converges to the one best particle within the current particle set without considering locations nearby Doucet et al. [2002]. This fact necessitates Step 7 where position noise or roughness is added to each particle Salmond et al. [1993]. In this work, uniform position noise of $\pm 0.5cm$ is used. Figures 39 and 40 illustrate the particle distribution before and after adding noise to each particle location. After completion of Step 7, the algorithm returns to Step 2 and the process is repeated for the next step in time. For this work, a time step of $1s$ is used. Figure 41 illustrates the particle distribution at $t = 330sec$.

Because the magnitude of the non-dimensional values in the matrix $Z_{i,t} - Ztrue_t$ in equation 56 ranges

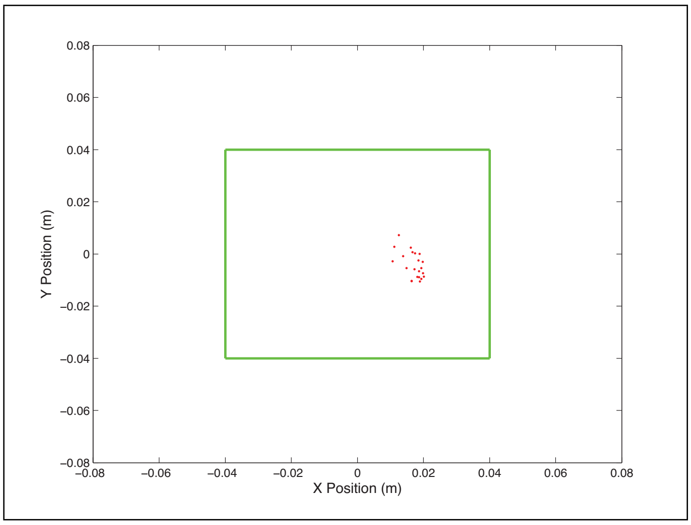


Figure 39: Particle locations $(m = 40)$ at $315s$.

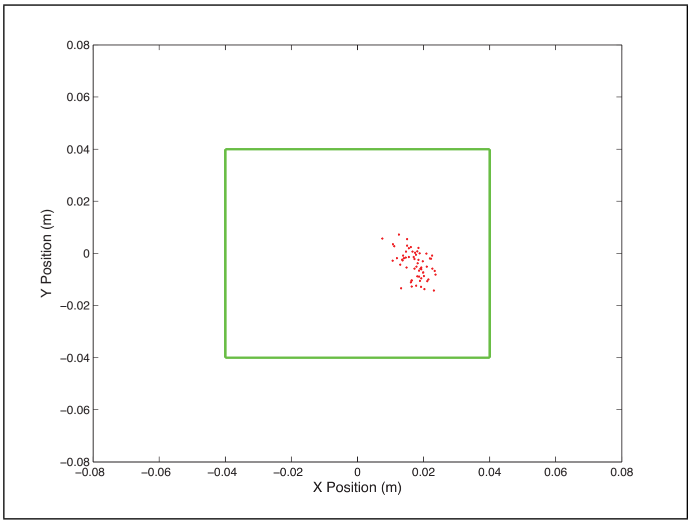


Figure 40: Particle locations $(m = 40)$ at $315s$ after adding noise to each particle location.

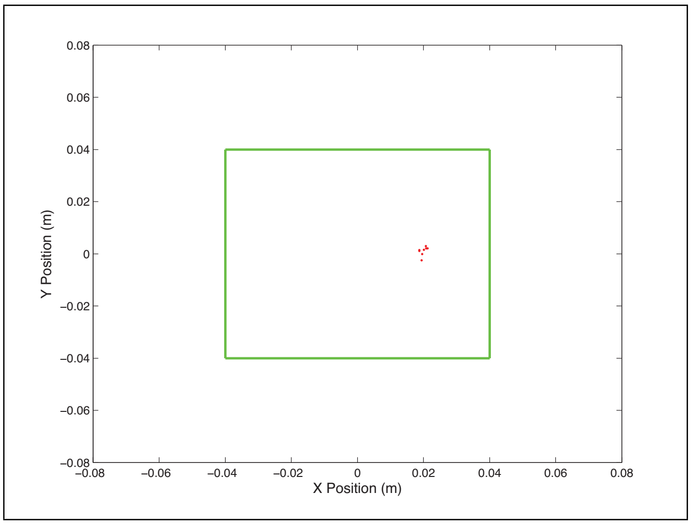


Figure 41: Particle locations $(m = 40)$ at $330s$.

from 0 to 0.008, using no *gain* ($gain = 1$) produces a value of 1 in the exponent portion of the equation for all particles. Thus, identical weights are computed for all particles (Figure 42). For the particle filter to function properly, it is imperative that the particles close to the actual heating source location receive the highest weight ($Z_{i,t} - Ztrue_t$ close to zero in Figure 42) and those particles far from the actual heating source location receive a weight close to zero. The non-dimensional sensor noise measured in the experiment above is approximately 6−5. Therefore, the applied gain should yield the largest weights for $Z_{i,t} - Ztrue_t$ magnitudes between 0 and the noise of 6−5 and should yield weights close to 0 for $Z_{i,t} - Ztrue_t$ magnitudes larger than the sensor noise. Illustrated in Figure 42, a gain of 14 is too small to produce weights close to zero for $Z_{i,t} - Ztrue_t$ magnitudes larger than the sensor noise, but a gain of 64 precipitates the desired effect. Figure 43 illustrates the effect of the number of particles m on the convergence behavior and performance of the particle filter. The filter is quite robust and Figure 43 demonstrates the filter's ability to converge to the correct location with only 10 particles. The particle filter used in the comparisons with the other localization methods in the next section is based on 40 particles.

6.4 Least squares with ultrasonic pulse one-way time of flight measurement model

The ordinary least squares method is sometimes called the ÃGauss method of minimizationÃ Woodbury [2003a]. For the current localization, the estimated time of flight G for each sensor pair for a particular time t, a 4×1 matrix, depends on a vector of two unknowns in the state X and the value of G at $X = X + \Delta X$ is obtained through the truncated Taylor's series as

$$G|_{X+\Delta X} \approx G|_X + \left.\frac{\partial G}{\partial X}\right|_X \Delta X. \tag{57}$$

The derivative in equation 57 is the 4×2 sensitivity matrix

$$B_t = \frac{\partial G}{\partial X} = \begin{bmatrix} \frac{\partial \overline{G}_1}{\partial x_1} & \frac{\partial \overline{G}_1}{\partial y_1} \\ \frac{\partial \overline{G}_2}{\partial x_2} & \frac{\partial \overline{G}_2}{\partial y_2} \\ \frac{\partial \overline{G}_3}{\partial x_3} & \frac{\partial \overline{G}_3}{\partial y_3} \\ \frac{\partial \overline{G}_4}{\partial x_4} & \frac{\partial \overline{G}_4}{\partial y_4} \end{bmatrix} \tag{58}$$

where t is time in seconds with a time step of $1s$, $\overline{G}_i$ with $i = 1, 2, 3, 4$ is the ultrasonic pulse time of flight with the heating source located at (x_s, y_s), and (x_i, y_i) with $i = 1, 2, 3, 4$ are the locations of four transducers. The experimentally obtained time of flight measurements are designated as Z_t, a 4×1 matrix. The desire is

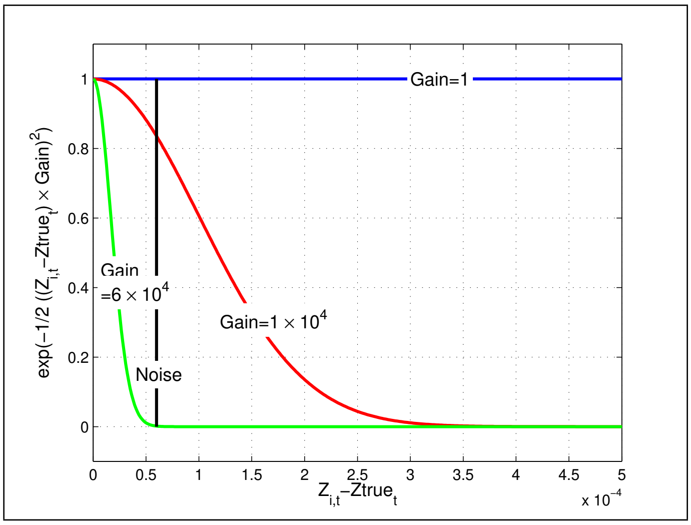


Figure 42: Comparison of selected particle filter gain values with sensor noise.

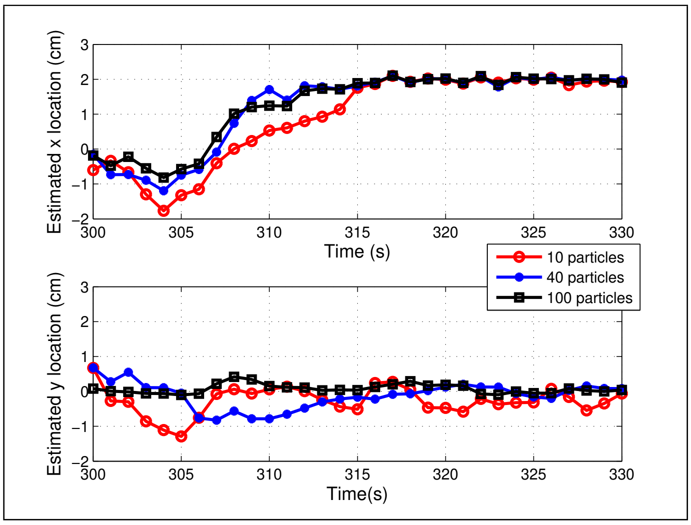


Figure 43: Particle filter convergence for selected numbers of particles with heating source located at $(x = 2cm,\ y = 0cm)$ and an initial guess of $(x = 0cm,\ y = 0cm)$.

to improve the estimate for the state X_t based on the observations Z_t. The ordinary least squares objective function is

$$S = (Z_t - G|_{X_t} - B_t \Delta X_t)^T (Z_t - G|_{X_t} - B_t \Delta X). \tag{59}$$

The minimizer of equation 59 is found by forcing to zero the derivative with respect to ΔX resulting in the estimator

$$\Delta X_t = (B_t^{\mathrm{T}} B_t)^{-1} B_t^{\mathrm{T}} (Z_t - G_t|_{X_t}). \tag{60}$$

This method consists of obtaining expected temperatures from the direct model, computing the average temperature between the transducers, and then computing an expected time of flight from equation 29. For the current analysis, the average temperature is computed along the path on the non-heated plate surface between the two sensors. The Jacobian B_t is constructed using the derivatives with respect to sensor position for convenience since this information can be obtained with one direct model simulation. The derivatives are obtained from the direct model using finite differences by independently varying the x and y positions of all sensors by $0.0001m$.

6.5 Least squares with ellipse from ultrasonic pulse one-way time of flight measurement model

This method uses the same ellipse model detailed above with the extended Kalman filter and employs least squares method detailed above to locate the source. Figure 27 illustrates the geometry of an ellipse, where the two sensors are assumed to be the foci for the ellipse. Since the distance between sensors is known, ellipse parameters c and d can be related to each other and the ellipse can be represented with just one parameter c.

$$r_{is} + r_{js} = 2c = \sqrt{r_{ij}^2 + 4d^2} \tag{61}$$

where i and j are sensors and s is heat source.

$$c = \frac{1}{2}\sqrt{r_{ij}^2 + 4d^2} = \frac{r_{is} + r_{js}}{2} \tag{62}$$

$$r_{is} = \sqrt{(x_i - x_s)^2 + (y_i - y_s)^2} \tag{63}$$

$$r_{js} = \sqrt{(x_j - x_s)^2 + (y_j - y_s)^2} \tag{64}$$

$$\frac{\partial c_i}{\partial x_s} = \frac{1}{2}\left[\frac{x_s - x_i}{r_{is}} + \frac{x_s - x_j}{r_{js}}\right] \tag{65}$$

$$\frac{\partial c_i}{\partial y_s} = \frac{1}{2}\left[\frac{y_s - y_i}{r_{is}} + \frac{y_s - y_j}{r_{js}}\right] \tag{66}$$

This measurement model consists of measuring the one-way ultrasonic pulse time of flight, using the direct model to obtain the average temperature between the transducers for a range of c values, using equation 29 to compute time of flight for the range of c values, and then interpolating the time of flight results using the spline method to obtain c for the measured time of flight. The sensitivity matrix B_t is then

$$B_t = \begin{bmatrix} \frac{\partial c_1}{\partial x_s} & \frac{\partial c_1}{\partial y_s} \\ \frac{\partial c_2}{\partial x_s} & \frac{\partial c_2}{\partial y_s} \\ \frac{\partial c_3}{\partial x_s} & \frac{\partial c_3}{\partial y_s} \\ \frac{\partial c_4}{\partial x_s} & \frac{\partial c_4}{\partial y_s} \end{bmatrix}, \tag{67}$$

where t is time in seconds with a time step of $1second$, c_i with $i = 1, 2, 3, 4$ is the ellipse parameter if the source is located at (x_s, y_s). The desire is to improve the estimate for the state X_t based on the observations c.

6.6 Results

Convergence behavior for the three inverse methods and both measurement models is compared in Figures 44 through 46. With the heating source located inside the sensor grid (Figure 44), the extended Kalman filter

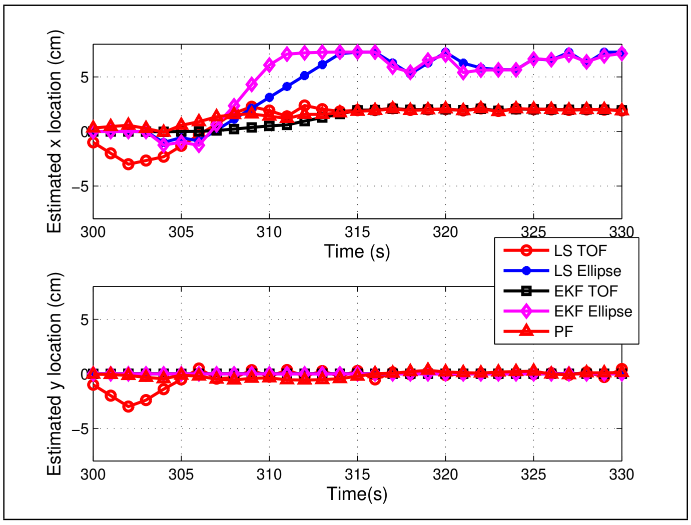


Figure 44: Least squares, extended Kalman filter, and particle filter convergence for both one-way ultrasonic pulse measurement models with the heating source located at $(x = 2cm,\ y = 0cm)$ and an initial guess of $(x = 0cm,\ y = 0cm)$.

and the least squares with the direct model and the particle filter converge to the correct location while the extended Kalman filter and the least squares with the ellipse model do not. Similar convergence behavior is found with the heating source located at the edge of the sensor grid (Figure 45), although convergence is much faster. With the heating source located outside of the sensor grid (Figure 46), none of the methods converge to the correct location. These examples started with an initial guess of $(x = 0cm,\ y = 0cm)$ for the heating source location.

Sensitivity to heating source location, explored in Section 5 and documented in previous work Myers et al. [2012b], can help explain these results. With the heating source outside the sensor grid, only one sensor pair would have sufficient sensitivity to heating source location and only then if the heating source is located close to the sensor pair. With only one sensor pair receiving usable information, the inverse routine has insufficient information to converge on the correct heating source location. With the heating source located inside the sensor grid, all sensor pairs receive usable information and the inverse routine is able to converge to the correct location. With the heating source located at the edge of the sensor grid, one sensor pair receives usable information almost instantaneously resulting in faster convergence.

The extended Kalman filter with ellipse and least squares with ellipse models use the least amount of wall time and but the most memory of the five methods. Wall time for each iteration, independent of re-computing the direct model for the updated parameters, is approximately three times longer for the least squares time of flight model and almost four times longer for the extended Kalman filter time of flight model than the wall time for both of the ellipse models. The particle filter requires approximately 37 times more wall time than the ellipse models. Memory usage is lowest for the extended Kalman filter time of flight model and the least squares time of flight model. The particle filter requires approximately 30% more memory and the ellipse models require approximately four times more memory than the time of flight models.

Repeating the experiment using multiplexing equipment and a complete sensor grid of four transducers instead of simulating the sensor grid with separate experiments using two transducers might produce different convergence behavior, especially for heating source locations outside the sensor grid. While the experiment is reproducible, simulating a sensor grid with separate experiments introduces uncertainties that could effect the results. Additionally, sensor and heating source placement introduce uncertainties when simulating the sensor grid.

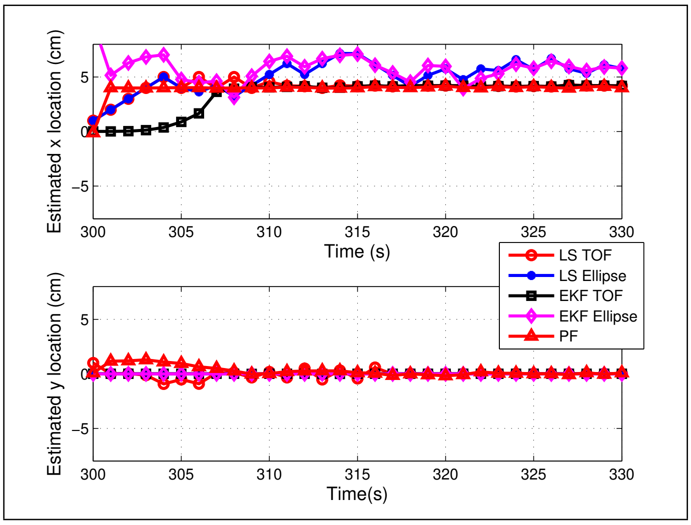


Figure 45: Least squares, extended Kalman filter, and particle filter convergence for both one-way ultrasonic pulse measurement models with the heating source located at $(x = 4cm,\ y = 0cm)$ and an initial guess of $(x = 0cm,\ y = 0cm)$.

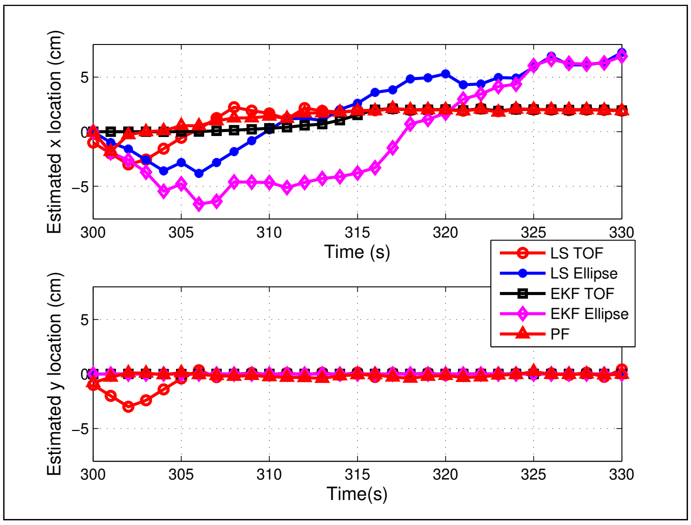


Figure 46: Least squares, extended Kalman filter, and particle filter convergence for both one-way ultrasonic pulse measurement models with the heating source located at $(x = 6cm,\ y = 0cm)$ and an initial guess of $(x = 0cm,\ y = 0cm)$.

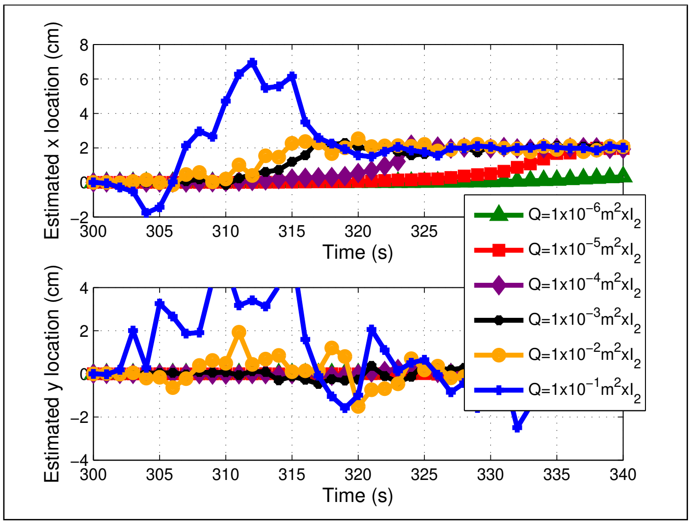


Figure 47: Extended Kalman filter convergence for a range of state model covariance values (Q) with constant measurement covariance values of $R = 4{-}7 \times I_4$, the heating source located at ($x = 2cm$, $y = 0cm$), and an initial guess of ($x = 0cm$, $y = 0cm$).

6.7 Adaptive extended Kalman filter

Figure 47 illustrates the sensitivity to the state model covariance by comparing values from $Q = 0.1m^2 \times I_2$ to $0.000001m^2 \times I_2$. Decreasing the state model covariance (Q) magnitude results in a damping effect on the convergence. Decreasing the magnitude too far causes the estimated position values to remain fairly constant and the solution fails to converge. Conversely, increasing the state model covariance (Q) magnitude increases the convergence rate. Increasing the state model covariance (Q) too far results in erratic position estimates and the solution fails to converge.

Figure 48 illustrates the sensitivity to the measurement covariance by comparing values from $R = 4{-}5 \times I_4$ to $R = 4{-}10 \times I_4$. Increasing the measurement covariance (R) magnitude results in a damping effect on the convergence. Increasing the magnitude too far causes the estimated position values to remain fairly constant and the solution fails to converge. Conversely, decreasing the measurement covariance (R) magnitude increases the convergence rate. Decreasing the measurement covariance (R) too far results in erratic position estimates and the solution fails to converge.

A trend is evident when comparing Figures 47 and 48 in that Q and R appear to be inversely correlated. Figure 49 illustrates the relationship. Decreasing Q by one order of magnitude or increasing R by one order of magnitude results in similar convergence behavior. Likewise, increasing Q by one order of magnitude or decreasing R by one order of magnitude also results in similar convergence behavior. For example, using $Q = 0.0001m^2 \times I_2$ and $R = 4{-}7 \times I_4$ as the baseline, decreasing the state model covariance to $Q = 0.00001m^2 \times I_2$ but keeping the measurement covariance at $R = 4{-}7 \times I_4$ results in similar convergence behavior if the state model covariance is kept at $Q = 0.0001m^2 \times I_2$ and the measurement covariance is increased to $R = 4{-}6 \times I_4$.

We can conclude from these observations that the state model covariance (Q) and the measurement covariance (R) are correlated for this heating source localization scenario. The measurement covariance is determined from sensor noise, a measurable quantity, and the state model covariance is unknown and not measurable. Therefore, a large uncertainty exists in the state model covariance while a small uncertainty exists in the measurement covariance. Selection of an appropriate state model covariance must then be obtained through a parameter sweep while observing convergence behavior.

Figure 51 illustrates the magnitude of each element in the Kalman gain (K_t), which, for this heating source localization, is a 2×4 matrix. Comparing Figure 51 with Figure 50, the $(1, 3)$ value from the Kalman gain (K_t) stands out as a possible source since it increases until convergence and then decreases rapidly. However, this value remains large even after convergence.

Figure 52 illustrates the normalized magnitude of the variance contained in the state covariance matrix (Σ) which is a 2×2 matrix in this heating source localization problem. The variance values are in the main diagonal

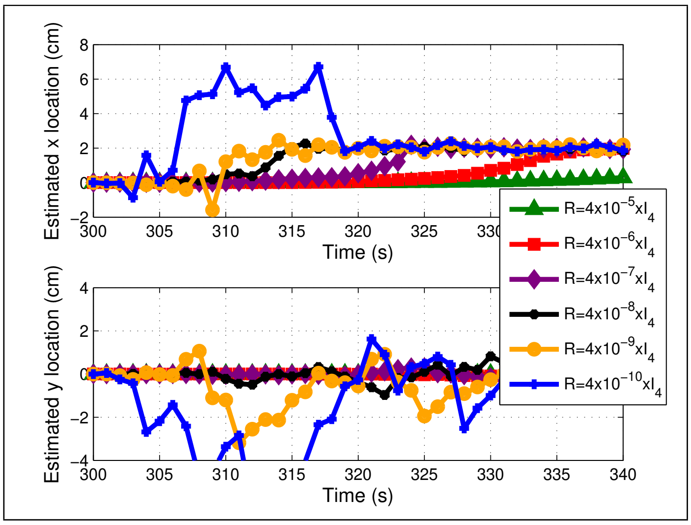


Figure 48: Extended Kalman filter convergence for a range of measurement covariance values (R) with constant state model covariance values of $Q = 1{-}4m^2 \times I_2$, the heating source located at ($x = 2cm$, $y = 0cm$), and an initial guess of ($x = 0cm$, $y = 0cm$).

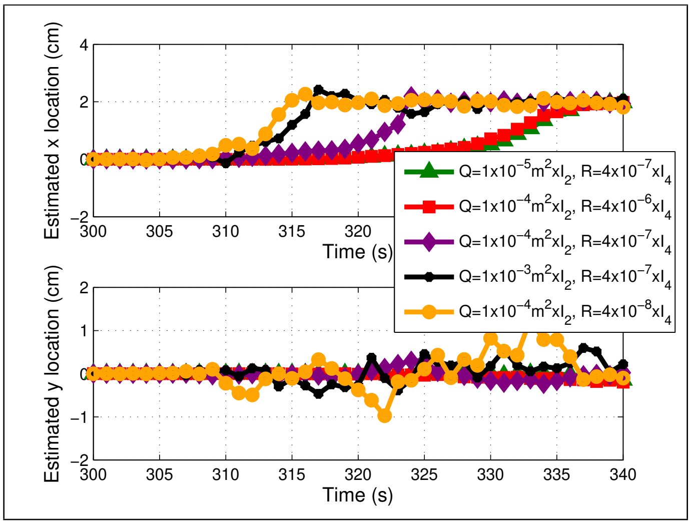


Figure 49: Extended Kalman filter convergence illustrating the correlation between the state model covariance matrix (Q) and the measurement covariance matrix (R). The heating source is located at ($x = 2cm$, $y = 0cm$) with an initial guess of ($x = 0cm$, $y = 0cm$).

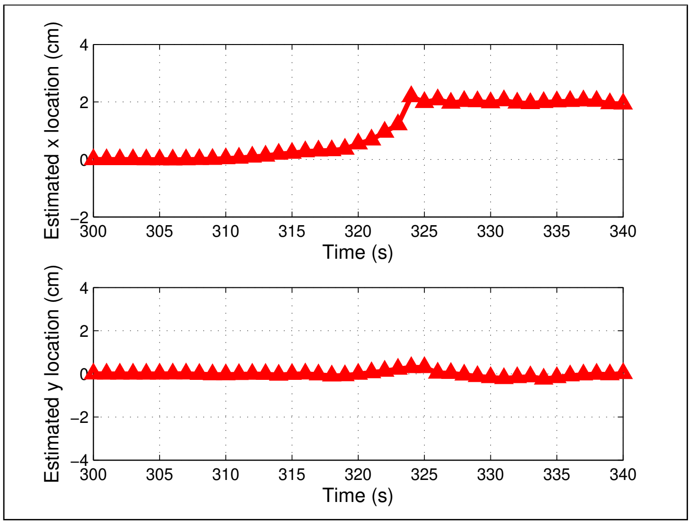


Figure 50: Extended Kalman filter convergence with the state model covariance values of $Q = 1{-}4m^2 \times I_2$, measurement covariance values of $R = 4{-}7 \times I_4$, heating source located at ($x = 2cm$, $y = 0cm$), and an initial guess of ($x = 0cm$, $y = 0cm$).

of the matrix and are identical. Comparing with Figure 50, we observe that the variance increases steadily, decreases rapidly just before and during convergence, and remains small after convergence. Figure 53 illustrates the normalized magnitude of the variance for a range of state model covariance values from $Q = 1{-}1m^2 \times I_2$ to $Q = 1{-}6m^2 \times I_2$ and a measurement covariance of $R = 4{-}7 \times I_4$. A comparison of Figure 53 with Figure 47 yields the observation that the variance increases steadily, decreases rapidly just before and during convergence, and remains small after convergence for every state model covariance examined.

The adaptive extended Kalman filter incorporates three new parameters. The predefined tolerance value $\Sigma_{tolerance}$ for changes to the state covariance Σ is based on the variance found in the first iteration and is defined as $\Sigma_{tolerance} = \Sigma_1$. The magnitude of Σ is dependent upon filter parameters including the state model covariance Q, thus basing the tolerance on the first iteration ensures the adaptive nature of the filter will smooth convergence near the converged solution. The predefined limit to convergence rate ΔX_{limit} for this work is defined as $\Delta X_{limit} = [\Delta x_{limit}, \Delta y_{limit}]^T = [1cm/sec, 1cm/sec]^T$. Figure 54 illustrates convergence for the adaptive extended Kalman filter with $Q_0 = 1{-}4m^2 \times I_2$, $R = 4{-}7 \times I_4$, and an adaptive gain of $M_t = 2$. The adaptive extended Kalman filter outperforms the extended Kalman filter in this example. Figure 55 illustrates the variance value in Q during convergence for the adaptive extended Kalman filter while Figure 56 illustrates the variance values from Σ for a range of initial state model covariance Q_0 values.

Figure 57 illustrates the effect that different initial state model covariance values has on the adaptive extended Kalman filter convergence. Comparing with Figure 47, the adaptive extended Kalman filter is able to converge quicker for a significant range of initial state model covariance values Q_0. Figure 58 illustrates the sensitivity to the adaptive extended Kalman filter gain M_t.

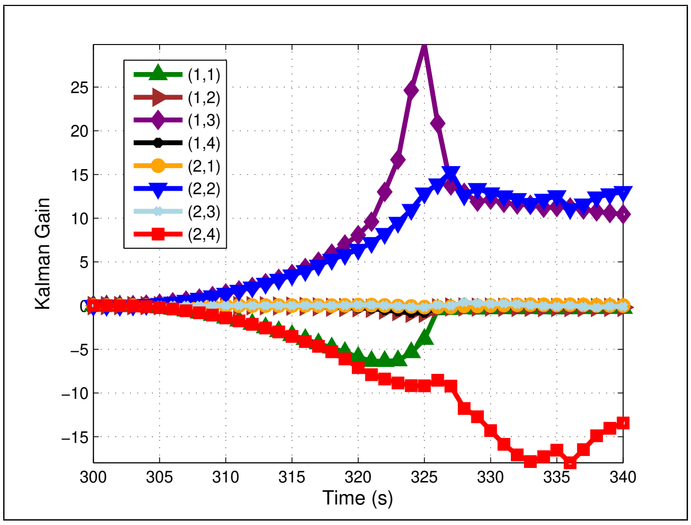


Figure 51: Kalman gain values during convergence for state model covariance of $Q = 1{-}4m^2 \times I_2$ and measurement covariance of $R = 4{-}7 \times I_4$. The heating source is located at $(x = 2cm,\ y = 0cm)$ with an initial guess of $(x = 0cm,\ y = 0cm)$. Legend entries refer to the matrix element in the Kalman gain which is a 2×4 matrix. Convergence (from Figure 50) is at $324s$.

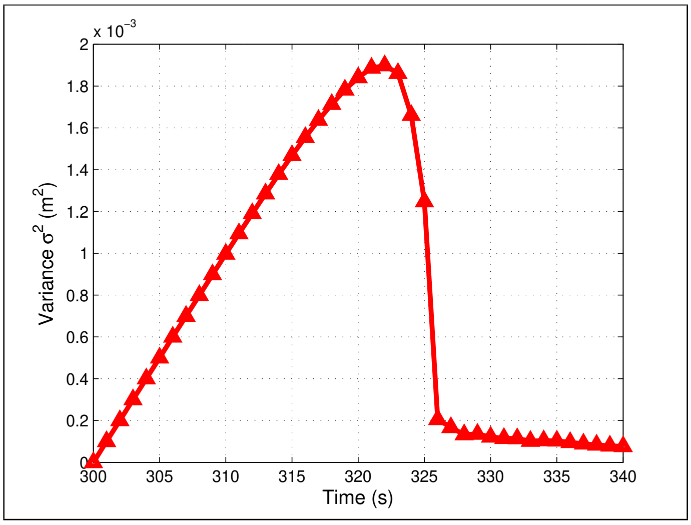


Figure 52: Variance (σ^2) from the state covariance matrix (Σ) for state model covariance of $Q = 1{-}4m^2 \times I_2$ and measurement covariance of $R = 4{-}7 \times I_4$. The heating source is located at $(x = 2cm,\ y = 0cm)$ with an initial guess of $(x = 0cm,\ y = 0cm)$. Convergence (from Figure 50) is at $324s$.

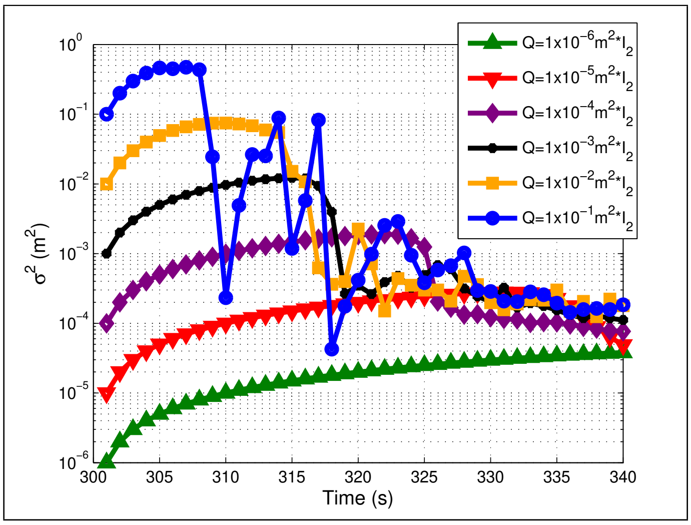


Figure 53: Variance (σ^2) from the state covariance matrix (Σ) for a range of state model covariance values (Q) and constant measurement covariance of $R = 4{-}7 \times I_4$. The heating source is located at ($x = 2cm$, $y = 0cm$) with an initial guess of ($x = 0cm$, $y = 0cm$). Convergence behavior can be found in Figure 47.

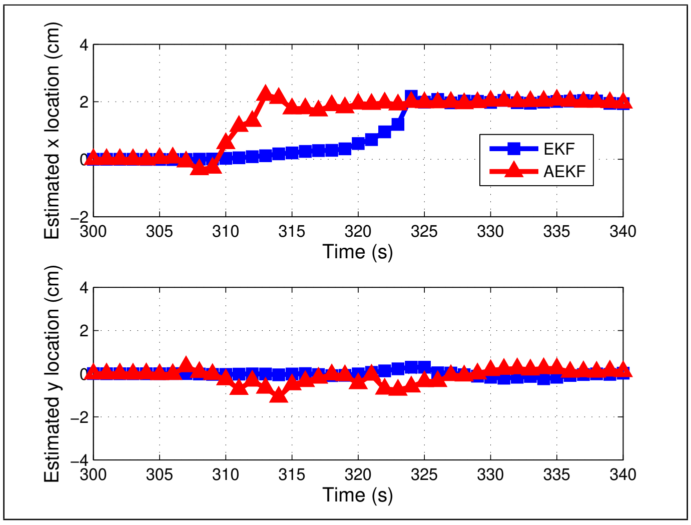


Figure 54: Extended Kalman filter and adaptive extended Kalman filter convergence with $Q_0 = 1{-}4m^2 \times I_2$, $R = 4{-}7 \times I_4$, and $M = 2$. The heating source is located at ($x = 2cm$, $y = 0cm$) with an initial guess of ($x = 0cm$, $y = 0cm$).

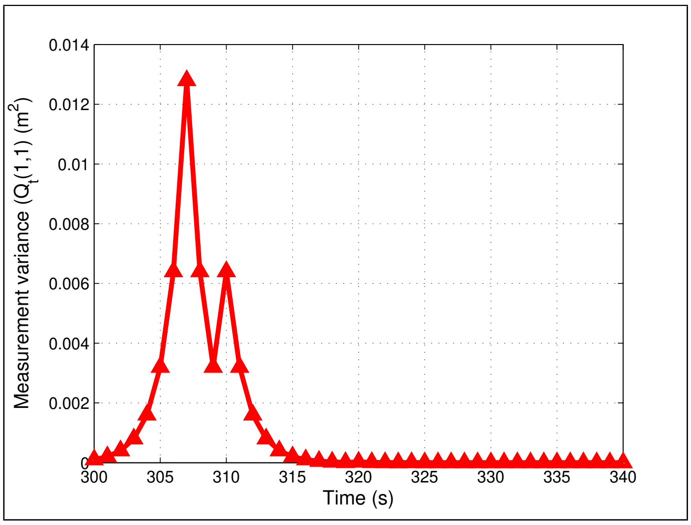


Figure 55: Adaptive extended Kalman measurement covariance (Q_t) values during convergence with $Q_0 = 1{-}4m^2 \times I_2$, $R = 4{-}7 \times I_4$, and $M_t = 2$. The heating source is located at ($x = 2cm$, $y = 0cm$) with an initial guess of ($x = 0cm$, $y = 0cm$).

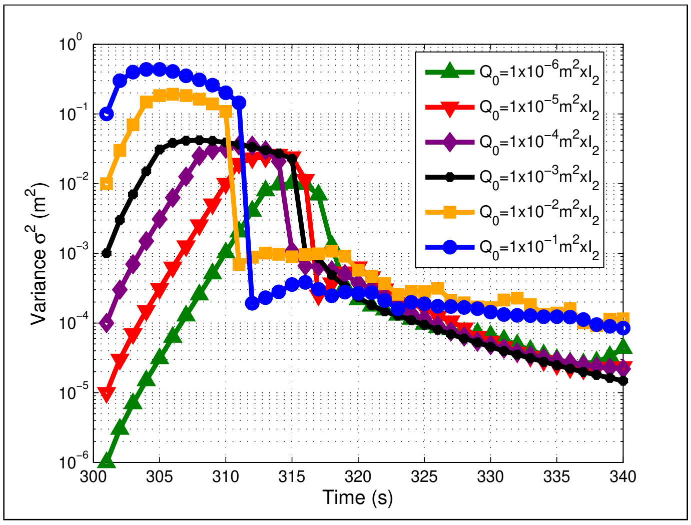


Figure 56: Adaptive extended Kalman filter variance (σ^2) from the state covariance matrix (Σ) for a range of starting state model covariance values (Q_0), constant measurement covariance of $R = 4{-}7 \times I_4$, and $M_t = 2$. The heating source is located at ($x = 2cm$, $y = 0cm$) with an initial guess of ($x = 0cm$, $y = 0cm$).

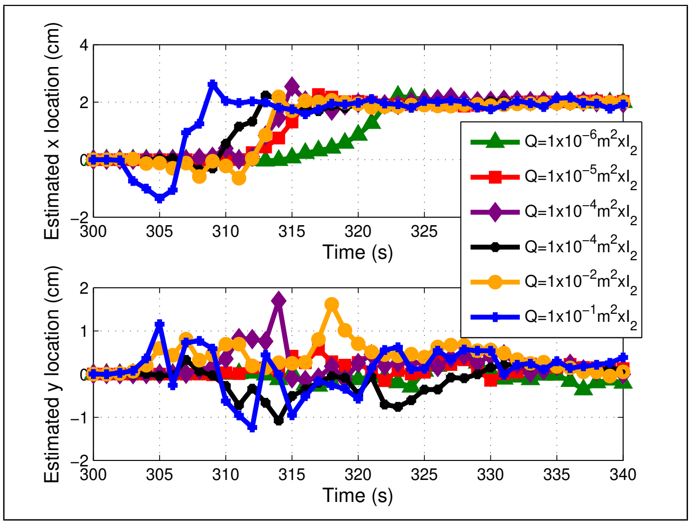


Figure 57: Adaptive extended Kalman filter convergence for a range of initial state model covariance values (Q_0), constant measurement covariance of $R = 4{-}7 \times I_4$, and a state model covariance gain of $M = 2$. The heating source is located at ($x = 2cm$, $y = 0cm$) with an initial guess of ($x = 0cm$, $y = 0cm$).

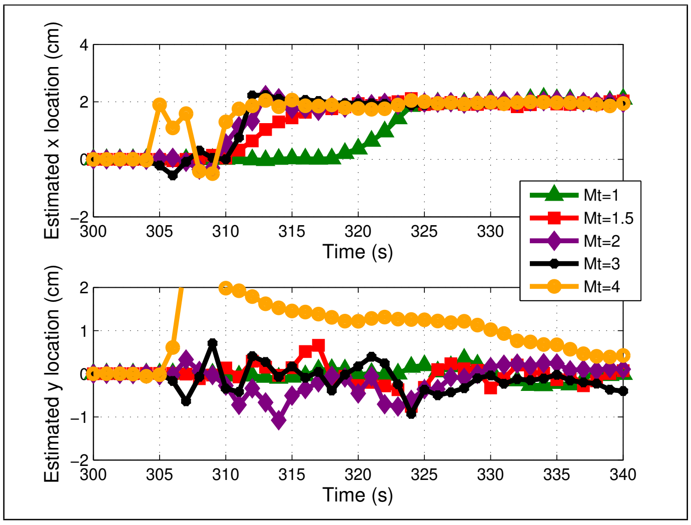


Figure 58: Adaptive extended Kalman filter convergence for a range of state model covariance gain values M, an initial state model covariance of $Q_0 = 1{-}4m^2 \times I_2$, and constant measurement covariance of $R = 4{-}7 \times I_4$. The heating source is located at ($x = 2cm$, $y = 0cm$) with an initial guess of ($x = 0cm$, $y = 0cm$).

7 Chapter Summary

This Chapter presents some fundamental concepts for detection, localization, and parameter estimation using inverse methods. These techniques can be used to characterize singularities, discontinuities, material properties, boundary conditions, loading, damage, structural changes, etc. Parameter estimation addresses the problem of estimating quantities that are not directly observable but that can be inferred from sensor data. Sensors carry only partial information and their measurements are corrupted by noise. Parameter estimation seeks to recover state variables from the sensor data using an iterative approach that incorporates a direct model.

References

M. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE transactions on signal processing*, 50(2):174–188, 2002.

