



DISSERTAÇÃO DE MESTRADO

**Projeto de controladores para um sistema  
de direção elétrica utilizando a  
metodologia de projeto baseado em modelos**

**Rafael Rodrigues da Silva**

**Brasília, agosto de 2017**

**UNIVERSIDADE DE BRASÍLIA**

**FACULDADE DE TECNOLOGIA**

UNIVERSIDADE DE BRASÍLIA  
Faculdade de Tecnologia

DISSERTAÇÃO DE MESTRADO

**Projeto de controladores para um sistema  
de direção elétrica utilizando a  
metodologia de projeto baseado em modelos**

**Rafael Rodrigues da Silva**

*Dissertação de Mestrado submetida ao Departamento de Engenharia  
Mecânica como requisito parcial para obtenção  
do grau de Mestre em Sistemas Mecatrônicos*

Banca Examinadora

Prof. Evandro Leonardo Silva Teixeira, Dr., \_\_\_\_\_  
FGA/UnB  
*Orientador*

Prof. André Murilo de Almeida Pinto, Dr., \_\_\_\_\_  
FGA/UnB  
*Coorientador*

Prof. Guilherme Caribé de Carvalho, Dr., \_\_\_\_\_  
ENM/UnB  
*Examinador interno ao PPMEC*

Prof. Max Mauro Dias Santos, Dr., \_\_\_\_\_  
DAELE/UTFPR-PG  
*Examinador externo ao PPMEC*

## FICHA CATALOGRÁFICA

SILVA, RAFAEL

Projeto de controladores para um sistema de direção elétrica utilizando uma metodologia de projeto baseado em modelos [Distrito Federal] 2017.

xvi, 84 p., 210 x 297 mm (ENM/FT/UnB, Mestre, Sistemas Mecatrônicos, 2017).

Dissertação de Mestrado - Universidade de Brasília, Faculdade de Tecnologia.

	1. MBD	2. EPAS
Departamento de Engenharia Mecânica	3. LQR	4. MPC
	I. ENM/FT/UnB	II. Título (série)

## REFERÊNCIA BIBLIOGRÁFICA

SILVA, RAFAEL. (2017). *Projeto de controladores para um sistema de direção elétrica utilizando uma metodologia de projeto baseado em modelos*. Dissertação de Mestrado, Departamento de Engenharia Mecânica, Universidade de Brasília, Brasília, DF, 84 p.

## CESSÃO DE DIREITOS

AUTOR: Rafael Rodrigues da Silva

TÍTULO: Projeto de controladores para um sistema de direção elétrica utilizando uma metodologia de projeto baseado em modelos.

GRAU: Mestre em Sistemas Mecatrônicos ANO: 2017

É concedida à Universidade de Brasília permissão para reproduzir cópias desta Dissertação de Mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. Os autores reservam outros direitos de publicação e nenhuma parte dessa Dissertação de Mestrado pode ser reproduzida sem autorização por escrito dos autores.

---

Rafael Rodrigues da Silva

Depto. de Engenharia Mecânica (ENM) - FT

Universidade de Brasília (UnB)

Campus Darcy Ribeiro

CEP 70919-970 - Brasília - DF - Brasil

# Dedicatória

Dedico este trabalho à todos os que torcem pelo meu desenvolvimento, principalmente família e amigos.

# Agradecimentos

Agradeço primeiramente a Deus, só Ele sabe o esforço empenhado para a realização deste trabalho. A Ele toda honra, a Ele toda glória a Ele todo louvor.

Agradeço a minha família pelo apoio incondicional durante a elaboração deste trabalho.

Agradeço aos orientadores prof. Dr. Evandro Leonardo Silva Teixeira e prof. Dr. André Murilo de Almeida Pinto por todos os ensinamentos e experiências trocadas durante este trabalho.

Agradeço ao prof. Dr. Max Mauro Dias Santos por abrir as portas do laboratório de sistemas automotivos da UTFPR-PG para que o desenvolvimento deste trabalho fosse possível. Agradeço ainda pela amizade durante todo esse período.

Agradeço aos amigos de mestrado Reurisson Silva e Pedro de Deus Peixoto pelo apoio e companheirismo durante esse período.

Agradeço aos amigos do laboratório de sistemas automotivos da UTFPR-PG pelo apoio durante o desenvolvimento deste trabalho. Um agradecimento especial para João Henrique Neme, Felipe Franco e Celson Silva.

Agradeço ao PPMEC pelos suportes necessários ao longo do desenvolvimento deste trabalho

Por fim, agradeço a FAPDF pelo fomento.

---

## RESUMO

A utilização e complexidade dos sistemas Eletroeletrônicos (E/E) automotivos têm crescido bastante ao longo dos anos. Esse crescimento é justificado pelos diversos benefícios que estes sistemas podem oferecer frente aos sistemas puramente mecânicos ou hidráulicos tais como: segurança, redução no consumo de combustível, redução na emissão de poluentes e aumento do conforto. Neste sentido, diversos subsistemas foram substituídos total ou parcialmente por dispositivos E/E. Paralelo ao desenvolvimento dos sistemas E/E automotivos, a complexidade dos softwares embarcados automotivos tem sido elevada consideravelmente. Pesquisas mostram que, atualmente, o desenvolvimento de software automotivo é responsável pelo maior custo no projeto de um veículo e o número de linhas de código podendo chegar a centenas de milhões. Devido a este aumento significativo na quantidade e complexidade dos softwares automotivos, os desenvolvedores têm buscado cada vez mais metodologias que acelerem e sistematizem o processo de desenvolvimento. A Metodologia de Projeto Baseado em Modelos (MBD) surge como uma alternativa para o desenvolvimento de software automotivo, uma vez que apresenta uma série de benefícios em relação ao desenvolvimento de software tradicional. Neste trabalho a metodologia MBD é aplicada ao desenvolvimento do software de controle de um Sistema de Direção Elétrica Assistida (EPAS). Para isto, duas estratégias de controle foram propostas. A primeira é a estratégia de controle Linear Quadrática Gaussiana (LQG) que é composta por um controlador Linear Quadrático (LQR) e um Filtro de Kalman para estimação dos estados. A segunda estratégia é o Controlador Preditivo Baseado em Modelos (MPC). Complementar à estratégia de controle MPC, foi desenvolvido um método de parametrização exponencial, o qual contribuiu significativamente com a redução do custo computacional. Os resultados apresentados mostraram que as estratégias de controle propostas cumpriram de forma satisfatória os requisitos de controle, tanto em nível de simulação quanto em nível de hardware. Neste sentido a metodologia MBD foi implementada no desenvolvimento do software de controle do sistema EPAS e os resultados mostraram que esta técnica proporciona uma série de benefícios e é capaz de acelerar e sistematizar o processo de desenvolvimento.

---

## ABSTRACT

The usage and complexity of automotive electro-electronic (E/E) systems have grown significantly over the years. This growth is justified by the many benefits these systems can offer in comparison to purely mechanical or hydraulic systems, such as: increased safety, reduced fuel consumption, reduced pollutant emissions and increased comfort. Thus, several subsystems have been totally or partially replaced by E/E devices. Simultaneously to the development of automotive E/E systems, the complexity of embedded automotive softwares has been raised considerably. Research shows that, today, automotive software development is responsible for the greatest cost in designing a vehicle, and the number of code-lines can reach hundreds of millions. Due to this significant increase in the quantity and complexity of automotive softwares, developers have increasingly sought methodologies that accelerate and systematize the development process. The Model-Based Design (MBD) methodology emerges as an alternative for the development of automotive software, since it presents a series of benefits in comparison to the development of traditional software. In this work, the MBD methodology is applied to the development of the control software of an Electric Power Assisted Steering (EPAS) system. In order to accomplish this, two control strategies were proposed. The first one is the Linear-Quadratic-Gaussian (LQG) control strategy, which is composed of a Linear-Quadratic-Regulator (LQR) and a Kalman Filter, for estimation of the states. The second strategy is the Model-Based Predictive Control (MPC). In addition to the MPC control strategy, an exponential parameterization method was developed, which contributes significantly to the reduction of computational cost. The results presented showed that the proposed control strategies fulfilled the control requirements satisfactorily, both at the simulation level and at the hardware level. This way, MBD methodology was implemented in the development of the EPAS control software and the results showed that this technique provides a series of benefits and is able to accelerate and systematize the development process.



# SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>1</b>
1.1	OBJETIVO GERAL	5
1.2	OBJETIVOS ESPECÍFICOS	5
1.3	ORGANIZAÇÃO DO TRABALHO	5
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA</b>	<b>7</b>
2.1	ARQUITETURA E/E AUTOMOTIVA	7
2.2	SISTEMA DE DIREÇÃO ELÉTRICA ASSISTIDA (EPAS)	8
2.2.1	PRINCÍPIO DE FUNCIONAMENTO	11
2.3	MODELO V PARA O DESENVOLVIMENTO DE SOFTWARE AUTOMOTIVO	13
2.3.1	ESPECIFICAÇÃO DOS REQUISITOS	14
2.3.2	PROJETO DO SISTEMA	15
2.3.3	PROJETO DO CONTROLADOR	15
2.3.4	CODIFICAÇÃO E IMPLEMENTAÇÃO	15
2.3.5	TESTE DA UNIDADE	15
2.3.6	TESTE DE INTEGRAÇÃO	15
2.3.7	VALIDAÇÃO DO SISTEMA	16
2.3.8	BENEFÍCIOS E DESAFIOS DO MODELO V	16
2.4	DESENVOLVIMENTO BASEADO EM MODELOS (MBD)	17
2.4.1	MODEL-IN-THE-LOOP (MIL)	19
2.4.2	SOFTWARE-IN-LOOP (SIL)	20
2.4.3	PROCESSOR-IN-THE-LOOP (PIL)	20
2.4.4	HARDWARE-IN-THE-LOOP (HIL)	21
2.5	DESENVOLVIMENTO TRADICIONAL VS DESENVOLVIMENTO BASEADO EM MODELOS (MBD)	22
<b>3</b>	<b>SISTEMA EPAS E ESTRATÉGIAS DE CONTROLE</b>	<b>25</b>
3.1	MODELO DINÂMICO DO EPAS	25
3.2	CONTROLADOR LINEAR QUADRÁTICO(LQR)	29
3.3	FILTRO DE KALMAN	32
3.4	REGULADOR LINEAR QUADRÁTICO GAUSSIANO (LQG)	33
3.5	CONTROLE PREDITIVO BASEADO EM MODELOS (MPC)	34
3.5.1	ESTRUTURA DO CONTROLADOR	37
3.5.2	FORMULAÇÃO DO CONTROLE PREDITIVO PARA SISTEMAS LINEARES	37
3.6	PARAMETRIZAÇÃO DO MPC	42