J. Beck and K. Woodbury. Inverse problems and parameter estimation: integration of measurements and analysis. *Measurement Science and Technology*, 9:839–847, Jun 1998. doi: 10.1088/0957-0233/9/6/001.

J. V. Beck and K. J. Arnold. *Parameter Estimation in Engineering and Science.* John Wiley & Sons, 1977.

K. Berger, S. Rufer, R. Kimmel, and D. Adamczak. Aerothermodynamic characteristics of boundary layer transition and trip effectiveness of the HIFiRE flight 5 vehicle. In *AIAA Proceedings*, number AIAA-2009-4055, June 2009.

C. A. Brebbia and J. Dominguez. *Boundary elements: an introductory course.* WIT press, 1994.

O. Bretscher. *Linear Algebra With Applications.* Prentice Hall, 1995.

A. Doucet, N. Gordon, and V. Krishnamurthy. Particle filters for state estimation of jump Markov linear systems. *IEEE Transactions on Signal Processing*, 49(3):613–624, 2002.

K. J. Dowding and B. F. Blackwell. Sensitivity analysis for nonlinear heat conduction. *Journal of Heat Transfer*, 123(1):1, Feb 2001. doi: 10.1115/1.1332780.

M. Engelhardt, G. E. Stavroulakis, and H. Antes. Crack and flaw identification in elastodynamics using kalman filter techniques. *Computational Mechanics*, 37(3):249–265, 2006.

T. J. Horvath, S. A. Berry, and B. R. Hollis. Boundary layer transition on slender cones in conventional and low disturbance Mach 6 wind tunnels. In *32nd AIAA Fluid Dynamics Conference and Exhibit*, number AIAA-2002-2743, St. Louis, MO, 24–27 Jun 2002. AIAA.

F. P. Incropera and D. P. DeWitt. *Introduction to Heat Transfer.* John Wiley and Sons, Hoboken, NJ, 4th edition, 2002.

V. Kozlov, V. Adamchik, and V. Lipovtsev. Local heating of an unbounded orthotropic plate through a circular and annular domain. *Journal of Engineering Physics and Thermophysics*, 57:1381–1391, Jan 1989. doi: 10.1007/BF00871278.

P. d. S. Lopes, A. B. Jorge, and S. S. Cunha Jr. Detection of holes in a plate using global optimization and parameter identification techniques. *Inverse Problems in Science and Engineering*, 18(4):439–463, 2010.

M. Myers, A. Jorge, D. Yuhas, and D. Walker. An adaptive extended Kalman filter incorporating state model uncertainty for localizing a high heat flux spot source using an ultrasonic sensor array. *CMES: Computer Modeling in Engineering & Sciences*, 83(3):221–248, 2012a. doi: 10.3970/cmes.2012.083.221.

M. R. Myers, D. G. Walker, D. E. Yuhas, and M. J. Mutton. Heat flux determination from ultrasonic pulse measurements. In *International Mechanical Engineering Congress and Exposition*, number IMECE2008-69054. ASME, 2–6 Nov 2008.

M. R. Myers, A. B. Jorge, D. G. Walker, and M. J. Mutton. A comparison of extended Kalman filter, extended information filter, and least squares approaches for parameter identification of a transient heat transfer problem. In *Inverse Problems Symposium*, East Lansing, MI, 6–8 Jun 2010a. Michigan State University.

M. R. Myers, A. B. Jorge, D. G. Walker, and M. J. Mutton. A comparison of extended Kalman filter approaches using non-linear temperature and ultrasound time-of-flight measurement models for heating source localization of a transient heat transfer problem. In *Inverse Problems, Design and Optimization Symposium*, number IPDO-027, Brazil, 25–27 Aug 2010b. IPDO.

M. R. Myers, A. B. Jorge, M. J. Mutton, and D. G. Walker. High heat flux point source sensitivity and localization analysis for an ultrasonic sensor array. *International Journal of Heat and Mass Transfer*, 55 (9-10):2472–2485, Apr. 2012b. doi: 10.1016/j.ijheatmasstransfer.2012.01.012.

M. R. Myers, A. B. Jorge, M. J. Mutton, and D. G. Walker. A comparison of extended Kalman filter ultrasound time-of-flight measurement models for heating source localization. *Inverse Problems in Science and Engineering*, 20(7):991–1016, 2012c. doi: 10.1080/17415977.2012.669272.

M. R. Myers, D. G. Walker, D. E. Yuhas, and M. J. Mutton. Heat flux determination from ultrasonic pulse measurements. *International Journal of Heat and Mass Transfer*, in review, 2010. (in review).

Recreational Aviation Australia. Flow regimes. http://www.auf.asn.au, 2010. [Online; accessed 10-March-2010].

H. L. Reed, R. Kimmel, S. Schneider, D. Arnal, and W. Saric. Drag prediction and transition in hypersonic flow. In *AGARD CONFERENCE PROCEEDINGS AGARD CP*, volume 3, pages C15–C15. AGARD, 1997.

D. Salmond, N. Gordon, and A. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings-F, Radar and signal processing*, 140(2):107–113, 1993.

S. Schneider. Flight data for boundary-layer transition at hypersonic and supersonic speeds. *Journal of Spacecraft and Rockets*, 36(1):8–20, 1999.

S. Schneider. Hypersonic laminar-turbulent transition on circular cones and scramjet forebodies. *Progress in Aerospace Sciences*, 40(1-2):1–50, 2004. doi: 10.1016/j.paerosci.2003.11.001.

G. Stavroulakis and H. Antes. Flaw identification in elastomechanics: Bem simulation with local and genetic optimization. *Structural optimization*, 16(2):162–175, 1998.

S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics.* The MIT Press, Cambridge, MA, 2006.

F. Vianna, H. Orlande, and G. Dulikravich. Prediction of the temperature field in pipelines with Bayesian filters and non-intrusive measurements. In *Proceedings of the 20th International Congress of Mechanical Engineering*, Gramado, RS, Brazil, 2009.

F. L. V. Vianna, H. Orlande, and G. Dulikravich. Temperature field prediction of a multilayered composite pipeline based on the particle filter method. *Proceedings of the 14th International Heat Transfer Conference*, 2010.

B. Vieira, A. Jorge, P. Lopes, and S. Cunha Junior. A Kalman filter model for an inverse problem of localization and identification of damage parameters on a 2-D structure. In *XXXII Cilamce-Iberian Latin-American Congress on Computational Methods in Engineering*, pages 1–18, 2011.

K. A. Woodbury, editor. *Inverse Engineering Handbook.* CRC Press, Boca Raton, FL, 2003a.

K. A. Woodbury. Sequential function specification method using future times for function estimation. In K. A. Woodbury, editor, *Inverse Engineering Handbook.* CRC Press, Boca Raton, FL, 2003b.

Annex: On the use of a Kalman Filter model for an inverse problem of localization and identification of damage parameters on a 2-D structure

An alternative to the finite element method detailed above, the boundary element method presents another tool for the direct model. In this annex, the boundary element method is used with a potential formulation and an elastostatic formulation for a plate with an inner hole representing damage to the plate. The damage here is treated as a circular hole, a geometric discontinuity, where the state, coordinates, and radius of the hole do not change with time, there is no control on the model, and the noise is considered to be white noise and low in amplitude.

Potential formulation

In the potential model, the plate and the hole are divided equally where its nodes should contain information on the x and y-directions on the temperature T and heat flux q. The number of elements that divide the plate and the number of elements that divide the hole need not be the same as this amount is the user's discretion, noting that the greater the division will produce more accurate results while also increasing computational cost.

This model consists of a temperature distribution on the surface of a square plate, six centimeters on each side. A temperature differential of 300 °C is applied between two edges with the other two edges and the hole adiabatic (i.e. the heat flow is zero). The potential model of the plate is illustrated by Figure 59.

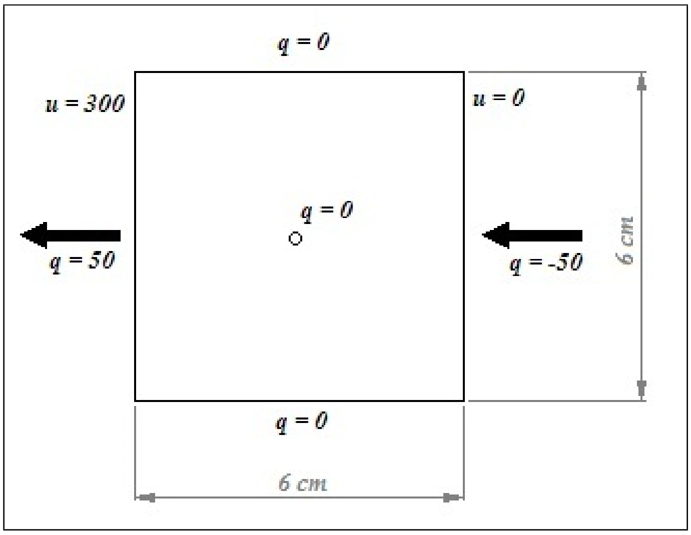


Figure 59: Plate model for the potential problem.

A hole of 0.06 cm of radius in the plate's center and the sensors were placed initially at 0.6 cm from the edges of the plate with a total of eight sensors. This sensor positioning was adopted, imagining a better utilization coupled with the possibility of easy positioning of thermocouples in the coordinates. With respect to the model developed in the boundary element method, the plate is divided into 12 elements and the hole with 24 elements, this division can be seen in Figure 60 along with the sensor positions.

Elastostatic formulation

As in the potential model, the plate and the hole are divided by elements, where its nodes should contain information on the x and y-directions, but now about the stress, shear τ or normal σ, or displacement δ prescribed itself. These stresses, normal and shear, cannot be used as a response of boundary element method in a damage detection problem, since they depend on the coordinate system used, or the normal direction of the cut plane that passes through the point of interest Lopes et al. [2010]. Therefore, it is necessary to use invariants of stress, which may be the average normal stress or octahedral stress.

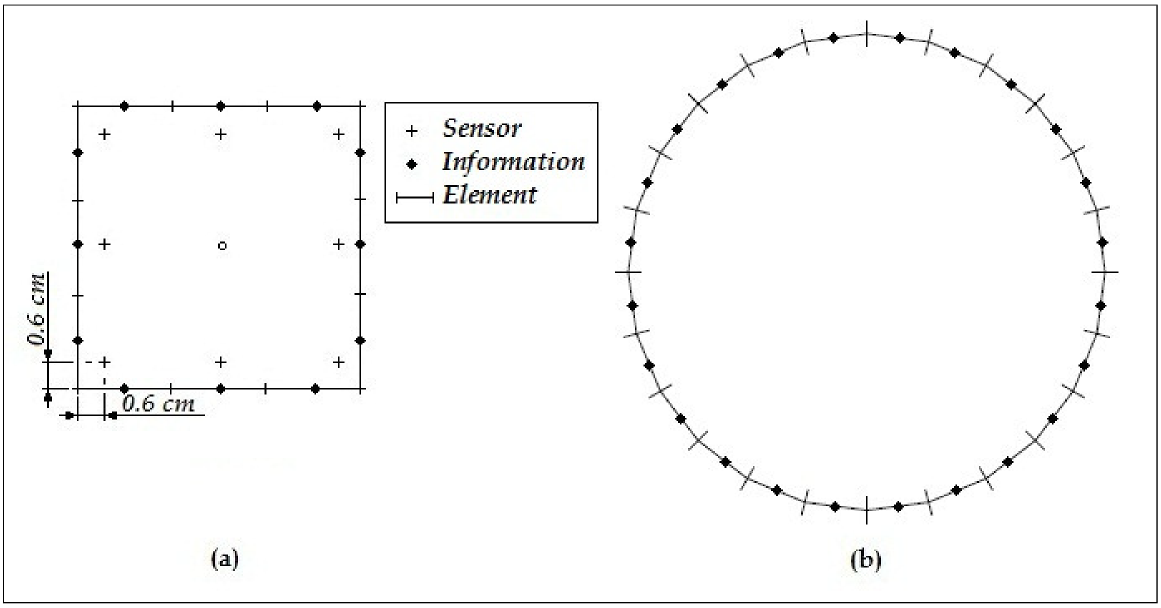


Figure 60: Potential model: (a) discretization of the plate and positioning of sensors, (b) discretization of the hole.

The average stress for the two-dimensional case is defined as:

$$\sigma_m = \frac{\sigma_x + \sigma_y}{2} \tag{68}$$

where σ_x is the normal stress in the axis direction x; σ_y is the normal stress in the axis y; and the octahedral stress, also in the two-dimensional case, is defined as:

$$\tau_{oct} = \sqrt{\frac{2}{9}\left[(\sigma_x + \sigma_y)^2 - 3(\sigma_x\sigma_y - \tau_{xy}^2)\right]} \tag{69}$$

The elastostatic model, consists of a plate of same dimensions as above with an applied load of 1,000 MPa traction on the plate towards the ordinate axis and leaving the other sides free. With respect to the hole, it is considered that there is no load. Figure 61 illustrates the model.

With respect to material of the plate, this is simulated considering a shear modulus of 94.5 GPa and a Poisson's ratio for plane strain of 0.1 Brebbia and Dominguez [1994], Lopes et al. [2010]. In this model, the hole used has a radius of 0.12 cm and sensors are positioned at 0.6 cm from the edges of the plate with a total of eight sensors. The plate is divided into 24 elements and the hole into 12 elements. The model is illustrated in Figure 62.

Potential model results

First the parameters were adjusted for the direct and inverse problem, where most parameters are determined by trial and error. Soon, after a few iterations, the following parameters were adopted:

$$R = \left(\frac{0.01 I_{8x8}}{3}\right)^2, \tag{70}$$

$$Q = \begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.1 & 0 \\ 0 & 0 & 0.01 \end{bmatrix}^2, \tag{71}$$

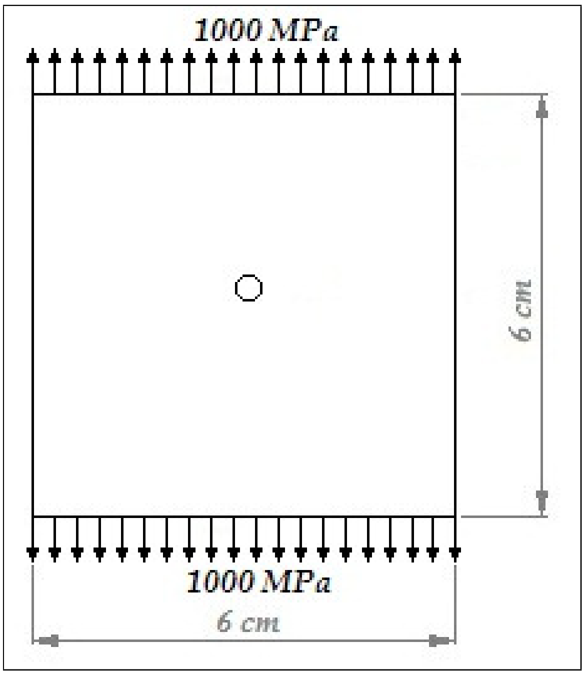


Figure 61: Plate model for the elastostatic problem.

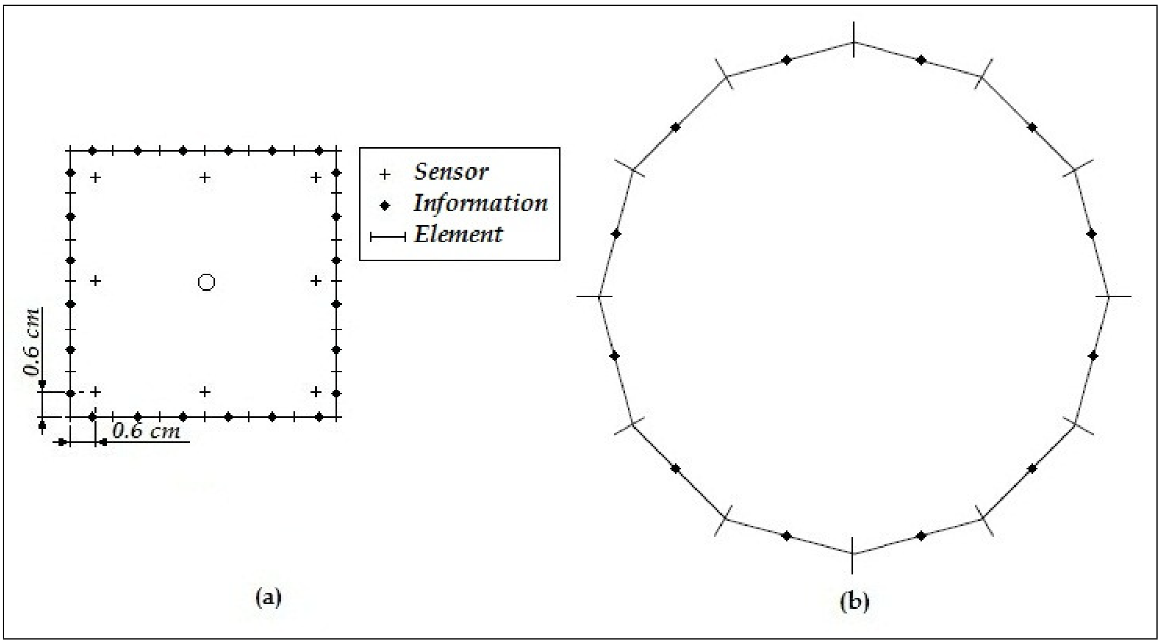


Figure 62: Elastostatic model: (a) discretization of the plate and positioning of sensors, (b) discretization of the hole.

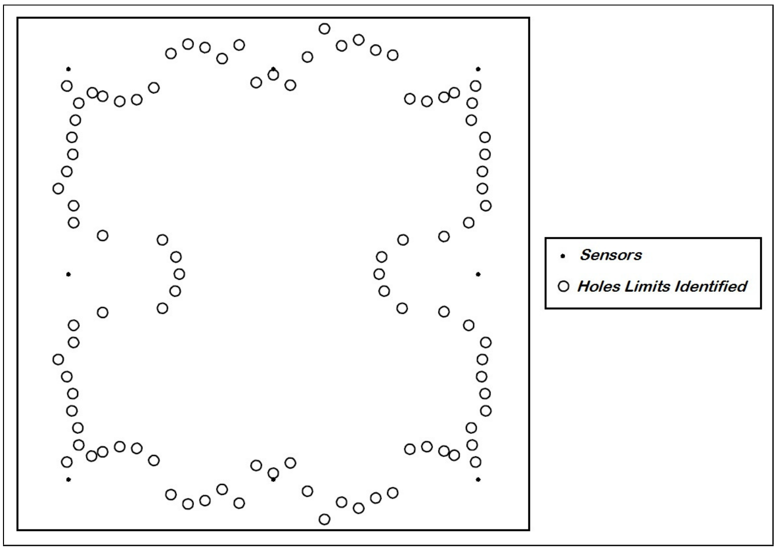


Figure 63: Region limit of identification of the filter for the sensor to 0.6 cm from the edges of a plate of 6x6 cm.

and the variations of the components of the state to calculate the sensitivity matrix

$$\Delta_x = \Delta_y = \Delta_r = 0.002 \tag{72}$$

The covariance matrix of the measurement noise was defined as the accuracy of a thermocouple of 0.01 °C, multiplied by the identity (each sensor has no effect on the other and they are not correlated). The covariance of the noise process, which also is not correlated, provides values only on the main diagonal, and the noise values being 10% and 1% of the state, to the coordinates and size of the hole, respectively.

With these parameters, using sensors at a distance of 0.6 cm from the edges and a maximum of 50 iterations, the Kalman filter obtained a result with error less than 0.3% for any of the three components of the state within a region that is illustrated by Figure 63, whose area is 20.56 mm^2.

When positioning the hole outside the region, which is 0.1 mm displacement, the error for the estimation of the filter jumps to extremely high values. This setting limit shows that the boundary conditions, as well as the sensors, interfere with the results of the estimation of the filter. The position of the sensors, therefore, is limited to a distance of 0.06 cm from the plate edges. The acceptable sensor region is illustrated in Figure 64 whose area is 21.48 mm^2 and the errors are less than 0.05%.

For this model, with these settings used, the positioning of sensors interferes in performance of the Kalman filter. For both configurations, the positioning of sensors has an increase of 4.47% of the area detectable by the Kalman filter. This inner region is where the filter locates and identifies the hole, since the external region have points that the filter can locate and identify, but randomly, being some isolated points and other forming small sub-regions.

For the positioning sensor to 0.06 cm from the edges, for a hole with coordinates (2.00;2.00) cm and radius equal to 0.06 cm, the Kalman filter produced results presented in Figures 65-68.

Elastostatic model

Figure 69 shows the results of the Kalman filter for the elastostatic model using as parameters:

$$Q = \left(\begin{bmatrix} 0.1 & 0 & 0 \\ 0 & 0.03 & 0 \\ 0 & 0 & 0.001 \end{bmatrix} * 10^3 \right)^2, \tag{73}$$

$$R = (\sigma_{z_k} I_{8x8})^2, \Delta_x = \Delta_y = 0.00035, \tag{74}$$

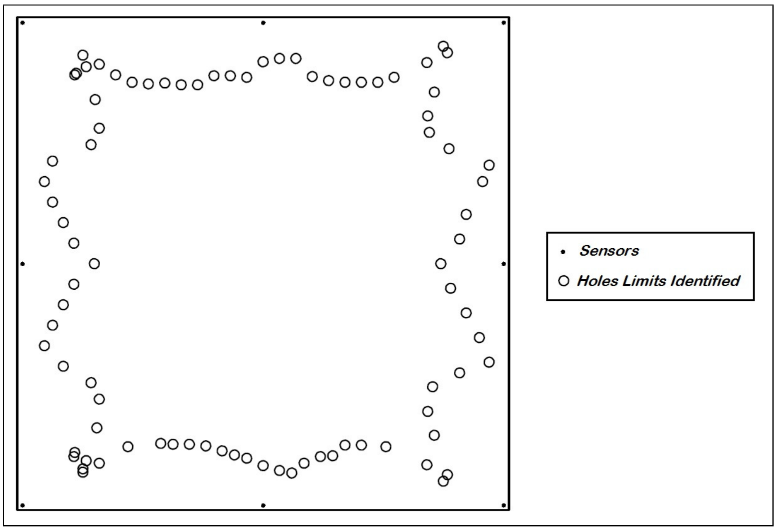


Figure 64: Region limit of identification of the filter for the sensor to 0.6 cm from the edges of a plate of 6x6 cm.

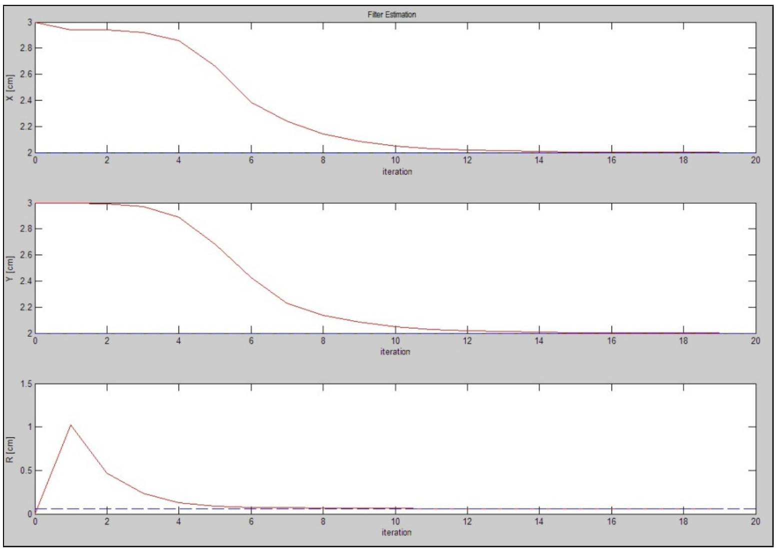


Figure 65: Estimation of components of state for a hole (2.00;2.00;0.06) cm for a potential model.

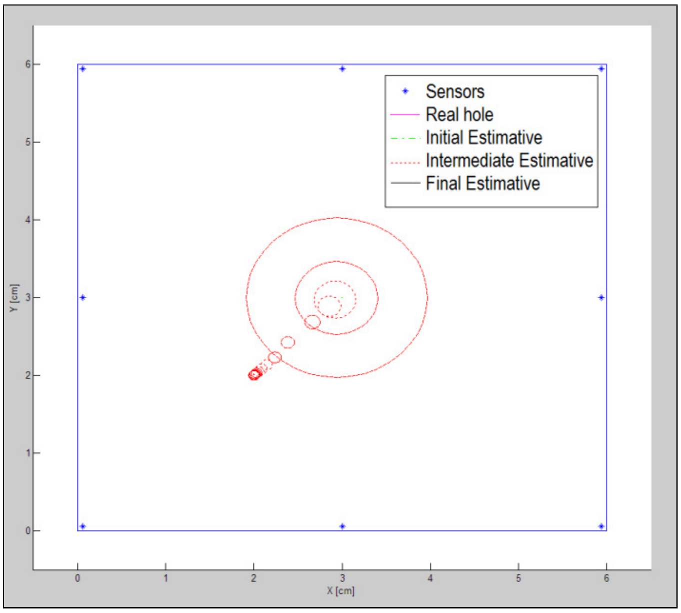


Figure 66: Performance of Kalman filter for a hole (2.00;2.00;0.06) cm for a potential model.

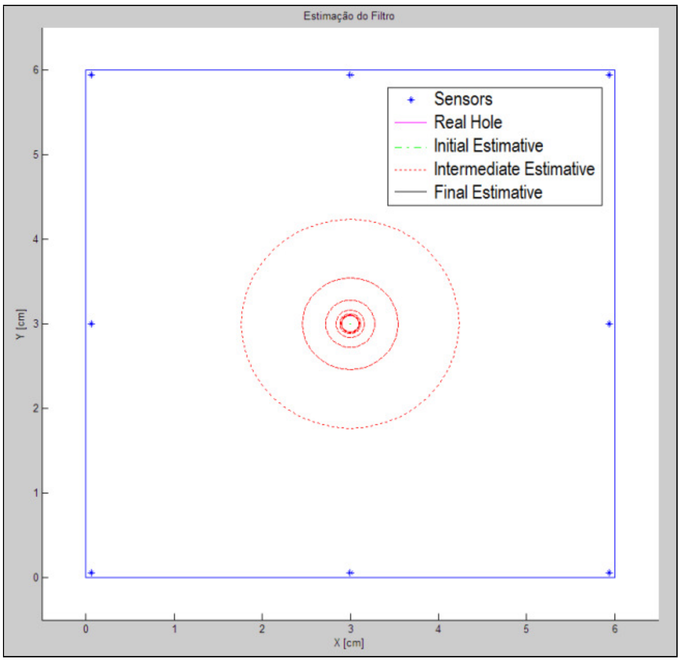


Figure 67: Performance of Kalman filter for a hole (3.00;2.00;0.06) cm for a potential model.

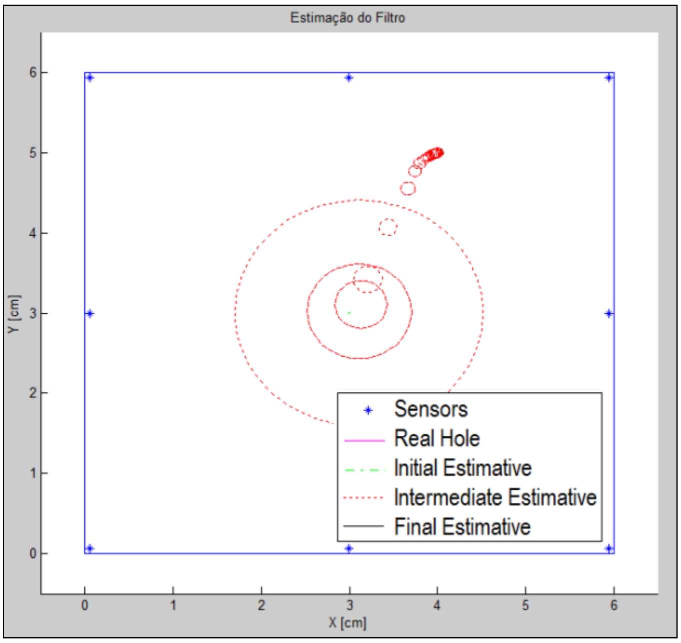


Figure 68: Performance of Kalman filter for a hole (4.00;5.00;0.06) cm for a potential model.

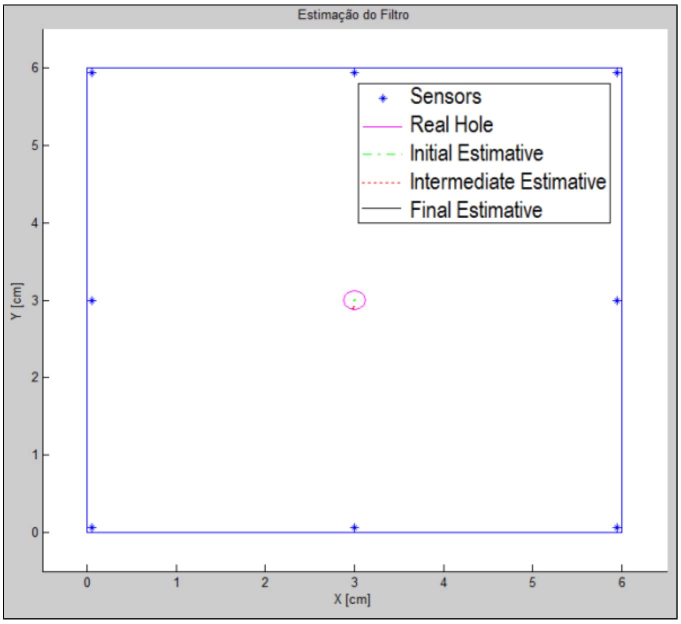


Figure 69: Performance of Kalman filter for a hole (3.00;3.00;0.06) cm for an elastostatic model.

and

$$\Delta_r = 0.002 \tag{75}$$

The tolerance for conversion is 1 x 10^{-3} and the maximum number of iterations equal to 75. In the measurement noise the covariance consists of the standard deviation of measured (σ_{z_k}) times the identity matrix (I_{8x8}).

Summary

In this annex, an inverse problem using the Kalman filter has been presented for localization and identification of damage parameters with a potential and an elastostatic formulation of a 2-D problem. A hybrid model of was presented where the state is described by a linear model and the measurements by a non-linear model, obtained from the numeric results of a boundary element method model of the direct problem.

In both formulations, the local information is obtained at an interior point (the temperature, for the potential case and the average stress or the octahedral stress for the elastostatics case), is a scalar quantity, and thus it is not dependent on the coordinate system. For the potential formulation with an adequate positioning of the sensors, there was an interior region in the domain in which the Kalman filter was able to identify and locate the hole. For the elastostatic formulation, the number and location of the internal points must be optimized to properly locate and identify the presence of the hole.

Chapter 12: Fundamental Concepts for Impedance-based Structural Health Monitoring

Chapter details

Chapter DOI:

https://doi.org/10.4322/978-65-86503-71-5.c12

Chapter suggested citation / reference style:

Finzi Neto, Roberto M., Moura Junior, Jose R. V. (2022). “Fundamental Concepts for Impedance-based Structural Health Monitoring”. *In* Jorge, Ariosto B., et al. (Eds.) *Model-Based and Signal-Based Inverse Methods*, Vol. I, UnB, Brasilia, DF, Brazil, pp. 443–471. Book series in Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity.

P.S.: DOI may be included at the end of citation, for completeness.

Book details

Book: Model-based and Signal-Based Inverse Methods

Edited by: Jorge, Ariosto B., Anflor, Carla T. M., Gomes, Guilherme F., & Carneiro, Sergio H. S.

Volume I of Book Series in:

Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity

Published by: UnB City: Brasilia, DF, Brazil Year: 2022

DOI: https://doi.org/10.4322/978-65-86503-71-5

Fundamental Concepts for Impedance-based Structural Health Monitoring

Roberto Mendes Finzi Neto*[1], Jose dos Reis Vieira de Moura Junior[2]

[1]Post-Graduate Program – Mechanical Engineering, Fed. University of Uberlandia, Brazil.
E-mail: finzi@ufu.br
[2]Post-Graduate Program – Modeling and Optimization, Fed. University of Catalao, Brazil.
E-mail: zereis@ufcat.edu.br

*Corresponding author

Abstract

Engineering has developed new approaches over the last 50 years in all its areas. Among the different areas, the design areas became more agile and computer-based, logistics developed new modes and interactions more Just in Time, while materials engineering revolutionized the options of possibilities in all sectors. However, the maintenance area gains special attention due to its objective to present limiting working conditions. While the production areas allow infinite productivity development, creating new processes and techniques, the maintenance area has a limit of improvement focused on the utopian zero defects. Thus, several techniques have been developed in recent decades, mainly with the advancement of computational power and integration. Techniques for assessing structural integrity have also developed, enabling the implementation of more precise techniques in the qualitative capacity of identifying a failure, as well as quantifying its severity and location. This chapter discusses the Structural Integrity Monitoring Method based on Electromechanical Impedance. This is one of the most explored techniques in recent decades by several research groups in Brazil and worldwide, having applications in aeronautics, space, oil and gas, bioengineering, civil construction, among others. In the next sections, some historical aspects of the technique are presented, as well as basic concepts and variables that influence this monitoring. Finally, a case study of the investigation of a failure simulated by machining an aluminum structure monitored in a climatic chamber is presented. As it is a chapter in which it is more concerned with the fundamentals and aspects that influence the use of the methodology, the conclusions of this case study do not cover any type of statistical modeling or based on machine learning as usually seen in scientific articles in the method. In this case, the effects of temperature on the impedance signature are focused, as well as the calculation of some damage metrics.

1. Introduction of the Impedance-based SHM

The SHM method described in this section is based on the electromechanical impedance (EMI), an ultrasonic technique that has been considered one of the most attractive nondestructive methods (NDE) for evaluating different types of structures.

The Impedance-based SHM technique (ISHM) was first proposed by Liang *et al.* (1994) and, subsequently, the method was extended by Chaudhry *et al.* (1995, 1996), Sun *et al.* (1995), Park *et al.* (1999, 2000, 2001, 2003), Moura and Steffen (2004), Peairs *et al.* (2004), Moura and Steffen (2006) and Palomino and Steffen (2009). This is an experimental technique and it is based on the piezoelectric effect, which couples the mechanical and the electrical domains.

In most applications of the ISHM, a piezoelectric transducer (PZT) is employed to interact with the monitored structure. The transducer must be bonded or embedded to the target structure and an electrical excitation will return information about the so-called "state of health" of the structure. Several types of pre/post-processing of the collected data can be employed to better evaluate the target structure. The next section describes the EMI technique in detail.

1.1. Physical principle of the EMI technique.

The EMI technique is considered to be a non-destructive method for evaluating different types of structures (NDE), as stated by Park G., *et al* [2003]. The only modification required is to bond the PZT to the target structure or to embed it (in the case a new structure is in the process of fabrication. In any case, since the transducer is very small and thin (thickness less than 1mm and less than 20mm in diameter for a disc patch), there are no major problems in incorporating these transducers into the target structure. Figure 1 presents a set of ceramic PZT, made of lead-zirconate-titanate (PZT) and produced by *MPI Ultrasonics,* and Figure 2 illustrates an aircraft aluminum window panel instrumented with six PZT patches.

Figure 1: Examples of ceramic PZT transducers. Source: www.mpi-ultrasonics.com).

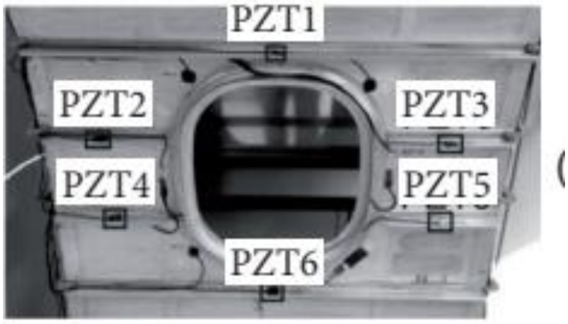


Figure 2: Aircraft aluminum window panel instrumented with six PZT patches (source: Maruo *et al*, 2015)

Independent of the shape, or even the material which the transducer is made of, the piezoelectric effect is what enables the electronic instrumentation of the target structure. This effect correlates the mechanical deformation of the transducer with a voltage at its terminals. It is possible to deform the transducer and obtain a differential voltage (direct effect) or to apply voltage and obtain mechanical deformation (converse effect). Figure 3 illustrates the direct and converse piezoelectric effects.

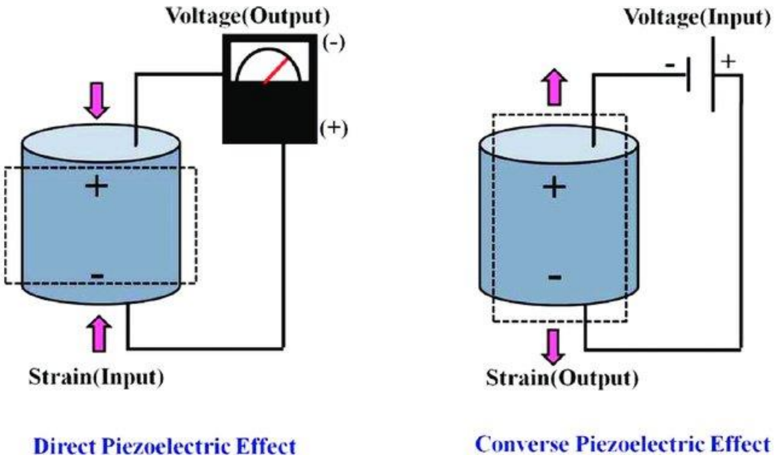


Figure 3: Direct and inverse piezoelectric effects. Source: Lakshmi *et al*, 2018.

In most cases, the PZT is fabricated to be sensitive to mechanical deformations applied to only one arbitrary direction, which can be perpendicular to the plane of bonding (or embedding) of the target structure. If the thickness of the bonding layer is considered to be not relevant, a simplified one degree of freedom (DOF) model may represent the bonded/embedded transducer. Figure 4 illustrates this 1 DOF model. The model reduces the complexity of the structure to a system represented by its mass, stiffness, and dampening. Any structural modification, like damage, will change at least one of those properties.

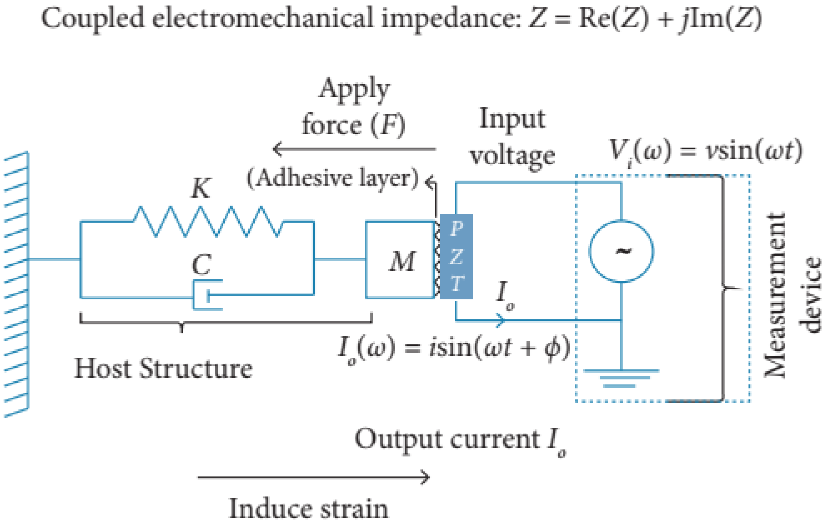

Coupled electromechanical impedance: $Z = \mathrm{Re}(Z) + j\mathrm{Im}(Z)$
Apply force (F)
Input voltage
$V_i(\omega) = v\sin(\omega t)$
(Adhesive layer)
K
C
M
PZT
I_o
Host Structure
$I_o(\omega) = i\sin(\omega t + \phi)$
Measurement device
Output current I_o
Induce strain

Figure 4: Mechatronic 1 DOF model of the coupled electromechanical transducer (source Maruo *et al* (2015)).

A sinusoidal voltage, with an arbitrary high-frequency band (tens to hundreds of kHz) and low amplitude (1V to 10V), is applied to the transducer. An equivalent strain will be applied to the coupled structure and the total circulating electrical current will be related to the transducer's electrical impedance and, also, the so-called **mechanical impedance** of the structure.

The **mechanical impedance** can be defined as a measure of how much a structure resists motion when subjected to a harmonic force. This property is also highly tied to other mechanical properties like mass, stiffness, and damping. In this case, the harmonic force is provided by the piezoelectric transducer (converse effect). So, the deformation of the bonded transducer will be directly affected by the mechanical impedance of the structure. The conclusion is that de equivalent impedance of the PZT will also be directly affected by the mechanical impedance. Liang *et al (*1994) modeled the so-called **electromechanical impedance** (EMI) of the coupled transducer as a function of frequency, as presented in equation (01).

$$Z_E(\omega)=\frac{V_i(\omega)}{I_O(\omega)}=\frac{h_a}{j\omega w_a l_a}\left[\overline{\varepsilon}^{T}_{33}(1-j\delta)-\frac{Z_S(\omega)}{Z_S(\omega)+Z_a(\omega)}d^2_{3x}\overline{Y}^{E}_{22}(1+j\eta)\right]^{-1} \qquad (01)$$

Where:

- $Z_E(\omega)$ represents the electrical admittance (inverse of electrical impedance); ω is the excitation frequency;
- $\bar{Y}^2_{11}$is the complex Young's modulus of the PZT at zero electric field;
- d^2_{3x} is the piezoelectric strain constant in the arbitrary x-direction at zero stress;
- $\varepsilon^T_{33}$is the dielectric constant at zero stress; and a is a geometric constant of the PZT;
- is the dielectric loss tangent of the piezoelectric patch;
- Z_s(W) represents the mechanical impedance of the monitored structure and any structural changes will represent new values for $Z_S(\omega)$ and, consequently, for $Z_E(\omega)$.

Usually, the frequency of the sinusoidal voltage applied to the PZT is between 20kHz and 1MHz, where the upper limit is defined by the impedance analyzer and the electrical characteristics of the PZT employed.

Measuring the impedance over a frequency band will return a waveform (impedance versus frequency) that can be used to identify the mechanical state of the target structure at the moment of that measuring. The next section will describe the way to evaluate those frequency-dependent waveforms to characterize de "state of health" of the target structure.

1.2. Impedance signatures and Damage indexes.

If external effects like temperature variations and static or dynamic loads, on the monitored structure, were not taken into account, an Impedance Signature (IS) is a bi-dimensional waveform that describes the structure's EMI over an arbitrary and limited frequency band. Instruments like the impedance analyzer HP4194A are employed to obtain de IS from any kind of PZT patch. The HP4194A can obtain an IS with 400 frequency points and with an upper frequency bound of 1GHz.

The definition of the lower and the upper bounds of the frequency band depends on the complexity of the structure and the characteristics of the damage that is subject to identification. This statement may appear very nebulous and imprecise but authors like Chaudhry *et al* (1995, 1996), Palomino *et al.* (2011), Finzi Neto *et al.* (2010), Liang *et al.*

(1994), Moura *et al.* (2004) Park *et al.* (1999, 2000, 2001, 2003), Sun *et al.* (1995) and many other confirm it. In reality, for large and complex structures, the frequency band may be defined through experimental testing. But there are a few indicators on how to proceed. Small damages, like cracks, can be better identified at higher frequencies (hundreds of kHz), and higher the frequency shorter the range of sensing of the PZT, Rabelo, *et al.* (2017).

As an example, consider the experiment proposed by Bitencourt and Steffen Jr. (2009) with the aluminum beam illustrated in Figure 5. There are two ceramic PZT patches, PZT_1 and PZT_2, bonded to that beam and a rivet inserted on the right side. The removal of the rivet was used to simulate damage on the beam. Besides the damage, this example also illustrates the attempt to "repair" the beam with the insertion of another rivet at the same position. Each PZT patch had there IS measured over arbitrary and different frequency bands. The experimental procedure is described as follows:

(1) – Measure the IS, of each PZT, with no change on the instrumented structure [BASELINE];
(2) – Remove de rivet and measure the IS of each PZT again [STATE1];
(3) – Insert a new rivet in the same position that it was removed.
(4) – Measure the IS of each PZT for the last time. [STATE 2].

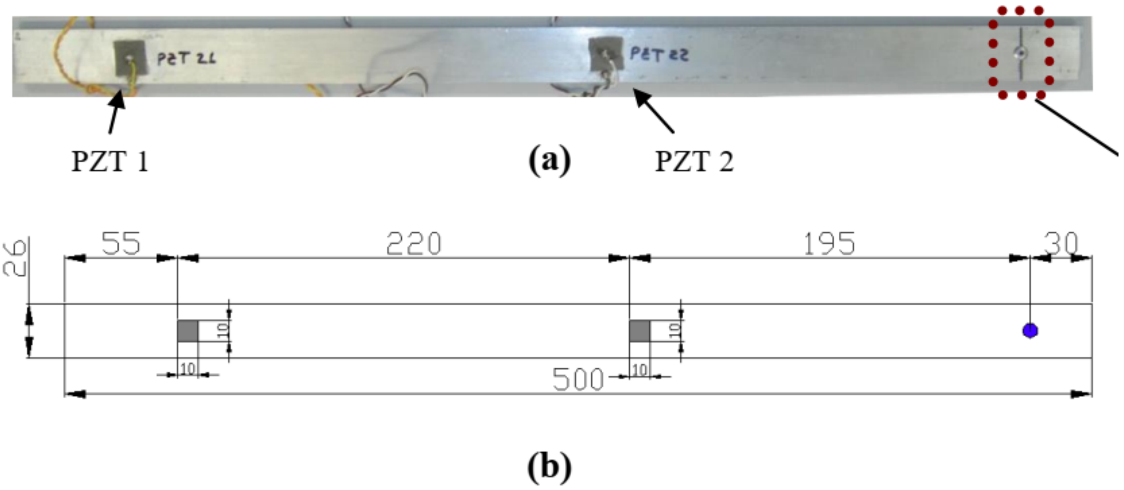


Figure 5: Aluminum beam (a) with ceramic PZT patches and a rivet used to simulate damage by its removal. The dimensions are presented at (b). Source: Bitencourt and Steffen Jr. (2009).

The collected IS for this experiment are presented in Figure 6. But, before trying to interpret the IS for the 3 states together, it is important to understand the IS by itself. The **characteristics of the quality** of an IS are presented as follows:

I. From Equation 01, it is well known that the EMI is a complex quantity (real part plus imaginary part). The imaginary part includes the reactive capacitance of the PZT and it is very influenced by temperature changes. So, to mitigate this effect, the IS are limited to only the real part of Z_E.
"Good" IS are the ones that present high and steep transitions in their values over their frequency range. This can be understood as a "reasonable" number of peaks and vales on the IS.

II. The defined frequency band should be related to the kind of damage/changes that are expected to happen on the target structure. Determining the frequency range, for a complex structure requires a set of experiments to better understand the EMI structure's behavior over the frequency. Nevertheless, studies are using mathematical models to help define the frequency range, Bhalla S., Soh (2004).

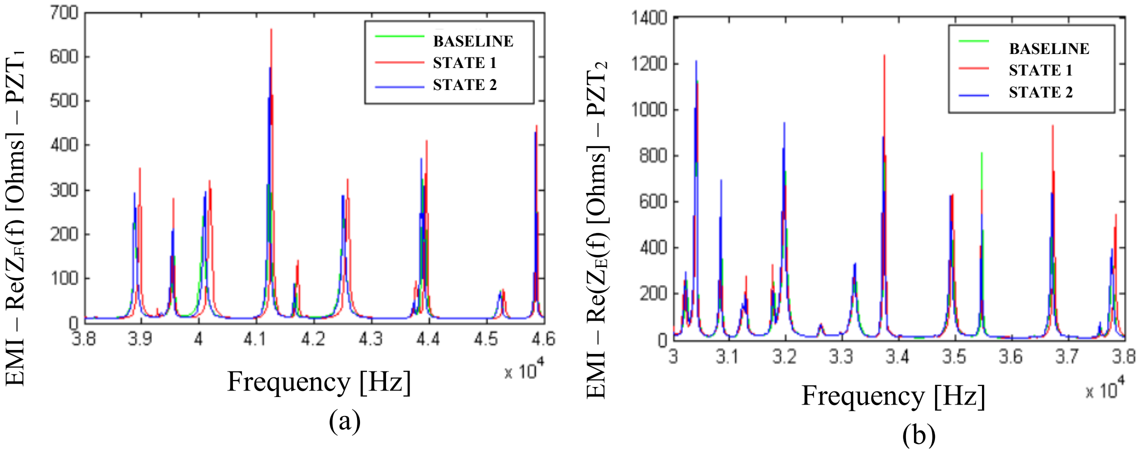


Figure 6: IS for PZT1 (a) and PZT2 (b) considering the three states [BASELINE, STATE 1, AND STATE 2]. Source: Bitencourt and Steffen Jr. (2009).

Taking into account those **characteristics of quality,** the following preliminary analysis can be done:

- All collected IS from Figure 6 are in accordance with (I) and (II). A "good" IS and only the real part of Z_E is considered;
- The differences in each of the three states, for PZT_1, can be clearly visualized. For PZT_2, this difference is more difficult to see.
- Even though a new rivet is inserted to replace that on it was removed, neither PZT presented an IS (for STATE 3) that was visually identical to their respective BASELINE. It must be understood that even a so-called "identical repair" is employed on the target structure the correspondent IS will never be the same as before de damage [Chaudhry *et al* (1995, 1996), Finzi Neto *et al.* (2010), Liang *et al.* (1994), Moura *et al.* (2004) Park *et al.* (1999, 2000, 2001, 2003), Sun *et al.* (1995)];

From the preliminary analysis, it became clear that only the visual inspection of the collected IS, in each state, is not enough to quantify the structural modifications (damage) on the aluminum beam.

A better way to analyze the IS is to mathematically quantify the difference between each state into a single number. A **damage index** is a mathematical way to quantify the difference between two IS obtained from the same bonded PZT. Palomino and Steffen Jr (2009) have evaluated several damage indexes for simulated damage on an aircraft aluminum panel. From those, the most commonly used ones are described as follows.

For every damage index described here, consider these definitions:

- $Re(Z_{1,i})$ is the array of impedance (resistance) points measured under "healthy conditions" (baseline);
- $Re(Z_{2,i})$ is the array of the real part impedance points measured in any other mechanical state than the baseline;
- $Re(\bar{Z}_1)$ mean value of the real part impedance points measured under "healthy conditions" (baseline);
- $Re(\bar{Z}_2)$ mean value of the real part impedance points measured in any other mechanical state than the baseline;
- $S_{Z_1,i}$ for a set of *k* IS collected at the baseline state, this is an array of standard deviation values, calculated with the *k* values, at each of the *n* array indexes;
- *n* is the number of impedance points.

RMSD – Root Mean Square Deviation.

$$RMSD = \sqrt{\sum_{i=1}^{n}\left(\frac{\left(Re(Z_{1,i}) - Re(Z_{2,i})\right)^2}{n}\right)} \qquad (02)$$

This is the most common damage index in the scientific literature. The RMSD number will always be positive and its magnitude shall not be influenced by the number of impedance points in the IS.

RMSD1 – Root Mean Square Deviation (alternative 1).

This is the first variant of the RMSD. Grisso (2005) has presented the RMSD1 as an alternative that is less sensitive to the amplitude of the impedance's amplitude. This variant, equation (03), presented good results on quantifying damages associated with variation in quantity and location of the impedance peaks and valleys in the IS for each state.

$$RMSD1 = \sqrt{\sum_{i=1}^{n}\left(\frac{\left(Re(Z_{1,i}) - Re(Z_{2,i})\right)^2}{Re(Z_{1,i})^2}\right)} \qquad (03)$$

RMSD2 – Root Mean Square Deviation (alternative 2).

The damage index presented in equation (04) has been used in a few studies comparing the sensibility of damage detection, Tseng *et al* (2002) and Giurgiutiu *et al* (2005). It presents a slightly different mathematical definition from the RMSD1.

$$RMSD2 = \sqrt{\frac{\sum_{i=1}^{n}\left(Re(Z_{1,i}) - Re(Z_{2,i})\right)^2}{\sum_{i=1}^{n} Re(Z_{1,i})^2}} \qquad (04)$$

RMSD3 – Root Mean Square Deviation (alternative 3).

Park *et al* provided another alternative for the original RMSD. In equation (05), the sum is outside the root mean square sign, unlike the definitions previously given. The authors claim that this version of the RMSD is more robust and less sensitive to impedance peaks variations.

$$RMSD3 = \sum_{i=1}^{n} \sqrt{\frac{\left(Re\left(Z_{1,i}\right) - Re\left(Z_{2,i}\right)\right)^2}{Re\left(Z_{1,i}\right)^2}} \tag{05}$$

RMSD4 – Root Mean Square Deviation (alternative 4).

This alternative includes the mean values of each IS analyzed. The authors claim that equation (06) is less sensitive to external effects, like temperature variations, that result in different kinds of variations in the IS.

$$RMSD4 = \sqrt{\sum_{i=1}^{n}\left(\frac{\left(\left(Re\left(Z_{1,i}\right) - Re\left(\bar{Z}_1\right)\right) - \left(Re\left(Z_{2,i}\right) - Re\left(\bar{Z}_2\right)\right)\right)^2}{n}\right)} \tag{06}$$

RMSD5 – Root Mean Square Deviation (alternative 5).

External effects, like temperature variations, are always present in industrial applications and must be mitigated with specific techniques that will be later described in this chapter. Meanwhile, the RMSD5 tries to mitigate this problem with a set of *k* IS collected at each state of interest (baseline, state1, …). Those differences, due to the external effects, will be mitigated in the damage index value.

$$RMSD5 = \sqrt{\sum_{i=1}^{n}\left(\frac{\left(\frac{Re\left(\bar{Z}_{1,i}\right) - Re\left(Z_{2,i}\right)}{S_{Z_1,i}}\right)^2}{n}\right)} \tag{07}$$

CCD – Correlation Coefficient Deviation.

A more statistical approach is presented by the CCD damage index described by Giurgiutiu (2014). It measures the linear relationship between two IS from different states and it is defined in equation (08).

$$CCD = 1 - \frac{\sum_{i=1}^{n}\left(Re\left(Z_{1,i}\right) - Re\left(\bar{Z}_1\right)\right)\left(Re\left(Z_{2,i}\right) - Re\left(\bar{Z}_2\right)\right)}{\sqrt{\sum_{i=1}^{n}\left(Re\left(Z_{1,i}\right) - Re\left(\bar{Z}_1\right)\right)^2 \sum_{i=1}^{n}\left(Re\left(Z_{2,i}\right) - Re\left(\bar{Z}_2\right)\right)^2}} \tag{08}$$

The main advantage of this damage index is that its values are always normalized, which could be helpful to verify the progression of the damage in some types of structures.

There are many others damage indexes (MAPD, ASD, M, etc.) but they are less present in the literature than those aforementioned in equations (02) to (08), Palomino (2009).

Returning to the evaluation of the IS presented in Figure 06, the RMSD damage index was used to quantify the difference between a baseline IS and the IS for STATE 1 and STATE 2, for each PZT. The following bar graphs illustrate the results.

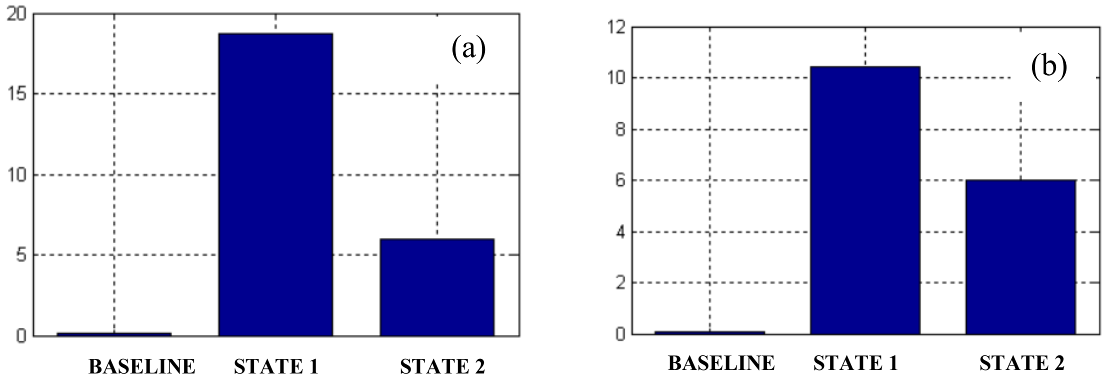


Figure 7: RMSD damage indexes for PZT1 (a) and PZT2 (b), considering the three states [BASELINE, STATE 1, AND STATE 2]. Source: Bitencourt and Steffen Jr. (2009).

What is most interesting, from the results illustrated in Figure 7, is that the RMSD values from PZT2 are lower than the ones from PZT1. Since PZT2 is nearer the location of the removed bolt (STATE 1), one would expect that those RMSD values should be higher for this PZT. Nevertheless, the difference in the band frequency applied to each PZT can help to explain these results. Another set of factors that can describe these results are the thickness of glue bonding each PZT and differences in PZT polarization due to the fabrication process.

In laboratory experiments, it is most common to use specialized equipment, like the expensive and bulky impedance analyzer HP4194A. This instrument can analyze only one connected PZT at a time but provides a rich set of information about the PZT's impedance. Industrial applications it is required a less costly solution that is capable of using many PZT transducers at the same time. The next section describes how the PZT´s impedance can be measured to build low-cost circuits capable of operating dozens of PZTs.

1.3. Modeling the electrical portion of the EMI.

Only the electrical portion of the EMI can be directly measured using conventional or more specialized instrumentation systems. It expresses a complex valued function dependent on the excitation frequency. For each corresponding frequency, the electrical part of the EMI can be represented in terms of the real and imaginary parts, or magnitude and phase in its polar form. Defining the sinusoidal excitation $v(\omega, t)$, with angular frequency $\omega = 2\pi f$, Figure 08(a) describes a PZT impedance as a simple resistor-capacitor circuit. In Figure 08(b) the waveforms of voltage and current are illustrated.

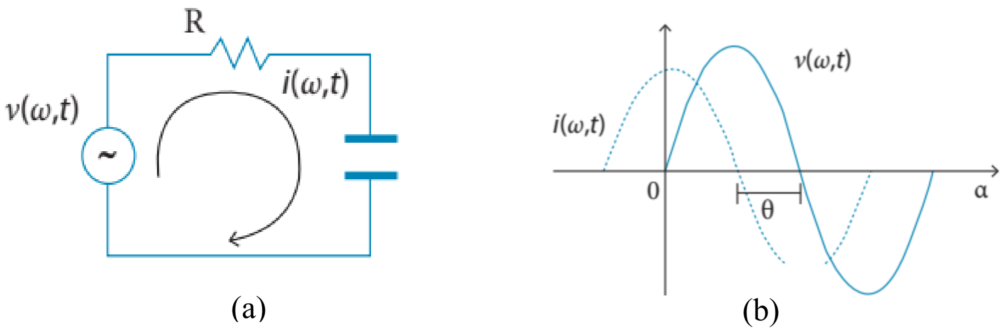


Figure 8: EMI equivalent circuit under sinusoidal excitation (a) and the waveforms of $v(\omega, t)$ and $i(\omega, t)$. Source: Maruo *et al* (2015).

Knowing that the more specialized impedance analyzers continually optimize the voltage amplitude applied to a DUT (Device Under Test) and amplitude of the resulting current is dependent on the EMI, equations (09) and (10) mathematically describe those waveforms.

$$v(\omega,t) = V(\omega)\,sin(\omega t) \tag{09}$$

$$i(\omega,t) = I(\omega)\,sin(\omega t + \theta) \tag{10}$$

Applying the well-known Ohm´s Law, the $Z_{EMI}(\omega)$ can be calculated measuring $i(\omega,t)$ and θ. Equations (11), (12) and (13) calculate the EMI and its complex/polar decomposition.

$$Z_{EMI}(\omega) = \frac{v(\omega,t)}{i(\omega,t)} = \frac{\mathcal{F}(v(\omega,t))}{\mathcal{F}(i(\omega,t))} = \frac{V(\omega)}{I(\omega)} = R(\omega) - iX_C(\omega) = |Z_{EMI}(\omega)|\angle\theta \tag{11}$$

$$Re(Z_{EMI}(\omega),\theta) = Z_{EMI}(\omega)\cos(\theta) = R(\omega) \tag{12}$$

$$Im(Z_{EMI}(\omega),\theta) = Z_{EMI}(\omega)\sin(\theta) = X_c(\omega) \tag{13}$$

For ISHM applications, it has already been stated that the resistive part of the EMI, equation (12), is more stable when temperature variations are taken into account. Nevertheless, there are cases where the reactive part of the EMI, equation (13), brings information about the PZT bonding, Grisso and Inman (2009).

1.4. The EMI measurement problem.

If you can use the impedance analyzer HP4194A your only problem is the space required to accommodate that bulky machine. Nevertheless, for industrial application, a less costly, smaller, and the lighter solution must be employed.

Through the years, the literature has presented different types of instrumentation systems capable of measuring and digitalizing the electrical impedance of a bonded PZT transducer, Finzi Neto *et al* (2011) and Baptista *et al* (2009). But, in all of them, the problem remains on measuring $i(\omega,t)$ and θ.

Measuring $i(\omega,t)$ is really simple and uses a well-known analog electronic circuit based on an operational amplifier. The electronic circuit illustrated in Figure 9 is a low-cost and simple solution. Using a precise and low value shunt resistor ($R_{shunt} \approx 100\Omega$), $i(\omega,t)$ can be calculated from $Vr(\omega,t)$ after being acquired by a data acquisition card (DAq).

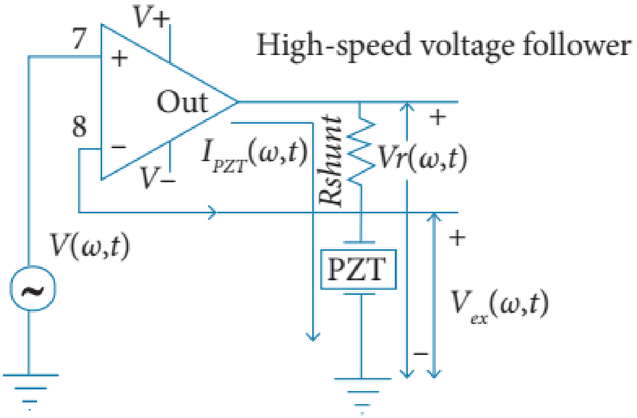


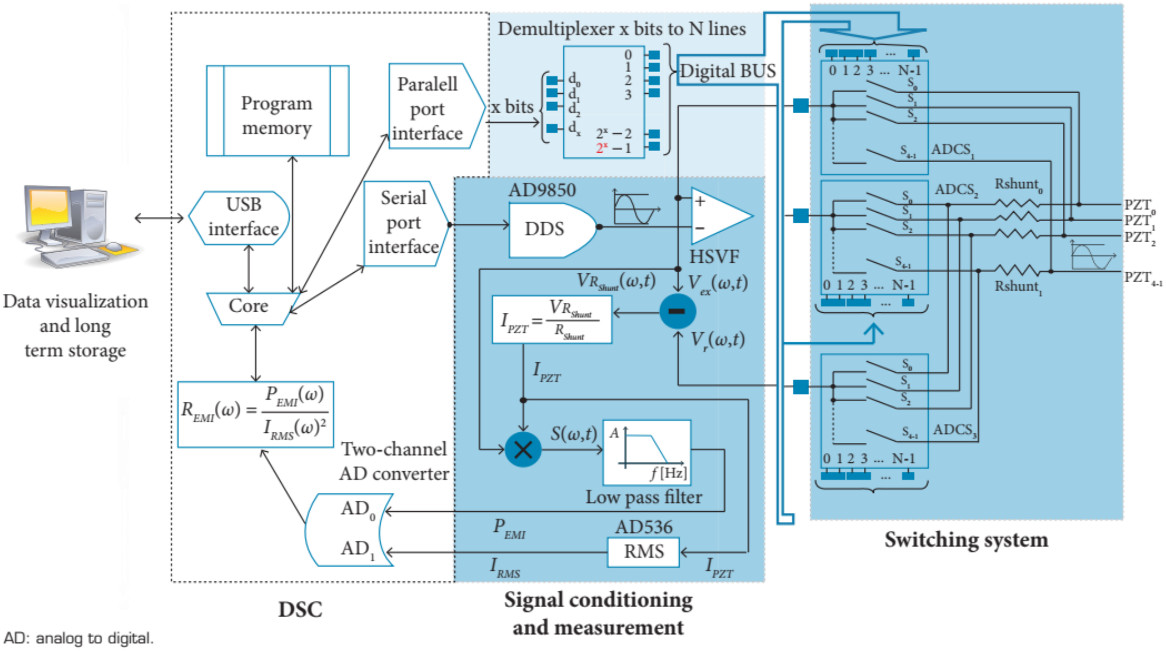
Figure 9: Electronic circuit used to measure $i(\omega,t)$. Source Maruo *et al* (2015).

From figure 9, *V(ω,t)* is any sinusoidal voltage source that can vary its frequency over an arbitrary frequency band. In most cases, the same DAq used to acquire *Vr(ω,t)* can generate *V(ω,t)*, Baptista (2009).

After acquiring *V(ω,t)* and *Vr(ω,t)*, it is required to transpose both of them to the frequency domain. A Fast Fourier Transform (FFT) can be easily applied to *V(ω,t)* and *Vr(ω,t)*, allowing for the use of equations (11) and (12) to calculate *R(ω)*.

Even though a DAq associated with one or more analogic circuits is less costly than the HP4194A, the ISHM will always be limited to the data acquisition rate of that card. There is some controversy about the use of the Nyquist Theorem as the time-domain requisite to process a time-domain signal in the frequency domain, Finzi Neto (2010) and Baptista (2009). The problem that arises is related to how accurate is θ, from equations (11) and (12), calculated near the upper limit of the Nyquist's frequency signal. Experimental results presented and discussed on the works of Finzi Neto *et al* (2010), Maruo *et al* (2015), Martins *et al* (2013) and Tsuruta *et al* (2017) shows and or discuss the deterioration of *R(ω)* near the upper limit of the signal's data acquisition frequency. Of course, there are plausible arguments that the so-called deterioration of *R(ω)* is neglectable, Baptista (2011).

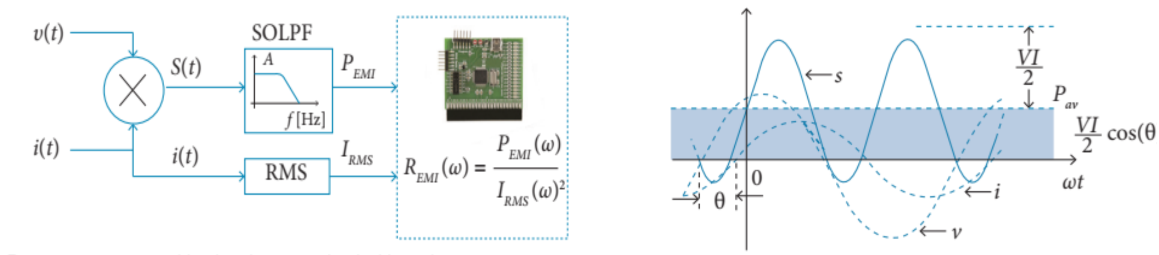
There are other architectures of electronic circuits where DAq's acquisition frequency and FFTs are no longer a limiting factor for ISHM applications. Maruo *et al* (2015) propose an architecture based on the digital signal controller (DSC) and specialized circuits for analog frequency processing. Beyond that, their architecture can monitor a large number of PZTs using analog multiplexed electronics. Figure 10 illustrates the proposed hardware.



AD: analog to digital.

Figure 10: Architecture of a multiplexed sensor array for ISHM applications. Source: Maruo *et al* (2015).

No DAq is employed to acquire *Vr(ω,t)* and or to generate *V(ω,t)*. The sinusoidal signal is generated by an integrated circuit, the AD9850, which is digitally programmed by the DSC. To avoid the use of temporal signals, the authors employed the dedicated integrated circuit AD536, from Analog Devices, to generate voltages proportional to the Root Mean Square (RMS) value of each sinusoidal waveform. To be able to calculate *R(ω)*, the author proposes to analogically calculate the mean power drained by the PZT using the proposed topology in Figure 11. The dedicated integrated circuit, AD633, multiply *Vr(ω,t)* and *i(ω,t)* to produce an analog waveform representing the apparent power, *S(ω,t)* consumed by the PZT. A Second-Order Low Pass Filter (SOLPF) is used to separate the mean power *P(ω)*.



P_{EMI}: power consumed by the electromechanical impedance;
R_{EMI}: resistive part of the electromechanical impedance; A: amplitude.

V: amplitude of waveform $v(\omega,t)$; I: amplitude of waveform $i(w,t)$;
P_{av}: mean value of $s(\omega,t) = P(\omega)$.

Figure 11: Topology to obtain the mean power $P(\omega)$ (a). Equivalent waveforms (b). Source: Maruo *et al* (2015).

Finally, the authors state that $R(\omega)$ is easily calculated from equation (14).

$$R_\omega = \frac{P_\omega}{I_{RMS}(\omega)^2} \tag{14}$$

Since $P(\omega)$ and $I_{RMS}(\omega)$ ideally have little to no alternating component, the authors states that an acquisition frequency as low as 1000 samples per second may be used to rapidly acquire PZT responses. The only operational limitation is related to the frequency band response of the integrated circuits AD633, AD536 and AD9850.

There are lots of other so-called low-cost topologies for the IEM measurement problem. It is up to the reader to choose the one that best suits the intended industrial application.

After collecting the IS for each state (baseline, state 1, …) it is important to pre-process each IS to identify and separate the influences of external effects, like temperature variations and static loading, from the real information in each IS. The next section will describe how these effects modify the IS and a few ways to mitigate this problem.

1.5. Influence of external effects on the IS.

When a new monitoring system is proposed it is necessary to evaluate the possibility of external influences deteriorating the signals collected from the transducers. External influences like temperature variations (TR), magnetic interference (MI), electromagnetic interferences (EMI), radio frequency interferences (RFI), mechanical static loads (MSL), mechanical dynamic loads (MDL), and an ionic environment (IEnv) may contaminate the electrical signal provided by the transducer or may alter the mechanical properties of the monitored structure. In any case, there are several ways to mitigate every external effect that is mentioned here.

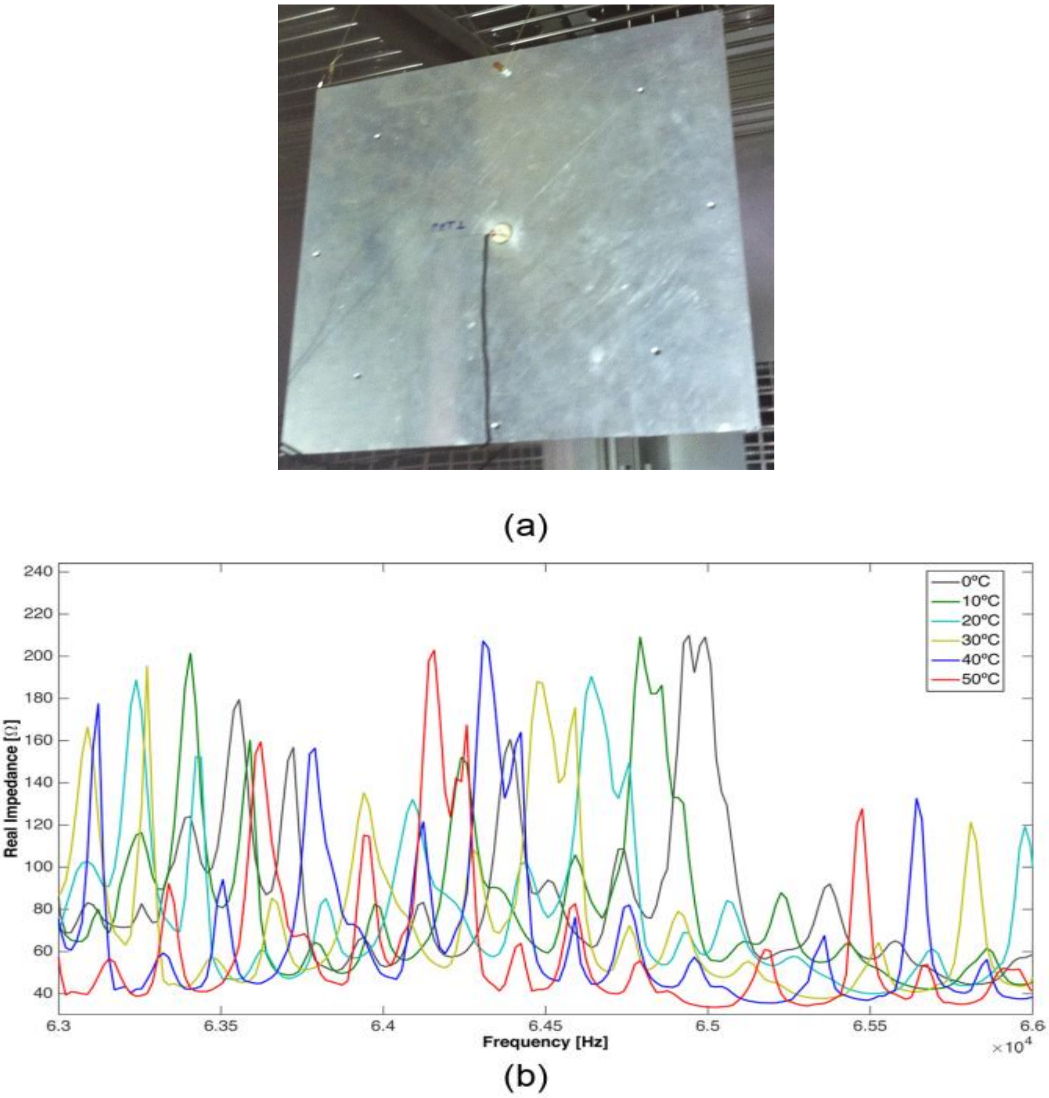
Palomino *et al* (2012) examined the influence of electromagnetic radiation, temperature and pressure variations, and the ionic environment under laboratory conditions. In this context, the major concern was to determine if the impedance responses are affected by these influences. In addition, the sensitivity of the method concerning the shape of the PZT patches was also evaluated. For this aim, two shapes of piezoelectric patches of the same size, namely circular and squared, have been tested in the laboratory. They were bonded to two different types of structures, namely a plate and a beam so that the impedance response was measured both for pristine and damaged conditions. Similar results were obtained for the two shapes of PZT patches tested. The results are summarized in Table 1, in which it can be observed that temperature is a major environmental issue in the context of ISHM.

Table 1 – Sensor shape and environmental influences. Source: Palomino *et al* (2012).

Sensor Geometry		Ionic Environment	
Squared PZT Patch	Circular PZT Patch	Without shielded PZT patch	With shielded PZT patch
=		YES	NO
Electromagnetic Radiation	Temperature		Pressure
NO	Low	High	NO
	NO	YES	

The temperature influence brings the most undesirable effect for industrial applications: **false positives** of damage detection**.** Under temperature variations changes in the stiffness and dampening, on the monitored structure, will be presented on the IS. Therefore, IS collected in different temperatures will be so different that any damage index would indicate severe damage on the monitored structure.

Before devising ways to mitigate the temperature effect problem, it is important to understand how the IS is altered due to those temperature variations. Consider the experiment performed by Rabelo *et al* (2017b). The authors have bonded a PZT in the center of an aluminum 2024-T3 plate measuring 305mmx305mmx2mm and used an environmental controlled chamber, ESPEC EPL-4H, to control the temperature, Figure 12(a). The authors varied the temperature inside the chamber from 0°C to 50°C, in steps of 10°C. At each temperature step, an IS was collected. All the IS, at each temperature step, are presented in Figure 12(b).


(a)

(b)

Figure 12: Temperature variation effects on impedance signatures: (a) instrumented Aluminium plate of 305 mm x 305mm x 3mm and (b) impedance signals shifted with temperature changes. Source: Rabelo (2017b)

The authors stated that the frequency band, 63kH to 66kHz, as defined by trial and error, looked for the already mentioned **characteristics of quality**. By analyzing Figure 12(b), the effects of temperature can be observed predominantly as horizontal shifts due to changes in the resonance frequencies of the system. Vertical shifts can also be seen, as well as changes in some peak amplitudes. The more temperatures get higher, the more shift horizontally to the left and vertically down can be seen. Analogous thinking can be done when temperatures get lower.

There are several ways to mitigate the temperature effect on the IS. Rabelo *et al* proposed a method based on horizontal frequency shifts to mitigate the most predominant effect of temperature variations. Figure 13 presents the compensated IS. Although the horizontal shifts were compensated, differences in peak values are still visible. These differences will limit the sensibility of the IEM method and the so-called **statistical threshold of detection** is proposed by the authors.

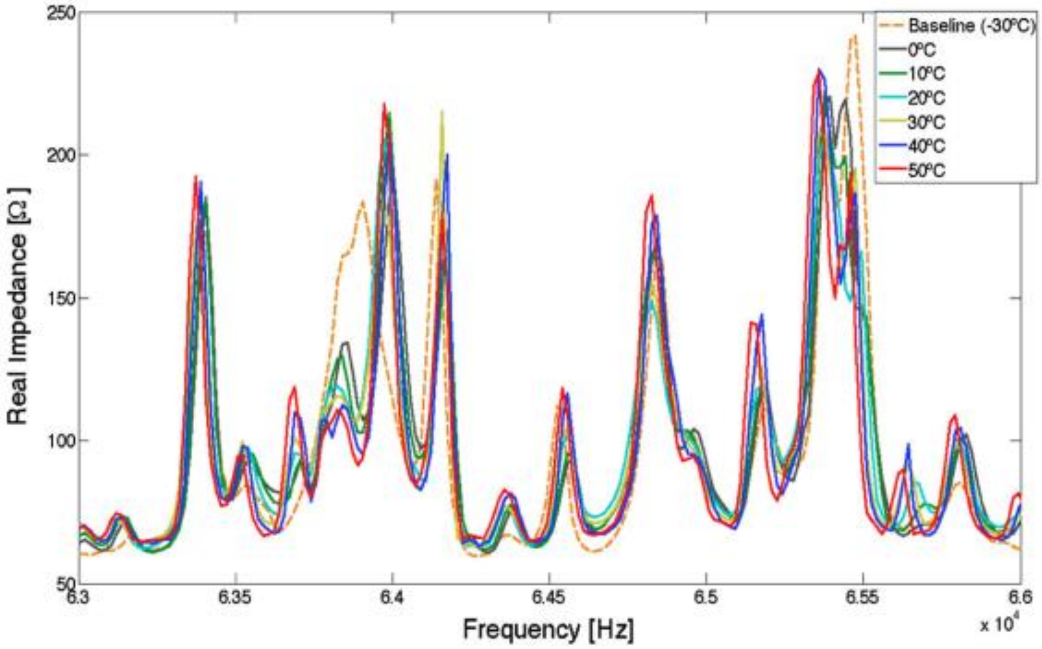


Figure 13: Impedance Signatures, from Figure 12, after the temperature compensation algorithm was applied. Source: Rabelo *et al* (2017b).

The fundaments behind the statistical process control allow for establishing limits in a control chart so that a threshold can be established using the upper control limit. These limits can be defined so that 95.45 or 99.73% of data from a normally distributed population remains if these control limits are established as expressed in equation (15), where x is the sample mean and s is the sample standard deviation.

$$\begin{gathered}\bar{x} \pm 2s - \mathit{for}\ 95{,}45\%\ \mathit{of\ confidence} \\ \bar{x} \pm 3s\ - \mathit{for}\ 99{,}73\%\ \mathit{of\ confidence}\end{gathered} \qquad (15)$$

From equation (15), it is the upper limit that must be considered to identify the so-called **threshold of damage detection** with a certain level of confidence. The authors continue to the argument that the sampled mean and sampled standard deviation are inferences from the population parameters (i.e., unknown values). Therefore, a more robust methodology should be proposed by using the upper limits of the confidence intervals for the population mean and standard deviation according to equations. (16) and (17), respectively.

$$\left[\bar{x} - \frac{st_{v;\alpha/_2}}{\sqrt{N}} \leq \mu_x \leq \bar{x} + \frac{st_{v;\alpha/_2}}{\sqrt{N}}\right] v = N - 1 \qquad (16)$$

$$\left[\frac{vs^2}{x^2_{v;\alpha/_2}} \leq \sigma^2_x \leq \frac{vs^2}{x^2_{v;1-\alpha/_2}} v = N - 1\right] \qquad (17)$$

where N is the sample size, μ_x and σ_x^2 are the population mean and variance, respectively, x and s^2 are the sample mean and variance, respectively, $t_{v;\alpha/2}$ has a *Student t* distribution with v degrees of freedom, α is the significance level and $x^2_{v;\alpha/2}$ has a Chi-Square distribution.

Hence, the upper limit of the confidence intervals was used and the threshold for each PZT transducer was determined according to equation (18).

$$PZT_{threashold} = \mu_{x\,max} + 3\sigma_{x\,max} \tag{18}$$

Where $\mu_{x\,max}$ is the upper limit for the population mean and $\sigma_{x\,max}$ is the upper limit for the population standard deviation, both obtained by choosing 5% of significance level α.

In practical terms, equation (18) will be applied to the damage indexes calculated for each temperature compensated IS, at the central temperature. Those sets of calculated damage indexes will have a mean value $\mu_{x\,max}$ a standard deviation of $\sigma_{x\,max}$. The threshold, a starting damage index value at which damage has the confidence of 99,73%, will be calculated with equation (18). Rabelo *et al* (2017b) present a set of experiments to prove the efficiency of the temperature effect compensation method and the threshold calculation. A more in-depth analysis may be found in their paper.

1.6. Final remarks

ISHM has been used in different industries over the years. The aircraft industry has been the pioneer. In Brazil, EMBRAER and many federal Universities have been developing new hardware and processing techniques for real-time and in-service applications.

The civil construction industries are following the same steps. Research works like the one developed by Silva, R.N *et al* (2020) are proving that ISHM is a low-cost alternative for monitoring aging and in-service concrete-based structures.

Several new alternatives have emerged to deal with the temperature compensation problem. Freitas et al (2021) propose a neuro-fuzzy model and Ferreira de Rezende et al (2020) propose a deep learning approach with convolutional neural networks. In this last approach, it is proposed that the temperature compensation in the pre-processing is not carried out like the other approaches. Thus, the identification and classification model must be able to understand different types of signatures as a baseline, as well as different possibilities for the same damage signature.

Effects like dynamic loading are not a factor to disregard ISHM, anymore. Research works like the ones developed by Rabelo *et al* (2017a and 2017b) and Cavalini, A. A. *et al* (2014) proved that there are efficient ways to mitigate these influences for in-service structure monitoring.

Finally, as an indication for future and innovative applications, Menegaz *et al* (2019) studied the application of ISHM in the detection of mammary inclusions (tumors). Furthermore, some mathematical-statistical approaches have emerged to complement the damage location capability as described by Golçalves et al (2021) who use geostatistical kriging techniques associated with the ISHM.

There are lots of paths to follow with the research in ISHM. The industry will decide which ones will become final products. But it is up to the researchers to present these many alternatives.

1.7. Experimental Example

To illustrate the procedure of the Impedance-based SHM Methodology and its main aspects, an experimental test was proposed as follows.

In this experiment, the specimens used were four aluminum beam structures monitored inside a Platinous EPL-4H series climatic chamber for temperature and humidity control, as shown in Figure 14. This chamber is installed in the Structural Mechanics Laboratory (LMEst) of the School of Mechanical Engineering (FEMEC) at the Federal University of Uberlandia (UFU).

In this experiment, aluminum beams measuring 500 mm in length, 38 mm in width, and 3.2 mm in thickness were used. In each of them, a PZT patch measuring 1 mm thick and 20 mm in diameter was glued at 100 mm from the edge of the structure.

Figure 14: Platinous EPL-4H series climatic chamber used in this test.

First, after selecting the specimens, the PZT patches are bonded to each one. In this case study, epoxy glues with temperature tolerance up to 60-70 °C were used. Right after the patches were bonded to the structure, the measuring terminals were soldered. In this PZT patch type, it was necessary to use copper tape on the lower surface to make a better connection. This can be observed in Figure 15.