<b>4</b>	<b>METODOLOGIA</b>	<b>45</b>
4.1	REQUISITOS DO SISTEMA	45
4.2	PROJETO DO SISTEMA	46
4.3	PROJETO DOS CONTROLADORES	46
4.4	MODEL-IN-THE-LOOP (MIL)	50
4.5	GERAÇÃO DE CÓDIGOS E SOFTWARE-IN-THE-LOOP (SIL)	50
4.6	PROCESSOR-IN-THE-LOOP (PIL)	51
4.7	TESTE DE INTEGRAÇÃO E HARDWARE-IN-THE-LOOP (HIL)	51
4.7.1	MICROAUTOBOX <sup>®</sup>	53
4.7.2	DS1104 <sup>®</sup>	53
4.7.3	CONTROLDESK	54
<b>5</b>	<b>RESULTADOS E DISCUSSÃO</b>	<b>57</b>
5.1	RESULTADOS LQG	57
5.1.1	MIL E SIL PARA O CONTROLADOR LQG	58
5.1.2	HIL PARA O CONTROLADOR LQG	63
5.1.3	MIL E SIL PARA O CONTROLADOR MPC	66
5.1.4	RESULTADOS DAS SIMULAÇÕES EM HIL PARA O CONTROLADOR MPC	69
<b>6</b>	<b>CONCLUSÕES E TRABALHOS FUTUROS</b>	<b>78</b>
	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b>	<b>80</b>
	<b>APÊNDICES</b>	<b>84</b>

# LISTA DE FIGURAS

1.1	Parcela dos subsistemas no custo total do veículo. Adaptado de (STELLA, 2015)..	2
2.1	Arquiteturas E/E automotiva centralizada e distribuída. Adaptado de (GUIMARÃES, 2003).....	8
2.2	Sistema de Direção Elétrica Assitida. Fonte: Day (2013). .....	9
2.3	Tipos de sistemas EPAS .....	9
2.4	Tipos de sistemas EPAS. Adaptado de (TECHOLOGIES, 2008) .....	10
2.5	Mapa de torque (MAROUF et al., 2012) .....	11
2.6	Modelo V para o desenvolvimento de software automotivo. Fonte: (KELEME-NOVÁ et al., 2013).....	14
2.7	MBD com o modelo no centro das fases de desenvolvimento. Adaptado de Kelemenová et al. (2013). .....	18
2.8	MBD integrado ao Ciclo V .....	19
2.9	Método de verificação: MIL .....	20
2.10	Método de verificação: SIL .....	20
2.11	Método de verificação: PIL .....	21
2.12	Método de verificação: HIL .....	21
2.13	Etapas tradicionais para o desenvolvimento de software automotivo. Fonte: (STELLA, 2015) .....	22
2.14	Etapas no desenvolvimento baseado em modelos. Fonte: (STELLA, 2015) .....	23
2.15	Relação entre o grau de utilização do MIL e a quantidade de erros encontradas no código. Adaptado de (BROY et al., 2013).....	24
3.1	Modelo da EPAS. Adaptado de Marouf et al. (2012). .....	26
3.2	Mapa de torque. Adaptado de Marouf et al. (2012). .....	27
3.3	Estrutura do compensador LQG. Adaptado de (KURUCSÓ et al., 2015). .....	33
3.4	Principais aplicações do MPC. Adaptado de Domahidi (2016) .....	35
3.5	Mapa de predição do MPC. Adaptado de Alamir (2013). .....	36
3.6	Estrutura do controlador MPC. Adaptado de (ABU-AYYAD; DUBAY, 2007) .....	37
3.7	Sequência de comandos para o caso parametrizado em que $n_r = 3$ . Adaptado de Alamir (2013). .....	43
4.1	Diagrama de blocos para a planta do sistema EPAS.....	46
4.2	Diagrama de blocos do controlador LQR.....	47
4.3	Diagrama de blocos do filtro de Kalman discreto.....	48
4.4	Estrutura do controlador MPC implementada em Simulink .....	48
4.5	Estrutura do controlador LQR em SIL implementado no Simulink. ....	51
4.6	Plataforma HIL .....	52

4.7	Diagrama de blocos da plataforma HIL.....	52
4.8	MBAX II.....	53
4.9	DS 1104.....	54
4.10	Interface de comunicação entre as plataformas MBAXII® e DS1104®.....	55
5.1	Referência e torque de assistência geral para o teste MIL.....	58
5.2	Referência e torque de assistência geral para o teste SIL.....	58
5.3	Estimação velocidade da coluna em MIL.....	59
5.4	Estimação velocidade da coluna em SIL.....	59
5.5	Estimação velocidade do motor em MIL.....	60
5.6	Estimação velocidade do motor em SIL.....	60
5.7	Referência e torque de assistência gerado para o teste MIL após variação dos parâmetros.....	61
5.8	Referência e torque de assistência geral para o teste SIL após variação dos parâmetros.....	61
5.9	Estimação velocidade da coluna após variação de parâmetros no teste MIL.....	62
5.10	Estimação velocidade da coluna após variação de parâmetros no teste SIL.....	62
5.11	Estimação velocidade do motor após variação de parâmetros no teste MIL.....	62
5.12	Estimação velocidade do motor após variação de parâmetros no teste SIL.....	63
5.13	Referência e torque de assistência para o teste em HIL.....	64
5.14	Performance do estimador de estados para a velocidade da coluna no teste HIL.....	64
5.15	Performance do estimador de estados para a velocidade do motor no teste HIL.....	65
5.16	Erro entre o valor medido e o valor estimado para a velocidade da coluna.....	65
5.17	Erro entre o valor medido e o valor estimado para a velocidade do motor.....	66
5.18	Referência e torque de assistência gerado para o teste MIL.....	67
5.19	Referência e torque de assistência gerado para o teste SIL.....	67
5.20	Estimação da velocidade da coluna em MIL.....	68
5.21	Estimação velocidade da coluna em SIL.....	68
5.22	Estimação velocidade do motor em MIL.....	68
5.23	Estimação velocidade do motor em SIL.....	69
5.24	Referência e torque de assistência gerado para o teste em HIL.....	70
5.25	Estimação velocidade da coluna em MIL.....	71
5.26	Estimação velocidade da coluna em SIL.....	71
5.27	Referência e torque de assistência gerado para o teste HIL após variação dos parâmetros.....	72
5.28	Referência e torque de assistência geral para o teste HIL após variação dos parâmetros com horizonte de predição reduzido.....	72
5.29	Estimação velocidade da coluna em HIL após a variação dos parâmetros.....	73
5.30	Estimação velocidade do motor em HIL após a variação dos parâmetros.....	73
5.31	Velocidade do motor após a implementação de restrições de $[-40... + 40]rad/s$ .....	74
5.32	Perfil do sinal de controle para o caso de restrição na velocidade do motor.....	74

5.33	Perfil do sinal de controle sob ação de restrições em $[-2.5... + 2.5]V$ .....	75
5.34	Perfil da variação do sinal controle para o caso de restrição na tensão do motor.....	75
5.35	Perfil da variação de controle sob ação de restrições em $[-0.15... + 0.15]V/s$ .....	76
5.36	Perfil do sinal de controle sob ação de restrições em $-3V$ .....	76
5.37	Comportamento do torque de assistência quando da tensão do motor sob ação de restrições.....	77

## LISTA DE TABELAS

3.1	Parâmetros do modelo do EPAS .....	28
4.1	Comparação dos <i>solvers</i> .....	50
4.2	Configurações MBAX II .....	54
4.3	Configurações DS1104.....	55
5.1	Parâmetros do EPAS utilizados nas simulações.....	57
5.2	Simulações com diferentes tempos de amostragem e horizonte de predição.....	69

# LISTA DE SÍMBOLOS

## Siglas

E/E	Eletroeletrônicos
MBD	Desenvolvimento Baseado em Modelos
EPAS	Sistema de Direção Elétrica Assistida
ECU	Unidade de Controle Eletrônico
LQG	Linear Quadrático Gaussiano
LQR	Controlador Linear Quadrático
MPC	Controle Preditivo Baseado em Modelos
MIL	<i>Model-In-the-Loop</i>
SIL	<i>Software-In-the-Loop</i>
PIL	<i>Processor-In-the-Loop</i>
HIL	<i>Hardware-In-the-Loop</i>
RCP	<i>Rapid Control Prototyping</i>
OEM	<i>Original Equipment Manufacturer</i>

# 1 INTRODUÇÃO

Para se destacar frente a um mercado consumidor cada vez mais exigente e sofisticado, é requisito fundamental para uma empresa se adaptar às rápidas mudanças tecnológicas e organizacionais a fim de fornecer produtos e serviços inovadores que se diferenciem frente aos já existentes. De fato, independente do segmento de engenharia no qual a empresa está inserida a inovação não é simplesmente um objetivo, é uma necessidade (AARENSTRUP, 2015).

No que diz respeito ao segmento automotivo, o termo inovação, atualmente, é diretamente relacionado à utilização de sistemas eletroeletrônicos (E/E), haja vista que, as principais funcionalidades que surgem nos veículos hoje em dia são direta ou indiretamente ligadas a estes sistemas. A utilização de sistemas E/E embarcados nos automóveis tem crescido exponencialmente nos últimos anos devido aos diversos benefícios que estes sistemas podem oferecer tais como: controle do veículo, segurança e conforto (BLAKE; LEADER, 2005).