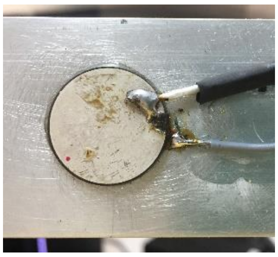


Figure 15: Specimen and PZT patch.

In this test, aluminum beams 500 mm in length, 38 mm in width, and 3.2 mm in thickness were used. In each of them, a PZT patch measuring 1 mm thick and 20 mm in diameter was bonded at 100 mm from the edge of the structure. Also, ABS plastic printing supports were designed to position the structures inside the chamber. Such structures have rounded bases to reduce the interference of the chamber floor with the structures and made it possible to adapt connectors to the samples. This type of connector embedded in the support structures allowed for less interference and noise addition to the cables during several removals of specimens from the chamber. Figure 16 illustrates one specimen, the support structure with the connector at the right end as well the PZT patch and damage position.

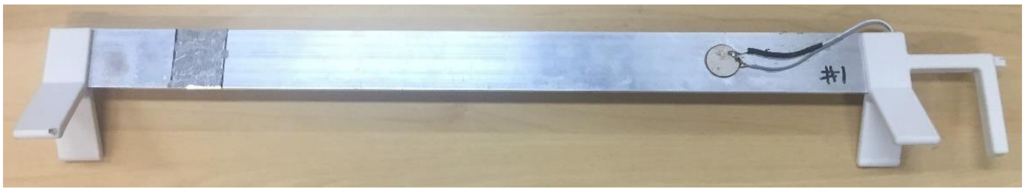


Figure 16: Specimen used in the test illustrating the PZT patch and damage position.

The simulated damage that was inserted into the structures was superficial and caused by grinding in a defined region. This machined region was 30mm wide at 70mm from the opposite end of the PZT patch. Thus, nine levels of damage were performed throughout the experiment, considering two reference levels and seven levels of gradual thickness removal. Throughout the experiment, only specimen 4 was kept without any changes for monitoring and controlling the process. However, specimen 4 was removed from the chamber together with the others at each machining process.

Figure 17 shows in better detail the connector attached to the structural support printed in ABS. This type of connector facilitates the process of removing and reinserting specimens for machining.

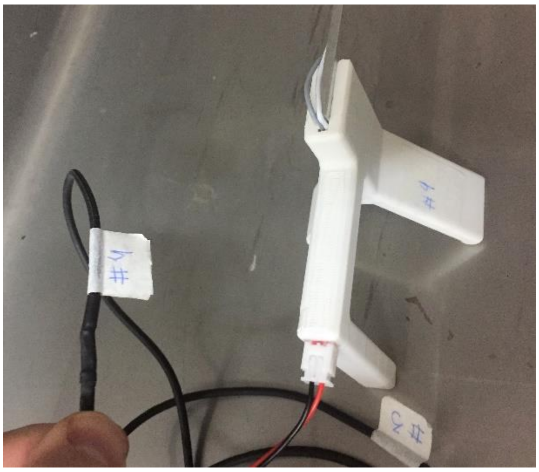


Figure 17: Specimen and connector.

In Figure 18, the test specimens are presented in the bi-supported condition. The support structures were designed to keep the specimens in this position because, during the thickness removal process, especially the terminal stages, the loss of structural stiffness could cause some kind of flexion at the two extreme support points.

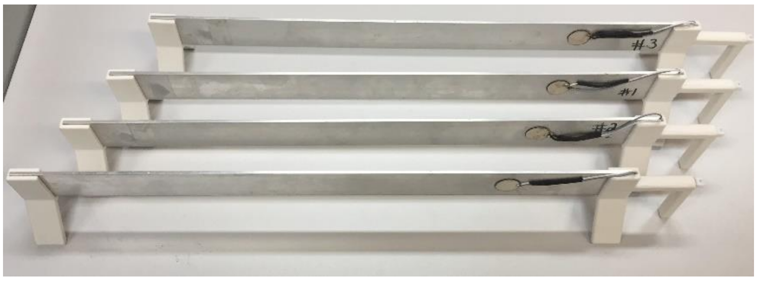


Figure 18: All specimens used in the test.

Since the temperature has a significant impact on the impedance monitoring process, this factor was controlled in this case study through the climate chamber. Figure 19 illustrates the specimens positioned inside the chamber. It is important to note that the specimens must be positioned inside the chamber in such a way that the effects of convection flows (blowing) on some specimens are reduced, causing fluttering or measurement noise.

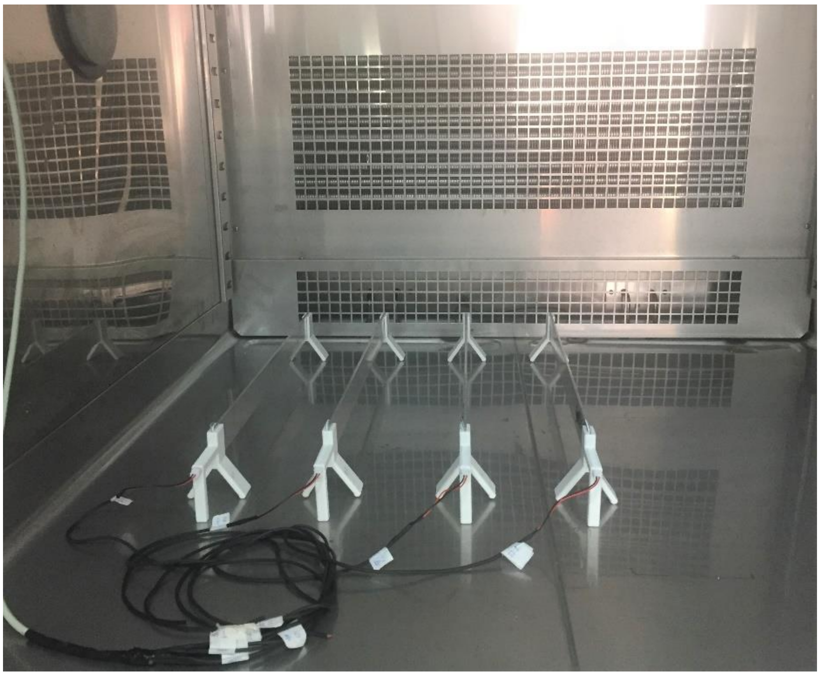


Figure 19: Specimens inside the chamber.

All specimens were monitored in ascending cycles of 3 ºC for a total of 11 temperature cycles. The impedance analyzer used for the acquisition and storage of signatures was connected externally to the chamber.

The temperature range used was from 10 to 40 ºC with levels of 3 ºC. As this level variation can be considered small for control and thermal stability purposes, a procedure was adopted. After the acquisition of each cycle of impedance signatures, the chamber increased its temperature by 3 ºC and remained for 30 minutes to stabilize the internal temperature and obtain the thermal balance between specimens and the environment.

For each configuration: specimen, temperature, and damage condition, 30 samples of impedance signatures were taken for process repeatability. Thus, in 11 temperature cycles, 9 damage levels, 4 specimens, and 30 repetitions, 11880 impedance signatures were acquired.

Considering that the process of reducing thicknesses by grinding is a manual task, two response variables of each specimen were selected for damage monitoring. The first response considered was the mass, considering the loss of mass about the previous state of integrity. In this case, a scale with two decimal places of the gram of precision was used. In addition, eight measurements were taken to obtain the average.

The second response variable was the thickness in the ground region, considering the loss of thickness from the previous condition. In this second answer, a micrometer with a resolution of 0.01mm was used and the average of 10 random measurements of thickness in the machined region was recorded. Figure 20 shows the ground region of one of the specimens during the thickness loss process.

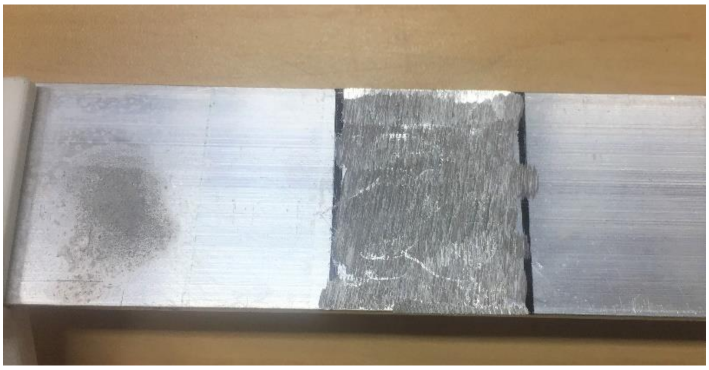


Figure 20: Specimen and ground position.

Next, Tables 1 and 2 are presented, which correspond, respectively, to the response variables of the experiment, with the results of the measurements of mass and thickness for each specimen.

Table 1: Results of beam mass losses (in *g*).

Condition	Specimen #1	Specimen #2	Specimen #3	Specimen #4
Baseline	191.36	192.42	192.86	195.89
Damage #1	191.2	192.27	192.69	-
Damage #2	191.16	192.02	192.58	-
Damage #3	190.95	191.82	192.39	-
Damage #4	190.41	191.48	192.02	195.87
Damage #5	189.96	190.54	191.33	-
Damage #6	189.16	189.42	189.78	-
Damage #7	187.44	188.57	188.37	195.83

Table 2: Results of beam thickness losses (in *mm*).

Condition	Specimen #1	Specimen #2	Specimen #3	Specimen #4
Baseline	3.17	3.17	3.19	3.18
Damage #1	3.09	3.11	3.09	-
Damage #2	3.01	2.96	2.99	-
Damage #3	2.97	2.93	2.95	-
Damage #4	2.85	2.85	2.89	3.17
Damage #5	2.72	2.49	2.63	-
Damage #6	2.44	2.23	2.09	-
Damage #7	1.93	1.92	1.83	3.18

As can be seen in Table 2, the damage inserted in the three specimens did not exceed 50% of the thickness. Furthermore, the evolution between damage levels occurred more severely in the last two stages. This variation has a specific purpose of evaluating greater variations in damage models the greater the damage severities.

Another observation can be seen by comparing Tables 1 and 2 in specimen column #4. From this comparison, it can be said that based on the case study in question and on the measuring devices used, the measurement of thickness loss can be a more rigorous measurement criterion.

Considering the impedance-based structural integrity monitoring of the aluminum beams, Figure 21 presents the impedance signatures for specimen #1. Since this case study focuses on observing the basic principles of the technique, results are presented for this specimen only, but similar results were obtained for specimens #2 and #3. It is believed that repetitive presentation of other specimens would increase the volume of information unnecessarily to present the technique.

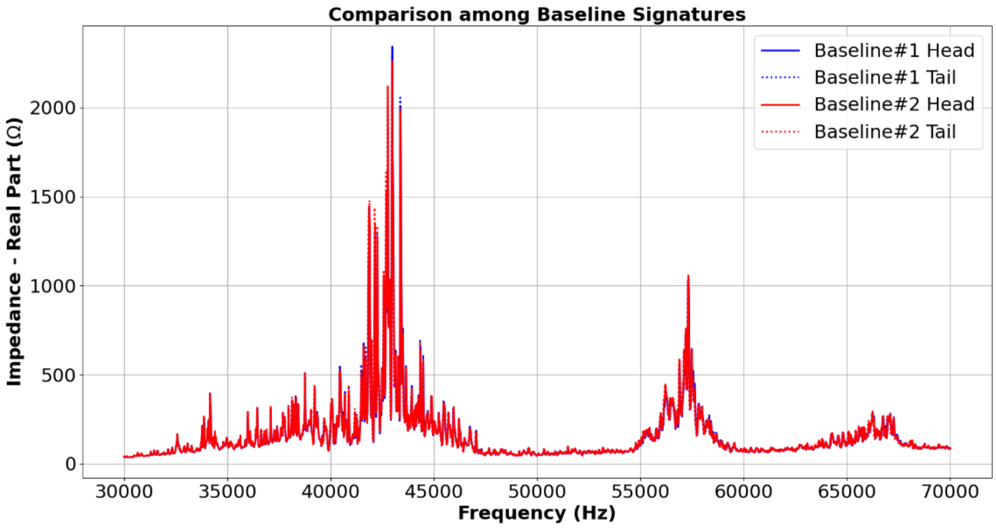


Figure 21: Specimen #1 and repetitive baseline signatures.

Figure 21 illustrates four impedance signatures from the experiment. There are presented the first and last impedance signature measured (head and tail) from the first temperature for the first and second group of baselines (Baseline #1 and #2). Explaining the objective of two sets of baseline measurements, a first group was measured, completing 30 repetitions for each of the 11 temperature levels, corresponding to 330 series. Then, all of the four beams were removed from the climate chamber, and it was transported along some buildings in the university to increase potential noise factors in a new measurement. Then, a few minutes later, the four structures were reinserted into the climate chamber and measured another set of baseline with 330 signatures.

According to Figure 21, it is possible to verify the repeatability of the signatures even with the potential insertion of small bending, small strains in the beams, and wired connections variations. This is one of the characteristics that make the use of the method applicable and reliable in real monitoring cases.

Figure 22, as described throughout the chapter, shows the temperature variation in the impedance signature under the same undamaged condition. The 11 temperature levels are not illustrated to facilitate an understanding of the phenomenon. However, the intermediate values between the ranges illustrated in the graph, present similar behavior.

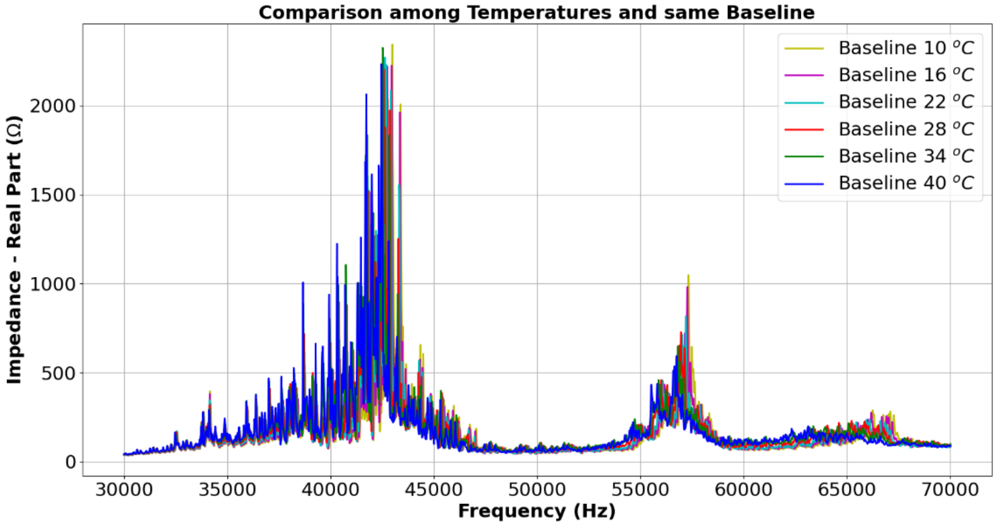


Figure 22: Specimen #1 and temperature changes for Baseline #1.

Since the frequency range in this experiment is wide, with 4000 sampling points, the effect of temperature in this figure may not be clear. In Figure 23, only the second region with the highest frequency peaks, around 54-60 kHz, is presented.

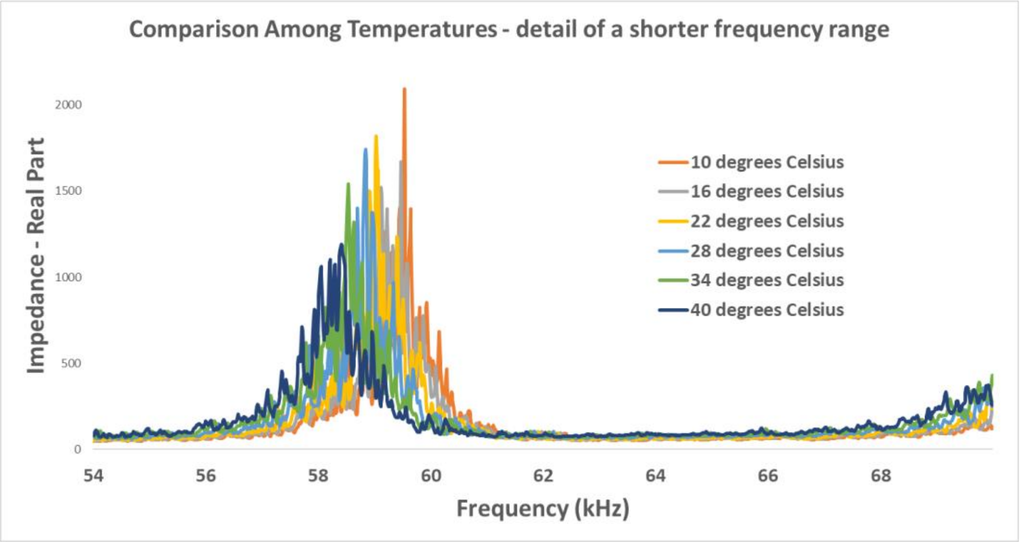


Figure 23: Specimen #1 and temperature changes for Baseline #1- shorter frequency range.

According to the detail in this frequency region, the displacement behavior in the frequency of the impedance peaks is more evident due to the temperature effect. Since the damage metrics work with the comparative calculation between the baseline signal and the monitoring signature, false positives and negatives can occur if there is no such concern with temperature. This can get worse if the threshold on the damage metric is small and measurements take place in very sudden temperature changes.

Figure 24 presents the final objective of monitoring the integrity of a structure, which is to visualize the variations in the impedance curve that are generated due to the process of loss of thickness located in the beam.

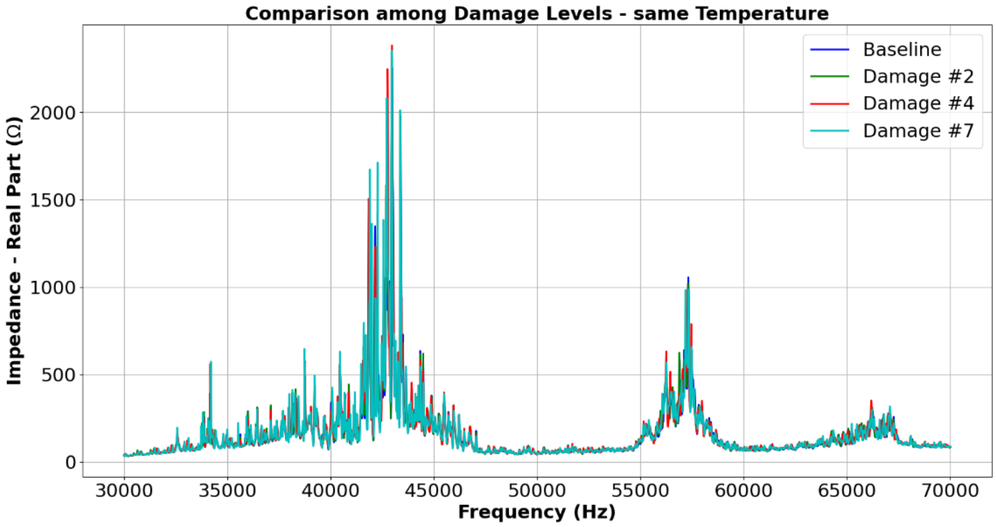


Figure 24: Specimen #1 in the same temperature but different damage conditions.

Again, only a few damage conditions are presented to support understanding of the phenomenon. However, it is possible to see from the image that the variations introduced by the damage are subtle and in specific regions.

Then, for metric calculations, only the set of 30 signatures from Baseline #1 was used. Figure 25 shows the RMSD damage metric for some individual subscriptions under the listed conditions.

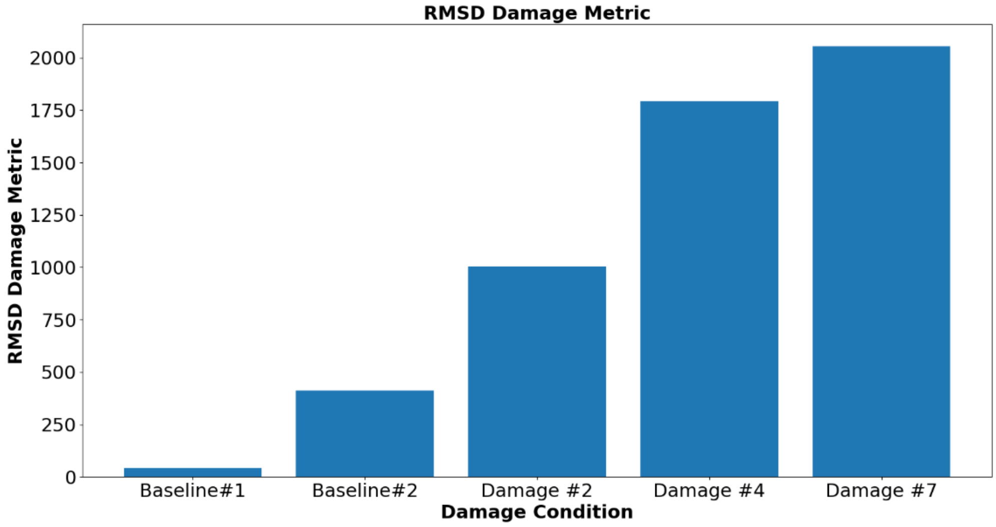


Figure 25: Specimen #1 and some RMSD damage metrics.

According to the figure, it is possible to see a non-null value for Baseline #1 because the comparative calculation of each signature is performed with the median of the 30 signatures of Baseline #1. So, while Baseline #1's damage metric value represents the deviation from the first measurement with its 30-repeat set, Baseline #2 presents the deviation between a measurement from the second reference group and the median of the first reference group.

Figure 26 presents the CCD damage metric. In it, it is possible to see better robustness making the Baseline #1 value practically null. In cases where a greater influence of temperature is perceived, this damage metric has been preferred over other damage metrics.

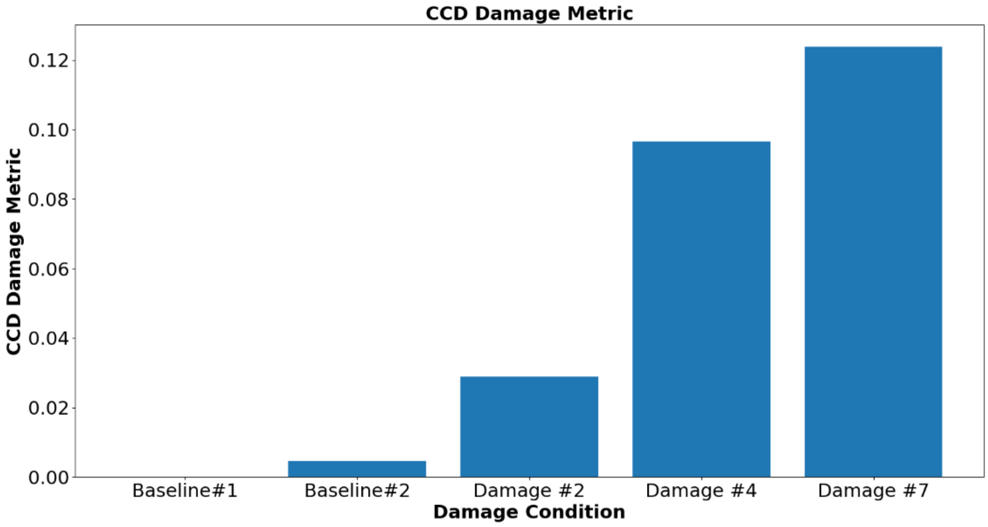


Figure 26: Specimen #1 and some CCD damage metrics.

It is important to note that the threshold value in fault detection needs to be considered around a safety margin. Likewise, the survey of a baseline signature must consider several measurement cycles, not just one with several repetitions. This expands the generalizability. Still, as seen, it is necessary to have baseline measurements at different temperature levels so that it is possible to apply adjustment techniques and make the fault identification more accurate. Several temperature compensation techniques have been studied over the last decades and recently some techniques based on Deep Learning have successfully managed to process data without temperature adjustment. However, even in these cases, the collection of baselines at different temperature levels is of paramount importance.

1.8. References

Baptista FG, Filho JV (2009) A new impedance measurement system for pzt-based structural health monitoring. *IEEE Instrumentation and Measuring* 58(10):3602-3608. DOI: 10.1109/TIM.2009.2018693.

Baptista, F. G.; Vieira Filho, J.; Inman, D. J. . Time-domain analysis of piezoelectric impedance-based structural health monitoring using multilevel wavelet decomposition. *Mechanical Systems And Signal Processing*, v. 25, p. 1550-1558, 2011.

Bhalla S., Soh, C. K. Electromechanical Impedance Modeling for Adhesively Bonded Piezo-Transducers. *Journal of Intelligent Materials*, 15(12):955-972, 2004. Horizonte Científico, Vol. 3, Nº 2, 2009.

Bitencourt, TF, Steffen Jr. Monitoramento Da Integridade Estrutural De Aeronaves, UFU, 2009.

Cavalini, A. A.; Finzi Neto, R. M.; Steffen Jr, V. Impedance-based fault detection methodology for rotating machines. *Structural Health Monitoring*, v. 1, p. 1, 2014.

Chaudhry, Z., Joseph, T., Sun, F. and Rogers, C. (1995), "Local-Area Health Monitoring of Aircraft via Piezoelectric Actuator/Sensor Patches", *Proceedings of the Smart Structures and Integrated Systems 1995*, San Diego-CA, USA, March.

Chaudhry, Z., Lalande, F., Ganino, A., Rogers, C., 1996, "Monitoring the Integrity of Composite Patch Structural Repair via Piezoelectric Actuators/Sensors", AIAA-1996-1074-CP

Finzi Neto, R. M.; Steffen, V.; Rade, D. A.; Gallo, C. A.; Palomino, L. V. . A low-cost electromechanical impedance-based SHM architecture for multiplexed piezoceramic actuators. *Structural Health Monitoring*, v. 1, p. 1-1, 2010.

Ferreira, de Rezende, S. W. F.; Moura Jr, J. R. V.; Finzi Neto, R. M.; Gallo, C. A.; Steffen Jr, V. . Convolutional neural network and impedance-based SHM applied to damage detection. *Engineering Research Express*, v. 1, p. 49-54, 2020.

Freitas, F.A; Jafelice, R.M; Silva, J.W; Rabelo, D.S; Nomelini, Q.S.S; Moura Jr, J.R.V; Gallo, C.A; Cunha, M.J; Ramos, J.E. A new data normalization approach applied to the electromechanical impedance method using adaptive neuro-fuzzy inference system. *Journal of the Brazilian Society of Mechanical Sciences and Engineering,* 43:475, 2021.

Gonçalves, D.R; Moura Jr., J.R.V; Pereira, P.E.C.; Mendes, M.V.A; Diniz-Pinto, H.S. Indicator kriging for damage position prediction by the use of electromechanical impedance-based structural health monitoring. *Comptes Rendus. Mécanique*, v. 349, p. 225-240, 2021.

Grisso, B. L., 2005, "Tailoring the Impedance-Based Structural Health Monitoring Technique to Composites and Wireless Systems", 13th March 2005-CMISS.

Grisso, B. L., Inman, D. J. "Temperature corrected sensor diagnostics for impedance-based SHM". *Journal of Sound and Vibration*. Ed. Elsevier. 2009.

Giurgiutiu, V.; Zagrai, A. Damage Detection in Thin Plates and Aerospace Structure with the Electro-Mechanical Impedance Method. *Structural Health Monitoring*. V. 4(2), p. 99-118, 2005.

Giurgiutiu, V. (2014), Structural health monitoring with piezoelectric wafer active sensors, Elsevier Inc, Academic Press, 2nd edition.

Lakshmi, M; Unnikrishnan, L. Kumar, S. Piezoelectric Polymer Composites for Energy Harvesting Applications: A Systematic Review. *Macromolecular Materials and Engineering* 304(1), 2013.

Liang, C., Sun, F.P. and Rogers, C.A. (1994), "Coupled Electromechanical Analysis of Adaptive Material Systems – Determination of the Actuator Power Consumption and System Energy Transfer" *J. Intell. Mater. Syst. Struct.*, **5**, 12-20.

Martins, L. G. A.; Finzi Neto, R. M.; Palomino, L. V.; Steffen Jr, V.; Rade, D. A. . Architecture of a remote impedance-based structural health monitoring system aiming at aircraft. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, v. XXXIV, p. 200-393, 2013.

Maruo, I. I.; Giachero, G.; Steffen Jr, V.; Finzi Neto, R. M.. Electromechanical Impedance - Based Structural Health Monitoring Instrumentation System Applied to Aircraft Structures and Employing a Multiplexed Sensor Array. *Journal of Aerospace Technology and Management* (Online), v. 7, p. 294-306, 2015.

Menegaz, G. L.; Tsuruta, K. M.; Finzi Neto, R. M.; Steffen Jr, V.; Araujo, C. A.; GUIMARAES, G. . Use of the electromechanical impedance method in the detection of inclusions: application to mammary tumors. *Structural Health Monitoring*, p. 147592171882513-105, 2019.

Moura Jr, J.R.V.; Steffen Jr, V.. Impedance-based Health Monitoring for Aeronautic Structures using Statistical Meta-modeling. *Journal Of Intelligent Material Systems and Structures*, v. 17, p. 1023-1036, 2006.

Nomelini, Q. S. Schroden; SILVA, J. W.; Gallo, C. A.; Finzi Neto, R. M.; Tsuruta, K. M.; MOURA JR, J. V. . Non-parametric Inference Applied to Damage Detection in the Electromechanical Impedance-based Health Monitoring. *International Journal Of Advanced Engineering Research and Science*, v. 7, p. 73-79, 2020.

Palomino, L.V. and Steffen Jr, V., "Damage Metrics associated with Electromechanical Impedance Technique for SHM applied to a Riveted Structure"; *Proc. of the 20th International Congress of Mechanical Engineering,* Gramado, Brazil, November 15-20, 2009

Palomino, L. V.; Moura Jr, J.R.V.; Tsuruta, K.M.; Rade, D.A.; Steffen Jr, V.. Impedance-based health monitoring and mechanical testing of structures. *Smart Structures and Systems*, v. 7, p. 15-25, 2011.

Palomino, L.V.; Tsuruta, K.M.; Moura Jr, J.R.V.; Rade, D.A.; Steffen Jr, V.; Inman, D.J., "Evaluation of the Influence of Sensor Geometry and Physical Parameters on Impedance-Based Structural Health Monitoring"; 2012, *Shock and Vibration*, Vol. 9, Nb. 5, pp. 811-823 (DOI: 10.3233/SAV-2012-0690)

Park, G., Kabeya, K., Cudney, H.H. and Inman, D.J. (1999), "Impedance- Based Structural Health Monitoring for Temperature Varying Applications", *SME Int .Journal. Ser A. Solid Mech. Mater. Eng. Soc. Mech. Engineers*, **42**(2), 249–258.

Park, G., Cudney, H. and Inman, D. J. (2000), "An Integrated Health Monitoring Technique Using Structural Impedance Sensors", *J. Intell. Mater. Syst. Struct.*, **11**(6), 448-455.

Park, G., Cudney, H. and Inman, D.J. (2001), "Feasibility of Using Impedance-Based Damage Assessment for Pipeline Systems", *Earthquake Engng Struct. Dyn.*, **30**, 1463–1474.

Park G, Sohn H, Farrar CR, et al. Overview of piezoelectric impedance-based health monitoring and path forward. *Shock Vib Digest* 2003; 35(6): 451–463.

Rabelo, Steffen Jr, Finzi Neto, R. M. "Impedance-based structural health monitoring and statistical method for threshold-level determination applied to 2024-T3 aluminum panels under varying temperature". *Structural Health Monitoring*. 2016, v1.

Rabelo, D. S.; Hobeck, J. D.; Inman, D. J.; Finzi Neto, R. M.; Steffen Jr, V. . Real-time structural health monitoring of fatigue crack on aluminum beam using an impedance-based portable device. *Journal Of Intelligent Material Systems And Structures*, p. 1045389X1770521-112, 2017(a).

Rabelo, D. S., Tsuruta, K. M., Oliveira, D. D., Cavalini Jr, A. A., Finzi Neto, R. M., Steffen Jr, V. "Fault Detection of a Rotating Shaft by Using the Electromechanical Impedance Method and a Temperature Compensation Approach". *Journal of Non-Destructive Evaluation*. 2017(b).

Silva, R. N. F.; Tsuruta, K. M.; Rabelo, D. S.; Finzi Neto, R. M.; Cavalini Junior, A. A.; Steffen Jr, V. . Impedance-based structural health monitoring applied to steel fiber-reinforced concrete structures. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, v. 42, p. 383-15, 2020.

Sun, F.P., Chaudhy, Z., Liang and C., Rogers, C.A. (1995), "Truss Structure Integrity Identification Using PZT Sensor–Actuator" *J. Intell. Mater. Syst. Struct.*, **6**, 134-139.

Tseng, K. K-H.; Naidu, A. S. K. Non-parametric damage detection and characterization using smart piezoceramic material. *Journal Smart Material and Structures*. V.11, p. 317-329, 2002.

Tsuruta, K. M.; Finzi Neto, R. M.; Rabelo, D. S.; Cavalini A. A.; Oliveira, D. D.; Steffen Jr, V. . Fault Detection of a Rotating Shaft by Using the Electromechanical Impedance Method and a Temperature Compensation Approach. *Journal of Nondestructive Evaluation*, v. 36, p. 1-13, 2017.

Chapter 13: Fundamental Concepts for Guided Lamb Wave-based Structural Health Monitoring

Chapter details

Chapter DOI:

https://doi.org/10.4322/978-65-86503-71-5.c13

Chapter suggested citation / reference style:

Finzi Neto, Roberto M., et al. (2022). “Fundamental Concepts for Guided Lamb Wave-based Structural Health Monitoring”. *In* Jorge, Ariosto B., et al. (Eds.) *Model-Based and Signal-Based Inverse Methods*, Vol. I, UnB, Brasilia, DF, Brazil, pp. 472–501. Book series in Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity.

P.S.: DOI may be included at the end of citation, for completeness.

Book details

Book: Model-based and Signal-Based Inverse Methods

Edited by: Jorge, Ariosto B., Anflor, Carla T. M., Gomes, Guilherme F., & Carneiro, Sergio H. S.

Volume I of Book Series in:

Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity

Published by: UnB City: Brasilia, DF, Brazil Year: 2022

DOI: https://doi.org/10.4322/978-65-86503-71-5

Fundamental Concepts for Guided Lamb Wave-based Structural Health Monitoring

Roberto Mendes Finzi Neto*[1], Stanley Washington Ferreira de Rezende[2], Jose dos Reis Vieira de Moura Junior[3]

[1]Post-Graduate Program – Mechanical Engineering, Fed. University of Uberlandia, Brazil. E-mail: finzi@ufu.br

[2]Post-Graduate Program – Mechanical Engineering, Fed. University of Uberlandia, Brazil. E-mail: stanley_washington@ufu.br

[3]Post-Graduate Program – Modeling and Optimization, Fed. University of Catalao, Brazil. E-mail: zereis@ufcat.edu.br

*Corresponding author

Abstract

This chapter presents the basic concepts for implementing the guided Lamb Wave method for damage monitoring in mechanical structures. This is one of the structural health monitoring techniques that have been employed in recent years using a network of transducers in order to inspect thin structures. Basically, the methodology allows damage monitoring by comparing the reference signals with the test signals. However, the acquisition systems commonly used in this technique have a high added cost and also require a better level of technical knowledge on the part of the analyst. Thus, this contribution aims to present the development of a low-cost and easy instrumentation system for monitoring structural integrity using Lamb Waves, enabling the use of the technique in field studies, as well as in the context of incipient research in schools.

1. Introduction

Several types of mechanical systems in use in engineering are subjected to critical work regimes in which, combined with possible design errors, can lead to failures, cracks, or damage (Moura Jr, 2008). In this way, the ability to monitor the useful lifetime in which these systems can maintain their operability has been emphasized in the field of structural engineering in recent years. These aspects concern economic, environmental, health, and safety factors.

Thus, in this chapter, we seek to carry out the evaluation of problems related to the scope of structural integrity monitoring, usually defined in the literature as Structural Health Monitoring (SHM). Such methods have the ability to detect and interpret changes in mechanical properties of the structures under study, in order to facilitate the management of the structural life cycle. These generally include steps of data acquisition, filtering, validation, and analysis (Kudela et al., 2018).

The basic principle used by the SHM methodology is that the presence of damage influences the physical and/or geometric properties of the structures, such as stiffness, mass, energy dissipation patterns, boundary conditions and system connectivity. All these elements can negatively affect the final performance of the structure, causing a change in the dynamic response of the system. To do so, it is necessary to compare the two fundamental states of the analyzed structure, that is, the healthy state condition and the damaged state condition (Saravanan; Gopalakrishnan; Rao, 2015).

In carrying out the process of monitoring the states of a mechanical structure, SHM techniques make use of so-called intelligent or adaptive materials, mechanically coupled to these structures (Cui; Liu; Soh, 2014).

Piezoelectric materials are among the main types of intelligent semi structures present in the literature, which are currently used with great efficiency both as sensors and actuators. These, added to a power source and subsequent fault assessment algorithms, allow the continuous monitoring of a given mechanical system over time (Santos, 2004).

In the same way as other non-destructive testing (NDT) techniques, the sensing employed by SHM methods preserve the primary characteristics of the structure after its completion (Soleimanpour; NG, 2017). Sensing can be performed passively, evaluating a certain state of a structure using passive sensors, which are continuously monitored in time and subsequently fed back into the structural system, or active, which employs actuators in order to interact with the interface of the mechanical system.

Piezoelectric wafers (PZT - Plumbum Zirconate Titanate) currently represent the main materials used in the active monitoring of SHM (Venugopal; Wang, 2015).

According to Franco (2009), the main challenge of monitoring structural integrity is the fact that changes in the dynamic response of the systems actually come from the presence of damage, as well as how to identify them. In this context, one of the main SHM methodologies in the time domain that has recently been shown to be efficient in predictive studies is the method based on the so-called Lamb Waves (Pohl et al., 2012).

Although its theoretical basis was founded in the 1920s by Horace Lamb, the Lamb Wave monitoring method has re-emerged in the last three decades as one of the most reliable methods for damage identification (Dalton, 2000). This event is due to the recent technological development of instrumentation systems as well as the need to develop low-cost integrity analysis techniques.

It should also be noted among the characteristics of the SHM technique by Lamb Waves, which is non-destructive, simple instrumentation, and has a broad perspective of applicability with intelligent control systems. These reasons attract their use in regions of difficult access and high implementation cost, such as in naval, automotive, space, and/or aeronautical structures (Leucas, 2009).

However, the physical experimentation of this monitoring technique is often reduced due to the process of building the scientific base. Such a situation often makes its understanding and application deficient. Therefore, the main objective of this chapter is to apply concepts already widespread in the scope of integrity analysis and a few practical details in SHM in an attempt to build a low-cost experimentation model of the structural integrity monitoring technique by Lamb Waves. (Moura Jr, 2008; Ledesma, 2015).

2. Smart Materials and Lamb Waves Technique

Piezoelectric transducers from their conception stage to modern times have been widely used in different types of electronic devices. This fact is due to that these semi structures have a low manufacturing cost and reduced dimensioning, combined with low operational energy consumption that makes them attractive in engineering applications (Kobayashi et al., 2009).

According to Afshari (2012), the piezoelectricity effect was developed at the end of the 19th century by Jacques and Pierre Curie. In their work, they identified that certain types of materials, when subjected to electric fields at high temperatures, showed a relative deformation of their dimensions. This property was later called Curie Temperature and is the temperature at which the material spontaneously loses its initial polarization and, consequently, its piezoelectric property. Thus, it has been called piezoelectric material any material that has the ability to relate different electrical potentials in mechanical efforts applied to it, or inversely, measure deformations generated in the material from subjected electrical potentials (Lu et al., 2017). The direct effect is the process of formation of a potential difference between the dipoles of a piezoelectric transducer when they are subjected to mechanical deformation. On the other hand, the inverse effect consists of the mechanical deformation of the transducer by imposing a potential difference (Lu et al., 2017). This relationship obtained between the applied electric field and the subsequent mechanical deformation of a given structure can be quantified according to equation (1).

$$\epsilon_{ij} = \rho_{ij}\left(\frac{v_i}{l}\right) \tag{1}$$

where ϵij is the piezoelectric modulus, *i* is the direction of the applied electric field, and *j* is the direction resulting from the normal strain; *v* is the voltage applied to the PZT patch in the electric field direction *i* and *l* is the patch thickness.

Through an initial formulation proposed by Devonshire, he determined that the characteristic relationships of dielectric materials precede the total energy of the system. This means that it is possible to understand the phenomenological relationships of the direct and inverse piezoelectric effects of a given electromechanical system (Fu; Cohen, 2000). Still, even considering the responses of these structural applications very close to linearity, when performing procedures with high levels of excitation, at high frequencies, the piezoelectric elements still showed non-linearity characteristics, making the modeling complex. Thus, applications involving piezoelectricity effects are generally restricted to the linear laws formulated by Voigt, where the direct and inverse effects follow, respectively, equations (2) and (3) (Franco, 2009).

$$D_m = \epsilon^T_{mn} E_n + d_{mkl}\sigma_{kl} \tag{2}$$

$$e_{ij} = s_{ijkl}\sigma_{kl} + d^T_{mij} E_m \tag{3}$$

where, *D* and *E*(n/m) represent, respectively, the displacement vector and the electric field vector of the PZT patch; $\epsilon^T{}_{mn}$ and *d* represent the material's dielectric tensor and the piezoelectric voltage tensor; the ceramic material strain and the sigma stress are related by the applied longitudinal tensor s.

At this moment, it is possible to understand that characteristics of the direct effect of piezoelectric materials allow their use as sensors, as the property resulting from the inverse effect allows the description of the device as an actuator (Islam; Huang, 2016). However, to perform an efficient fault identification, the electromechanical coupling of such structures must be designed in a way that does not significantly influence the dynamic response of the structural system.

Among the main types of piezoelectric materials presented in the literature that have subsequently been used efficiently in fault diagnosis are:

- lead titanate-zirconate (PZT) patches: ceramic material that generally has greater stiffness than the holding structure, allowing an effective electromechanical coupling to identify damage.
- Polyvinylidene Fluoride (PVDF) polymers: greater ductility compared to PZT patches and also greater elasticity than common engineering structures, making their use as actuators inefficient.

In the literature, there are two possible approaches that correlate the Lamb Waves technique to the use of piezoelectric materials for the fault identification process, especially in relation to the use of PZT patches. Such approaches are called pulse-echo and pitch-catch (Mei; Giurgiutiu, 2018), and both are respectively represented in Figure 1.

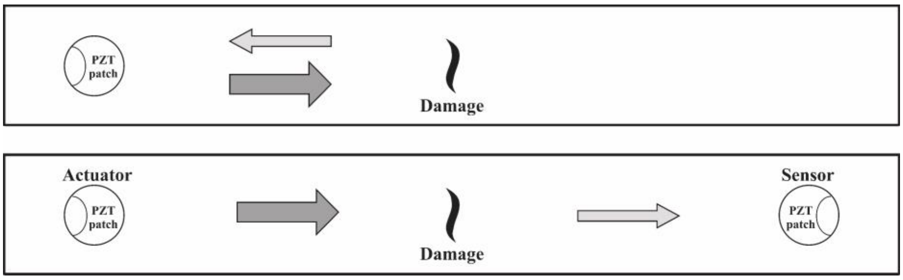


Figure 1: Pulse-echo and pitch-catch approaches in Lamb Waves techniques.

The pulse-echo approach employs only a single patch in order to interact with the host structure and receive its dynamic response. The pitch-catch, on the other hand, employs two or more piezoelectric transducers, interacting alternately (one as sensor and the other as actuator) in the gathering process (Zhang et al., 2016).

However, some noise can be acquired during the data collection procedure due to intrinsic (electronic components) and environmental (temperature) aspects that involve the experimental procedure. These noises can interfere to indicate false positives in the damage inference process. Thus, it is necessary to apply extraction methods to obtain information from the wave propagation medium in the structures under study. In this context, Wavelet Transforms have been shown to be very useful for the analysis of non-stationary signals, making their use widespread in the most diverse areas, including the analysis of structural integrity.

3. Continuous Wavelet Transform (CWT)

Wavelets are representations of wave functions of short duration with sudden changes in amplitude, proportional to their fast decay in time. This feature of signal energy limitation gives Wavelets a compact aspect in their use, being useful to signal processing, especially to non-stationary signals (Park et al., 2007).

The wavelet transform mathematical approach is based on scalar representation and maps the signal into the resolution-scale domain, alternatively to the frequency domain of classical Fourier analysis. Thus, all scales in the resolution-scale domain, without exception, have their frequency equivalence in the time-frequency domain (Debnath; Shah, 2002).

One of the main tools that make up the Wavelet theory, the Continuous Wavelet Transform (CWT) describes the union of a set of base functions resulting from different displacement and dilation operations of the main Wavelet in the scale resolution domain. This main Wavelet is called Mother Wavelet $\Psi_{(a,b)}$ and is described by equation (4).

$$\Psi_{(a,b)}(t) = \frac{1}{\sqrt{a}}\Psi\left(\frac{t-b}{a}\right) \tag{4}$$

where a is the scale parameter that are compressed or extended versions of the Mother Wavelet function and b is the displacement parameter, positioning the function in the temporal domain (Domingues et al., 2016). These compressed and extended versions of the Mother Wavelet function can be obtained by convoluting functions belonging to Wavelets families in time.

Therefore, the decomposition of a signal $f(t)$ resulting from the application of CWT at different frequencies allows us to obtain a family of scales in the Wavelet Domain. This decomposition is carried out according to equation (5).

$$CWT_{(a,b)}(t) = \frac{1}{\sqrt{a}}\int_{-\infty}^{\infty} f(t)\Psi\left(\frac{t-b}{a}\right)dt \tag{5}$$

As the displacement parameter changes, the signal is analyzed locally around it. This means that the Wavelet function molds itself to different parts of the signal in order to allow detection of the characteristic frequencies of each region.

Figure 2 illustrates the relationship between the multiple scales of the Wavelet domain and their subsequent equivalence in the power spectrum. Furthermore, in this image, the process of abstraction of a time-varying signal by the CWT technique is also conceptually presented.

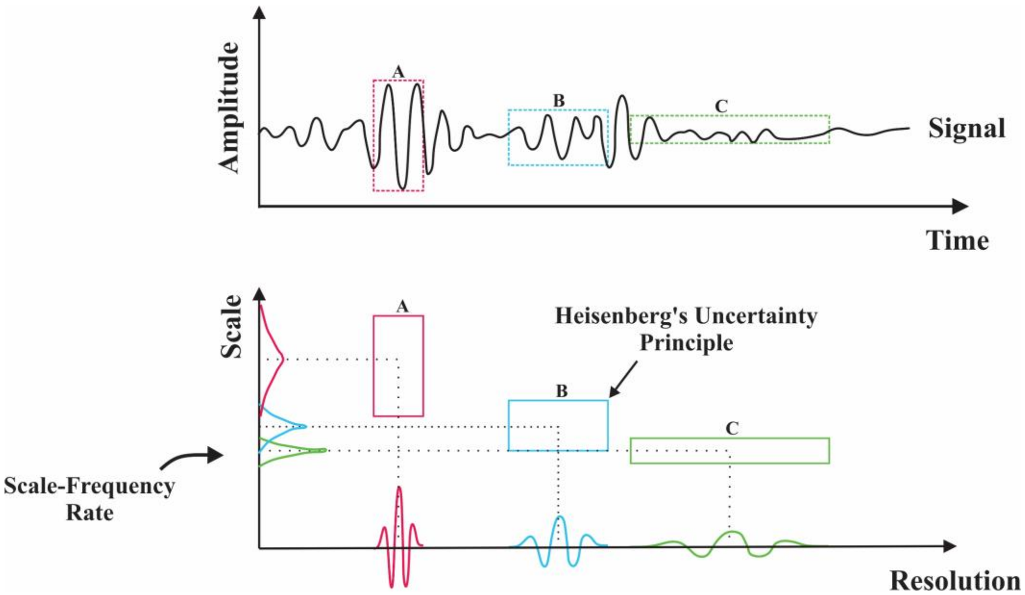


Figure 2: the process of abstraction of a time-varying signal by the CWT technique.

In general, the most used continuous wavelets in the signal analysis are the functions of the Complex Morlet family. These functions provide information about the phase, modulus, and discontinuous periods of the signal. Morlet Wavelets can be obtained by multiplying a complex exponential with a modeling Gaussian window, according to equation (6).

$$\Psi(t) = \int_{-\infty}^{\infty} \left(e^{-iwt}\right)\left(e^{\frac{t^2}{2\sigma^2}}\right) dt \tag{6}$$

where σ is the Gaussian standard deviation correlated to the Heisenberg uncertainty principle present in the Wavelet function. This parameter provides information on the quality of the time-frequency relationship of the signal analysis windows. High values for σ allow obtaining better resolutions in the frequency domain, while small values lead to better temporal resolutions of the signal (Debnath; Shah, 2002). Figure 3 illustrates Morlet's Wavelet function ranging from 0 to 1000 seconds. It can be seen in Figure 3 that the Morlet function presents a sudden scalar variation in a short period of time (around -2 and 2) and null at other moments.

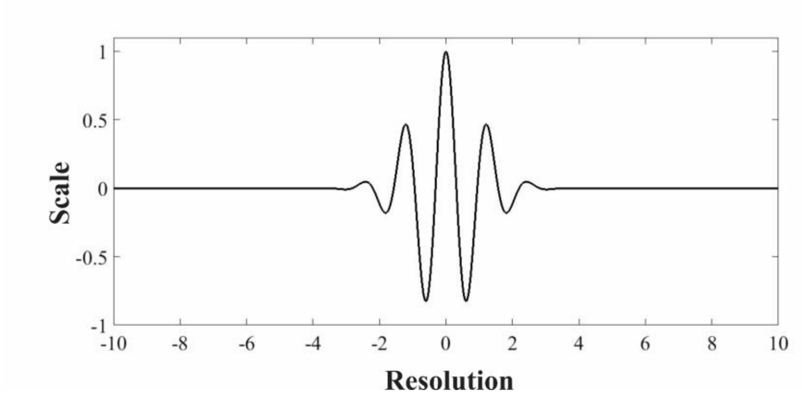


Figure 3: Morlet's Wavelet function.

The Wavelet approach differs from the Fourier methodology because it is locally restricted to a single region, while Fourier uses infinite oscillatory functions obtained from families of sine and cosine. Thus, the approach using CWT suggests being able to extract different frequency bands, as well as their respective energy contributions in the formation of the vibration signal of a given structure (Domingues et al., 2016). Thus, the application of CWT with integrity monitoring techniques based on Lamb Waves allows monitoring failure behaviors.

4. Fundamentals of Lamb Waves

Recently seen as one of the most widely used methods in identifying damage in structural dynamics, Lamb waves use a system composed of transducers, usually piezoelectric. This system sends mechanical voltage waves in the host structure and, based on the comparison of variations between the received signals, it monitors the presence of damage (Rocha, 2017).

Based on the movement of the structure, the Lamb Wave method can be understood as the superposition of two basic modes of vibration. These are the longitudinal wave propagation mode and the transverse wave propagation mode.

In longitudinal propagation, the movement applied to the material particles is defined as parallel to the direction of the force that runs through the structure. In transverse propagation, this movement is perpendicular to its direction. Thus, the coupling of such basic vibration modes enables the method the possibility of inspecting a large coverage area, making it attractive to several engineering applications (Sun; Zhang; Rose, 2005).

There are several mathematical equations capable of obtaining the modes: longitudinal and transverse wave propagation, each with its own particularity (Possani et al., 2017; Santos, 2004). However, among the main methods used efficiently in the literature, there is one based on scalar and vector potential fields of the wave. These are expressed according to equations (7) and (8), respectively:

$$\frac{\partial^2\varphi}{\partial x^2}+\frac{\partial^2\varphi}{\partial z^2}=\frac{\partial^2\varphi}{v_L^2\partial t^2} \tag{7}$$

$$\frac{\partial^2\psi}{\partial x^2}+\frac{\partial^2\psi}{\partial z^2}=\frac{\partial^2\psi}{v_T^2\partial t^2} \tag{8}$$

where φ and Ψ represent the potential displacement functions; v_L and v_T are, respectively, the longitudinal and transverse propagation velocities of the wave in the material; x is the coordinates of the wave propagation direction, z is the coordinates of the normal direction on the surface of the structure, and t is the propagation time. The longitudinal and transverse propagation velocities can be obtained, respectively, according to equations (9) and (10) (Santos, 2004).

$$v_L = \sqrt{\frac{E(1-\sigma)}{\rho(1+\sigma)(1-2\sigma)}} = \sqrt{\frac{2\mu(1-\sigma)}{\rho(1-2\sigma)}} \tag{9}$$

$$v_T = \sqrt{\frac{E}{2\rho(1+\sigma)}} = \sqrt{\frac{\mu}{\rho}} \tag{10}$$

where *E* is the modulus of elasticity, *μ* is the shear modulus, *σ* is the Poisson's coefficient and *ρ* is the density of the material that constitutes the structure.

Using the Euler formula and subsequent simplification processes, the sinusoidal solutions found for the potentials expressed in equations (7) and (8) are given by equations (11) and (12).

$$\varphi = \left(A_1 sin\left(pz\right) + A_2 cos\left(pz\right)\right) e^{-i(kx-\omega t)} \tag{11}$$

$$\psi = \left(B_1 sin\left(qz\right) + B_2 cos\left(qz\right)\right) e^{-i(kx-\omega t)} \tag{12}$$

where *A₁*, *A₂*, *B₁,* and *B₂* are constants defined according to the boundary conditions of the system under study, *k* represents the number of pulses that, as well as the parameters *p* and *q*, are obtained by equations (13), (14) and (15), respectively.

$$k = \frac{2\pi}{\lambda} \tag{13}$$

$$p = \sqrt{\frac{\omega^2}{v_L} - k^2} \tag{14}$$

$$q = \sqrt{\frac{\omega^2}{v_T} - k^2} \tag{15}$$

where *λ* represents the wavelength and *ω* the angular frequency obtained by *2πf*, where *f* is the cutoff frequency.

In this scenario, the literature describes two patterns for the combination of these propagation modes in an isotropic material, simultaneously, correlating them to the structural characteristic movement, called symmetric and anti-symmetric patterns.

Considering an isotropic plate as propagation material, the system boundary conditions can be easily obtained using the Rayleigh-Lamb equations (Corrêa, 2014). The Rayleigh-Lamb equations for symmetric and antisymmetric modules are expressed as equations (16) and (17), respectively.

$$\frac{tan\left(qh\right)}{tan\left(ph\right)} = \frac{-\left(k^2 - q^2\right)^2}{4qpk^2} \tag{16}$$

$$\frac{tan\left(qh\right)}{tan\left(ph\right)} = \frac{-4qpk^2}{\left(k^2 - q^2\right)^2} \tag{17}$$

In symmetrical modules, the elements that move in the material have a certain conformity in relation to the median plane of the structure, and these are commonly used for failure analysis in metallic structures. In antisymmetric modules, the elements interact alternately in relation to the median plane of the structure, being used in composite structures (Shen, 2014). Both patterns can be graphically visualized in Figure 4.

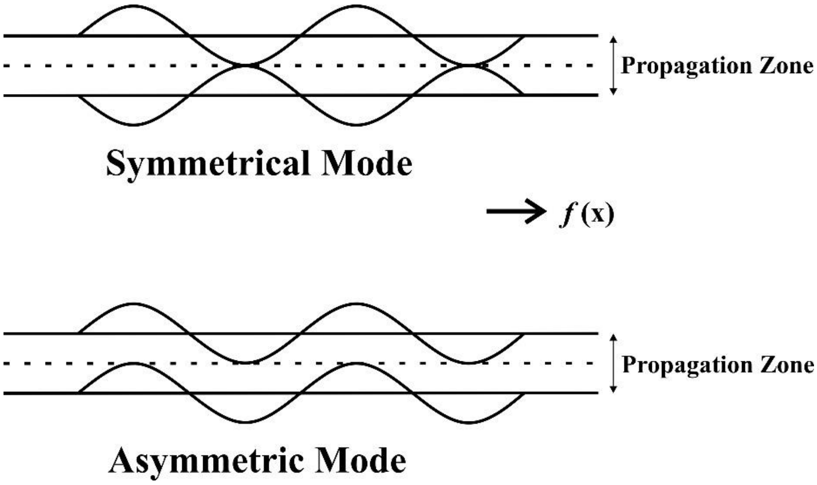


Figure 4: Symmetric and anti-symmetric patterns.

There are infinite wave propagation modes, symmetric and anti-symmetric, able to traverse structural materials. They can be obtained by different cutoff frequencies while becoming dependent on the specific properties of the materials that compose them (Qiao; Fana, 2014). The propagation modes are commonly represented in the literature by S_i (symmetric) and A_j (anti-symmetric), where *i* and *j* are the orders of each module.

Cardoso et al. (2012) used the A_1 and S_0 propagation modes in order to inspect possible corrosion in carbon steel tubes and aluminum plates. Therefore, the displacement curves of the propagation modes in these two structures and the groups of longitudinal propagation modules in the tubes were qualitatively evaluated. In addition, a quantitative analysis of phase velocities was performed, using a difference factor. Oliveira (2015) compared the same symmetric S_0 mode of Lamb Waves to the quasi-Scholte mode in studies of metallic plates submerged in a viscous fluid. Both described that the Lamb Wave integrity monitoring method is more effective than the frequency domain-based methods in terms of location, severity, and type of damage.

However, it is important to note that the abstraction of information from the waves obtained in the execution of experimental processes becomes the main hesitation in applying the method. This is due to its assessment, which is a complex task, as Lamb's guided waves are relatively influencing of the wavelength and cut-off frequency (Mechbal; Rebillat, 2017). In this sense, in order to abstract characteristics regarding the propagation medium, signal processing algorithms can be used, such as the Fast Fourier Transform (FFT) and the CWT. The use of this type of algorithm enhances the capabilities of the Lamb Waves fault identification method, enabling the extraction of characteristic frequencies in the case of

FFT, or even finding signal energy scales in the time domain for CWT (Franco, 2009). Also, it should be noted that the current data acquisition systems used in monitoring by Lamb Waves have a high operational cost added to a robustness that complicates its use in field analysis, as well as its large-scale evaluation.

5. Damage Index for Lamb Waves

In this section, we will briefly discuss the measurement of structural integrity using lamb waves. Initially, it is considered that the mechanical system is properly instrumented with piezoelectric transducers (actuator-sensor) with adequate electromechanical coupling. The experimental procedure of integrity inspection by Lamb Waves starts with the determination of the waveform to be used for system excitation (Leão et al., 2012).

Therefore, to configure the waveform to be used in the mechanical excitation process, three main factors of the wave must be taken into account, namely: amplitude, frequency, and propagation period. The first factor is defined considering that the insertion of energy into the system allows the propagation of the signal throughout the evaluated structure, without its complete attenuation (Leucas, 2009).

It should be noted that the material used in PZT patches has a relative linearity limit, thus making the signal amplitude typically lower than *5Vpp*. For the other wave parameters, although there are optimization methods to define them, in practice these are obtained by trial-and-error processes.

Having performed the mechanical excitation of the evaluated structure, the response signals are then measured by the sensor transducer and subsequently processed by data extraction or processing tools, such as CWT. This treatment leads to obtaining a signal energy scalogram, in which the scale with the greatest contribution is directly associated with the structural characteristic movement.

From the wavelet scale with the greatest contribution to the composition of the signal, it is possible to abstract several characteristics such as signal energy, phase and group velocities, propagation period, and peaks of maximum and minimum of the wave. Therefore, using such tools, it is possible to relate the different states of the structure from the variation of these wave aspects (Palmos, 2009).

However, to perform a quantitative analysis of these aspects of the signal, failure indexes called damage metrics are commonly used. The most common metric for the Lamb Waves method is the Damage Index (DI) as presented by equation (18) (Cheraghi; Taheri, 2007).

$$DI = \left| 1 - \frac{\int_{t_0}^{t_n} CWT_D(t, a_0)\,dt}{\int_{t_0}^{t_n} CWT_B(t, a_0)\,dt} \right| \tag{18}$$

where *a* is the scale with the greatest contribution to signal composition, *D* index represents the signal to be evaluated and *B* index represents the baseline signal.

According to the equation, it can be seen that the DI metric performs a comparison of the propagated energy variation between the different states of the structure. In this way, the metric values will vary between *0* and *1*, with values close to *1* meaning the presence of damage.

6. Lamb Waves Experimental Setup

In recent decades, there has been a growing search for new instrumental approaches (portable and with lower added cost) for monitoring structural integrity. However, alternative systems that are currently described in the literature still require a great deal of technical knowledge on the part of analysts and researchers, as well as on the areas related to the maintenance and set of electronic systems (Moura Jr, 2008; Wang et al., 2018). In the following section, some works developed in the last decade will be presented to illustrate some possible experimental approaches.

This section will clarify the process of structural health monitoring using the Lamb Wave method, covering all the necessary tools for its application. Since there are different possible equipment setups for the application of the technique, this section will present an elementary and low-cost approach, easy to implement by the user, without the need for great technological knowledge.

Initially, it is necessary a source of energy generation to promote the excitation of the structure, an analyzer to measure the sensor signal, and a computational interface to store it. This approach will also require PZT patches such as sensors and actuators. Aiming at the aspects of low cost and ease of implementation mentioned above, the following items were used in this work:

- UDB11008S wave generator;
- 2530 B&K Precision digital oscilloscope;
- 2 PZT patches (20 mm in diameter by 3 mm thick);
- computer (Software: EasyScope) for data collection, storage, and processing.

The UDB110x (S) series consists of frequency generators up to *8MHz* with an integrated circuit in the FPGA category. They are direct digital synthesis (DDS) alternating voltage generators with standard sine, square, triangle, and sawtooth waveforms. These generators are characterized by having high stability and low distortion, in addition to being able to regulate the amplitude and DC polarization of the output signal.

The UDB11008S generator has a sweep function, allowing you to freely define the interval and the sweep time (sweep). The output signal amplitude can reach a maximum of *9Vpp* and a minimum of *10mVpp*. However, because the PZT patches used in this example have high stiffness and small thickness, this function was set to a minimum output (*10mVpp*) (Tsuruta et al., 2008).

In the measurement step, the B&K Precision 2530 oscilloscope combines characteristics such as high performance and low cost. Since this part of the system in the literature shows itself as the one with the highest aggregate cost, this should be the item with the most careful choice. This equipment is capable of reading signals with a relative bandwidth of up to *25MHz*, with a sampling of *500 MSa/s*.

In the acquisition step, the integration between the oscilloscope and the EasyScope software allows you to explore functionalities such as collection, filtering, and storage (up to *32* automatic measurements) of signals with sizes of *500* (high-pass) or *32000* (low-pass) points. It is still possible to perform an FFT on the sample signal with different decomposition windows in order to perform both time and frequency domain analysis.

In this next step, the generator and the analyzer are connected to the two PZT inserts coupled to the structure. In the case under study, the pitch-catch configuration is used, as presented in section 2 (YU et al., 2012). With the connection of the acquisition system to the computer through a USB port, full manipulation of the system via software is possible.

An important feature of the UDB11008S generator is that it operates with a *5V* supply, allowing its use in-field analysis. Figure 5 presents the scheme of the acquisition system developed in this work, where the monitored structure is illustrated by a plate.

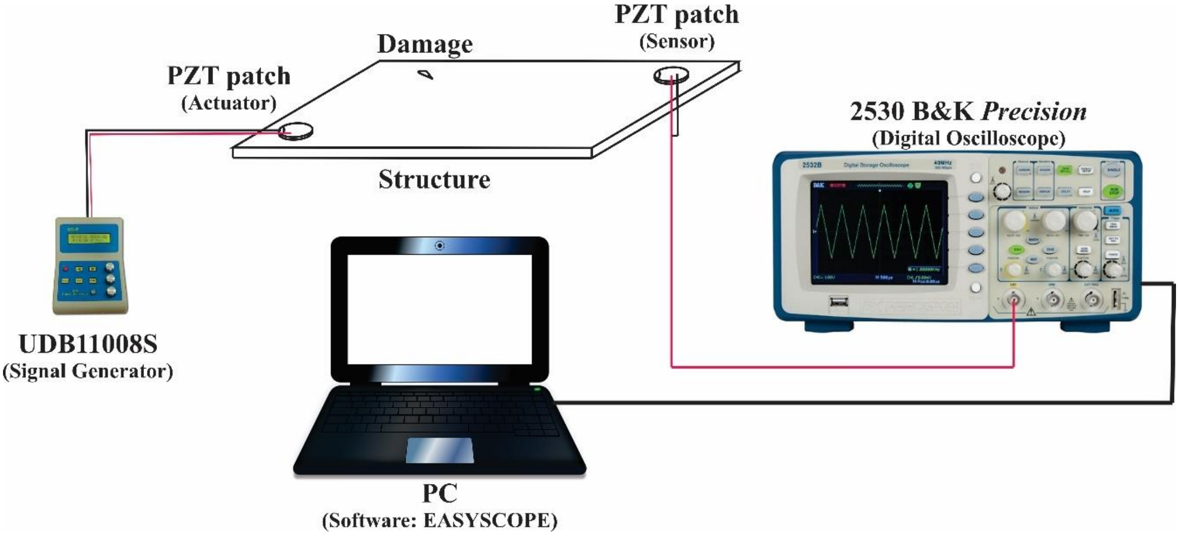


Figure 5: Low-cost Lamb Waves Experimental Setup.

The first instrumentation step of the acquisition system developed in this work consists of the electromechanical coupling between the piezoelectric chips and the structure to be monitored. For this, both PZT ingots must be bonded to the structure with a thin layer of epoxy adhesive, taking approximately 24 hours for effective curing. A simplified description of the piezoelectric patch bonding process can be seen in Figure 6.

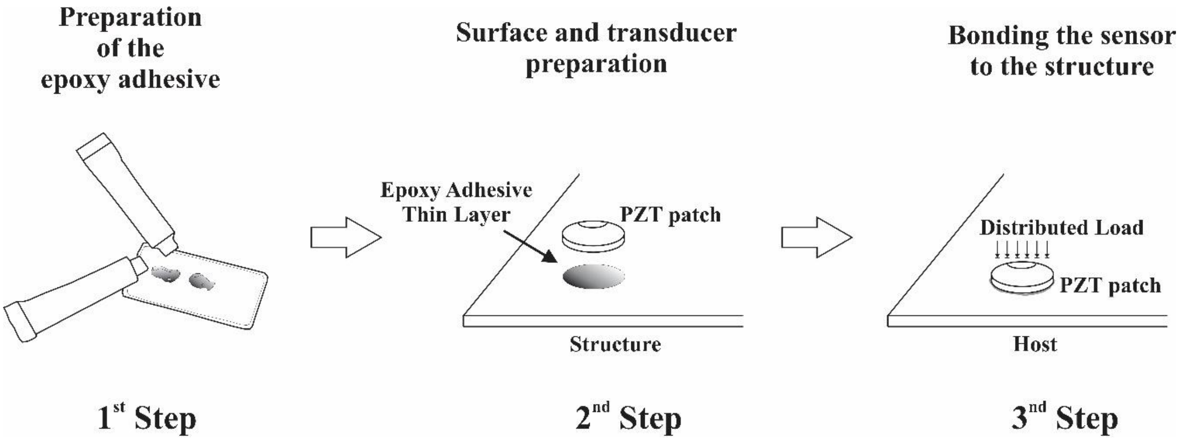

Preparation of the epoxy adhesive
Surface and transducer preparation
Bonding the sensor to the structure
Epoxy Adhesive Thin Layer
PZT patch
Structure
Distributed Load
PZT patch
Host
1st Step
2nd Step
3rd Step

Figure 6: Piezoelectric patch bonding process.

The epoxy adhesive was used due to its mechanical properties such as 1) high stiffness compared to other types of glue, allowing an efficient electromechanical coupling; 2) resistance to high temperatures, remaining unchanged up to 70º C (Overly; Park; Farrar, 2007). Furthermore, the adhesion of the transducers to the structure must be performed in a way that does not significantly interfere with the dynamic response of the system, thus ensuring that the presence of damage is the only factor for the variation of Lamb Waves.

The definition of the best coupling positions of the transducers in the structure takes into account its geometry, dimensions, contour shapes, the excitation capacity of each PZT patch, and the prerogatives of the pitch-catch configuration, that is, linear positioning between the actuator and sensor.

After the step of bonding the patches, two wires must be soldered to the electrodes of each piezoelectric component. Then, the PZT patch actuator wires are connected to the probes of the UDB11008S wave generator, which are connected to its output terminal (OUT). Now, the PZT patch sensor wires are connected to one of the channels (CH1 or CH2) of the digital oscilloscope, and this is also done with the aid of probes. Figure 7 presents the schematic representation of the connections of the transducers to the electronic components.

After the acquisition system is properly connected, the process of calibration and configuration of its electronic components begins, that is, the configuration of the wave generator and digital oscilloscope.

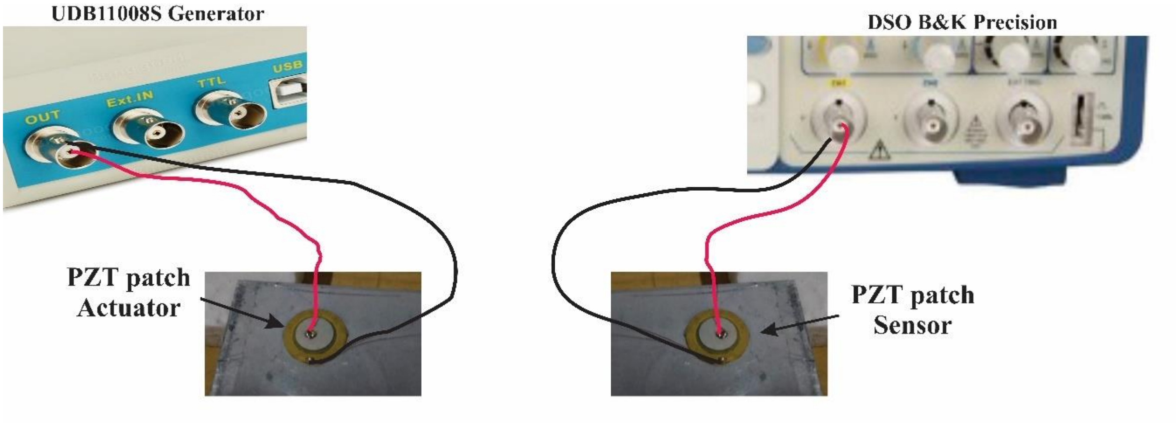


Figure 7: Graphic representation of the process of connections between PZT inserts and acquisition system equipment.

Wave generators currently play an important role in the study of electronic measurement circuits. They make it possible to analyze and manipulate the parameters of interest from previously generated reference signals. Also, with the development of digital technology, analog-to-digital converters began to create high-precision signals, mainly through the use of DDS technology. In this context, the frequency generators belonging to the UDB110x (S) series have shown great applicability in engineering and computing studies (Chen; Chen, 2011).

The UDB11008S signal generator, belonging to the commented series, allows creating alternating voltage waves in standard sine, square, triangular and sawtooth waveforms with high stability and low distortion. Figure 8 presents the schematization of the features of this generator, and each feature is described later.

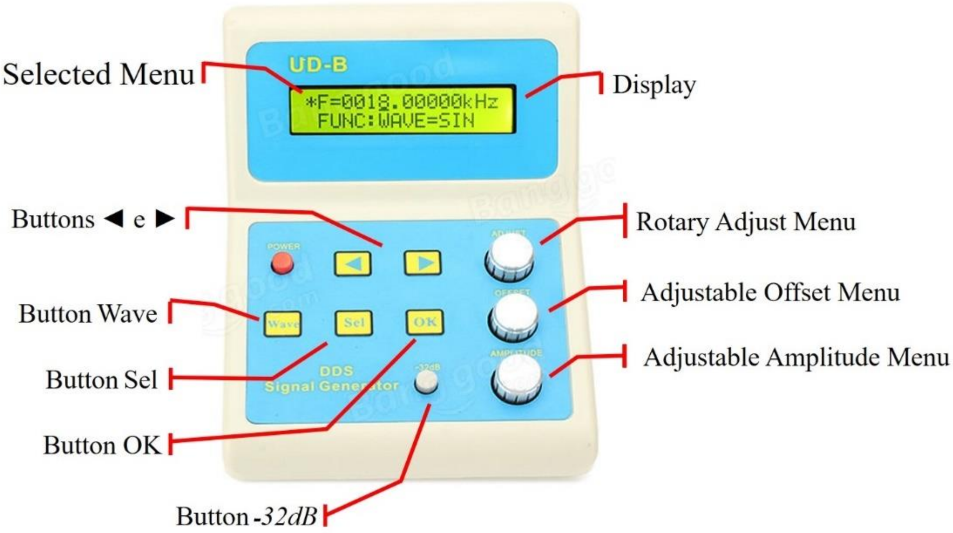


Figure 8: Function Generator - UDB11008S model.

The *Sel* key carries out the selection procedure between the function menu and the output frequency, in which the chosen functionality is represented by the * operator on the display. When the selected operation is the output frequency setting, the left and right arrows allow scrolling between the chosen frequency numbers. Here, the *Adjust* menu allows the increase or decrease of this value.

Pressing the *OK* button switches between the frequency measurement scales that the signal generator is able to generate (Hz, kHz, and MHz). It is worth mentioning that the UDB11008S generator is capable of generating waves with a frequency of up to 8MHz.

When the functions menu is chosen, the left and right arrows allow switching between functions previously stored in the generator's memory. Then they modify the control characteristics of the output wave and the external measurement. The functions available in the generator's memory are:

- *WAVE* – allows modifying the generated wave pattern by selecting and pressing the *OK* button. The UDB110x (S) series is capable of generating standard sine (*SIN*), square (*SQR*) and triangular (*TRI*) waveforms;
- *DUTY* – adjusts the duty cycle of square and triangle waveforms. When the triangle waveform is chosen, imposition values greater than 50% cause the waveform to change to a rising sawtooth, while values less than 50% cause the waveform to change to a descending sawtooth;
- *COUNTER* – represents a counter that starts from an external pulse being measured by the *Ext.IN* input terminal. The value starts to be shown on the display, while the *OK* button allows resetting the counter;

- *EXT.FREQ* – allows measurement of external frequencies from the *Ext.IN* input terminal;
- *SAVE* – allows the storage of base parameters. The stored data is the current frequency value, waveform, and duty cycle. The UDB11008S model has 10 memory storage positions, which can be changed and loaded at any time. For data storage, the following steps must be followed:
  1. Choose the current frequency and data (WAVE and DUTY);
  2. Choose the memory location to store the data;
  3. Press *OK* button to save.

  It is worth mentioning that the memory position *M0* is the wave generator's default, being called at all times during its initialization. Furthermore, the memory variables *M1* and *M2* (frequencies stored in the same way as described) are the frequency values used in the sweep (without using the *WAVE* and *DUTY* data).

- *LOAD* – loads the data stored in a given memory by the *SAVE* function;
- *TIME* – sets the sweep time to be used in the sweep;
- *SWEEP* – performs a sweep, adding the frequency value gradually over time. The start frequency value is delimited by the variable *M1* while the stop frequency value is delimited by the variable *M2*. These are obtained according to the *SAVE* submenu. The increment frequency value is defined according to the scan time previously defined in the *TIME* function of the submenu. To generate a single pulse, the values of variables *M1* and *M2* must be the same, and if these variables are not defined in memory, the standard sweep will be performed starting from *0kHz* to *10kHz* with a step of *0.1Hz*;

The *Wave* key on the function generator allows direct switching between the different waveforms (sine, square, and triangle). Meanwhile, the *OFFSET* and *AMPLITUDE* adjustment menus allow, respectively, to adjust the DC polarization and amplitude of the output signal. The *-32 dB* button attenuates the output signals to values below 10mVpp.

Based on the physical instrumentation process of the developed system, the generator calibration will depend on the wave properties to be studied. This by itself can be (also known as time-of-flight - TOF), or by means of a frequency sweep, in order to check the variation of resonant frequencies.

Figure 9 presents a flowchart for carrying out the wave generator configuration in each particular case. However, in both cases, frequencies higher than 10kHz and lower than 40kHz are chosen, since in this frequency range it is possible to assess incipient damage (MOURA JÚNIOR, 2008).

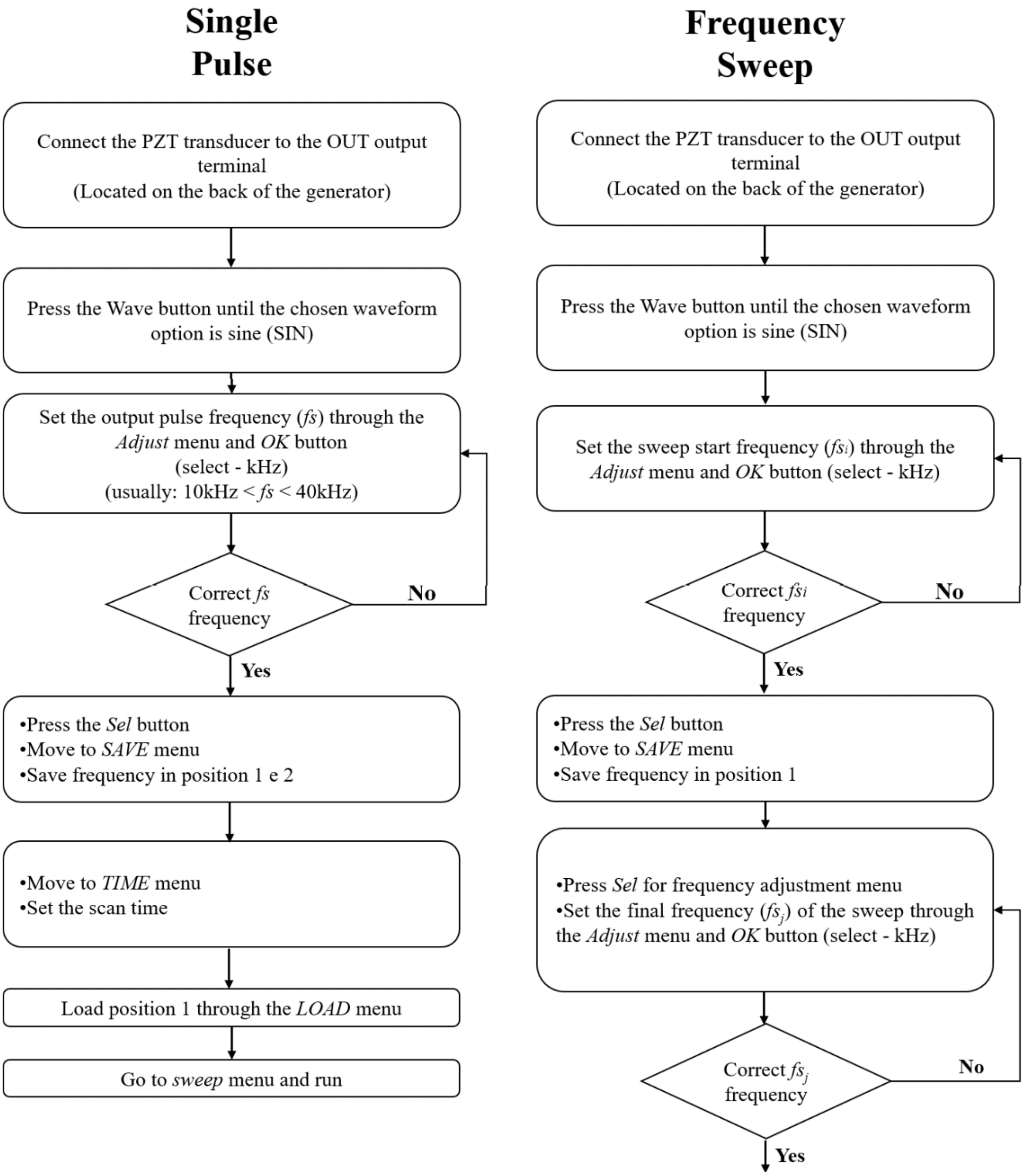

Single Pulse
Connect the PZT transducer to the OUT output terminal (Located on the back of the generator)
Press the Wave button until the chosen waveform option is sine (SIN)
Set the output pulse frequency (fs) through the Adjust menu and OK button (select - kHz) (usually: 10kHz < fs < 40kHz)
Correct fs frequency
No
Yes
•Press the Sel button
•Move to SAVE menu
•Save frequency in position 1 e 2
•Move to TIME menu
•Set the scan time
Load position 1 through the LOAD menu
Go to sweep menu and run
Frequency Sweep
Connect the PZT transducer to the OUT output terminal (Located on the back of the generator)
Press the Wave button until the chosen waveform option is sine (SIN)
Set the sweep start frequency (fsi) through the Adjust menu and OK button (select - kHz)
Correct fsi frequency
No
Yes
•Press the Sel button
•Move to SAVE menu
•Save frequency in position 1
•Press Sel for frequency adjustment menu
•Set the final frequency (fsj) of the sweep through the Adjust menu and OK button (select - kHz)
Correct fsj frequency
No
Yes

Figure 9: flowchart for setting up the UDB11008S wave generator.

During the studies of Lamb Waves, the standard sine wave format is generally used, since it makes it possible to obtain the phase and period of the wave more easily compared to other patterns.

After a reference signal is sent by the function generator and subsequently propagated through the structure under study, it will be captured by the PZT patch sensor coupled directly to the channels of the digital oscilloscope. This oscilloscope will graphically display the measured signal and then store it. Therefore, it is necessary to configure and define the parameters of this other device.

The Digital Storage Oscilloscope (DSO) is a flexible instrument that aims to measure measurements and data regarding wave aspects. This makes it possible to graphically analyze the electrical signals in the time domain and later store them in external memory (USB) (Cardoso; Silva; Segundo, 2017).

In most applications, the DSO acquires signal samples and represents them virtually on the display, where the vertical axis expresses the measured signal amplitude (Volts/Div) and the horizontal axis the wave scan time (Sec/Div). Figure 10 presents a schematic of the digital oscilloscope parts while the functions are given in Table 1.

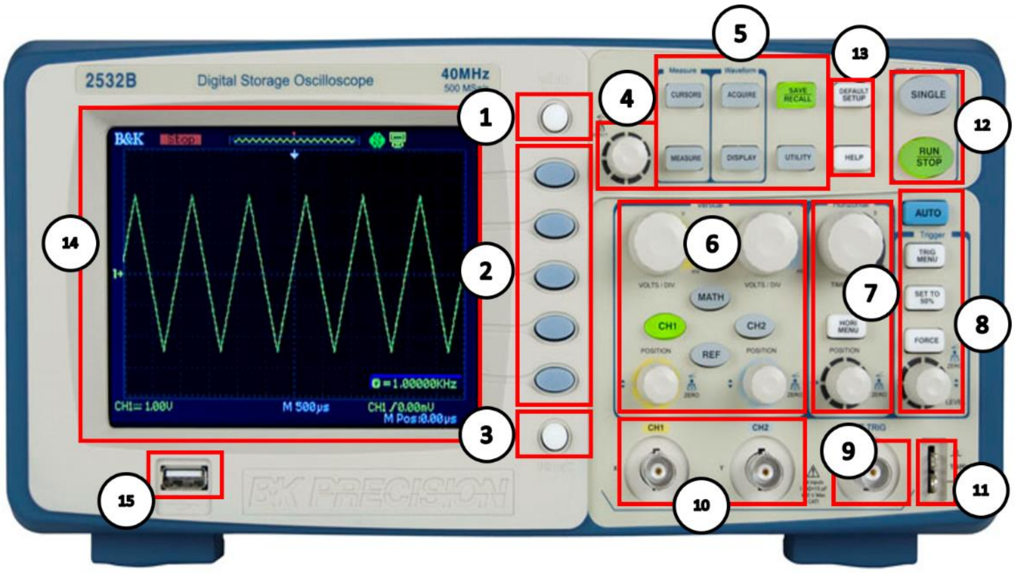


Figure 10: B&K Precision Oscilloscope – measurement of the sensor.