Um estudo realizado pela IBM *Institute for Business Value* mostra que existem cinco fatores que impulsionam as empresas do segmento automotivo a investirem recurso e esforço no desenvolvimento e implementação de sistemas E/E nos veículos (GUMBRICH; KOPPINGER, 2004).

- Competição - Empresas estão competindo cada vez mais pela preferência dos consumidores e os clientes cada vez mais críticos em relação ao produtos que desejam.
- Diferenciação do produto - Criar produtos com características únicas, que se destaquem frente aos concorrentes.
- Legislação - Legislações ambientais e de segurança que fomentam o desenvolvimento de novos sistemas.
- Expectativas dos clientes - O desejo por uma condução cada vez mais segura e de alta performance encoraja os clientes a buscarem produtos mais desenvolvidos tecnologicamente.
- Inovações tecnológicas - Novas tecnologias, que surgem constantemente no setor automotivo ou em outros segmentos e podem ser implementadas em veículos. Necessitando cada vez mais da utilização de sistemas embarcados.

Este estudo evidencia os principais motivos da busca por funcionalidades que impressionem o mercado consumidor e faça com que os fabricantes se diferenciem frente aos concorrentes. A implementação destas funcionalidades em veículos é possibilitada devido ao constante desenvolvimento de componentes e sistemas E/E. De fato, atualmente, os componentes E/E podem contribuir com cerca de 40% do custo total de um veículo dentre os quais a maior parcela é relacionado ao desenvolvimento de software (BROY et al., 2013). A figura 1.1 mostra a contribuição dos diferentes sistemas automotivos no custo total do automóvel. Observa-se ao longo dos anos



o aumento significativo da utilização de software e sistemas E/E em face da redução dos outros subsistemas.

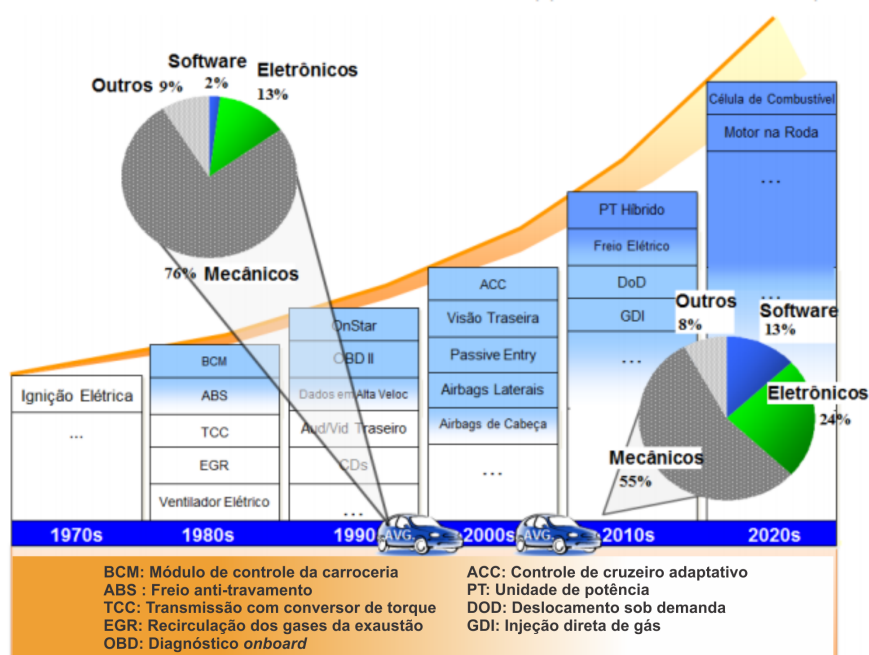


Figura 1.1: Parcela dos subsistemas no custo total do veículo. Adaptado de (STELLA, 2015).

Grande parte destas mudanças foi motivada pelos diversos benefícios que os sistemas E/E podem oferecer frente aos sistemas puramente mecânicos tais como: segurança, controle, redução de tamanho e desempenho. Como resultado disso, novas funcionalidades puderam ser desenvolvidas e implementadas o que levou ao crescimento da quantidade de dispositivos E/E utilizadas nos veículos.

Em geral um subsistema eletroeletrônico automotivo é composto por sensores, Unidades de Controle Eletrônico (ECUs) e atuadores. Estes componentes são responsáveis, respectivamente, por: detectar, controlar e atuar nos mais diversos subsistemas existentes no veículo. As ECUs utilizadas para o controle dos subsistemas são constituídas basicamente por uma interface de entrada, softwares de controle e interface de saída. Os softwares são algoritmos responsáveis por processar os dados de entrada e atuar no sistema segundo uma lógica de controle pré-estabelecida.

O desenvolvimento de sistemas embarcados elevou a complexidade e consequentemente o custo dos softwares automotivos. De fato, a utilização de software para controle de sistemas automotivos tem crescido consideravelmente nas últimas décadas (BROY et al., 2013). Krüger e Meisinger (2006) afirmam que em média 90% de todas as inovações que ocorrem em sistemas automotivos são diretas ou indiretamente ligadas aos *softwares* embarcados. Os automóveis atuais chegam a conter mais de 2500 funções controladas podendo conter, em alguns casos até 10 milhões de linhas de códigos (LIU; ZHANG; ZHU, 2016). Esses números mostram o quão complexo tem se tornado o desenvolvimento de software automotivo nos veículos modernos.

Atualmente, o desenvolvimento de software automotivo é realizado por meio da cooperação

entre fabricantes de veículos e fornecedores. Sangiovanni-Vincentelli e Natale (2007) definem que este grupo de empresas é composto por:

- Fabricantes (OEMs) – GM, Ford, Toyota e as empresas que fornecem o produto final aos consumidores, ou seja, os automóveis.
- Fornecedores (Tier 1) – Bosch, Magneti Marelli, Delphi e demais empresas responsáveis por fornecer subsistemas eletroeletrônicos que compõem o automóvel tais como: sistema de gerenciamento do motor, gerenciamento da suspensão, direção elétrica, sistema de diagnósticos dentre outras funcionalidades.
- Fornecedores (Tier 2) – Fabricantes de componentes eletrônicos tais como: CIs, placas e interfaces. Dentre essas empresas estão Freescale, Infineon e ST Microelectronics
- Fabricantes - Empresas como Flextronics e TSMC, responsáveis pela integração de componentes eletrônicos e capazes de fornecer produtos inteiros ou parcialmente montados.

Nesse sentido, as funções E/E automotivas têm sido cada vez mais complexas, fazendo com que os fabricantes e fornecedores sejam conduzidos a mudanças no paradigma de desenvolvimento, uma vez que os métodos tradicionais de desenvolvimento de software apresentam grandes desafios para sistemas de complexidade elevada. Dessa forma, diferentes métodos e ferramentas têm sido utilizados afim de garantir a qualidade do produto, redução de tempo e custo de desenvolvimento.

Uma prática de desenvolvimento de software que tem sido bastante utilizada é o modelo em V (*V-Cycle*). O modelo em V é um método de desenvolvimento de software que é subdividido em etapas bem delimitada e que cobrem todo o ciclo necessário para o projeto do software de um subsistema E/E, caracterizando-se portanto como um método de desenvolvimento *end-to-end* (SHAHBAKHTI; LI; HEDRICK, 2012). Este método pode oferecer diversos benefícios em relação ao métodos tradicionais, fato este que o tornou amplamente utilizado, principalmente, pela indústria automotiva.

Outra metodologia de desenvolvimento que tem sido bastante utilizada na industria automotiva é método de Projeto Baseado em Modelos (MBD). O MBD é uma metodologia que utiliza recursos gráficos e diagramas de blocos, contendo equações matemáticas a fim de obter o modelos funcionais dos sistemas a serem desenvolvido (MATHWORKS, 2014). A partir do desenvolvimento e avaliação destes modelos, ferramentas específicas possibilitam a geração automática de códigos, que podem então serem embarcados em *hardwares* para o controle de subsistemas e funcionalidade do veículo (PRABHU; MOSTERMAN, 2004). A utilização do MBD integrado ao ciclo de desenvolvimento em V oferece uma série de benefícios aos desenvolvedores, podendo impactar diretamente na qualidade do software gerado, redução de tempo e custo de desenvolvimento, maior integração entre as diferentes empresas e equipes envolvidas (BROY et al., 2013).

No MBD, tanto controlador quanto o modelo da planta física do sistema devem ser representadas através de modelos funcionais, geralmente, como diagrama de blocos que podem ser

modelados em *softwares* específicos. Neste trabalho, utilizou-se os softwares *Mathworks Matlab/Simulink* para o desenvolvimento dos modelos funcionais bem como para implementação dos algoritmos de controle.

De fato, a utilização de sistemas E/E embarcados tornou-se fundamental tanto para a indústria automotiva quanto para o consumidor, uma vez que estes sistemas proporcionam uma série de benefícios tais como: aumento de performance, redução no consumo de combustível, redução na emissão de gases poluentes, conforto e segurança quando comparados à sistemas puramente mecânicos ou hidráulicos (BLAKE; LEADER, 2005). Dentre as diversos subsistemas automotivos controlados eletronicamente, o Sistema de Direção Elétrica Assistida (EPAS) é um sistema capaz de proporcionar aumento no conforto, economia de combustível, redução de peso e segurança na condução de um automóvel (CHABAAN; WANG, 2001).

Devido aos diversos benefícios proporcionado pelo EPAS, este sistema tem substituído os sistemas de direção convencionais como o mecânico e o hidráulico. No EPAS, o esforço empregado pelo condutor para promover a mudança de direção do veículo é consideravelmente reduzido, uma vez que o torque necessário para realizar o esterçamento das rodas passa a ser complementado por um motor elétrico. Observa-se que como medida de segurança, o torque de assistência gerado pelo motor elétrico é reduzido gradativamente a medida que a velocidade do veículo aumenta (Kim & Song, 2002). Diante da importância e da criticidade de um EPAS, diferentes estratégias de controle tem sido investigadas em diversos trabalhos.

Neste sentido, o principal objetivo deste trabalho é aplicar a metodologia de projeto baseado em modelos (MBD) e o fluxo de trabalho estabelecido no método V para o desenvolvimento do *software* de controle de um sistema EPAS. Para o projeto de controladores de funções automotivas utilizando o MBD é fundamental obter um modelo da planta física a ser controlada, uma vez que no MBD a planta do sistema está presente em todas as fases do processo. Neste trabalho, limitou-se a uma planta física de um sistema EPAS apresentada no trabalho de (MAROUF et al., 2012), em que os autores apresentam as equações dinâmicas do EPAS bem como os valores dos parâmetros utilizados para a simulação.

No que tange as estratégias de controle, dois diferentes controladores serão abordadas sendo estes: o Controlador Linear Quadrático Gaussiano (LQG) e o Controlador Preditivo Baseado em Modelos (MPC). Inicialmente, estes sistemas de controle serão implementados em nível de simulação, em seguida os algoritmos de controle serão convertidos em códigos em linguagem C que são gerados automaticamente por meio de ferramentas específicas. Em seguida, será realizada a implementação dos modelos de controlador e planta em *hardwares* específicos para aplicações automotivas. Os *hardwares* utilizados foram a *MicroAutobox II* onde foram embarcados os controladores e a *DS1104* onde embarcou-se o modelo da planta, estes *hardwares* combinados com *softwares* específicos para calibração e monitoramento de sinais e variáveis configura, portanto, uma plataforma *Hardware-In-the-Loop* em tempo real.

## 1.1 OBJETIVO GERAL

Desenvolver e implementar as estratégias de controle Linear Quadrático Gaussiano (LQG) e Controle Preditivo Baseado em Modelos (MPC) para o controle de um Sistema de Direção Elétrica Assistida (EPAS) utilizando a metodologia de Projeto Baseado em Modelos (MBD), explorando as diversas etapas bem como as principais ferramentas do MBD.

## 1.2 OBJETIVOS ESPECÍFICOS

- Apresentar o fluxo de trabalho e as ferramentas de trabalho utilizadas no desenvolvimento de software para funções automotivas;
- Apresentar o ciclo de desenvolvimento em V;
- Apresentar a metodologia de projeto baseado em modelos
- Apresentar a dinâmica e funcionamento de um sistema de direção elétrica assistida;
- Desenvolver as estratégias de controle LQG e MPC para o sistema EPAS;
- Aplicar as fases de testes MIL, SIL e HIL;
- Fazer uma análise detalhada dos sinais de controle do EPAS.

## 1.3 ORGANIZAÇÃO DO TRABALHO

Este trabalho é organizado em quatro capítulos, referências e apêndices.

No segundo capítulo é realizada a revisão bibliográfica a cerca de sistemas e arquiteturas E/E automotivas, direção elétrica (EPAS), método V para o desenvolvimento de software e uma comparação entre os métodos de desenvolvimento tradicional e o MBD.

No capítulo três é apresentado o modelo EPAS utilizado neste trabalho, são discutidos a estrutura do modelo, o funcionamento, a dinâmica e por fim é desenvolvido o equacionamento a ser utilizado como modelo da planta.

No capítulo quatro os dois controladores são projetados. Inicialmente é projetado o controlador LQG em que são desenvolvidas as equação do LQR e do Filtro de Kalman. Em seguida o controlador MPC é discutido e as equações pertinentes são apresentadas.

No capítulo cinco é apresentada a metodologia utilizada nestes trabalho, ressaltando os passos utilizados durante todas as etapas do ciclo de desenvolvimento V, bem como no MBD. Neste capítulo são mostrados as ferramentas e plataformas utilizadas e as técnicas, são mostradas as principais dificuldades inerentes a utilização do MBD e como estas podem ser solucionadas.

Por fim são apresentados os resultados das simulações, as conclusões e sugestões de trabalhos futuros.

## 2 REVISÃO BIBLIOGRÁFICA

### 2.1 ARQUITETURA E/E AUTOMOTIVA

A arquitetura de um subsistema eletroeletrônico (E/E) automotivo é, geralmente, composta por sensores, Unidades de Controle Eletrônico (ECUs) e atuadores (NAVET et al., 2005). Os sensores têm a função de leitura e conversão de parâmetros físicos tais como temperatura, pressão e rotação em sinais elétricos que possam ser interpretados pelas ECUs. Os atuadores são controlados pelos sinais de comando enviados pela ECU, e de acordo com estes sinais, realizam as operações necessárias para o controle do subsistema.

Por exemplo, no sistema de injeção eletrônica do tipo LE-Jetronic, a ECU obtém dados dos sensores de temperatura, medidor do fluxo de ar, sensor de posição da borboleta, rotação e carga do motor e então calcula a quantidade de combustível que precisa ser injetada no cilindro através dos atuadores, neste caso os atuadores são as válvulas injetoras (BOSCH, 2014). De fato, os sistemas E/E têm sido cada vez mais utilizados nos veículos devido aos diversos benefícios que podem oferecer. Ao passo que novas funcionalidades são implementadas nos veículos o número de ECUs também aumenta. Atualmente, automóveis como o Volvo XC90 podem contar com mais de 40 ECUs (NAVET et al., 2005) e, em alguns veículos, de luxo este número pode ultrapassar as 70 unidades (SENTHILKUMAR; RAMADOSS, 2011).

Devido a aumento da quantidade de ECUs, as arquiteturas E/E também tiveram que ser reestruturadas. Nas primeiras implementações de sistemas E/E nos automóveis havia, geralmente, apenas uma ECU para controlar uma determinada funcionalidade. Com o surgimento de novas funcionalidades, diversas ECUs são utilizadas para o controle dos subsistemas. Em alguns casos, as ECUs necessitam trocar dados entre-si a fim de compartilhar informações, principalmente, no que diz respeito à leitura de sensores. Em casos que mais de uma ECU precise dos mesmos dados para suas operações, a unidade que está ligada diretamente ao sensor responsável por esta informação pode adquirir o sinal e disponibilizá-lo na rede para que as demais ECUs tenham acesso. Dessa forma, as arquiteturas podem ser do tipo centralizadas em que toda leitura, processamento e atuação é feita por uma ECU, ou distribuída, em que as unidades podem trocar informações entre si (GUIMARÃES, 2003). A figura 2.1 ilustra estes dois tipos de arquiteturas E/E, em que as setas indicam o fluxo de dados nas unidades.

As principais vantagens da arquitetura centralizada são a simplicidade e disponibilidade de informações durante toda a operação do sistema. Os desafios desta arquitetura são a quantidade elevada de cabos numa mesma unidade e a dificuldade em expandir a arquitetura. No que tange a arquitetura distribuída, as principais vantagens são a redução na quantidade de cabos, escalabilidade e robustez. Os desafios dessa arquitetura concentra-se basicamente na necessidade de um software baseado em algum protocolo de comunicação para realizar o controle da troca de informações na rede (GUIMARÃES, 2003).

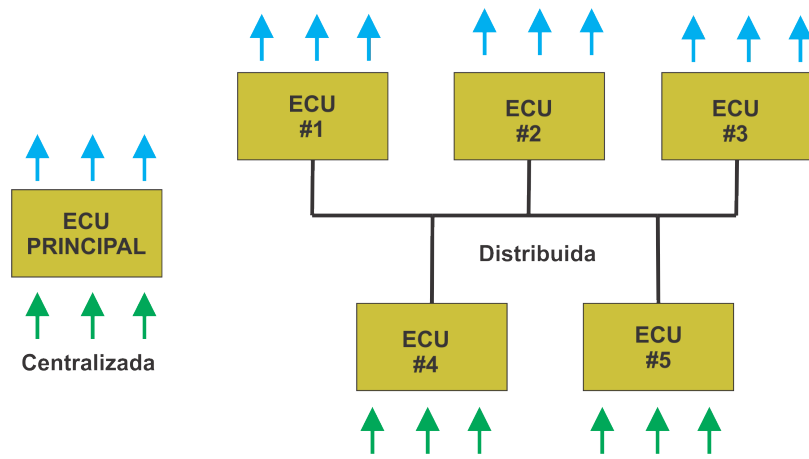


Figura 2.1: Arquiteturas E/E automotiva centralizada e distribuída. Adaptado de (GUIMARÃES, 2003).

Desta forma, as arquiteturas e os sistemas E/E têm se tornado cada vez mais desenvolvidos ao passo que os sistemas mecânicos e/ou hidráulicos são substituídos integral ou parcialmente por dispositivos E/E. Dentre as diversas soluções E/E nos veículos, um dos sistemas que tem sido bastante utilizado nos veículos atuais é o Sistema de Direção Elétrica Assistida (EPAS), as principais vantagens desse sistema são aumento no conforto, economia de combustível, redução de peso e maior segurança. (CHABAAN; WANG, 2001).

## 2.2 SISTEMA DE DIREÇÃO ELÉTRICA ASSISTIDA (EPAS)

O EPAS é considerado uma evolução nos sistemas de direção, uma vez que apresenta diversos benefícios em relação aos sistemas tradicionais como o mecânico e hidráulico (CHABAAN; WANG, 2001). Basicamente, o EPAS é composto por uma coluna de direção, um sistema de engrenagem, um sensor de torque também com a função de sensor de posição, uma Unidade de Controle Eletrônico (ECU) e um motor DC (Ren et al., 2008). Além dos demais componentes utilizados na direção mecânica, isto é, um volante, um pinhão, uma cremalheira e terminais de direção. No EPAS, parte do torque necessário para provocar a mudança na direção do veículo é fornecido por um motor elétrico. Isto reduz sensivelmente o esforço do condutor, em particular, em situações que exijam um demasiado esterçamento das rodas (como, por exemplo, as necessárias para estacionar um veículo). A figura 2.2 mostra um EPAS atualmente comercializado no mercado.

Em geral, os sistemas de direção elétrica podem ser divididos em três grupos que se diferem de acordo com a localização do motor elétrico (SHWETHA; RAMESH; SHANKAPAL, 2013). A definição do local de instalação é um requisito de projeto e é definida, geralmente, de acordo com o peso e dimensões do veículo. A Figura 2.3 mostra os diferentes tipos de sistemas EPAS.