Table 1: B&K Precision Oscilloscope functions

Item	Function Name
1	On/Off Menu
2	Selection Buttons
3	Print
4	Universal Menu
5	Measurement and Acquisition Functions
6	Vertical Controls (CH1 and CH2)
7	Horizontal Control
8	Trigger Menu
9	Channel for External Calibration
10	Input Channels (CH1 and CH2)
11	Compensation terminal
12	Acquisition Control
13	Default and Help Buttons
14	LCD Display
15	USB input

The first step to be adopted, before any measurement test, is the calibration of the system to be used. This procedure can be performed by pressing the *Default Setup* button, located at 13, and later, by adjusting the signal obtained by the probe, which must be connected to the *Compensation Terminal*.

Vertical Controls modify both the magnification of the waveforms, which are obtained by the *Input Channels* and their positioning on the vertical axis. Also, the *CH1* and *CH2* buttons allow the visualization and application of digital filters on the sampled signals. The relationship between different filter types and menu options is shown in Table 2.

Table 2: B&K Precision Oscilloscope filters

Menu icon	Filter type
⊓‗‗→f	Low-pass
‗‗⊓→f	High-pass
‗⊓‗→f	Band-pass
⊓‗⊓→f	Notch/Band-reject

The choice of filter type is directly linked to the frequency range to be used in the experiment. Thus, because SHM methods commonly work at high frequencies, this functionality must be set to the high-pass filter type (Leucas, 2009).

In addition, the *MATH* button allows the use of mathematical operations between the multiple channels of the oscilloscope, one of which is the fast Fourier transform. The *REF* submenu displays up to two reference signals for the data to be measured, thus enabling a direct qualitative comparison between the states of the structure, with and without damage.

The *Horizontal Control* provides temporal manipulation of the sampled data on the LCD display. Here, the *Time/Div* and *Position* buttons allow, respectively, the definition of the sampling rate of the signal and its subsequent positioning in the time domain.

The *Menu Trigger* is responsible for synchronizing the acquisition of waveforms when they exceed a given threshold amplitude value. Such functionality allows the sampling of periodic signals, as well as their stabilization in the data acquisition process. Among the forms of synchronization in the *Menu Trigger* are:

- *Trigger Edge* – represents the process of acquiring a given signal when one of its points exceeds the threshold value in a specific direction, be it rising edge or falling edge. The deflection edge Trigger mode is the most used in Lamb Wave SHM studies.
- *Trigger Pulse Width* – delimits the measurement process based on the data collection trigger time with the width of the sampled wave pulse.
- *Trigger Video* – uses NTSC (525 lines of resolution) or PAL (625 lines of resolution) video standards to carry out the data verification process.
- *Trigger Slope* – relates the data acquisition process to the ascent or slope velocity of the measured signal.

When the *SET TO 50%* key is pressed, the digital oscilloscope will perform a quick wave stabilization process based on the measured voltage midpoint. However, such functionality only becomes applicable when a signal is emitted in the *Channel for External Calibration*. The *FORCE* button allows the direct acquisition of waveforms, without the need to detect a trigger threshold value.

The set of buttons present in *Measurement and Acquisition Functions* enable analytical manipulation of the signals sampled on the display, as well as data storage on a flash drive.

To carry out the storage process via physical device, the following steps must be performed on the instrument:

Initially, the flash drive, to be used for storage, must be plugged into the *USB Input* of the digital oscilloscope and wait until it is recognized. If recognition does not take place, a quick audible alert will be emitted by the DSO and, later, the message *"USB Flash Drive is not connected!"* will appear in the lower corner of the display.

With the flash drive recognized by the DSO, as a second step, press the *SAVE/RECALL* key in 5. This is responsible for displaying the storage functions menu.

Different storage options can be performed by DSO B&K Precision from this step, having different peculiarities. Table 3 briefly presents the relationship between the storage options and the subsequent steps that must be taken to save the data.

Table 3: B&K Precision Oscilloscope storage options.

Option	Details	Next Steps
Config	This option stores data in the DSO's internal memory. The B&K Precision oscilloscope has up to 20 positions available for data storage.	1. Select the *Device* option; 2. Choose the memory option to be stored; 3. Select the *Save* option.
Waveform	This option stores the signal image in the DSO's internal memory. The B&K Precision oscilloscope has up to 10 positions available for image storage.	1. Select the *Device* option; 2. Choose the memory option to be stored; 3. Select the *Save* option.
Image	This option stores the image of the signal sampled on the display in a flash drive memory.	1. Press *Save. image*.
CSV	This option stores the wave data (amplitude, time and others) on the flash drive with .csv extension	1. Set the data length option to Display; 2. Press the *Save* option; 3. Choose the name and directory on the flash drive; 4. Press *Save*.
Factory	This option restores the factory parameters of the DSO's internal memory, that is, the option performs the cleaning of the internal memory.	1. Press *Restore*.

Assuming that Lamb Waves use feature extraction methods (CWT) and subsequent damage quantification algorithms, the most used storage form in SHM procedures is the CSV option. This enables the physical storage of the points that compose the signals for further analysis in a computational environment.

When pressing the *CURSORS* key on the DSO panel, the *Selection Buttons* allow the transition between the different cursor options available. With these buttons, it is possible to measure both the signal amplitude variation and the wave periods.

The *MEASURE*, *DISPLAY*, and *UTILITY* buttons allow, respectively, the visualization of the current wave data, the format of displaying the equipment interface, and the configuration of the firmware and system utilities (language, audio, and others).

Figure 11 shows the flowchart for performing the B&K Precision digital oscilloscope configuration via physical equipment.

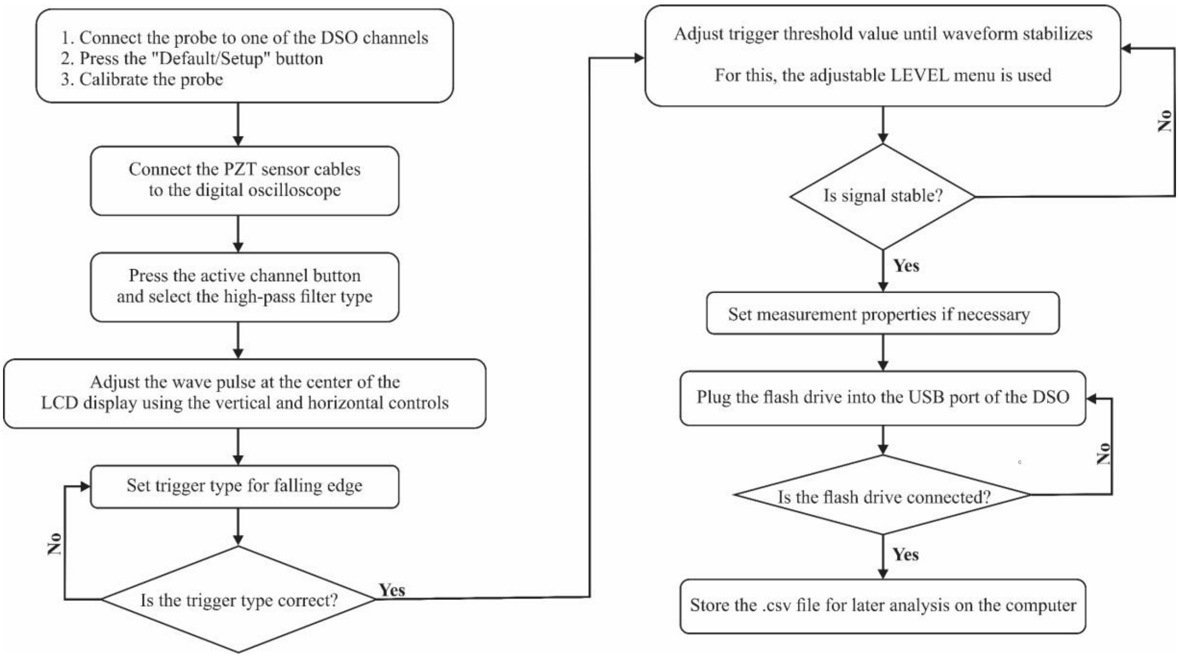

1. Connect the probe to one of the DSO channels
2. Press the "Default/Setup" button
3. Calibrate the probe
Connect the PZT sensor cables to the digital oscilloscope
Press the active channel button and select the high-pass filter type
Adjust the wave pulse at the center of the LCD display using the vertical and horizontal controls
Set trigger type for falling edge
No
Is the trigger type correct?
Yes
Adjust trigger threshold value until waveform stabilizes
For this, the adjustable LEVEL menu is used
No
Is signal stable?
Yes
Set measurement properties if necessary
Plug the flash drive into the USB port of the DSO
No
Is the flash drive connected?
Yes
Store the .csv file for later analysis on the computer

Figure 11: flowchart for B&K Precision Oscilloscope configuration.

7. Experimental Case Study

The experimental procedure adopted in this case study aimed to identify the presence of damage in an aluminum plate with 2 mm thickness and geometry as shown in Figure 12. To monitor this structure, three buzzer-type PZT patches were used, with a geometry of 20 mm in diameter and 0.35 mm in thickness, considering the pitch-catch coupling configuration (Figure 1).

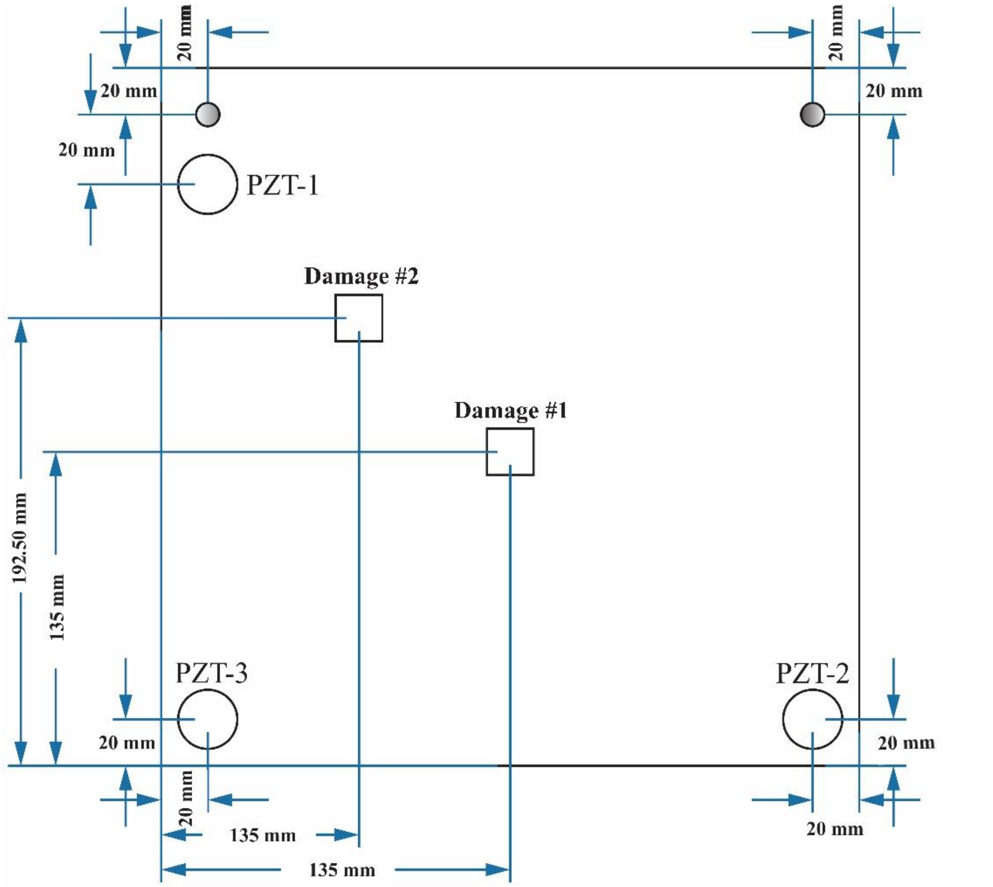


Figure 12: flowchart for B&K Precision Oscilloscope configuration.

The structure was hung vertically by thin fishing lines so that there was no greater interference from boundary conditions and the weight of the structure (holes 20 mm from the top edge). The type of virtual damage applied to the structure was by adding mass to the system in different positions. Thus, three cubic neodymium magnets (10 mm side) were mechanically coupled to the structure, two of them being coupled stacked on one side and the third on the opposite surface of the plate, in order to generate an increase in local stiffness in the two simulated positions.

The different positions used for the damage simulation aimed to visualize the potential of the Lamb Wave SHM method regarding the physical location of the damage. Thus, the position was defined with the objective of carrying out a process of triangulation of the position of the damage, contributing to the understanding of the results. The positions considered for the fault conditions can also be seen in Figure 12.

It should be noted that the weight of each magnet used is 2.1g, thus representing an increase in mass of about 1.24% compared to the total weight of the whole mechanical system. This configuration was adopted in order to verify the potential of the method in terms of identifying incipient damages.

In general, two inspection paths were adopted for the acquisition of Lamb Waves during the experiment, the PZT-1 patch was considered as an actuator in both cases and the other transducers were used as sensors in each particular path.

Then, the PZT-1 transducer was connected to the OUT-output terminal of the function generator and the other patch (PZT-2 and PZT-3) was connected to one of the channels of the 2530 B&K Precision digital oscilloscope. The connection of the transducers to the electronic equipment was made according to Figure 6.3, with the aid of probes.

The way of excitation of the PZT-1 actuator was based on a sweep of frequencies whose used range was 20-30kHz with a step of 33Hz. In all, 20 samples from each state were collected for each inspection path, thus totaling a population of 120 signal samples. Thus, 40 signatures were collected for a baseline: 20 samples PZT-1/PZT-2 and 20 samples PZT-1/PZT-3. Likewise, 40 signatures for damage 1 and 40 for damage 2.

After the acquisition step, CWT and DI algorithms were used in order to abstract the characteristics of the signals and later quantify the presence of damage in this structure. Such steps were applied in order to validate the acquisition system developed in this work.

Through the analysis of the Lamb Wave sets obtained by the experimental procedure, it was verified the need to apply an algorithm that allows the abstraction and separability of the states of the structure. This is because incipient or light damage promotes a very subtle variation in the vibration signature. Therefore, CWT was applied to the set of signals in order to extract the different characteristic frequency scales.

Figures 13 and 14 illustrate the average frequency scales obtained by applying the CWT in the different data sets, taking into account their respective inspection path.

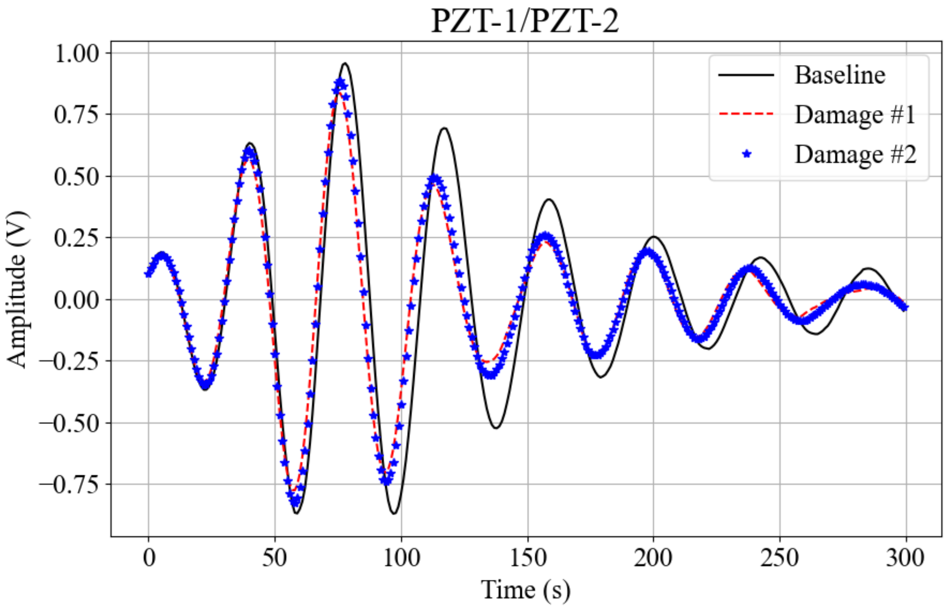


Figure 13: Averages of the characteristic wavelet scales of the states of the structure - Path PZT-1/PZT-2.

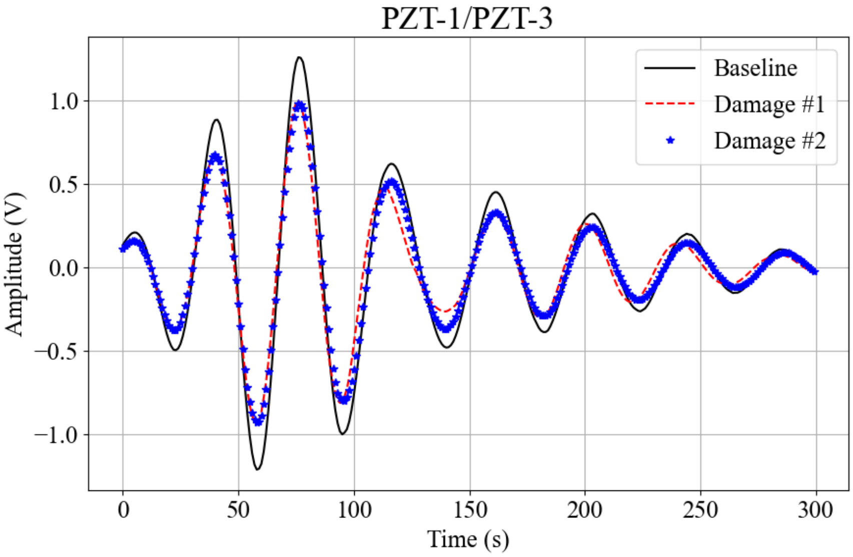


Figure 14: Averages of the characteristic wavelet scales of the states of the structure - Path PZT-1/PZT-3.

It can be seen that with the application of the CWT algorithm in the different data sets, it was possible to define the different state conditions of the structure, as well as the qualitative evaluation of the presence and severity of the damage.

However, the process of graphical analysis of the averages of the signals (performed previously) only allows the verification of the presence of damage and not its quantification and characterization in terms of severity. Therefore, an algorithm in Python was developed and applied, as a second evaluation step, of the Damage Index as a damage metric, as previously described. The DI metric values obtained as a result of the developed algorithm are presented in the box plots of Figure 15.

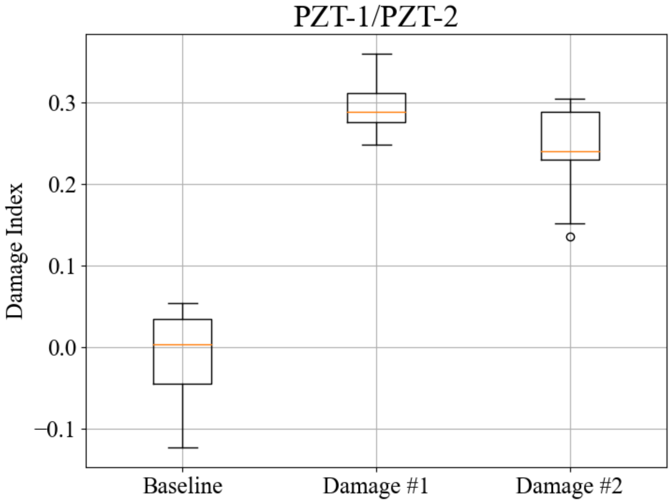


Figure 15: Boxplot of DI metric values for inspection path PZT-1/PZT-2.

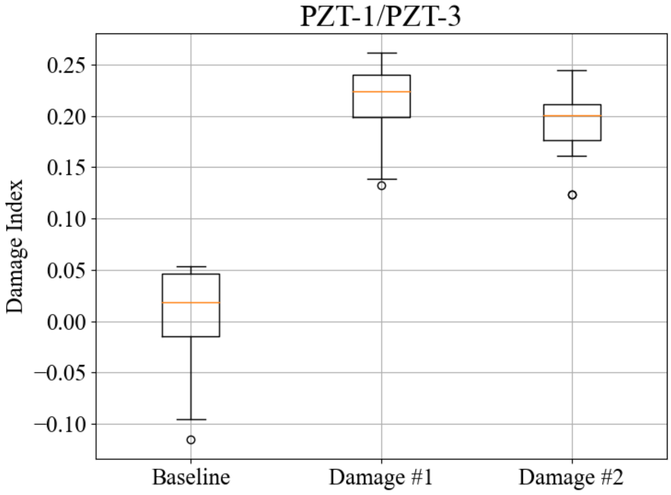


Figure 16: Boxplot of DI metric values for inspection path PZT-1/PZT-3.

From the boxplots shown in Figures 15 and 16, it can be seen that the proposed method was able to efficiently identify the presence of damage in the structure under study. Still, it can also be evaluated through the first graph that, although they are not linearly separable, the severity of Damage #1 is, in general, greater than that of Damage #2. This condition suggests that the location of this damage state is closer to the sensor than the Damage #2 state. Also, in an automatic system with a sensor network, it is possible to exploit this information through the inverse measures PZT-2/PZT-1.

The metric values obtained for the second inspection path (PZT-1/PZT-3) define that both damages are linearly equal or that their severity in relation to the inspection path is, reasonably, indifferent. This is due to the fact that the Lamb Waves are linearly influenced by the insertion mode, that is, their efficiency is more considerable in a perpendicular path between the actuator and the sensor.

Therefore, with this experiment, the applicability of the acquisition system can be evidenced, as well as the potential of the Lamb Wave SHM method in structural integrity analysis. However, it is necessary to remember that the applicability of the technique it depends on: an integrated sensor/actuator network, statistical or AI-based models for locating and quantifying the severity, and monitoring being focused on thin structures.

7.1. Final remarks – Brief State of the Art

This section presents a brief state of the art of acquisition systems used in the Structural Health Monitoring by Lamb Waves considering values in BRL in the year 2019. The currency presented is due to the high number of Brazilian colleagues who have acquired an interest in the technique.

At first, the literature describes several works that employ different fault acquisition systems by ultrasonic inspection. However, these systems usually have a high aggregate cost (over BRL 30,000) in addition to being robust compared to frequency domain acquisition systems (LEDESMA, 2015). Thus, although they have high precision, their characteristics make them a limiting factor regarding the use of these systems in-field analysis, as well as their use by schools with few financial resources. As a result, in order to develop practical and portable systems for integrity analysis by Lamb Waves, new instrumental configurations have been studied in recent decades (Farias et al., 2011; Asadi et al., 2017).

In this sense, Table 4 presents the recent instrumentation setups of the guided Lamb Waves inspection technique, linking them to their respective authors and the year of publication.

Table 4: Different authors and respective setups.

Setup	Authors	Cost
Wave Generator: PXI-5421 Module, Oscilloscope: PXI-5105 module, Signal Amplifier: Module A-303 A.A. Transducers: PZT (7.56 mm diameter per 1mm wide)	Keulen, Yildiz and Suleman (2014)	BRL 56,590
Wave Generator: PXI-5412 Module, Oscilloscope: PXI-5122 Module, Signal Amplifier: Falco WMA-320 Transducers: P-876 Patch DuraAct	Hettler et al. (2015)	BRL 60,302
Wave Generator: PXI-5412 Module, Oscilloscope: PXI-5105 Module, Transducers: PZT 5H (15 mm diameter per 0.5mm wide)	Rocha, Finzi Neto and Steffen Jr (2017)	BRL 43,415
PSV-400-3D-M Doppler Vibrometer	Soleimanpour (2016) Ji et al. (2018)	On-demand

In the work developed by Keulen, Yildiz and Suleman (2014), a network of PZT sensors was used to identify laminations in a carbon fiber-epoxy composite panel. For that, a PXI-5421 wave generator module and an A-303 A.A. signal amplifier were used in order to transmit Lamb Waves in the structure at a cutoff frequency of 265 kHz. Subsequently, the dynamic responses were collected on the PXI-5105 oscilloscope and a process of triangulation of the damaged region was performed using the RAPID (Reconstruction Algorithm for Probabilistic Inspection of Damage) tomographic reconstruction algorithm.

Hettler et al. (2015) used a system consisting of a PXI-5412 modular function generator and an NI PXI-5122 digital oscilloscope module to generate guided Lamb Waves on carbon fiber polymer (CFRP) reinforced plates. This function generator can generate standard waveforms (sine, triangle, square, and ramp) with bandwidths up to 20 MHz and amplitude ranging from -6V to 6V. The signals imposed on the actuator network were amplified through a WMA-320 laboratory amplifier and later captured through the sensor, both actuators and sensors are PVDF P-876 patches. Subsequently, a RAPID tomographic reconstruction algorithm was applied in order to obtain the parameters of location and extent of impact damage.

Rocha, Finzi Neto and Steffen Jr (2017) studied the influence of physical uncertainties (electronic components) of data acquisition systems, as well as environmental factors (temperature) on the integrity analysis procedure by Lamb Waves. For this, a 2024-T3 aluminum plate was instrumented in the pitch-catch configuration, that is, using two PZT transducers at different ends of the structure. A PXI-5412 function generator was used to apply a sine wave with a frequency of 30kHz and an amplitude of 10 Vpp. The dynamic response was measured with the PXI-5105 module with a sampling rate of 30MSa/s, with 30 thousand samples in total. It was observed that the temperature variation is a limiting factor for the damage inference process. Santos et al. (2016), observed the same relationship in the study of beams. Furthermore, it can be observed in the literature the use of this configuration in other works, such as (Moura Jr, 2008; Leucas, 2009).

Soleimanpour (2016) investigated the nonlinearity of Lamb Waves in composite beams. For this, experimental results were compared with the results obtained in a three-dimensional finite element model. The PSV-400-3D-M laser optical system was used to perform a triangulation of the surface modal nodes. This was possible through the variation of frequencies resulting from the lights of the three lasers that compose it. This configuration was replicated by Ji et al. (2018) for thin structures.

Still, alternative systems for inspection of structural integrity by Lamb Waves can be cited in the literature, which has lower cost instrumentation. The WSHM system developed by Ledesma (2015) uses transducers remotely controlled through a ZigBee network. However, there is greater complexity in using this system due to greater technical expertise in instrumentation.

7.2. References

Afshari, M. Vibration-and Impedance-based Structural Health Monitoring Applications and Thermal Effects. Ph.D. dissertation —Virginia Tech, 2012.

Asadi, S. et al. Implementation of a novel efficient low-cost method in structural health monitoring. *Smart Materials and Structures*, IOP Publishing, v. 26, n. 5, p. 055032, 2017.

Bailey, J. et al. Evidence relating to object-oriented software design: A survey. In: IEEE. null. [S.l.], 2007. p. 482–484.

Cardoso, L. B. et al. Estudo dos modos de propagação das ondas guiadas em estruturas cilindricas de aço carbono. In: *VII CONNEPI-Congresso Norte Nordeste de Pesquisa e Inovação*. [S.l.: s.n.], 2012.

Cardoso, L. F.; Silva, V. M.; Segundo, F. C. G. Osciloscópio de baixo custo utilizando a plataforma Arduino. *Anais do Encontro de Computação do Oeste Potiguar ECOP/UFERSA* (ISSN 2526-7574), v. 1, n. 1, 2017.

Chen, X.; Chen, J. Design of an arbitrary waveform signal generator. *Procedia Engineering*, Elsevier, v. 15, p. 2500–2504, 2011.

Cheraghi, N.; Taheri, F. A damage index for structural health monitoring based on the empirical mode decomposition. *Journal of Mechanics of Materials and Structures*, Mathematical Sciences Publishers, v. 2, n. 1, p. 43–61, 2007.

Cooper, I. D. What is a “mapping study?”. *Journal of the Medical Library Association: JMLA*, Medical Library Association, v. 104, n. 1, p. 76, 2016.

Corrêa, L. d. A. Estudo de propagação de ondas em tubos epóxi reforçado com fibra de vidro. 2014.

Cui, L.; Liu, Y.; Soh, C. K. Identification of crack size and orientation in continuous cylindrical structure using macro-fiber composite. *Journal of Intelligent Material Systems and Structures*, Sage Publications Sage UK: London, England, v. 25, n. 5, p. 596–605, 2014.

Dalton, R. P. The propagation of Lamb waves through metallic aircraft fuselage structure. Tese (Doutorado)—Department of Mechanical Engineering, Imperial College London 2000., 2000.

Debnath, L.; Shah, F. A. Wavelet transforms and their applications. [S.l.]: Springer, 2002.

Domingues, M. et al. Explorando a transformada wavelet contínua. Caderno Brasileiro de Ensino de Física, v. 38, n. 3, 2016.

Farias, C. et al. Estudo da propagação das ondas de Lamb em chapas de alumínio com furos de diferentes profundidades. In: *5th Pan American Conference For NDT*, COPAEND, Cancun, México. [S.l.: s.n.], 2011.

Franco, V. R. Monitoramento da integridade em estruturas aeronáuticas. Universidade Estadual Paulista (UNESP), 2009.

Fu, H.; Cohen, R. E. Polarization rotation mechanism for ultrahigh electromechanical response in single-crystal piezoelectrics. *Nature*, Nature Publishing Group, v. 403, n. 6767, p. 281, 2000.

Hettler, J. et al. Application of a probabilistic algorithm for ultrasonic guided wave imaging of carbon composites. *Physics Procedia*, Elsevier, v. 70, p. 664–667, 2015.

Islam, M.; Huang, H. Effects of adhesive thickness on the lamb wave pitch-catch signal using bonded piezoelectric wafer transducers. *Smart Materials and Structures*, IOP Publishing, v. 25, n. 8, p. 085014, 2016.

Ji, H. et al. Investigations on flexural wave propagation and attenuation in a modified one-dimensional acoustic black hole using a laser excitation technique. *Mechanical Systems and Signal Processing*, Elsevier, v. 104, p. 19–35, 2018.

Keulen, C. J.; Yildiz, M.; Suleman, A. Damage detection of composite plates by Lamb wave ultrasonic tomography with a sparse hexagonal network using damage progression trends. *Shock and Vibration*, Hindawi, v. 2014, 2014.

Kobayashi, M. et al. Structural health monitoring of composites using integrated and flexible piezoelectric ultrasonic transducers. *Journal of Intelligent Material Systems and Structures*, Sage Publications Sage UK: London, England, v. 20, n. 8, p. 969–977, 2009.

Kudela, P. et al. Structural health monitoring system based on a concept of Lamb wave focusing by the piezoelectric array. *Mechanical Systems and Signal Processing*, Elsevier, v. 108, p. 21–32, 2018.

Leão, R. J. et al. Simulação da propagação de ondas ultrassônicas longitudinais em materiais estruturais aeroespaciais. [sn], 2012.

Ledesma, N. E. C. Desenvolvimento de um sistema de SHM sem fio e com compensação automática de temperatura. PhD Dissertation – Universidade Estadual Paulista, 2015.

Leucas, L. d. F. Utilização das técnicas de Impedância eletromecânica e ondas de Lamb para identificação de dano em estruturas com rebites. Master Thesis – Universidade Federal de Uberlândia, 2009.

Lu, G. et al. Characterization of ultrasound energy diffusion due to small-size damage on an aluminum plate using piezoceramic transducers. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 17, n. 12, p. 2796, 2017.

Mechbal, N.; Rebillat,M. Damage indexes comparison for the structural health monitoring of a stiffened composite plate. A. Güemes, A. Benjeddou, J. Rodellar and Jinsong Leng, 2017.

Mei, H.; Giurgiutiu, V. Effect of structural damping on the tuning between piezoelectric wafer active sensors and lamb waves. *Journal of Intelligent Material Systems and Structures*, SAGE Publications Sage UK: London, England, v. 29, n. 10, p. 2177–2191, 2018.

Moura Jr, J. d. R. V. Uma contribuição aos sistemas de monitoramento de integridade estrutural aplicada a estruturas aeronáuticas e espaciais. PhD Dissertation – Universidade Federal de Uberlândia, 2008.

Oliveira, A. E. T. d. Uso das ondas de Lamb e Scholte para caracterização de líquidos. Universidade Estadual Paulista (UNESP), 2015.

Overly, T. G.; Park, G.; Farrar, C. R. Development of signal processing tools and hardware for piezoelectric sensor diagnostic processes. In: International Society For Optics And Photonics. Sensor Systems and Networks: Phenomena, Technology, and Applications for NDE and Health Monitoring 2007. [S.l.], 2007. v. 6530, p. 653018.

Palmos, E. Modeling of Lamb waves and application to crack identification. [S.l.], 2009.

Park, H. W. et al. Time reversal active sensing for health monitoring of a composite plate. *Journal of Sound and Vibration*, Elsevier, v. 302, n. 1-2, p. 50–66, 2007.

Pohl, J. et al. Experimental and theoretical analysis of lamb wave generation by piezoceramic actuators for structural health monitoring. *Experimental Mechanics*, Springer, v. 52, n. 4, p. 429–438, 2012.

Possani, D. et al. Ondas ultrassônicas: teoria e aplicações industriais em ensaios não destrutivos. Revista Brasileira de Física Tecnológica Aplicada, v. 4, n. 1, 2017.

Qiao, P.; Fana,W. Lamb wave-based damage imaging method for damage detection of rectangular composite plates. *Structural Monitoring and Maintenance*, Techno-Press, v. 1, n. 4, p. 411–425, 2014.

Rocha, L.; Finzi Neto, R.; Steffen JR, V. SHM baseado em ondas de Lamb e métodos estatísticos para o limiar de detecção de dano aplicado a estruturas de aeronaves. 2017.

Rocha, L. A. d. A. Identificação de dano em estruturas utilizando uma metodologia que integra a técnica da impedância eletromecânica e ondas de Lamb. PhD Dissertation – Universidade Federal de Uberlândia, 2017.

Santos, J. G. d. F. et al. Monitoramento utilizando a técnica de ondas de Lamb para detecção de dano em estruturas metálicas. In: *XXXVII Iberian Latin American Congress on Computational Methods in Engineering*. [S.l.: s.n.], 2016. v. 2, n. 30, p. 46–54.

Santos, M. J. S. F. d. Ondas ultra-sonoras guiadas na caracterização e controlo não destrutivo de materiais. PhD Dissertation – Faculdade de Ciências e Tecnologia de Coimbra, Portugal, 2004.

Saravanan, T. J.; Gopalakrishnan, N.; Rao, N. P. Damage detection in structural element through propagating waves using radially weighted and factored rms. *Measurement*, Elsevier, v. 73, p. 520–538, 2015.

Shen, Y. Structural health monitoring using linear and nonlinear ultrasonic guided waves. 2014.

Soleimanpour, R. Damage detection of defects using linear and nonlinear guided waves. PhD Dissertation – University of Adelaide, Australia, 2016.

Soleimanpour, R.; NG, C.-T. Locating delaminations in laminated composite beams using nonlinear guided waves. Engineering Structures, Elsevier, v. 131, p. 207–219, 2017.

Sun, Z.; Zhang, L.; Rose, J. L. Flexural torsional guided wave mechanics and focusing in pipe. *Journal of Pressure Vessel Technology*, American Society of Mechanical Engineers, v. 127, n. 4, p. 471–478, 2005.

Tsuruta, K. M. et al. Monitoramento de integridade estrutural de materiais compostos sujeitos a impactos empregando a técnica da impedância eletromecânica. Master Thesis – Universidade Federal de Uberlândia, 2008.

Venugopal, V. P.; Wang, G. Modeling and analysis of lamb wave propagation in a beam under lead zirconate titanate actuation and sensing. Journal of Intelligent Material Systems and Structures, SAGE Publications Sage UK: London, England, v. 26, n. 13, p. 1679–1698, 2015.

Viana, D. D.; Formoso, C. T.; Kalsaas, B. T. Waste in construction: a systematic literature review on empirical studies. In: ID Tommelein & CL Pasquire, 20th Annual Conference of the International Group for Lean Construction. San Diego, USA. [S.l.: s.n.], 2012. p. 18–20.

Wang, W. et al. Experimental and numerical validation of guided wave phased arrays integrated within standard data acquisition systems for structural health monitoring. *Structural Control and Health Monitoring*, Wiley Online Library, v. 25, n. 6, p. e2171, 2018.

Yu, L. et al. Corrosion detection with piezoelectric wafer active sensors using pitch-catch waves and cross-time–frequency analysis. *Structural Health Monitoring*, Sage Publications Sage UK: London, England, v. 11, n. 1, p. 83–93, 2012.

Zhang, L. et al. Health monitoring of cuplok scaffold joint connection using piezoceramic transducers and time reversal method. *Smart Materials and Structures*, IOP Publishing, v. 25, n. 3, p. 035010, 2016.

Chapter 14: Machine Learning and Pattern Recognition: Methods and Applications for Integrity Monitoring of Civil Engineering Structures

Chapter details

Chapter DOI:

https://doi.org/10.4322/978-65-86503-71-5.c14

Chapter suggested citation / reference style:

Alves, Vinicius N., et al. (2022). “Machine Learning and Pattern Recognition: Methods and Applications for Integrity Monitoring of Civil Engineering Structures”. *In* Jorge, Ariosto B., et al. (Eds.) *Model-Based and Signal-Based Inverse Methods*, Vol. I, UnB, Brasilia, DF, Brazil, pp. 502–535. Book series in Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity.

P.S.: DOI may be included at the end of citation, for completeness.

Book details

Book: Model-based and Signal-Based Inverse Methods

Edited by: Jorge, Ariosto B., Anflor, Carla T. M., Gomes, Guilherme F., & Carneiro, Sergio H. S.

Volume I of Book Series in:

Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity

Published by: UnB City: Brasilia, DF, Brazil Year: 2022

DOI: https://doi.org/10.4322/978-65-86503-71-5

Machine Learning and Pattern Recognition: Methods and Applications for Integrity Monitoring of Civil Engineering Structures

Vinicius N. Alves[1a], Alexandre A. Cury[*2], Ney Roitman[3a], Carlos Magluta[3b]

[1]Graduate Program in Civil Engineering, Federal University of Ouro Preto, Brazil. E-mail: vinicius.alves@ufop.edu.br
[2]Graduate Program in Civil Engineering, University of Juiz de Fora, Juiz de Fora, Brazil. E-mail: alexandre.cury@ufjf.edu.br
[3]Federal University of Rio de Janeiro, Civil Engineering Department, COPPE, Rio de Janeiro, Brazil. E-mail: roitman@coc.ufrj.br, magluta@coc.ufrj.br

*Corresponding author: alexandre.cury@ufjf.edu.br

Abstract

Structural Health Monitoring (SHM) is a problem that can be addressed at many levels. One of the most promising approaches used in damage assessment problems is based on pattern recognition. The idea is to extract features from data that characterize only the normal condition and to use them as a template or reference. During structural monitoring, several data are measured, and appropriate features need to be extracted and compared to a baseline. Any significant deviations could be considered as structural novelty or possible damage. This chapter presents a collection of novelty detection approaches where the concept of Symbolic Data Analysis (SDA) is used to manipulate raw vibration data (i.e., acceleration measurements). These quantities (transformed into symbolic data) are combined to unsupervised - hierarchy-agglomerative, dynamic clouds, and soft c-means clustering – and supervised classification techniques - Bayesian decision trees, artificial neural networks, and support vector machines applied to SHM. To attest the robustness of these approaches, experimental tests are performed on a simply supported beam considering different damage scenarios, and on a motorway bridge, in France, where thermal variation effects also played a major role. The results obtained confirm the efficiency of the proposed methodologies. Finally, the authors present some practical recommendations about the discussed techniques.

1. Introduction

Studies related to early damage detection are of special concern for civil engineering structures. It is common knowledge that if a damage process is not identified in time, structural systems may undergo serious safety and economic consequences. Traditional methods of damage detection and health monitoring are often based on the variation of structural vibration characteristics, i.e., natural frequencies, damping ratios and mode shapes. Any modification of mechanical properties must be detectable through changes in the modal parameters (Moughty and Casas [2017]). Research in vibration-based damage identification has been rapidly expanding over the last few years. In their survey, Doebling et al. [1996] have addressed several critical issues for future research in damage identification and health monitoring. Although this literature survey is now almost 30 years old, most of the conclusions are still valid despite the large research work done since then.

As previously mentioned, most of the techniques used for damage identification are essentially based on the determination of modal properties through an identification process (Cury et al. [2011]; Hakim and Razak [2014]). Nevertheless, the identification of modal parameters is a sort of filtering process, leading to a loss of information compared to raw data (acceleration measurements). This compression process can erase any small changes due to a structural modification. Furthermore, and this is certainly the major drawback when using modal parameters, modal components are essentially describing an equivalent linear behavior, a feature which may be not exact for the analysis of specific degraded systems (Finotti et al. [2019]). In turn, using raw dynamic measurements (especially if high sampling frequencies are used) leads to the storage of large data sets. Dynamic measurements can easily contain over thousands of values, making an analysis process extensive and prohibitive. Few damage detection methods present in the literature are based on signature principles, but they usually fail when making them practical. In this sense, despite the current processing power of computers, the necessary computational effort to manipulate large datasets remains a problem. A lot of effort has been put in the development of new procedures of damage detection involving not only engineers, but also researchers of different areas such as mathematics, physics, statistics, among others, to improve formulations of well-known techniques or to develop new ones (Cardoso et al. [2019]; Santos et al. [2020]; Zhou et al. [2019]).

The arising problem often faced by structural engineering when it comes to health monitoring is the great amount of data acquired through dynamic tests. More important than gather this information is, of course, the interpretation of these data. It is difficult, however, to analyze this type of (raw) information although extremely important in some cases where the engineer/stakeholder needs to know, in real time, the condition of a given structure. Thus, it would be imperative to assess structural information directly from raw data i.e., accelerations measured *in situ*.

The development of high-performance sensors, precision signal conditioning, analog-to-digital converters, optical or wireless networks, global positioning systems and so on, has drastically changed the vision of structural monitoring, giving engineers a large amount of data and consequently performance indicators. In connection with advanced software for a structural analysis, significant developments can be expected regarding the detection of deterioration mechanisms. These developments have opened the way for a wide range of applications dedicated to efficient operation and maintenance of civil engineering structures.

Detecting structural changes in a timely manner and understanding the mechanical behavior are critical to ensure that the resulting disruption and the economic management issues are optimized. This explains why many expectations have been placed in vibration-based monitoring for structural behavior characterization and novelty detection (diagnosis of abnormal behavior) (Cury and Crémona [2012]). Many novelty techniques have been proposed in the last few years to detect structural damage. The use of most of these techniques yielded interesting and encouraging results. In fact, in some applications, it was possible to identify several structural conditions correctly i.e., to discriminate undamaged conditions from damaged behaviors using modal parameters (natural frequencies, mostly) (Alvandi and Cremona [2006]; Wang [2013]). However, the use of these techniques applied to raw data (accelerations) remains a challenge. To overcome these limitations, this chapter presents the use of a special set of data manipulation techniques. Data mining is the process of extracting hidden patterns or features from data. As more data are gathered in monitoring, data mining is becoming an increasingly important tool to transform these data into information and is being used in a wide range of profiling practices, such as marketing, fraud detection and scientific discovery.

Different types of data can be employed and manipulated in data mining, such as single quantitative or categorical values, interval-valued data, multi-valued categorical data, and modal multi-valued (histograms) (Cury and Crémona [2012]). These types of data are generally called “symbolic data” and they allow representing the variability and uncertainty present in each variable. The development of new methods for data analysis suitable for treating this type of data is the main issue of Symbolic Data Analysis (SDA). This chapter shows how the combination of SDA with classification methods can be used to separate different structural states. The major advantage of such combined approach is that enhanced - yet raw - information is used, i.e., histograms, intervals, etc. and they can be applied to manipulate vibration data (acceleration measurements, for example). When these quantities are converted into symbolic data, this piece of information will be applied to three clustering methods: hierarchy-agglomerative, dynamic clouds and soft c-means clustering and three supervised classification techniques: Bayesian decision trees, artificial neural networks, and support vector machines. The main objective here is to assess structural modifications due to damage or any other abnormal event, such as reinforcement procedures, different types of traffic loads, among others. However, when applying vibration-based damage detection to SHM, changes in vibration signatures are not only based on changes in any physical property. Environmental changes, notably temperature variations, can have a significant effect and it is necessary to take this into account for the evaluation of structural integrity (Martins et al. [2014]; Xia et al. [2013]). Encouraging results are obtained and they show strong evidence that environmental effects play an important role in the field of SHM.

This chapter is organized as follows: Section 1 presents a brief state-of-art about SHM techniques in the framework of civil engineering structures. Section 2 delves into some thoughts on machine learning techniques. Initially, it explains the idea behind the concept of Symbolic Data Analysis. Then, it presents a contextualized explanation about unsupervised and supervised classification methods applied to structural novelty detection. Section 3 focuses on the applications and the results obtained using the proposed approach. Finally, Section 4 presents the final remarks and some recommendations about the practical use of the techniques discussed in this chapter.

2. Machine Learning Techniques Overview

This section intends to give the reader a general perspective about how vibration data can be compressed while keeping their intrinsic characteristics necessary to provide enough information about the structure's dynamic behavior. Next, it explains how these condensed data is inputted to classification techniques, which ultimately will discriminate abnormal structural conditions.

2.1 Symbolic Data Analysis

In general, data acquisition campaigns in civil engineering structures gather thousands of accelerations values measured by several sensors. Consequently, analyzing all these data (classical data) directly may usually be time-consuming or even prohibitive. In this sense, transforming this massive quantity of data into a compact but also rich descriptive type of data (symbolic data) becomes an attractive approach. Let us consider, for instance, a signal X (which is part of a dynamic test) containing 5,000 acceleration values measured by one single sensor (see Fig.1 on the left). There are several ways to transform classical data into symbolic data.

This signal can be represented by:

- a *k*-category histogram: X={1(0.0025), 2(0.0721), 3(0.8546), 4(0.0626), …, k(0.0082)};
- an interquartile interval: X = [-0.012; 0.015];
- a min/max interval: X = [-0.025; 0.025].

Figure 1 (on the right) shows how a classical signal (one sensor) is converted to a symbolic representation. In this case, all acceleration values are projected to the y-axis of coordinates and a 20-category histogram is constructed. In fact, it must be noted that the same representation could be applied to modal parameters, i.e., natural frequencies and mode shapes. In other words, both quantities can be represented by intervals or histograms. Transforming classical data to symbolic data is carried out almost instantaneously, which does not prohibit or make difficult the use of this methodology for a large ensemble of dynamic tests.

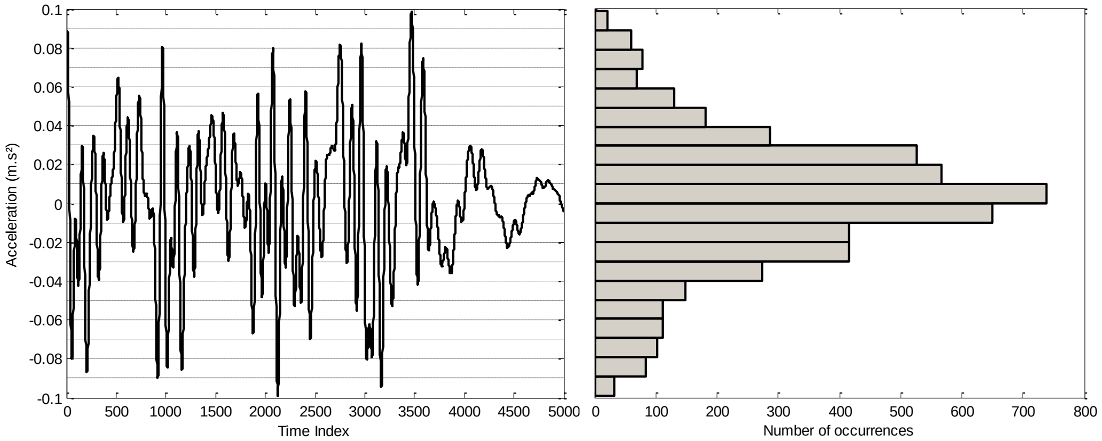


Figure 1 – Example of transforming classical signal to symbolic signal (20-category histogram).

In fact, when this transformation procedure is carried out, two important aspects must be considered. The first one relies on the conservation of some statistical properties of the original data, i.e., the moments of first order (mean value) and second order (variance). Higher order moments (skewness and kurtosis) are not considered here. The second aspect refers to the number of nonzero categories. It is not of interest to keep categories with values equal to zero since they will not contribute to the classification procedures. Thus, in this chapter, 10-category histograms are used in the SDA process, since it has proven to be the most efficient transformation for this type of analysis (Alves et al. [2015].

2.2 Unsupervised classification methods

Data clustering is a common technique for statistical data analysis, which is used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics (Madhulatha [2012]). A clustering procedure can be defined as a way of classifying several objects into different groups. More precisely, it can be described as the partitioning of a data set into subsets (clusters), so that the data in each subset share some common properties. For an appropriate clustering, it is necessary to minimize the within-cluster variation to obtain the most homogeneous clusters as possible and, as a natural consequence, to maximize the between-cluster variation to obtain the most dissimilar clusters among each other. To define these clusters and determine the proximity (or similarity) among tests, it is necessary to define suitable dissimilarity measures. In a common sense, the lower these values are the more similar the objects are and thus, they are gathered in the same cluster. Conversely, the objects allocated into different clusters are the ones that have greater distances between them. Dissimilarity measures can take a variety of forms and some applications might require specific ones. More details can be found in (Billard and Diday [2006]).

In this chapter, three clustering methods are used to discriminate structural conditions: hierarchy-agglomerative, dynamic clouds and fuzzy c-means. The first two are briefly described here since reference (Finotti et al. [2019]) contains additional details. Thus, their formulations are omitted in this chapter. The third method, since it is less documented in this field of research, is described later.

2.2.1 Hierarchy-Agglomerative

Hierarchy-Agglomerative is a bottom-up clustering process, i.e. the process starts with q clusters containing one single test, and proceeds by merging two sub-clusters (C^1 and C^2, say) into one new cluster C. The sub-clusters are merged according to similar criteria for minimizing the within-cluster variation and for maximizing the between-cluster variation. More details can be found in (Billard and Diday [2006]; Finotti et al. [2019]). This clustering method provides a measure of proximity between clusters when using the "difference in height" between them. Fig. 2 illustrates a hierarchy-agglomerative clustering procedure. In this example, clusters 1, 2 and 3 are highlighted. The heights "H(1,2)" and "H(2,3)" represent the distances among them. As "H(1,2)" is greater than "H(2,3)", clusters 2 and 3 are closer (or more similar) than clusters 1 and 2. For this method, the categorical symbolic distance (Billard and Diday [2006]) and the centroid linkage clustering were adopted. These parameters were chosen after adequate results obtained by (Alves et al. [2015]).

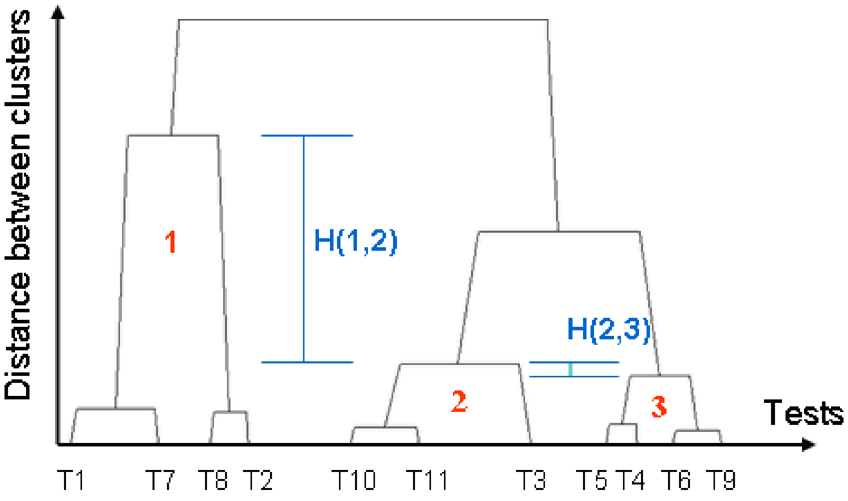


Figure 2 – Example of a hierarchy-agglomerative clustering.

2.2.2 Dynamic Clouds

This clustering method is based on a generalization of the classical dynamical clusters' method (Billard and Diday [2006]). This method consists in minimizing a general optimized criterion that measures the adequacy between the partition and the representation of the clusters, denoted *prototype*. A prototype is a symbolic description model representing a cluster or, in other words, the "average" concept of a cluster. It is used as a reference to calculate the distances among the concepts and to define each cluster.

The algorithm initiates with a set of *k* random prototypes and iteratively applies an allocation phase to place each concept in the cluster where the proximity between concept and prototype is minimal. In this process, a representation phase is performed where the prototypes are updated according to the allocation phase results. This is realized by computing and storing the total sum of the distances between concepts and the prototype in one cluster. The new cluster prototype is the one minimizing this sum. These two phases procedure is repeated until convergence, that is, when the adequacy criterion reaches a stationary value (or at least when reaching a maximum number of iterations).

In general, the dynamic cluster algorithm converges in a few iterations. To improve the quality of the clustering, the algorithm is executed with different initial partitions, and the best configuration is chosen among all the results. Fig. 3 shows a simplified scheme of this clustering method. For this method, the categorical symbolic distance (Billard and Diday [2006]) was adopted. These parameters were chosen after adequate results obtained by (Alves et al. [2015]).

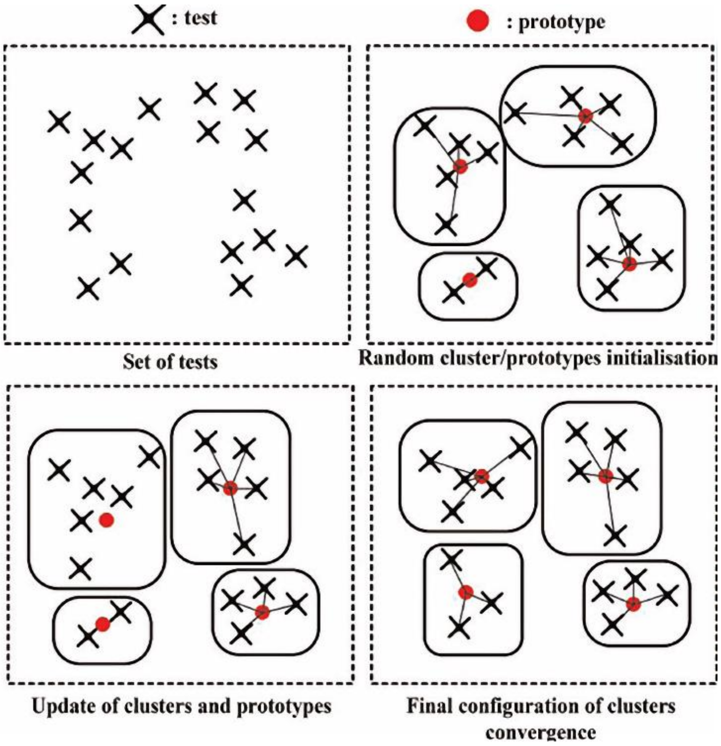

: test
: prototype
Set of tests
Random cluster/prototypes initialisation
Update of clusters and prototypes
Final configuration of clusters convergence

Figure 3 – Scheme representing the dynamic clouds algorithm.

2.2.3 Fuzzy c-means method (FCM)

A fuzzy set is a class of objects that has a continuum of grades of membership. Such a set is characterized by a membership (characteristic) function that assigns a grade of membership that ranges between zero and one to each object (Kroszynski and Zhou

[1998]; Song et al. [2006]). The FCM is an iterative algorithm clustering method that produces optimal c-partitions by minimizing the weighted dissimilarity function (Bezdek [1981]).

Let T_i $(i = 1,,2,...,n)$, which represents a *m*-dimensional vector (*m* is the number of categories of each dynamic test transformed into symbolic data). This notation is used to determine the cluster centers CC_{qr} for the q^{th} cluster and its r^{th} dimension by using the expression given below:

$$CC_{qr} = \frac{\sum_{i=1}^{n} U_{iqr}^{f} T_{ir}}{\sum_{i=1}^{n} U_{iqr}^{f}} \tag{1}$$

where C is the number of the cluster to be made $(2 \le C \le N)$, and f is an appropriate level of cluster fuzziness $f > 1$. U is the membership matrix, which has the size $n \times C \times m$ and is first initialized randomly such that $U_{iqr} \in [0,1]$ and $\sum_{q=1}^{C} U_{iqr} = 1$, for each i and a fixed value of *r*.

Then, the Euclidean distance is evaluated between the i^{th} data point and the q^{th} cluster with respect to the r^{th} dimension with the equation below:

$$D_{iqr} = \left\| T_{ir} - CC_{qr} \right\| \tag{2}$$

In the following step, CC_{qr} is updated in Eq.(1) by recalculating the fuzzy membership matrix U according to D_{iqr} (if $D_{iqr} > 0$), as follows:

$$U_{iqr} = \frac{1}{\sum_{c=1}^{C} \left(\frac{D_{ir}}{D_{icr}} \right)^{f^{2}-1}} \tag{3}$$

This updating iteration is repeated until the changes in U are sufficiently small (such that $(U \le \varepsilon)$, where ε is a predefined termination criterion.

The main difference between hard (hierarchy agglomerative and dynamic clouds) and soft-clustering (FCM) relies on the fact that hard-clustering methods can only assign a given object to a unique cluster. Conversely, soft-clustering methods allow classifying a given object as a part of different the clusters. This allows defining the so-called *pertinence values.* The higher these values are, more likely a given object is to be classified into a cluster. In this chapter, the aforementioned procedure was coupled with symbolic data transformation within Matlab® software framework. In fact, fuzzy c-means procedures are built-in in the Fuzzy Toolbox. The Euclidian distance was used, and the level of fuzziness was set to 2.

2.2.4 Determination of the optimal number of clusters

A common limitation of the clustering methods cited before is the necessity of predefining the number of clusters. In practice, however, this number is usually unknown. To circumvent this drawback, many different stopping rules for determining the optimal number of clusters have been published in the scientific literature. The most detailed and complete comparative study has been carried out by (Milligan and Cooper [1985]). They performed a Monte Carlo evaluation of thirty indexes for the determination of the optimal number of clusters and they investigated the extent to which these indexes were able to detect the correct number of clusters in a series of simulated data sets containing a known structure. In this section, the three best indexes are presented: the CH index, the C^* index and the Γ index (Billard and Diday [2006]).

The general methodology for the evaluation of these three indexes is based on calculating each one of them for a partition $P_q = (C^1, \cdots, C^q)$ containing a different number of clusters. The number of clusters q is arbitrarily chosen, but it is usually greater than the number of clusters considered in the analysis. Since $q = 1$ represents the partition P_1, which only contains the initial cluster, the indexes are not considered in this case.

The CH index is given by:

$$CH(P_j) = \frac{B(P_j)}{W(P_j)} \times \frac{(n-j)}{(j-1)}, j = 2,...,q \tag{4}$$

where $B(P_j)$ is the between-cluster variation, $W(P_j)$ is the total within-cluster variation, n is the total number of tests and j is the number of clusters in the partition P_j. Further details can be found in (Finotti et al. [2019]). When CH has its **maximal absolute value**, then the optimal partition (i.e., the optimal number of clusters) is obtained.

The C^* index can be calculated as:

$$C^*(P_j) = \frac{1}{n}\sum_{k=1}^{j} n_k \frac{(S^k - S^k_{\min})}{(S^k_{\max} - S^k_{\min})}, j = 2,...,q \quad C^* \in [0,1] \tag{5}$$

where n is the total number of tests, n_k is the number of tests of a cluster C^k, S^k represents the sum of distances among the k tests within a cluster C^k, $S^k_{\min}$ is the sum of the k smallest distances among all tests, $S^k_{\max}$ is the sum of the k largest distances among all tests. The optimal partition is given for C^* **minimal absolute value**.

The Γ index is given by:

$$\Gamma(P_j) = \frac{\Gamma_+(P_j) - \Gamma_-(P_j)}{\Gamma_+(P_j) + \Gamma_-(P_j)}, \; j = 2,...,q \qquad \Gamma \le 1 \tag{6}$$

where $\Gamma_+(P_j)$ represents the number of within-cluster distances smaller than the between-cluster distances and $\Gamma_-(P_j)$ is the number of within-cluster distances larger than the between-cluster distances. The optimal partition is given for **4 maximal absolute value**.

2.3 Supervised classification methods

This section presents an overview of three supervised classification methods. Firstly, some concepts regarding Bayesian Decision Trees (BDT) and its applicability are explained, followed by a brief discussion about how Neural Networks (NN) are used in this study. Finally, a general idea of Support Vector Machines (SVM) applied to classification problems is presented. These methods were selected following the works of Martins et al. [2014] and Xia et al. [2013]. In those references, several supervised classification techniques coupled with SDA were tested using either artificial or real controlled (labeled) data. In general, BDT, NN and SVM achieved the best correct classification ratios.

2.3.1 Bayesian Decision Trees

Bayesian Decision Trees (BDT) are a decision procedure that can solve classification problems (Billard and Diday [2006]). The general idea of this method is to classify a particular object (dynamic test, in this case) into one of the classes (groups) previously defined i.e., in the training set. For instance, let $\Omega = \{T_1, T_2, ..., T_n\}$ be a set of n dynamic tests and C a class variable containing values varying within $\{1,...,m\}$ where m is the number of classes. This discriminant analysis tries to predict the unknown value C for a given test $\tilde{T}$ according to its p features (the symbolic representations of sensors) and a training set. The steps of this symbolic classification procedure are to represent a given partition in the form of a BDT and create a rule capable of assigning a new test to one class of a prior partition.

Now, let us consider that each test T_i is described by two types of variables, considering symbolic signals:

- p sensors described as k-category histograms;
- a class variable C: this variable specifies the class of a test in the training set in the form of a unique value (1, 2,…, m).

Let T_1 denote a test belonging to class 1 and T_2 a test belonging to class 2 within the training set. The goal is to classify a test $\tilde{T}$ into one of these two classes. In this example, all tests are described by two sensors only (s_1, s_2) .

In the framework of the recursive algorithm, each node division step is performed according to a single variable (suitably chosen) and to "yes/no" answers to specific binary questions. The set of binary questions that will be useful for the recursive partition procedure is based on the Bayesian rule (Billard and Diday [2006]). This rule can be enounced as follows: suppose that for a set of tests Ω, the test $\tilde{T}$ is distributed with density functions:

$$f_j(\tilde{T}), \quad j=1,...,m \tag{7}$$

Density functions are generally unknown and need to be estimated. One way to solve this issue is to use the Kernel method (Kroszynski and Zhou [1998]), which can reasonably approximate density functions. The Kernel estimator is defined as:

$$f_j(\tilde{T}) \cong \frac{1}{n_j}\sum_{s=1}^{n_j}\sum_{r=1}^{p}\sum_{q=1}^{k}\frac{1}{h}K\left(\frac{\tilde{T}^{r,q}-T_s^{r,q}}{h}\right),\ j=1,...,m \tag{8}$$

where n_j is the number of tests within the class *j*, $h>0$ is a smoothing parameter and *K* is called kernel, which is symmetric, continuous and can be evaluated by:

$$K\left(\frac{\tilde{T}^{r,q}-T_s^{r,q}}{h}\right)=\frac{1}{\sqrt{2\pi}}e^{-\frac{(\tilde{T}^{r,q}-T_s^{r,q})^2}{2h^2}},s=1,...,n_j;r=1,...,p;q=1,...,k \tag{9}$$

where $T_s^{r,q}$ is the *q*th category of the *r*th sensor corresponding to the *s*th test of a given class *j*.

By definition, the Bayesian rule assigns $\tilde{T}$ to the class with the largest value of $P_i \times f_j(\tilde{T})$, where P_j $(j=1,...,m)$ represent the prior probabilities of each class. For the previous example introducing two classes, the following question is considered (Billard and Diday [2006]):

$$\text{Is } P_1 \times f_1(\tilde{T}) > P_2 \times f_2(\tilde{T}) \ ? \tag{10}$$

If the first term is greater than the second one, then the test $\tilde{T}$ is considered to belong to class 1; otherwise, the test is assigned to class 2. Although this example contains only two groups, this procedure can easily be extended to problems which have several classes. In fact, if there are *m* possible classes, one will have $m-1$ questions to assign a new test to a given class.

In addition, if prior probabilities are unknown, two choices are available to determine them:

- uniform prior probabilities based on the number of classes:

$$p_j=\frac{1}{m}, \qquad j=1,...,m \tag{11}$$

- prior probabilities based on the proportions observed in the training set:

$$p_j=\frac{n_j}{n} \quad \text{with} \quad n=\sum_{j=1}^{m} n_j \tag{12}$$

where *n* is the number of tests in the training set.

As already mentioned, tests are classified into different nodes according to “cut rules” and splitting criteria. To define each split, the most discriminant feature (sensor) must be used, that is, the feature capable to optimally separate tests into different classes. This “optimal” feature is called *cut variable* and it leads to the “purest” nodes of the Bayesian tree. The purity of a node is the number of tests that are correctly classified in it (by leave-one-out and/or bootstrap methods), where the classification is verified in relation to the class variable *C*. This number can be obtained by the Bayesian rule computed for each feature. The choice of the most discriminant *cut variable* will result in selecting the feature that minimizes the impurity measure i.e., the number of misclassified tests. Finally, the *cut variable* and its associated *cut value* will form the *cut rule*, which will be used to properly classify a new test (Song et al. [2006]).

2.3.2 Artificial Neural Networks

Neural networks have proven themselves as proficient classifiers and are particularly well suited for addressing non-linear problems. Given the non-linear nature of real-world phenomena, like the classification of dynamic tests, neural networks are a potential approach for dealing with this problem. Commonly, neural networks are adjusted or trained, so that a particular input leads to a specific target output (*supervised learning*). In this case, the network is adjusted based on a comparison of the output and the target, until the network output matches the target. Supervised learning is a machine learning technique for deducing mapping functions from a training dataset consisted of input-output pairs. The goal is to predict output values of the mapping function for any valid input after having seen some training examples (i.e., pairs of inputs and target outputs). To achieve this, the network must generalize from the training data to unseen situations in a "reasonable" way.

Mapping functions are obtained through an optimization scheme based on the evaluation of the mean-squared error (explained later). This scheme tries to minimize the average squared error between the network's output and the target value over all the training dataset pairs. In this chapter, a feed-forward multilayer perceptron neural network is used for classifying dynamic tests. Multilayer networks use a variety of learning techniques, the most popular being back-propagation. In this case, the output values are compared with the correct answer to compute the value of some predefined error-function and the error is then fed back through the network (Bezdek [1981]). Using this information, the algorithm adjusts the weights of each connection to reduce the value of the error function by some small amount. In this study, training automatically stops when generalization stops improving, which is indicated by an error increase in the validation samples. At this point, it is said that the network is “trained”.

Let us consider, a one-hidden-layer MLP with *N* hidden neurons where the inputs are the dynamic tests $T_i \ (i = 1,...,n)$ which are symbolic representations of signals as described in section 2. They are defined by $p \times k$ matrices, where p is the number of sensors and k is the number of categories. Like the BDT method, outputs are set to represent the target vectors (labels) corresponding to each class i.e., $\{1, 2, ..., m\}$. These outputs are transformed into a binary notation according to the number of classes used. Target vectors have m elements, where for each target vector, one element is 1 and the others are 0. This defines a problem where inputs are to be classified into m different classes.

The general formulation for a neural network can be written as follows: each input x_i (which can either be an input of the network or of a layer) is multiplied by adjustable weights denoted w_{il} before being fed to the l^{th} neuron in the hidden/output layers, yielding (Milligan and Cooper [1985]):

$$o_j = f\left(\sum_{i=1}^{N} w_{il} x_i + b_i\right) \quad j = 1, ..., n \tag{13}$$

where o_j, b_i and f represents the output (either from a layer or from the network), the bias of each perceptron and the activation function, respectively.

To adjust weights properly, a general method for non-linear optimization called *gradient descent* is applied (Milligan and Cooper [1985]). Briefly, the derivative of the error function with respect to the network weights is calculated and the weights are then changed such that the error decreases (thus going downhill on the surface of the error function). Eq. 14 shows the expression to evaluate the updated weights of this network:

$$w_{il}(t+1) = w_{il}(t) - \eta \frac{\partial J}{\partial w_{il}} \tag{14}$$

where η is the learning rate, t is the iteration step and J is mean error for a perceptron, written as:

$$J = \frac{1}{2n} \sum_{j=1}^{n} (C_j - y_j)^2 \tag{15}$$

where C_j represent the desired outputs (targets) and y_j the observed outputs (evaluated by the neural network). For the simulations presented in this chapter, a free Matlab toolbox named Netlab developed by Bezdek [1981] was used. This toolbox allows performing several types of supervised multi-class classifications. The architecture of the NN consisted of a 20-neuron, one hidden-layer network using sigmoid activation function (hidden-layer) and linear function (output layer).

2.3.3 Support Vector Machines

Support Vector Machines (SVM) are a useful technique for data classification problems. As usual, the objective is to separate two different classes by a function which is induced from available examples (training dataset). SVMs were first suggested in the 1960s for classification and have recently become an area of intense research due to developments in the techniques and theory coupled with extensions to regression and density estimation (Alves et al. [2015b]; van Overschee and de Moor [1996]). This technique came up from statistical learning theory and is based on the structural risk minimization principle. Commonly, SVMs are used for two-class classification problems. However, this can be extended from 2-class classifications to m-class classification problems by constructing m two-class classifiers. Thus, for multi-class SVM methods, either several binary classifiers must be constructed, or a larger optimization problem is needed. In the work of Hsu et al. [2002], a decomposition implementation for "all-together" methods, "one-against-all", "one-against-one" and Directed Acyclic Graph Support Vector Machines (DAG-SVM) were tested and compared.

In this section, a very brief review of SVM classification is given; further information can be found in references (Alves et al. [2015b]; van Overschee and de Moor [1996]). Let us consider $\{x_i, y_i\}_{i=1}^{n}$ a training dataset, where x_i are the input samples (symbolic representations of signals) and $y \in \{+1, -1\}$ the labels of classes and n the number of samples, respectively. According to Vapnik's original formulation, the hyperplane (HP) is defined as $wx + b = 0$, where x is a point lying on the HP, w determines the orientation of the HP, and b is the bias of the distance of the HP from the origin. If this HP maximizes the margin, then the following inequality is valid for all input data:

$$(b + w^T x_i) y_i \geq 1, \text{for all } x_i,\ i = 1,2, \dots, n \tag{16}$$

The margin of the HP is equal to $2/\|w\|$, any training turples that fall on HP_1 or HP_2 (i.e., the sides defining the margin) are called support vectors. Thus, the problem is the maximization of the margin by minimizing $\|w\|/2$ subject to Eq.(16).

Lagrange multipliers $\alpha_i(\alpha_i > 0, 1 = 1, \dots, n)$ are used to solve $J_p = -\sum_{i=1}^{n} \alpha_i [(b + w^T x_i) y_i - 1] + \|w\|^2/2$. After minimizing J_p with respect to both $\boldsymbol{w}$ and b, the optimal values are given by: $w^* = \sum_{i=1}^{n} \alpha_i^* y_i x_i$. The so-called dual problem can be described as:

$$J_p(\alpha) = -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i x_j + \sum_{i=1}^{n} \alpha_i \tag{17}$$

Thus, the linear decision function is created by solving the dual optimization function, which can be obtained by:

$$f(x) = sgn\left(\sum_{i=1}^{n} \alpha_i^* y_i x_i x^T + b^* \right) \tag{18}$$

where a_i^* are the optimal Lagrange multipliers.

For input data with a high noise level, SVM uses soft margins that can be expressed as follows with the introduction of non-negative slack variables ξ_i, $i = 1, \dots, n$:

$$\left(b + \boldsymbol{w}^{*T} x_i\right) y_i \geq 1 - \xi_i, \text{ for } i = 1,2, \dots, n \tag{19}$$

To obtain the optimum separation HP, it should be minimized the $\psi = \mathrm{C}\sum_{i=1}^{n} \xi_i + \frac{1}{2}\|w\|^2$ subject to Eq. (19), where C is the penalty parameter. In the nonlinearly separable cases, the SVM maps the training points, nonlinearly, to a high dimensional feature space using kernel function $K(x_i, x) = \varphi(x_i).\varphi(x)$, where linear separation may be possible. The kernel function, $K(x, x_i)$, typically has multiple alternatives. In this chapter, the Radial Basis Function (RBF) is considered:

$$K(x_i, x) = \exp\left(-\|x_i - x\|^2 / 2g^2\right) \tag{20}$$

where $g \epsilon R^+$ is a constant.

After a kernel function is selected, Eq. (20) becomes:

$$f(x) = sgn\left[\sum_{i=1}^{n} a_i^* y_i K(x_i, x) + b^*\right] \tag{21}$$

In general, RBF kernel is a reasonable first choice in training SVM, thus this kernel was used in this study. The selection of parameters C (the penalty term) and g (the basis width of the kernel) for the SVM model influences the classification accuracy significantly. In this work, an iterative algorithm was used during the validation phase to determine their optimal values. The SVM model in this chapter was implemented using the software LibSVM developed by Chang and Lin [2021].

3. Experimental applications and results

3.1 Simply supported beam

This section presents the experimental tests conducted at COPPE/UFRJ laboratory on a simply supported steel beam depicted in Figure 4.

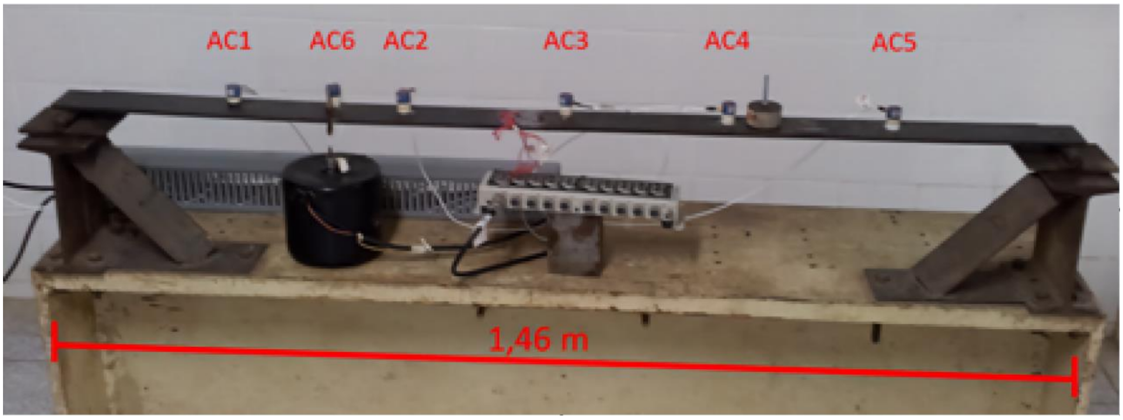


Figure 4: Instrumented steel beam.

The beam is 1.46 m long with rectangular cross-section (76,2 x 8,0 mm) and was instrumented with six piezoelectric accelerometers (PCB, 336C31). Data acquisition was carried out using Lynx ADS2002 equipment, which essentially is a conditioning/amplifier regulating system.

The present study considered two types of dynamic excitation: impact tests (using an impact hammer) and random vibration tests (using a shaker). For the impact tests, an impact hammer was used at each 10 seconds of the dynamic tests. The random excitation was applied throughout the duration of the tests. Six acquisition campaigns were performed, and, for each campaign, three tests were conducted. Thus, 18 dynamic tests recorded in total. Each one of them lasted for about 10 minutes with a sampling rate of 0,00025s (4000 Hz). This means that, for each test, 2.4 million values were recorded per sensor. Figure 5 shows an example of the dynamic tests performed under impact excitation (left) and random vibration (right).

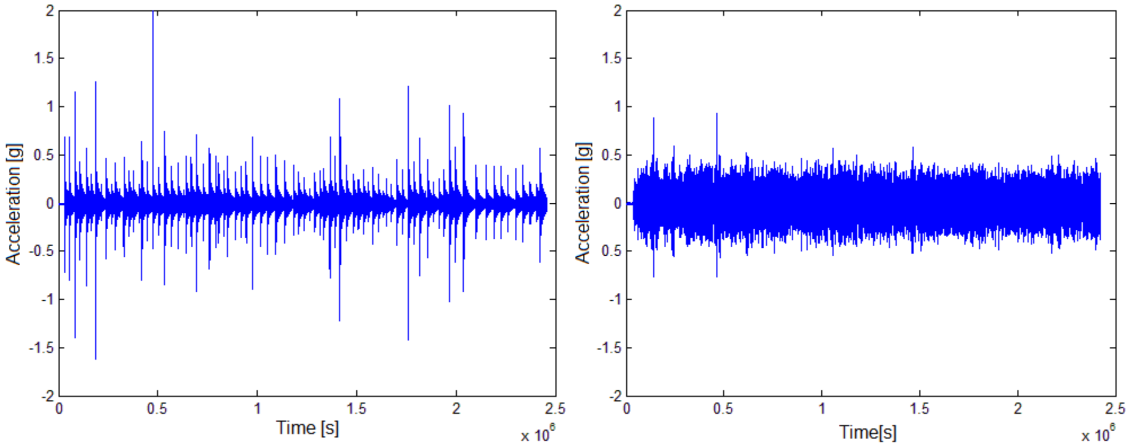


Figure 5: Example of vibration tests performed in laboratory: impact (left) and random (right).

The first campaign comprised the beam without damage. For the second campaign (Level 1), an eccentric mass (0.5 kg) was positioned at 102.7cm from the left support of the beam (between accelerometers AC5 and AC6). In fact, this second campaign was conducted so it could produce a reversible, non-permanent, “damage scenario”. The third campaign (Level 2) considered a small damage inflicted to the beam (a 12 mm round hole) imposed at the same position of the mass, which was removed beforehand. The fourth campaign (Level 3) consisted in increasing the hole to 16 mm. During the fifth and sixth campaigns (Levels 4 and 5), the hole was increased to 22.5 mm and 32 mm diameter, respectively. Figure 6 depicts the 12 mm hole imposed to the beam. It should be noted that these campaigns simulated very small structural damage, which makes this study even more challenging.

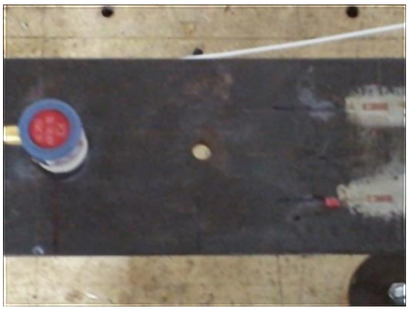


Figure 6: 12-mm hole imposed to the beam.

Modal identification was performed using Stochastic Realization Method algorithm (van Overschee and de Moor [1996]). Tables 1 and 2 present the mean values and respective standard deviations for the first five natural frequencies identified under impact and random vibration. Standard deviation values were rather low, which corroborates the overall quality of the modal identification procedure. In general, it is possible to observe a slight decrease of all five natural frequencies as damage increases. Exceptions occur and may be due to the modal identification procedure, mainly in the higher vibration modes where the signal/noise ratio is low. This agrees with results commonly presented in the literature. If one considers, however, confidence intervals i.e., mean values ± standard deviation, the values for natural frequencies from all damage scenarios would superpose themselves. Thus, from a statistical point-of-view, it is not possible to state that the deviation of these frequencies indicates a structural change. These results show that a more detailed analysis is necessary, but also considering a probabilistic approach, such as SDA.

Table 1. Frequency deviation according to each damage level (mean value and standard deviation - impact vibration).

Damage Level	Freq.#1	Freq. #2	Freq.#3	Freq.#4	Freq.#5
Undamaged	8,30 ± 0,170	33,31 ± 0,025	73,99 ± 0,011	134,90 ± 0,219	205,83 ± 0,135
Level 1	7,90 ± 0,196	31,40 ± 0,036	73,56 ± 0,062	131,63 ± 0,054	195,20 ± 0,023
Level 2	8,26 ± 0,060	33,24 ± 0,025	74,30 ± 0,013	136,23 ± 0,091	205,10 ± 0,041
Level 3	8,25 ± 0,064	33,21 ± 0,026	74,31 ± 0,0106	135,63 ± 0,158	205,57 ± 0,148
Level 4	8,25 ± 0,078	33,23 ± 0,056	74,66 ± 0,044	132,63 ± 0,112	205,90 ± 0,099
Level 5	8,23 ± 0,098	33,01 ± 0,064	74,19 ± 0,077	134,07 ± 0,165	204,30 ± 0,107

Table 2. Frequency deviation according to each damage level (mean value and standard deviation - random vibration).

Damage Level	Freq.#1	Freq. #2	Freq.#3	Freq.#4	Freq.#5
Undamaged	8,30 ± 0,170	33,31 ± 0,025	73,99 ± 0,011	134,90 ± 0,219	205,83 ± 0,135
Level 1	7,90 ± 0,196	31,40 ± 0,036	73,56 ± 0,062	131,63 ± 0,054	195,20 ± 0,023
Level 2	8,26 ± 0,060	33,24 ± 0,025	74,30 ± 0,013	136,23 ± 0,091	205,10 ± 0,041
Level 3	8,25 ± 0,064	33,21 ± 0,026	74,31 ± 0,0106	135,63 ± 0,158	205,57 ± 0,148
Level 4	8,25 ± 0,078	33,23 ± 0,056	74,66 ± 0,044	132,63 ± 0,112	205,90 ± 0,099
Level 5	8,23 ± 0,098	33,01 ± 0,064	74,19 ± 0,077	134,07 ± 0,165	204,30 ± 0,107

Mode shapes are omitted, since their study is not the focus of this chapter, but they followed a series of sinusoidal curves, as one should expect for simply supported beams.

3.1.1 – Results (unsupervised classification methods)

The procedure conducted henceforth in this chapter follows these steps:

1. Convert acceleration measurements into symbolic data (10-category histograms) as explained in section 2;
2. Use symbolic histograms (step 1) as inputs for the clustering techniques (hierarchy-agglomerative, dynamic clouds and FCM) considering different number of clusters (greater than 2);
3. Evaluate the optimal number of clusters using indexes CH, C^* and Γ applied to dynamic clouds and FCM methods (these are the only clustering methods in this chapter that require an initial number of clusters);
4. Retrieve partition of clusters obtained considering the optimal number of clusters (step 3).

Furthermore, it is important to emphasize that this entire procedure strongly depends on the quality of the input data. In this case, if accelerations measurements present any type of problem (bad sampling, missing data, incorrect measurement, etc.), the results obtained from the clustering methods will be compromised. Thus, it is imperative to assure, in firsthand, that the data used in the analysis is adequate.

The proposed approach is thus applied to accelerations measured during the experimental tests carried out in laboratory. Let us recall that the main objective here is to try to separate the six structural conditions (undamaged and damaged levels 1, 2, 3, 4 and 5) into six different clusters, using raw data (signals) only. Moreover, each type of forced vibration is considered separately.

The first simulations comprehend dynamic tests obtained under impact vibration only. As explained at the beginning of section 3, the proposed approach follows four steps. Once the clustering procedure is performed considering different number of clusters (in this case, it varied from 2 to 10), the indexes are evaluated.

Table 3 contains the first results. The last column shows the optimal number of clusters according to each index. It can be observed that both CH and Γ indicate 6 clusters (C^*, however, indicates 7). It can also be seen that all tests corresponding to structural conditions "undamaged", "damaged – level 1" and "damaged – level 3" are correctly classified. However, for the cluster "damaged – level 2", only one third of tests is classified properly. In general, results are very good and show that it is possible to extract information from raw data. Moreover, all clustering methods performed quite similarly. This does not mean, however, that all techniques will always yield the same results.

To attest the efficiency of the fuzzy c-means method, it is possible to evaluate the pertinence values, which quantify the certainty of the classification. If this index is 1, it means that the method is completely sure about its classification (which does not imply that the result is correct). Otherwise, if the pertinence value is close to 0, the method is "not sure" about its classification. In this sense, it is possible to observe that pertinence values for the classification showed in Table 3 are relatively high, validating the clustering procedure and, in this case, the results obtained.

Table 3: Percentages of correct classification (impact vibration).

(%)	Dynamic Clouds	Hierarchy Agglomerative	FCM	Pertinence value	Indexes
Undamaged	100	100	100	0,72	$CH = 6$
Level 1	100	100	100	0,92	$C^* = 7$
Level 2	33	33	33	0,79	$\Gamma = 6$
Level 3	100	100	100	0,83	
Level 4	100	0	100	0,81	
Level 5	66	66	66	0,74	
Average	**83**	**67**	**83**	-	

When it comes to random vibration tests, results are not as adequate as observed with impact vibration tests. Table 4 shows the percentages of correct classification for the six clusters and the respective pertinence values for the fuzzy c-means method. Once more, both CH and Γ indicate 6 clusters but C^*, however, indicates 9. This latter tends to indicate a higher number of clusters, as observed in references (Billard and Diday [2006]; Cury and Crémona [2012]; Finotti et al. [2019]).