Figura 2.2: Sistema de Direção Elétrica Assitada. Fonte: Day (2013).



Figura 2.3: Tipos de sistemas EPAS

### Tipo coluna

O EPAS tipo coluna é, geralmente, utilizado em veículos compactos que possuem motores (elétrico ou a combustão) de potência reduzida, e também caracterizados por dispor de pouco espaço no compartimento do motor. Por isso, o motor auxiliar da direção elétrica bem como as engrenagens de transmissão são instaladas dentro do habitáculo do veículo logo após o volante. Como é um projeto voltado para veículos leves e o motor elétrico não realiza tanto esforço, portanto, instalar o sistema dentro do compartimento dos ocupantes não acarreta em maiores desconfortos ao mesmos, uma vez que o ruído ocasionado pela rotação do motor é baixo.



## Tipo pinhão

Nesta arquitetura, o motor elétrico é instalado junto ao eixo do pinhão, que se localiza no compartimento do motor. Este tipo é, geralmente, utilizado em veículos maiores, mas que ainda apresentam limitações de espaço no compartimento do motor. Nesta arquitetura, o motor auxiliar pode ser mais robusto e fornecer um torque superior quando comparado ao tipo coluna, uma vez que, o ruído provocado não afeta o interior do veículo.

## Tipo cremalheira

No tipo cremalheira, o motor elétrico é acoplado diretamente à cremalheira do sistema de direção. O fato de o motor poder ser instalado em diferentes locais ao longo da cremalheira promove grande flexibilidade ao projeto do sistema de direção. Devido à localização, o sistema apresenta baixa inércia e garante uma excelente sensação de dirigibilidade, que consiste na percepção que o condutor possui da resposta do sistema de direção em relação ao piso.

Na prática, a arquitetura do EPAS a ser utilizada depende da força que o motor elétrico e as relações de engrenagem conseguem fornecer à cremalheira. A Figura 2.4 mostra as principais aplicações dos diferentes tipos de sistemas EPAS de acordo com a força que o motor auxiliar pode realizar na cremalheira do veículo.

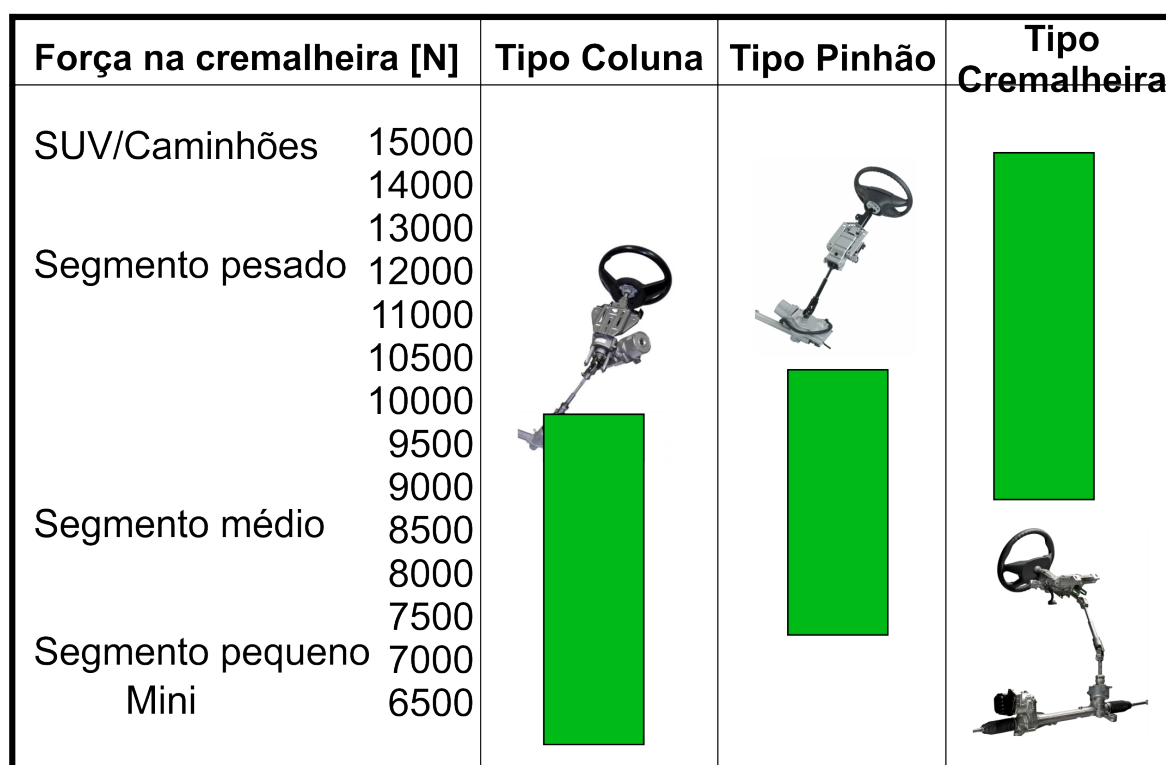


Figura 2.4: Tipos de sistemas EPAS. Adaptado de (TECHOLOGIES, 2008)

Portanto, quanto mais próximo à cremalheira mais robusto o motor elétrico pode ser e consequentemente mais esforço pode ser realizado pelo mesmo.

## 2.2.1 Princípio de funcionamento

O EPAS funciona, basicamente, da seguinte forma: Um sensor de torque é acoplado à coluna de direção a fim de medir o torque aplicado pelo condutor ao realizar uma manobra no veículo. Este sinal de torque é então combinado com o sinal de velocidade do veículo e por meio de um mapa, o torque de assistência a ser realizado pelo motor elétrico é definido. Portanto, o torque realizado pelo motor elétrico é uma função do esforço inicial realizado pelo condutor e da velocidade do veículo. A Figura 2.5 mostra um exemplo de mapa de torque.

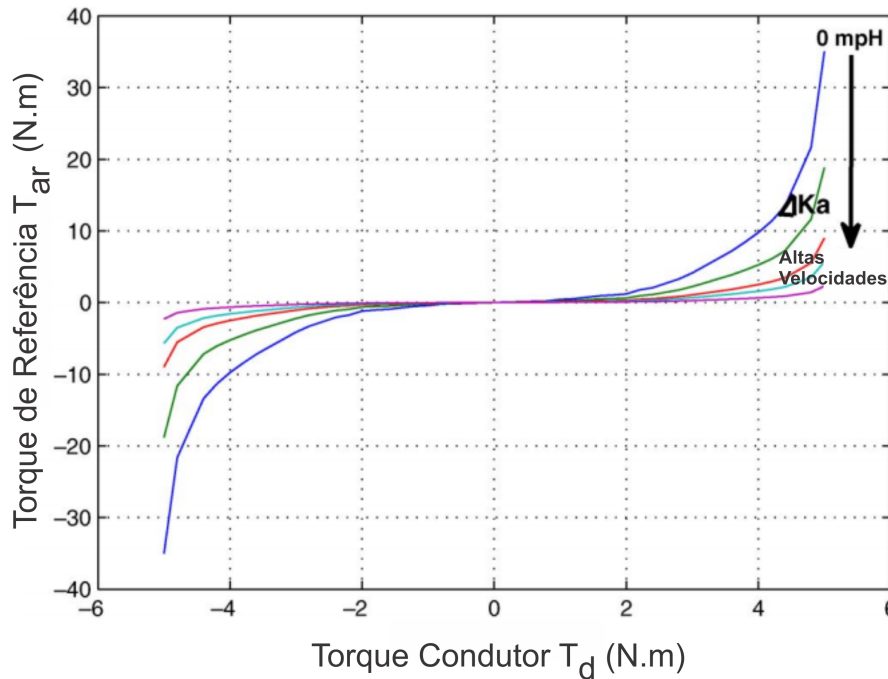


Figura 2.5: Mapa de torque (MAROUF et al., 2012)

O mapa de torque fornece uma relação não linear entre o torque medido do motorista e o torque de assistência realizado pelo motor elétrico (LEE; KIM; KIM, 2017). A forma do mapa de torque determina como o motorista sente a direção do veículo. Este tipo de estratégia baseada em um mapa de torque é de grande importância uma vez que define o torque de assistência a ser realizado pelo motor elétrico de acordo com as condições de condução do veículo. Por exemplo: para o veículo parado ou em baixas velocidades, situações que comumente demandam grande esforço do condutor e que geralmente ocorrem ao manobrar o veículo em um estacionamento, é desejável que a maior parte do torque seja realizado pelo motor auxiliar em vez do condutor ocasionando, deste modo, redução no esforço do condutor e maior conforto ao mesmo.

Por outro lado, nas situações em que o veículo estiver em altas velocidades, por questões de segurança, é necessário que o condutor realize um esforço maior para promover o esterçamento das rodas. Isto significa que com o aumento da velocidade do veículo, o torque de assistência produzido pelo motor elétrico é reduzido a fim de enrijecer a direção e naturalmente fazer com que o condutor realize um esterçamento mais moderado no volante ocasionando, desta forma, uma condução mais segura.

Uma vez que o torque é obtido, um sinal elétrico correspondente é enviado para a ECU da direção. A ECU então, utilizando o mapa de torque, determina o torque de assistência necessário a ser aplicado na coluna de direção. Observa-se que a estratégia de controle deve ponderar o torque de assistência, considerando a velocidade do veículo e controla o motor elétrico por meio da tensão no terminal do motor (KIM; SONG, 2002).

Contudo, para se obter o correto e eficiente funcionamento do EPAS é de fundamental importância a implementação de uma estratégia de controle. Diversos trabalhos propõem estratégias de controle para direções elétricas. (KIM; SONG, 2002) apresentam uma estratégia clássica de controle baseada em um controlador Proporcional Integral Derivativo (PID). Esta estratégia tem como objetivo controlar um motor para a redução do esforço exercido pelo condutor e melhorar o retorno do volante ao centro após a realização das manobras. No referido trabalho, também foi utilizado um sistema *Hardware-In-the-Loop* (HIL) a fim de verificar e validar a interação planta-controlador. Song et al. (2004) e Zang e Liu (2007) também utilizaram controladores PID para atingir os objetivos de controle, todavia, os pesquisadores reconhecem que os diferentes parâmetros de sintonia limitam a utilização prática. Outra limitação comumente reportada é o fato de controladores PIDs, em algumas situações, serem sensíveis a perturbações externas provenientes, principalmente, do atrito pneu-solo e vibrações o que pode comprometer o funcionamento do sistema.

Uma forma de contornar estes problemas é apresentada em Dong et al. (2010) e Parvathy e Daniel (2013). Nestes trabalhos, um controlador ativo é utilizado para aumentar a rejeição de perturbações de forma que o EPAS seja insensível a influências externas. A ideia principal da estratégia de controle apresentada pelos pesquisadores foi utilizar um Observador de Estado Estendido (ESO) de forma a estimar as perturbações generalizadas e, em seguida, neutralizar as perturbações em tempo real. Desse modo, o modelo passa a ser representado de forma mais simples podendo então ser controlado utilizando um tradicional Proporcional Derivativo (PD). Zhao et al. (2013) utilizam um controlador  $H_\infty$  e realizam comparações entre o controlador proposto e um PID tradicional. Ele mostra que o  $H_\infty$  apresenta resultados superiores ao PID principalmente no que se refere ao tempo de resposta, além de ser insensível a perturbações externas.

Em Chabaan e Wang (2001) é utilizado, também, um controlador  $H_\infty$  para fornecer estabilidade e minimizar os efeitos de perturbações no torque de assistência. Desta forma, implementou-se um estimador que através das informações do torque no pinhão, da variável de comando e das funções de transferência do sistema é capaz de estimar o torque aplicado pelo condutor no volante do veículo.

De fato, diferentes estratégias de controle têm sido investigadas para aplicação em sistemas de direção elétrica. Todavia, nota-se a ausência de discussões acerca de diversos tópicos tais como: detalhamento claro e objetivo da dinâmica, dos parâmetros e das variáveis envolvidas nos modelos de sistemas EPAS, e também pouca informação a respeito dos parâmetros de sintonia do LQR e do Filtro de Kalman. Portanto, este trabalho aborda, com profundidade, os tópicos ausentes em trabalhos anteriores, além de, propor uma estratégia de controle digital robusta, por

meio da discretização do controlador e do observador, o que é fundamental para utilização dos sistemas em hardwares específicos.

### 2.3 MODELO V PARA O DESENVOLVIMENTO DE SOFTWARE AUTOMOTIVO

Diante do aumento na quantidade e complexidade dos sistemas E/E, fabricantes e fornecedores têm sido cada vez mais desafiados a desenvolver software de alta qualidade para os mais diversos subsistemas automotivos a fim de garantir, principalmente, competitividade no mercado, lançar produtos inovadores bem como atender aos requisitos legais. Neste sentido, o desenvolvimento de sistemas E/E tem se tornado cada vez mais complexo e, portanto, metodologias que acelerem e sistematizem o processo de desenvolvimento destes sistemas podem trazer diversos benefícios aos desenvolvedores.

O modelo V é uma metodologia de desenvolvimento de software que tem sido amplamente utilizada pela indústria automotiva e aeroespacial nas mais diversas aplicações (PRABHU; MOSTERMAN, 2004). Isso se dá pelo fato de o método apresentar diversos benefícios, principalmente no que diz respeito à modelagem sequencial, detecção de erros nas fases iniciais e testes contínuos onde uma fase desenvolvimento é sempre finalizada antes de a próxima começar (SHAHBAKHTI; LI; HEDRICK, 2012). Essa metodologia engloba as principais fases de definição e desenvolvimento de softwares e pelo fato de possuir etapas de testes e validações correspondentes a cada uma destas fases anteriores, garante que o software gerado represente de forma precisa a descrição conceitual bem como atenda aos requisitos estabelecidos.

Diversos trabalhos relatam a utilização do modelo V para o desenvolvimento de software embarcado automotivo. Em (JI et al., 2009) foi desenvolvido o controlador para um motor de veículo elétrico movido a célula combustível do tipo *Proton Exchange Membrane* (PEM). Neste trabalho os autores utilizaram o modelo V no ciclo de desenvolvimento, desde a fase de requisitos até as etapas de validação e testes. Por fim, concluíram que o método de desenvolvimento utilizado foi bastante eficaz acelerando e sistematizando o processo. Em (KATTANKULATHUR, 2013) uma estrutura de modelo V é apresentada para o desenvolvimento do sistema de controle de um sistema *Anti-Lock Braking* (ABS). Os autores mostram todo o fluxo de trabalho para o projeto de um controlador para este subsistema e mostram como a técnica pode beneficiar o processo de desenvolvimento e testes. Em seguida o controle proposto é implementado em hardware e o sistema de controle é validado. (AWEDIKIAN; YANNOU, 2010) mostram que o modelo V é adotado pela empresa Johnson Controls® como fluxo de desenvolvimento de software desde a fase de levantamento dos requisitos até os testes e verificações finais. Portanto, diversos autores e empresas utilizam este método de desenvolvimento em suas aplicações.

A Figura 2.6 mostra as etapas do modelo V para o desenvolvimento de software para um sistema de controle automotivo. Nota-se que este método é decomposto em duas partes principais sendo, geralmente, definido pelo lado esquerdo e o direito. O lado esquerdo do modelo trata

da especificação do sistema, de forma que nesta fase busca-se entender as reais necessidades do cliente. Estas necessidades são descritas em forma de requisitos. Uma vez que os requisitos do produto estão claros e bem definidos do ponto de vista do cliente, eles podem então ser traduzidos em modelos funcionais que representem o sistema real com as necessidades requeridas. Já o lado direito do modelo trata das fases de integração dos componentes desenvolvidos, a fim de obter as arquiteturas especificadas e dos métodos de testes utilizados para validação do sistema (BERGSTROM; GORANSSON, 2016).

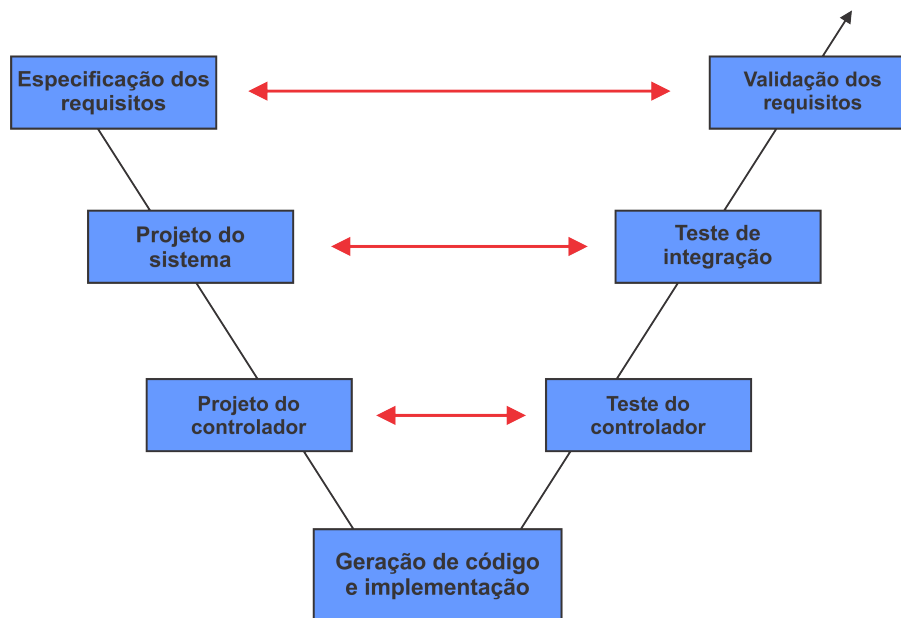


Figura 2.6: Modelo V para o desenvolvimento de software automotivo. Fonte: (KELEMENOVÁ et al., 2013)

A simetria entre o lado esquerdo e direito é uma das principais características do modelo V. Além disso, a conexão entre os dois lados do diagrama mostra como os requisitos do projeto são desenvolvidos nas fases subsequentes e assim os desenvolvedores podem verificar em todas as fases se o sistema atende aos requisitos previamente estabelecidos. Abaixo são descritas as etapas do modelo.

### 2.3.1 Especificação dos Requisitos

O primeiro passo do modelo V consiste em levantar os requisitos do sistema do ponto de vista do cliente. Os requisitos contêm informações sobre a operação do sistema, necessidades e aplicações. Uma vez que todo o desenvolvimento do software automotivo é baseado nos requisitos estabelecidos, nesta fase, o objetivo principal é entender de forma clara e objetiva as reais necessidades do cliente. A especificação dos requisitos é considerada desde as fases iniciais do ciclo de desenvolvimento, como a modelagem do sistema, até as fases finais do ciclo, através das verificações e validações.

### **2.3.2 Projeto do sistema**

Nesta etapa os requisitos especificados são analisados pelos desenvolvedores de forma a compreender, do ponto de vista técnico, as características e o funcionamento do sistema proposto. Nesta etapa, geralmente, busca-se desenvolver um modelo do sistema a ser controlado, ou seja, a planta física é representada através de equações ou diagramas de blocos que visam representar de forma satisfatória o sistema real. Para isso são levadas em considerações as características dinâmicas do modelo e as interfaces de entrada e saída. Alguns requisitos podem ser verificados já nesta etapa e caso não sejam exequíveis tecnicamente o usuário é comunicado.

### **2.3.3 Projeto do controlador**

Nesta etapa, a estratégia de controle do sistema é projetada. Para o projeto dos controladores alguns requisitos são levados em consideração de forma a obter estratégias de controle que satisfaçam algumas das necessidades de projeto exigidas. Ainda nesta etapa, os componentes do sistema podem ser subdivididos em sub-componentes tais como: controladores, estimadores e arquivos de preparação.

### **2.3.4 Codificação e implementação**

Nesta etapa, as funções são implementadas em alguma linguagem de programação, seja através da programação manual ou através da geração automática de códigos utilizando ferramentas específicas.