Table 4: Percentages of correct classification (random vibration).

(%)	Dynamic Clouds	Hierarchy Agglomerative	FCM	Pertinence value	Indexes
Undamaged	66	66	66	0,69	$CH = 6$
Level 1	33	33	66	0,63	$C^* = 9$
Level 2	33	100	33	0,58	$\Gamma = 6$
Level 3	66	33	66	0,67	
Level 4	100	100	100	0,77	
Level 5	33	33	33	0,61	
Average	**55**	**61**	**61**	-	

In this case, only one cluster is pure (for damage level 4). All other clusters are mixed, showing a difficulty for those methods to discriminate these structural states under random vibration. Although the classification ratios are lower, it is interesting to notice that the pertinence values also yield low scores (except for the cluster corresponding to level 4).

As a further analysis of this experimental application, the clustering procedures are carried out considering only damage levels 2, 4 and 5 (which corresponds to holes with 12, 22.5 and 32mm diameter). Once again, it should be noted that these campaigns simulated very small structural damage, which makes this study even more challenging.

Table 5 presents the classification ratios obtained considering impact vibration tests. It can be observed that the results improve significantly. Now, almost all classifications are perfect, except for the last cluster where only one third of the tests were classified incorrectly. In this analysis, all indexes indicated 4 clusters as the optimal partition.

Table 5: Percentages of correct classification (impact vibration).

(%)	Dynamic Clouds	Hierarchy Agglomerative	FCM	Pertinence value	Indexes
Undamaged	100	100	100	0,74	$CH = 4$
Level 2	100	100	100	0,91	$C^* = 4$
Level 4	100	100	100	0,84	$\Gamma = 4$
Level 5	66	66	66	0,74	
Average	**92**	**92**	**92**	-	

Similarly, even when random vibration tests are considered, classification results improved as well (see Table 6). In this case, however, index C^* has again pointed out a higher number of optimal clusters (5).

Table 6: Percentages of correct classification (random vibration).

(%)	Dynamic Clouds	Hierarchy Agglomerative	FCM	Pertinence value	Indexes
Undamaged	66	33	33	0,68	$CH = 4$
Level 2	100	100	66	0,76	$C^{*} = 5$
Level 4	100	100	100	0,77	$\Gamma = 4$
Level 5	66	66	66	0,74	
Average	**83**	**75**	**77**	-	

These last results can be partially explained by the fact that all these damage levels represent a more discriminative sequence of structural degradation. If damage levels are too similar, the proposed approach might not yield perfect classification scores. Nonetheless, it must be kept in mind that these results were obtained using raw data with no signal processing procedure whatsoever.

In general, better classification ratios were achieved when the structure was under impact vibration. In fact, when random vibration is considered, the transformation procedure to symbolic data (histograms) carries the noise from the excitation, thus misrepresenting the description of the dynamic test. Then, these “noisy” descriptions may mislead the clustering procedures into wrong classifications.

3.1.2 Results (supervised classification methods)

The procedure conducted henceforth follows two steps: i) transformation of acceleration measurements into symbolic data (10-category histograms) as explained in section 2; ii) use of these symbolic representations of acceleration measurements as inputs for the classification techniques (BDT, NN and SVM).

As previously mentioned, each dynamic test has 2.4 million points recorded per sensor. To make the classification analysis more robust, each test is subdivided into 24 ‘subtests’ having 1.0 million points per sensor. Thus, instead of employing the 18 original dynamic tests for classification, now there are 18x24 = 432 tests to classify. This is important because all classification methods use training, validation and testing groups. If these groups are small or poorly rep-resented, the classification results may be affected.

The architecture of the NN consisted of a 20-neuron, one hidden-layer network using sigmoid activation function (hidden-layer) and linear function (output layer). This architecture was chosen after previous simulations, considering different numbers of layers and neurons. The learning rate was set to 0.001. For SVM classifier, the RBF kernel was used, and its parameters were chosen via an iterative algorithm during the validation phase.

The first simulations correspond to the impact excitation case and use only 30% of the set of dynamic tests for training, 10% for validation and 60% for testing. This means that the 432 tests are distributed over three groups: 130 tests for the training dataset, 43 tests for the validation dataset and 259 tests for the testing dataset. Since this distribution can be randomly performed, 10000 simulations (10000 different groups of training, validation, and testing) are generated. This strategy was used following the work of references (Billard and Diday [2006]; Milligan and Cooper [1985]). Table 6 summarizes the results, showing the best, average, and worst classification ratios for each simulation. Considering the first column of this table, for example, it is possible to observe that the BDT has its best performance with 89% of correct classification. In average, it classifies 76% of tests correctly. Its worst performance occurs when it classifies only 51% of tests correctly among the 10000 simulations.

When the training dataset increases to 40%, the testing dataset reduces to 50%. Now, the probabilities of true detection improve significantly (average and worst ratios). Finally, when 50% of tests are used for training, all methods achieve their best results. It can be observed that the SVM reaches better results than the other two techniques. This might be because the SVM could better adjust "separation thresholds" for each damage state. Also, the BDT yield slightly worse probabilities of true detection compared to the NN. This could be explained by the fact that the BDT are not a true learning procedure. In fact, new tests are classified according to logical questions. If the six groups (corresponding to the damage levels) do not provide a meaningful representation of each structural state, the classification may be compromised.

In summary, what is important to extract from these simulations is the average values of true classification. Thus, both NN and SVM yield very satisfactory results. This shows that the pro-posed approach can classify structural conditions using only raw data recorded *in situ*.

Similarly, the simulations are performed using tests recorded under random excitation. Once again, as the training dataset increases, correct classification rates also improve. Moreover, both NN and SVM methods yield higher classification ratios.

In general, better classification ratios are obtained when the structure was under impact vibration. In fact, when random vibration is considered, the transformation procedure to symbolic da-ta (histograms) carries the noise from the excitation, thus misrepresenting the description of the dynamic test. Then, these "noisy" descriptions may mislead the classification procedures into wrong classifications. The authors acknowledge this drawback and further research are pointing into this direction.

Tables 7 and 8 omit, however, an important information: the number of occurrences for "worst" and "best" classification ratios. In all simulations, considering the impact excitation, the number of occurrences for worst ratios is lower than 10% of the total (967 simulations). This means that from all the 10000 simulations, less than 1000 reach their worst result. For the best ratios, 1654 simulations reach their best ratio (16,5%). The remaining simulations yield results between worst and best ratios with the corresponding average value presented in Table 6. Regarding the random excitation tests, results are slightly worse. In that case, 12,7% (1275) of the simulations yield the worst ratio as only 9,8% (982) produce the best ratios. Again, the remaining results oscillate between worst and best ratios with the corresponding average value presented in Table 7.

Table 7. Probabilities of true classification using raw dynamic measurements (impact excitation).

	30% Tr., 10% V, 60% T			40% Tr., 10%V, 50% T			50% Tr., 10%V, 40% T		
Method	BDT	NN	SVM	BDT	NN	SVM	BDT	NN	SVM
Best (%)	89	100	100	94	100	100	94	100	100
Average (%)	76	85	88	68	90	91	67	93	95
Worst (%)	51	57	64	60	56	65	58	68	73

Table 8. Probabilities of true classification using raw dynamic measurements (random excitation).

	30% Tr., 10% V, 60% T			40% Tr., 10%V, 50% T			50% Tr., 10%V, 40% T		
Method	BDT	NN	SVM	BDT	NN	SVM	BDT	NN	SVM
Best (%)	88	96	96	90	99	99	91	100	100
Average (%)	65	83	84	65	89	89	64	91	94
Worst (%)	45	47	54	41	50	57	58	60	69

3.2 PI-57 motorway bridge

The PI-57 Bridge is a double-deck bridge located near the town of Senlis in France, crossing the Oise River, and carrying the A1 motorway, which connects Paris to Lille (Fig. 7). The bridge, a 116.50 m long, cast-in-place, post-tensioned segmental structure built in 1965, consists of three continuous spans of 18.00 m, 80.50 m, 18.00 m (Fig. 8).

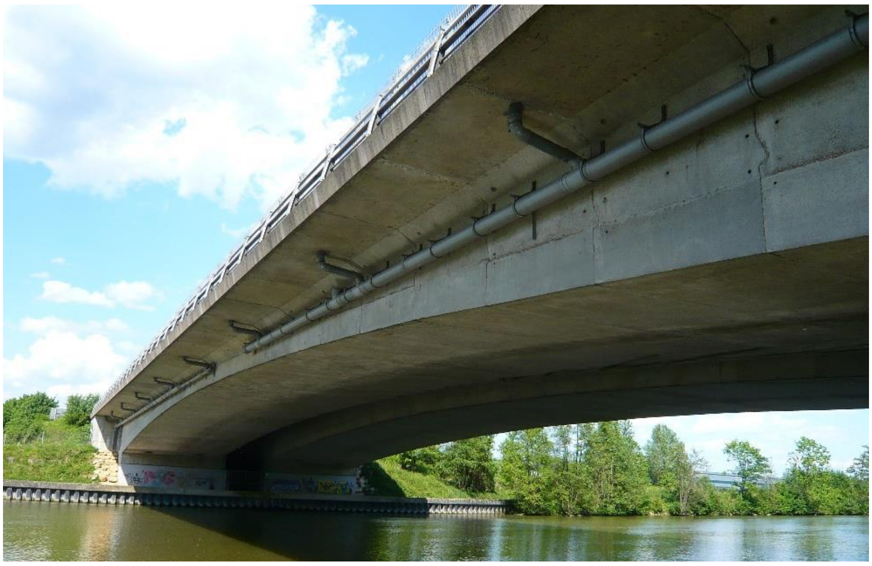


Figure 7. PI-57 Bridge.

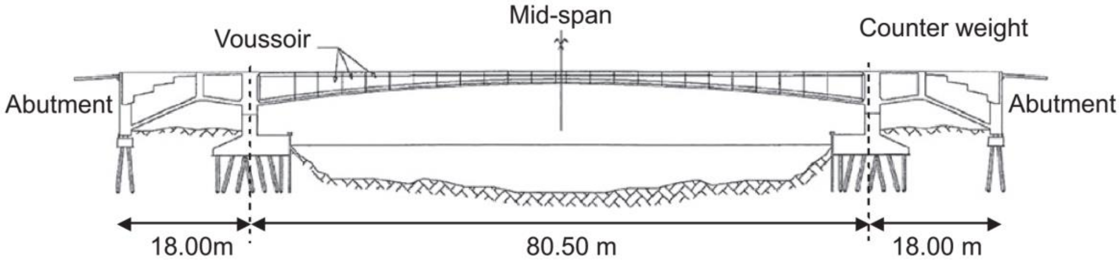


Figure 8. Bridge elevation.

The two lateral spans play the role of counterweights. This slender and elegant structure experienced various problems and distresses during and after its construction, resulting in localized cracking and increasing deflection in the central part. These problems were mainly due to insufficient prestressing because of lowering shrinkage and creep effects and a limited knowledge regarding thermal stresses at the time of construction. Because of the potential risk of cracking in the deck, numerical studies showed that the long-term integrity could be affected if corrective measures were not immediately taken. Based on these technical evaluations and considering the structure's importance, the concessionary motorway company (SANEF) decided to strengthen the two decks. Additional longitudinal prestressing (32,000 kN) would correct the lack of sufficient prestressing. The reinforcement works took place during the summer of 2009 (Alves et al. [2015b]). The strengthening consisted in reducing the tensile stresses under live loads to 1.5 MPa at the bottom part of the bridge cross-sections. These tensile stresses could reach 5.10 MPa at the mid-span cross-section. The external pre-stressing should induce at least 6.60 MPa compressive stresses (with a straight profile for the external prestressing cables). The anchorages were placed on the backside of the cross-girders located on the bridge piles. The total prestressing force has been evaluated to 32,000 kN corresponding to eight 19T15S cables (Fig. 9).

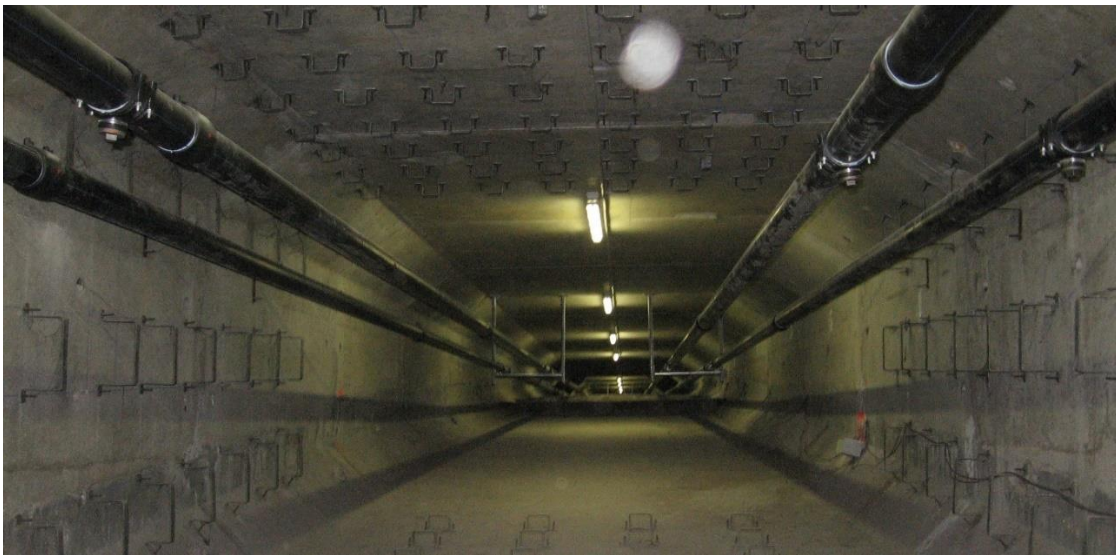


Figure 9. General view of the external prestressing cables installed in the bridge deck.

With this strengthening, the displacement at midspan decreased from 2.44 cm to 0.69 cm under SLS dead and live load effects. A small longitudinal displacement was expected (4.02 mm). No extra displacements occurred on the piles. To check on one hand the variability of the structural behavior due to thermal effects and on the other hand, to evaluate the efficiency of the strengthening procedure, a vibration-based monitoring was conducted: it consisted in installing accelerometers before and after reinforcement. The first campaign of measurements took place between October 15, 2008, and April 3, 2009. The second campaign, after reinforcement, started on October 15, 2009, and ended on April 3, 2010. Dynamic tests were performed under ambient excitation: the traffic was used as a source of excitation. Sixteen piezo-electric accelerometers (Bruel & Kjaer 4507B-005 with sensitivity 1 V/g, frequency range from 0.4 to 6000 Hz, maximum operational level 75g, temperature range from 54 to 100ºC) and seven temperature sensors (Pt100 class B) have been installed on the most deficient bridge deck (Lille/Paris – Figs. 10 and 11).

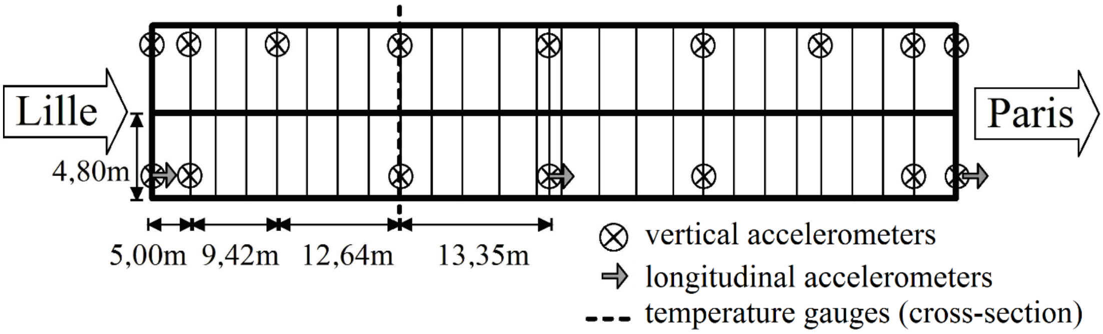

Lille
Paris
4,80m
5,00m
9,42m
12,64m
13,35m
⊗ vertical accelerometers
⇨ longitudinal accelerometers
--- temperature gauges (cross-section)

Figure 10. Plane view of the bridge with monitoring system.

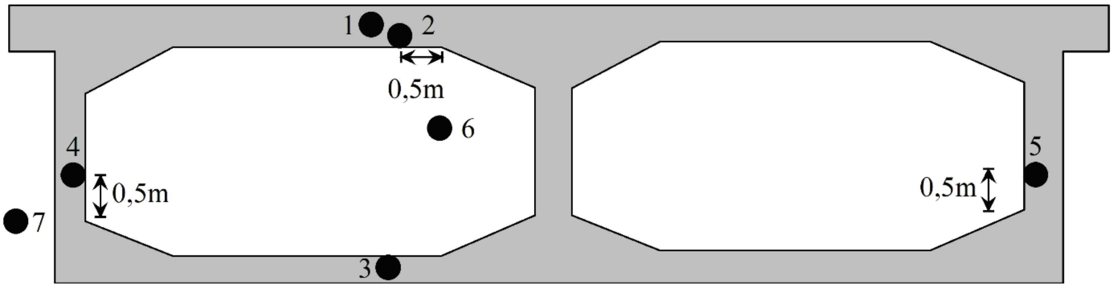


Figure 11. Location of the temperature sensors.

The data acquisition system was com-posed of two separate data acquisition systems. For the acceleration recording, a data programmable controller Gantner E-PAC DL was used and connected to an 8 GB USB flash drive. Data was transferred by a TCP/IP modem. For the temperature recording, a data logger Gantner IDL100 was used, and data was transferred by a GSM modem. Accelerations were filtered within the 0–30 Hz frequency range and sampling was set to 0.004 s during 5 min. To make the data processing amenable, structural data were only recorded every 3 h over a 24-h period and stored on a buffer hard disk. For the first campaign, 1174 tests have been recorded. The second campaign has had a total of 1316 tests recorded. Sixteen accelerometers were measuring vertical accelerations and three longitudinal accelerations (Fig. 10). The instrumentation scheme allowed identifying flexural vertical modal shapes, torsional and longitudinal vibration modes. Temperature was measured on seven different locations across a bridge deck cross-section (Fig. 11). Table 9 regroups the first five natural frequencies, showing mean values and respective standard deviations before and after reinforcement. Standard deviation values were rather low, which corroborates the overall quality of the modal identification procedure. Once again, if one considers confidence intervals i.e., mean values ± standard deviation, the values for natural frequencies from the first and second campaigns would superpose themselves. Thus, from a statistical point-of-view, it is not possible to state that the deviation of these frequencies indicates a structural change. Again, these results show that a more detailed analysis is necessary, but also considering a probabilistic approach, such as SDA.

Table 9. Frequency deviation according to each damage level (mean value and standard deviation).

Structural condition	Freq.#1	Freq. #2	Freq.#3	Freq.#4	Freq.#5
Before reinforcement	2,23 ± 0,053	4,89 ± 0,173	6,84 ± 0,111	8,48 ± 0,173	11,00 ± 0,165
After reinforcement	2,29 ± 0,067	4,95 ± 0,088	6,93 ± 0,161	8,51 ± 0,135	11,08 ± 0,176

Fig. 12 depicts a typical set of temperature data for the gauges located in the mid-span section. These data were collected every hour, over a 24-h period, before the bridge strengthening (from November 2008 to April 2009). In this figure, the two lower curves are for temperature gauges located on the side of the bridge close to the other bridge deck (east) and outside the instrumented deck. As expected, this later temperature gauge has a larger variability than the other gauges. The highest curve is for the air temperature inside the girder box; T5 (lateral west) sensor is very close to the evolution of the inside air temperature with a time lag of 2–3 h. Sensors T1 and T2 are very close and the gauge depth in the concrete deck does not appear to be significant. There is also a time lag of approximately 2–3 h between the peak air temperature outside and the peaks for the inside and lateral west temperatures. Gauges T1, T2 and T3 are following very closely the time history of the inside temperature. In this study, the temperatures are used as a basis for comparison with the vibration information.

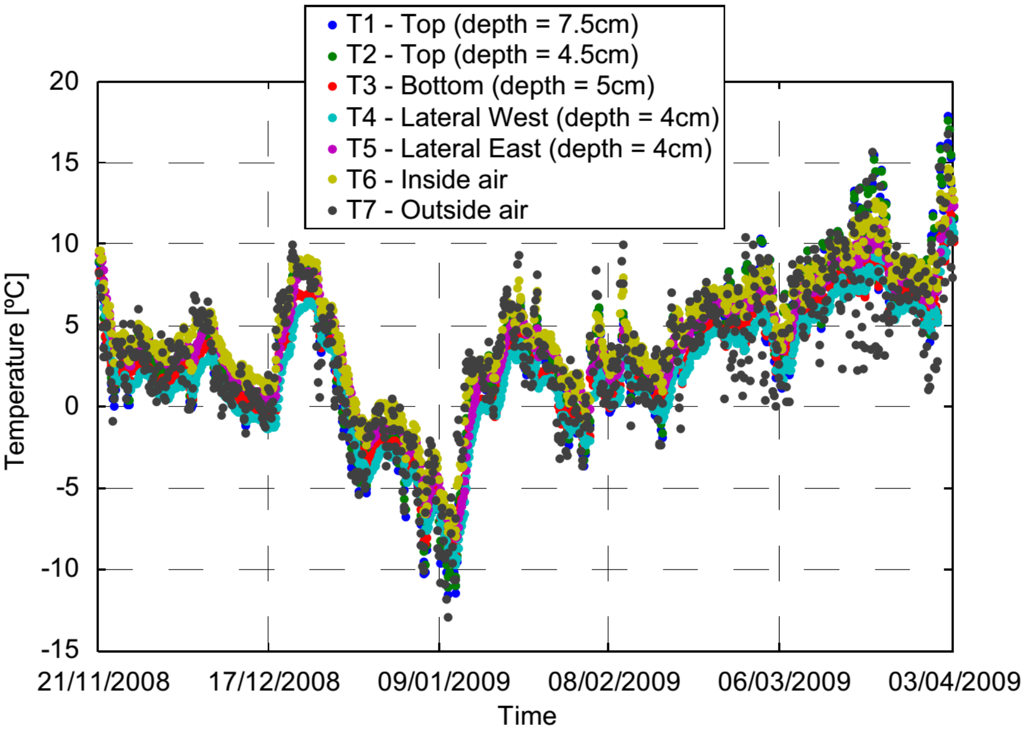


Figure 12. Time history of temperatures (before strengthening).

3.2.1 Results (unsupervised classification methods)

The procedure conducted henceforth in this section follows the steps described in section 3.1.1. Initially, the proposed approach is applied to all tests transformed into symbolic data (10-category histograms). The first campaign recorded 1284 tests as the second one collected 1476 tests. The objective of this study is to classify each test according to its campaign (before and after prestressing works). Thus, the target value must be intuitively ‘‘2’’ since two states (before and after strengthening) are expected to occur.

Table 10 shows the results obtained. In general, the percentages of correct classification are rather low for both dynamic clouds and hierarchy-agglomerative methods. Fuzzy c-means yield better results with moderate pertinence values. It must be noted that the proposed approach is being applied to the entire raw dataset of measurements, mixing dynamic tests registered upon different temperatures. This is directly reflected by the results of the indexes: in this analysis, all indexes indicate more than two clusters as the optimal partition. In fact, it turns out that some clusters might comprehend groups with similar profiles of temperature or traffic, for instance.

Table 10: Percentages of correct classification.

(%)	Dynamic Clouds	Hierarchy Agglomerative	FCM	Pertinence value	Indexes
Before	54	52	63	0,66	$CH = 4$
After	61	45	67	0,67	$C^{*} = 5$
					$\Gamma = 3$

Thus, a more detailed analysis can be conducted. Instead of using the entire dataset of tests from both campaigns, only data recorded during the same month are used. The objective now is to compare tests recorded in a month in 2008 with those registered at the same month in 2009 (for the months of January, February, March and April, the results correspond to the years of 2009 and 2010). The idea is to reduce, in some way, the uncertainties related to changes in temperature and due to traffic indirectly. For the month of October 2008 and 2009, 182 tests were recorder; for November 2008/2009, 444 tests; 458 in December 2008/2009; 455 in January 2009/2010; 386 in February 2009/2010; 439 in March 2009/2010 and 126 in April 2009/2010. Table 11 gathers the results obtained.

Table 11: Percentages of correct classification.

	Dynamic Clouds		Hierarchy-agglomerative		FCM			Indexes		
	Before	After	Before	After	Before	After	Pertinence Value	*CH*	C^*	Γ
Oct08/Oct09	100	100	100	100	100	100	0,88	2	2	2
Nov08/Nov09	56	58	54	75	67	67	0,65	3	4	3
Dec08/Dec09	65	69	59	52	62	70	0,71	3	3	2
Jan09/Jan10	63	54	53	62	67	67	0,64	2	3	3
Feb09/Feb10	59	63	63	55	65	69	0,60	2	4	2
Mar09/Mar10	61	57	74	61	70	69	0,70	2	4	3
Apr09/Apr10	57	58	67	58	67	68	0,60	2	3	2
Average	**66**	**66**	**67**	**66**	**71**	**73**	**0,68**	**-**	**-**	**-**

Overall, the classification results for the monthly-basis analysis are adequate and significantly better than those obtained using all tests concomitantly (Table 8). It is important to notice that for the months of October all tests were classified correctly. This result shows that it is indeed possible to extract useful information from raw vibration data. For the further months, the percentages are not as high. The fuzzy c-means method achieves better results with higher pertinence values compared to those of Table 8. It can also be observed that the average percentages of correct classification are also higher than those obtained in the previous simulation. Finally, indexes indicate optimal partitions equal or closer to two clusters. This corroborates the idea that the temperature variation plays an important role when it comes to the classification of structural behaviors.

3.2.2 Results (supervised classification methods)

The procedure carried out in this section mirrors the steps described in section 3.1.2.

The results show that both NN and SVM are again more robust for classifying tests compared to BDT. In general, NN achieves higher rates of correct classification (greater than 85%). In this study, it is noted that increasing the number of tests in the training group did not significantly alter the rates of correct classification. This may indicate that the proposed approach is sufficiently discriminative, even if few tests are used in the learning phase.

Like the previous study, Table 12 does not show the number of occurrences for the most extreme classification ratios (worst and best). In this case, 14,8% (1483) of the simulations yield the worst ratio and 7,3% (734) produce the best ratios. Again, the remaining results oscillate be-tween worst and best ratios with the corresponding average value presented in Table 12.

Table 12. Probabilities of true classification using signals for all dynamic tests.

	30% Tr., 10% V, 60% T			40% Tr., 10%V, 50% T			50% Tr., 10%V, 40% T		
Method	BDT	NN	SVM	BDT	NN	SVM	BDT	NN	SVM
Best (%)	74	87	86	80	89	90	85	92	92
Average (%)	65	75	74	69	75	76	74	78	77
Worst (%)	31	46	749	45	58	46	52	60	55

However, a more detailed analysis can be conducted. Instead of using the entire dataset of tests from both campaigns, only data recorded during the same month are used. The objective now is to compare tests recorded in a month in 2008 with those registered at the same month in 2009 (for the months of January, February, March and April, the results correspond to the years of 2009 and 2010). The idea is to reduce, in some way, the uncertainties related to changes in temperature and due to traffic. For the month of November 2008 and 2009, 444 tests were record-ed; 458 in December 2008/2009; 455 in January 2009/2010; 386 in February 2009/2010; 439 in March 2009/2010 and 126 in April 2009/2010.

Figure 13(a) summarizes the results obtained when only 30% of the tests (each month, separately) are retained in the training group. Once more, both NN and SVM are the most efficient classification methods. Indeed, the coupling of these methods to the symbolic representation of signals (histograms) is more sensitive to external effects (traffic, temperature, wind, etc.). However, using a monthly basis analysis, one observes the reduction of these effects, yielding better classification ratios. Figures 13(b) and 13(c) show the mean values obtained for each classification method considering the training groups with 40% and 50% of the tests, respectively. In general, the average classification rates are in the range of 80-90% considering the three methods and the three configurations of testing/training groups.

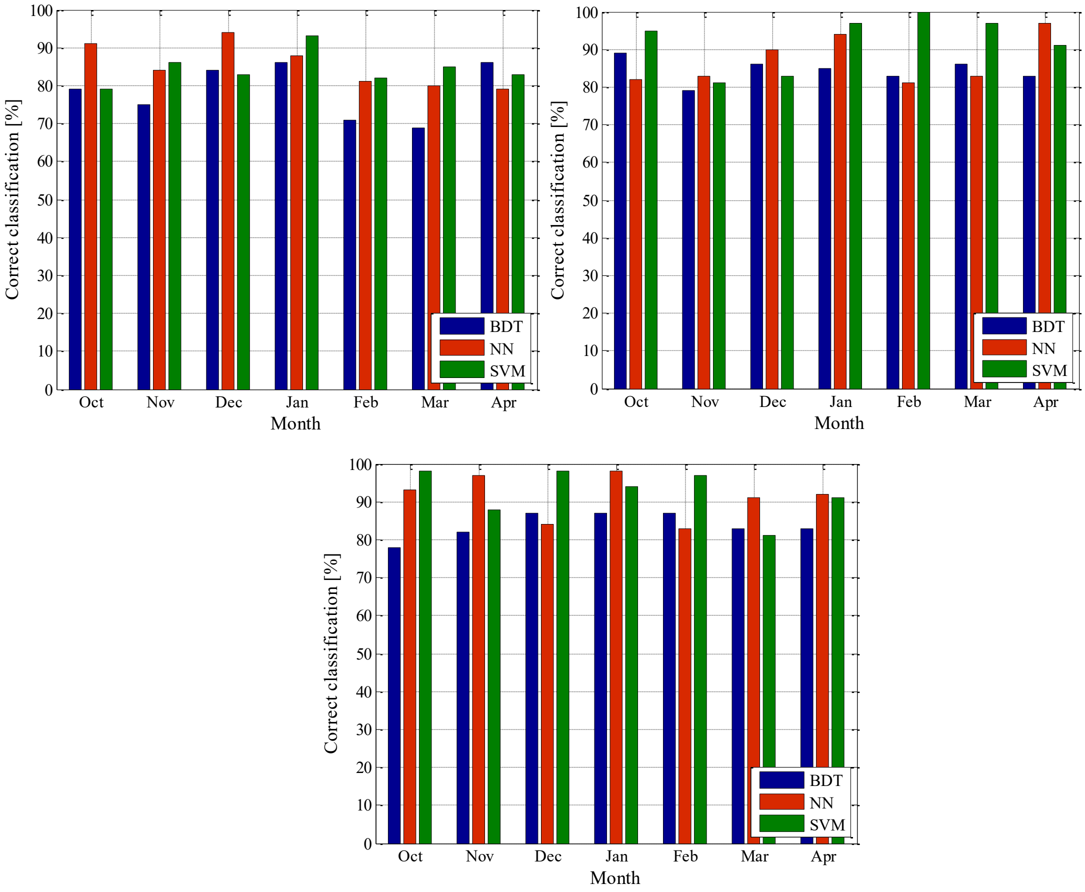


Figure 13. Average rates of correct classification. a) 30% training, 10% validation, 60% testing; b) 40% training, 10% validation, 50% testing; c) 50% training, 10% validation, 40% testing.

4. Final remarks and recommendations

This chapter presented a collection of approaches based on the coupling of Symbolic Data Analysis with three unsupervised and supervised classification methods. The main goal was to discriminate different structural behaviors using only raw information for feature extraction.

To attest the robustness of the proposed approaches, an experimental application performed at COPPE Structural Laboratory was studied. This application comprised a simply supported beam instrumented with six accelerometers, tested under two different excitation conditions: impact and random vibration. Moreover, six damage scenarios were considered: one undamaged scenario and five others corresponding to holes with different diameters. The main objective was to discriminate those structural states using only measured accelerations. Results obtained showed that the SDA methods were quite efficient to classify and discriminate structural modifications, even when raw data are used. In addition, better results were obtained when impact excitation was considered.

Furthermore, this chapter presented an experimental application concerning reinforcement works performed at a motorway bridge in France. Dynamic tests were registered under two structural states – before and after reinforcement. The results obtained showed that the SDA methods were efficient to classify and to discriminate structural modifications considering raw vibration data. To consider the effects of temperature variation, a more detailed study was conducted: instead of using the entire dataset of tests from both campaigns, only data recorded during the same month were used. In that case, results have improved significantly, showing evidence that this environmental effect plays an important role in the field of SHM.

In summary, the proposed combined approaches can be applied to continuous structural monitoring analyses when:

- reference behaviors are unknown → unsupervised learning methods. This is major advantage when SHM are performed over the years and a constant attention is necessary when it comes to structural safety. The main drawback might be the number of false positive alarms due to the lack of proper "learning" of the structure's behavior.
- one (or more) reference behaviors are known → supervised learning techniques. The main advantage is to provide the machine learning techniques a comprehensive database about the structure's dynamic behavior. Then, it allows mitigating the number of false positive alarms. The main drawback, on the other hand, is that it is not always possible to know the structure's actual (or former) condition *a priori*.

Acknowledgments

The authors would like to thank CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Finance Code 001), CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico – Grant 304329/2019-3), FAPEMIG (Fundação de Amparo à Pesquisa do Estado de Minas Gerais – Grant PPM-00001-18), and CIDENG-CNPq (Research Group on Data Science in Engineering). Moreover, the authors would like to thank to IFSTTAR (former LCPC - Laboratoire Central des Ponts et Chaussées) and SNCF (Société Nationale des Chemins de fer Français) - project 01V0527 RGCU "Evaluation dynamique des ponts" - for the data used in this chapter.

References

A. Alvandi and C. Cremona. Assessment of vibration-based damage identification techniques. Journal of Sound and Vibration, volume 292, no. 1–2, pages 179–202, 2006. DOI: 10.1016/j.jsv.2005.07.036.

V. Alves, A. Cury, N. Roitman, C. Magluta, and C. Cremona. Novelty detection for SHM using raw acceleration measurements. Structural Control and Health Monitoring, volume 22, no. 9, 2015a. DOI: 10.1002/stc.1741.

V. Alves, A. Cury, and C. Cremona. On the use of symbolic vibration data for robust structural health monitoring. Proceedings of the Institution of Civil Engineers -

Structures and Buildings, volume 169, issue 9, pages 715–723, 2015b. DOI: 10.1680/jstbu.15.00011.

J. C. Bezdek, Pattern Recognition with Fuzzy Objective Function Algorithms. Boston, MA: Springer US, 1981. DOI: 10.1007/978-1-4757-0450-1.

L. Billard and E. Diday, Symbolic Data Analysis. Chichester, UK: John Wiley & Sons, Ltd, 2006. DOI: 10.1002/9780470090183.

R. de A. Cardoso, A. Cury, F. Barbosa, and C. Gentile. Unsupervised real-time SHM technique based on novelty indexes. Structural Control and Health Monitoring, volume 26, no. 7, 2019. DOI: 10.1002/stc.2364.

A. A. Cury, C. C. H. Borges, and F. S. Barbosa. A two-step technique for damage assessment using numerical and experimental vibration data. Structural Health Monitoring, volume 10, no. 4, pages 417–428, 2011. DOI: 10.1177/1475921710379513.

A. Cury and C. Crémona. Assignment of structural behaviours in long-term monitoring: Application to a strengthened railway bridge. Structural Health Monitoring, volume 11, no. 4, pages 422–441, 2012. DOI: 10.1177/1475921711434858.

S. W. Doebling, C. R. Farrar, M. B. Prime, and D. W. Shevitz. Damage identification and health monitoring of structural and mechanical systems from changes in their vibration characteristics: A literature review. Los Alamos, NM, 1996. DOI: 10.2172/249299.

R. P. Finotti, A. A. Cury, and F. de S. Barbosa. An SHM approach using machine learning and statistical indicators extracted from raw dynamic measurements. Latin American Journal of Solids and Structures, volume 16, no. 2, 2019. DOI: 10.1590/1679-78254942.

S. J. S. Hakim and H. A. Razak. Modal parameters based structural damage detection using artificial neural networks - a review. Smart Structures and Systems, volume 14, no. 2, 2014. DOI: 10.12989/sss.2014.14.2.159.

U. Kroszynski and J. Zhou. Fuzzy Clustering - Principles, Methods and Examples. 1998.

T. S. Madhulatha. An overview on clustering methods. volume 2, no. 4, pages 719–725, 2012.

N. Martins, E. Caetano, S. Diord, F. Magalhães, and Á. Cunha. Dynamic monitoring of a stadium suspension roof: Wind and temperature influence on modal parameters and structural response. Engineering Structures, volume 59, pages 80–94, 2014. DOI: 10.1016/j.engstruct.2013.10.021.

G. W. Milligan and M. C. Cooper. An examination of procedures for determining the number of clusters in a data set. 1985.

J. J. Moughty and J. R. Casas. A state-of-the-art review of modal-based damage detection in bridges: Development, challenges, and solutions. Applied Sciences (Switzerland), volume 7, no. 5, 2017. DOI: 10.3390/app7050510.

P. van Overschee and B. de Moor. Subspace Identification for Linear Systems. Boston, MA: Springer US, 1996. DOI: 10.1007/978-1-4613-0465-4.

C.W. Hsu, C.-J. Lin. A comparison of methods for multi-class support vector machines, *IEEE Transactions on Neural Networks*, 13, 415-425, 2002.

C.C. Chang, C.J. Lin, LIBSVM: a library for support vector machines, 2021. https://github.com/cjlin1/libsvm.

J. Santos, C. Crémona, and P. Silveira. Automatic Operational Modal Analysis of Complex Civil Infrastructures. Structural Engineering International, volume 30, no. 3, 2020. DOI: 10.1080/10168664.2020.1749012.

Y.-S. Song, D.-C. Park, C. N. Tran, H.-S. Choi, and M. Suk. Fuzzy C-Means Algorithm with Divergence-Based Kernel. 2006. DOI: 10.1007/11881599_12.

S. Wang. Iterative modal strain energy method for damage severity estimation using frequency measurements. Structural Control and Health Monitoring, volume 20, no. 2, pages 230–240, 2013. DOI: 10.1002/stc.495.

Y. Xia, B. Chen, X. Q. Zhou, and Y. L. Xu. Field monitoring and numerical analysis of Tsing Ma suspension bridge temperature behavior. Structural Control and Health Monitoring, volume 20, no. 4, pages 560–575, 2013. DOI: 10.1002/stc.515.

Y. Zhou, M. Wahab, N. Maia, L. Liu, and E. Figueiredo, Data Mining in Structural Dynamic Analysis. Singapore: Springer Singapore, 2019. DOI: 10.1007/978-981-15-0501-0.

About the Editors

Ariosto Bretanha Jorge *(Book Series Leading Editor)*

Visiting Professor at Post-Graduate Program - Integrity of Engineering Materials, University of Brasilia, Brazil

Research interests include: helicopter technologies, mechanical vibrations, computational mechanics, numerical methods, optimization, reliability, aircraft structures, fracture mechanics, inverse problems.

More info: lattes.cnpq.br/3558866397613277, orcid.org/0000-0002-8631-1381

✉ *ariosto.b.jorge@gmail.com, ariosto.jorge@unb.br*

Carla Tatiana Mota Anflor

Professor at Post-Graduate Program - Integrity of Engineering Materials, University of Brasilia, Brazil

Research interests include: optimization, boundary element method, mechanical vibrations and solid mechanics

More info: lattes.cnpq.br/0526742760439036, orcid.org/0000-0003-3941-8335

✉ *ctanflor@gmail.com; anflor@unb.br*

Guilherme Ferreira Gomes

Professor at Mechanical Engineering Institute, Federal University of Itajubá, Itajubá, Brazil.

Research interests include: structures, vibration and modal testing, structural health monitoring, composite structures, optimization and applied artificial intelligence

More info: guilherme.unifei.edu.br, lattes.cnpq.br/4963257858781799, orcid.org/0000-0003-0811-6334

✉ *guilhermefergom@gmail.com; guilhermefergom@unifei.edu.br*

Sergio Henrique da Silva Carneiro

Collaborating Professor at Post-Graduate Program - Integrity of Engineering Materials, University of Brasilia, Brazil.

Research interests include: modal testing, structural analysis, finite element method, dynamics, fracture mechanics and damage detection.

More info: lattes.cnpq.br/6280300531787552, orcid.org/0000-0001-6669-2255

✉ *shscarneiro@gmail.com, shscarneiro@unb.br*

Book Series in Discrete Models, Inverse Methods, & Uncertainty Modeling in Structural Integrity

VOLUME I

MODEL-BASED AND SIGNAL-BASED INVERSE METHODS

VOLUME II

FUNDAMENTAL CONCEPTS AND MODELS FOR THE DIRECT PROBLEM

VOLUME III

UNCERTAINTY MODELING: FUNDAMENTAL CONCEPTS AND MODELS

> *This book series is an initiative of the Post Graduate Program in Integrity of Engineeering Materials from UnB, organized as a collaborative work involving researchers, engineers, scholars, from several institutions, universities, industry, recognized both nationally and internationally. The book chapters discuss several direct methods, inverse methods and uncertainty models available for model-based and signal based inverse problems, including discrete numerical methods for continuum mechanics (Finite Element Method, Boundary Element Method, Mesh-Free Method, Wavelet Method). The different topics covered include aspects related to multiscale modeling, multiphysics modeling, inverse methods (Optimization, Identification, Artificial Intelligence and Data Science), Uncertainty Modeling (Probabilistic Methods, Uncertainty Quantification, Risk & Reliability), Model Validation and Verification. Each book includes an initial chapter with a presentation of the book chapters included in the volume, and their connection and relationship with regard to the whole setting of methods and models.*

The Book Series is an initiative supported by:

UNIVERSITY OF BRASILIA - UnB
www.unb.br

With the kind encouragement of:

BRAZILIAN ASSOCIATION OF COMPUTATIONAL METHODS IN ENGINEERING - ABMEC
www.abmec.org.br

BRAZILIAN SOCIETY OF MECHANICAL SCIENCES AND ENGINEERING – ABCM
www.abcm.org.br

LATIN AMERICAN JOURNAL OF SOLIDS AND STRUCTURES - LAJSS
www.lajss.org