### **2.3.5 Teste da Unidade**

Refere-se aos testes efetuados na unidade após a rotina de controle ser embarcada na mesma. Uma unidade é a menor parte do sistema que pode ser utilizada de forma independente para controle de um sistema ou funcionalidade, são exemplos de unidades: Unidades de Controle Eletrônico (ECU) ou hardwares dedicados. Estes testes são responsáveis por verificar o sistema, de forma que podem ser detectados erros de software que surgem após o código ser embarcado, erros no sistema de controle não detectados em etapas anteriores, bem como erros no próprio dispositivo físico. Podem ser realizados também testes a fim de avaliar o desempenho do hardware, seja através requisitos temporais, de memória ou de processamento. Os testes a serem realizados nesta etapa são elaborados na etapa correspondente do lado esquerdo do modelo V, neste caso: Projeto da lógica de controle.

### **2.3.6 Teste de integração**

Neste etapa os testes são realizados com a unidade incorporada ao sistema. A unidade é então integrada aos dispositivos e sistemas físicos com os quais irá comunicar, podendo haver também

troca de informações de diversas unidades entre si. Portanto, são avaliados nesta fase o comportamento do subsistema com todos os componentes necessários para sua operação. Novamente podem ser encontrados erros que passaram por etapas anteriores, avaliação de requisitos temporais, software e hardware. O plano de testes para etapa é definido na etapa do lado esquerdo correspondente: Projeto da arquitetura.

### **2.3.7 Validação do sistema**

A última etapa do modelo V tem como objetivo avaliar o desempenho do sistema frente aos requisitos estabelecidos na primeira etapa: Requisitos do sistema. Uma vez que estes requisitos são definidos em conjunto pelos clientes e desenvolvedores, os testes realizados nesta etapa são realizados por usuários de ambas as partes, e geralmente são usuários treinados para tais finalidades além de serem efetuados em ambientes e situações reais ou bem próximas a realidade. Ao término, é avaliado se o produto gerado atende todas as necessidades requeridas bem como se o mesmo pode ser por fim utilizado em aplicações reais.

De fato, o desenvolvimento de software automotivo tem se tornado cada vez mais complexo devido, principalmente, ao elevado grau de complexidade dos sistemas projetados e também da grande cadeia de empresas envolvidas no processo de desenvolvimento (OEMs e Tiers dos diferentes níveis). Por isso, a utilização de metodologias que acelerem e sistematizem o processo de desenvolvimento e garantam a qualidade do software gerado tornou-se uma busca constante por parte dos desenvolvedores. Dessa forma, o modelo V é amplamente utilizado para o desenvolvimento de *software* automotivo devido aos diversos benefícios e desafios que este método pode oferecer.

### **2.3.8 Benefícios e desafios do modelo V**

Os principais benefícios do modelo V é a simplicidade e facilidade de implementação do método, em que as atividades a serem executadas em cada etapa são muito bem delimitadas o que permite aos desenvolvedores saber claramente o trabalho a ser executado. Devido a essa simplicidade o método V pode ser facilmente incorporado a outras metodologias de desenvolvimento. Além disso, é um modelo linear, em que cada fase é essencialmente finalizada antes de avançar para a próxima permitindo assim que erros sejam detectados logo nas primeiras etapas. (BUCANAC, 1999).

No que tange aos desafios, o principal consiste no fato de que todo o sistema ser projetado no início do ciclo de desenvolvimento. Dessa forma, caso sejam necessárias alterações nas etapas posteriores é preciso voltar para a descrição dos requisitos, refazer a documentação e percorrer as demais etapas novamente. Isto se torna ainda mais crítico no desenvolvimento de sistemas muito complexos, onde alguns componentes, interações ou outros elementos podem não ser conhecidos desde o início (BUCANAC, 1999).

## 2.4 DESENVOLVIMENTO BASEADO EM MODELOS (MBD)

Atualmente, alguns desenvolvedores utilizam o MBD incorporado ao método V. O MBD oferece um método gráfico que permite a modelagem e implementação de sistemas de controle E/E através de modelos funcionais. Desta forma, as etapas do modelo em V podem ser executadas com todas as vantagens do MBD garantindo assim uma série de benefícios quando comparado aos métodos tradicionais de desenvolvimento.

Assim como um software CAD pode modelar peças, reproduzindo de forma exata suas características (medidas, materiais, parâmetros geométricos), a metodologia MBD permite aos desenvolvedores projetar o modelo funcional de um sistema de forma que seja possível verificar continuamente se este modelo cumpre os requisitos especificados. Além do mais, ferramentas de Desenho Assistido por Computador (CAD) permitem a geração automática de códigos numérico, tal como Comando Numérico Computadorizado (CNC) para produção de peças em uma fresa ou torno. De forma análoga as ferramentas do MBD proporcionam a geração automática dos códigos de controle a partir do modelo projetado (MATHWORKS, 2014). Dessa forma, o MBD tem se mostrado uma alternativa eficaz para o desenvolvimento de software automotivo e apoiado ao modelo de desenvolvimento em V é capaz de fornecer um fluxo de desenvolvimento contínuo e verificável ao longo das diversas etapas.

Diversos trabalhos utilizaram o MBD para o desenvolvimento de sistemas de controle embarcado automotivo. Em Neme, Santos e Teixeira (2015) o MBD foi utilizado para o desenvolvimento do controle do sistema de iluminação externa de um veículos. Os autores desenvolveram o sistema de controle através de modelos funcionais, em seguida realizaram todos os testes inerentes ao MBD, verificando e validando o controle. Para os autores o MBD pode acelerar significativamente o processo de desenvolvimento. Eyisi et al. (2013) apresentam um metodologia sistemática e as ferramentas utilizadas no desenvolvimento de um sistema de gerenciamento aplicado a um controle de cruzeiro adaptativo utilizando o MBD. Eles destacaram MBD como fundamental para superar os desafios encontrados na práticas atuais de desenvolvimento de software automotivo como detecção de erros e maior agilidade no processo.

Seo et al. (2009) apresentam uma metodologia de desenvolvimento de software automotivo utilizando o MBD para um sistema de controle baseado no padrão OSEK/VDX. OSEK/VDX é um padrão de desenvolvimento de ECUs automotivas que tem por objetivo uniformizar as arquiteturas das ECUs visando a portabilidade e reusabilidade das unidades entre os diferentes fabricantes. Portanto, os autores aplicaram com sucesso o MBD no desenvolvimento de software de acordo com um protocolo específico, demonstrando ainda mais a versatilidade do MBD. Por fim, concluem que o MBD é mais eficiente que os métodos tradicionais no que diz respeito à redução de tempo e custo de desenvolvimento. Em (CICEO et al., 2015), o MBD foi usado para implementar um método de estimativa de consumo de energia (ECE) da bateria em um veículo elétrico para diferentes cenários.

Diversos trabalhos mostram que o MBD pode ser uma excelente alternativa no desenvol-



vimento de software, principalmente quando comparado às técnicas tradicionais. Os métodos tradicionais de desenvolvimento de software automotivo diferem da técnica do MBD principalmente pelo fato de grande parte de suas etapas serem realizadas manualmente, principalmente na fase de levantamento dos requisitos que é feita, geralmente, através de documentos e reunidos em planilhas além da codificação manual. Estas práticas além de impactarem no tempo de desenvolvimento, podem resultar em erros (KELEMENOVÁ et al., 2013).

Na metodologia de projeto baseado em modelos, muitas dessas técnicas são descartadas, uma vez que o MBD mantém o modelo do sistema no centro do processo durante todas as etapas de desenvolvimento. A Figura 2.7 ilustra este conceito, em que partindo do levantamento dos requisitos, através do projeto, implementação do código, testes e verificações contínuas até a geração automática de códigos o modelo é sempre mantido no centro do processo de desenvolvimento. (SMITH; PRABHU; FRIEDMAN, 2007).

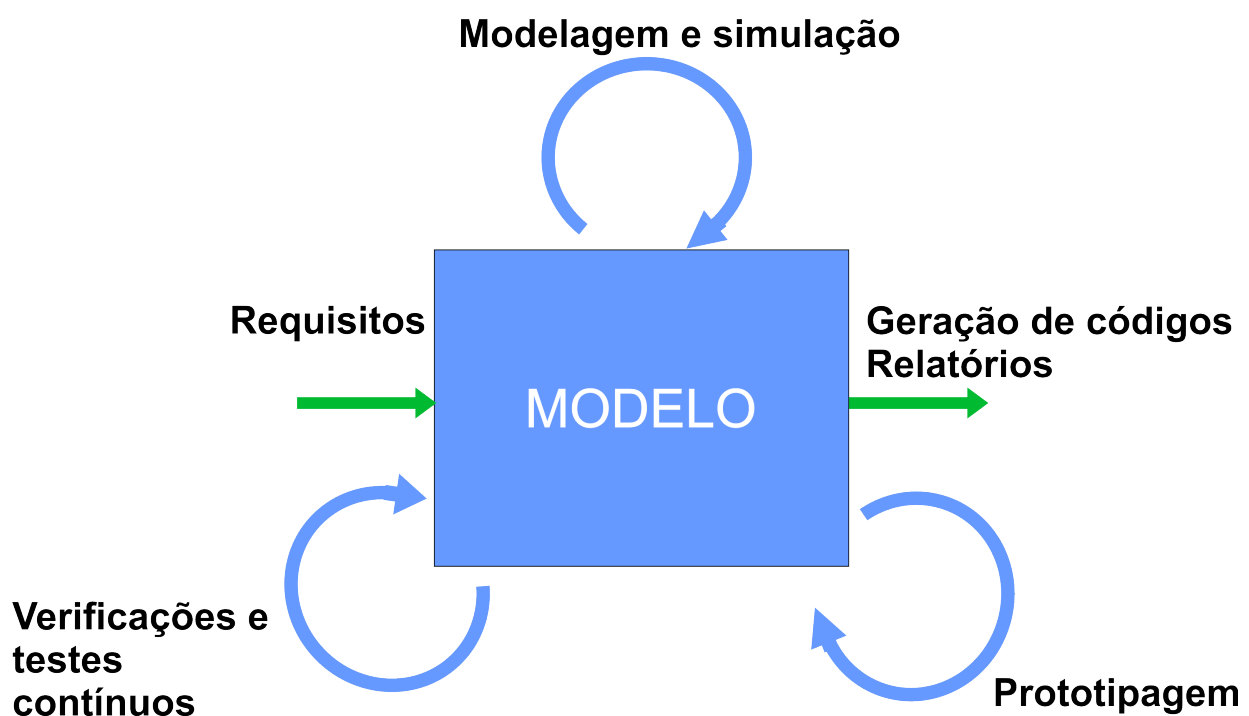


Figura 2.7: MBD com o modelo no centro das fases de desenvolvimento. Adaptado de Kelemenová et al. (2013).

De fato, uma das principais vantagens do MBD é a capacidade de serem executadas verificações e testes ao longo de todo o processo de desenvolvimento. As formas e intensidade das verificações e testes dependem, geralmente, da complexidade do sistema a ser desenvolvido. Em outras palavras, para sistemas que apresentem elevadas complexidade, geralmente, são realizados testes mais robustos a fim de minimizar a quantidade de erros nas etapas finais, uma vez que estas etapas demandam mais tempo e recurso. Por outro lado, em sistemas de complexidade reduzida, geralmente, utiliza-se de testes mais curtos a fim de reduzir o tempo de desenvolvimento. Os principais benefícios dos testes e verificações contínuas são:

- Identificação de erros nas fases iniciais, reduzindo custo e tempo de desenvolvimento;

- Identificação de erros que impactam diretamente na qualidade do software gerado;
- Possibilita a realização de testes em diversos cenários;
- Proporciona maior entendimento do sistema, uma vez que é possível compreender o comportamento do sistema a medida em que está sendo desenvolvido.

Os principais métodos de verificação e validação (V&V) no MBD são: *Model-In-the-Loop* (MIL), *Software-In-the-Loop* (SIL), *Processor-In-the-Loop* (PIL) e *Hardware-In-the-Loop* (HIL). A Figura 2.8 mostra estes métodos quando integrados ao ciclo de desenvolvimento em V.

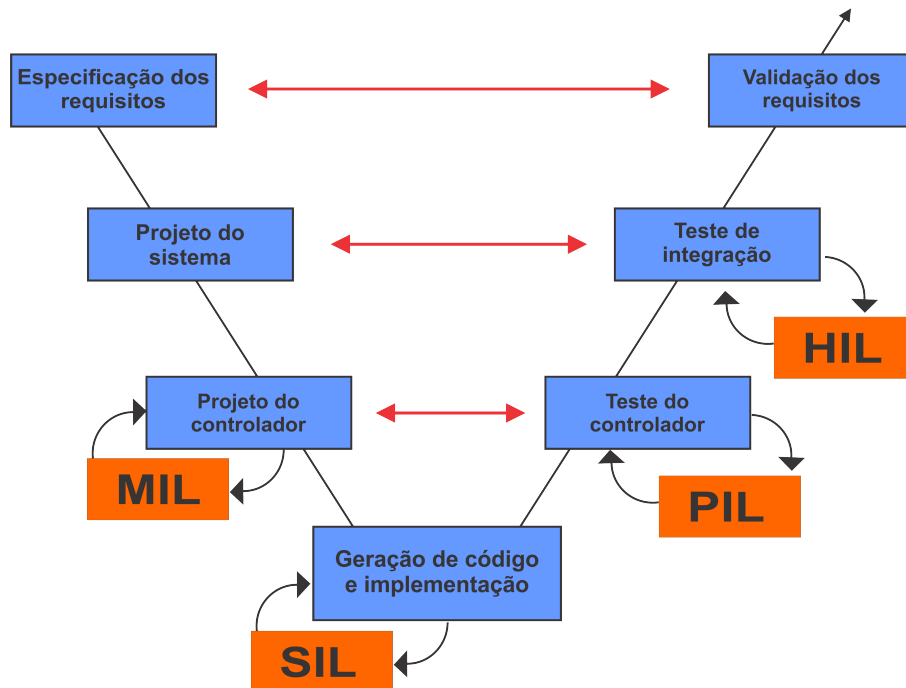


Figura 2.8: MBD integrado ao Ciclo V

Em seguida os métodos de verificação e testes são detalhados.

#### 2.4.1 Model-In-the-Loop (MIL)

Este é o primeiro teste do ciclo de V&V no MBD. Nesta etapa, tanto a estratégia de controle quanto a planta do sistema são desenvolvidas em ambiente virtual através de blocos funcionais. Os softwares Matlab/Simulink<sup>®</sup> são bastante utilizados para esta finalidade. A principal vantagem deste teste é a capacidade de reproduzir as funcionalidades do sistema sem a necessidade de componentes de hardware ou equipamentos específicos. A Figura 2.9 mostra o modelo de um sistema utilizado na fase do MIL.

O objetivo do MIL é verificar o funcionamento e a lógica de controle do sistema através da avaliação dos sinais de entrada e saída e dessa forma avaliar se o sistema de controle modelado atende de forma satisfatória aos requisitos previamente especificados. Sendo assim, a utilização

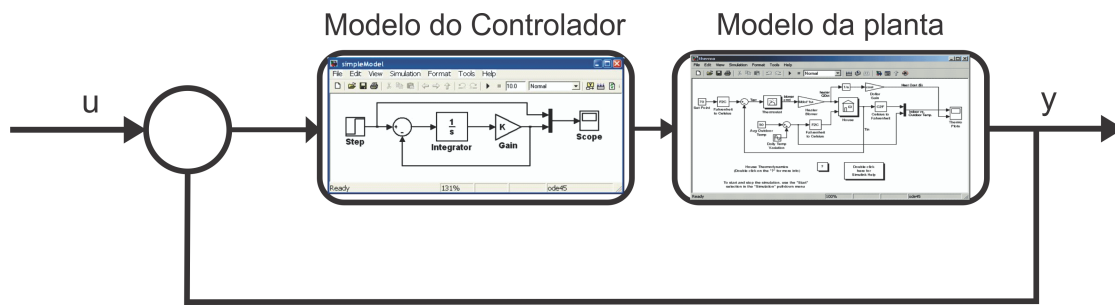


Figura 2.9: Método de verificação: MIL

deste método possibilita aos desenvolvedores a detecção de erros de modelagem e de requisitos nos primeiros estágios do projeto, o que permite uma rápida correção dos problemas detectados

### 2.4.2 Software-In-Loop (SIL)

Seguindo o fluxo do MBD, o SIL consiste novamente em verificar e validar o comportamento do sistema. Todavia o modelo do controlador é convertido em código fonte, enquanto que a planta do sistema ainda é representada por um modelo funcional. Conforme ilustra a Figura 2.10, o controlador é representado pelo código em linguagem C ou C++ e a planta segue sendo representada por um diagrama de blocos de Matlab/Simulink.

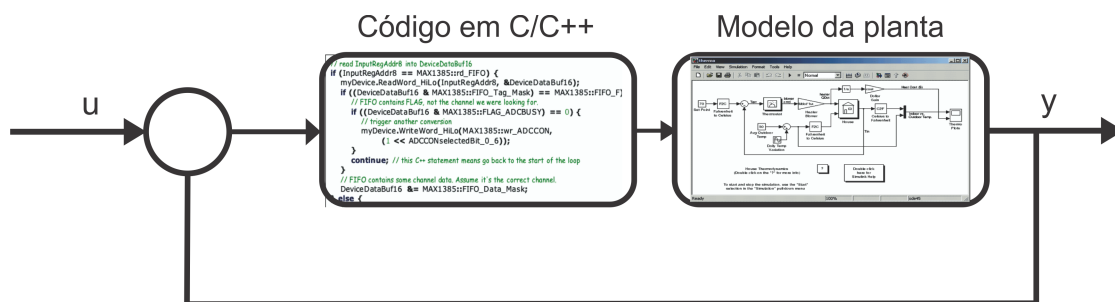


Figura 2.10: Método de verificação: SIL

Nesta etapa, o algoritmo de controle representado em código fonte é avaliado quanto a sua correspondência ao controlador em diagrama de blocos (caso anterior). Esta fase possibilita aos desenvolvedores detectarem erros de programação, principalmente em partes que a codificação é feita de forma manual. Além de verificar se o modelo gerado pode ser completamente convertido em código de forma automática. Para respostas satisfatórias, resultados semelhantes aos obtidos nas simulações em MIL são esperados.

### 2.4.3 Processor-In-the-Loop (PIL)

Neste estágio é realizado o PIL em que os algoritmos de controle, gerados de forma automática ou não, são embarcado em uma placa controladora e interagem com a planta do sistema (ainda na forma de diagrama de blocos) através de interfaces específicas.

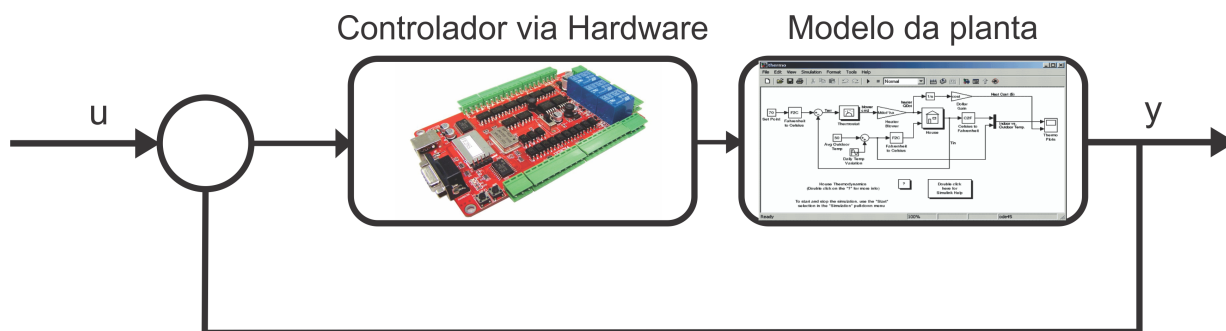


Figura 2.11: Método de verificação: PIL

A utilização do PIL é essencial uma vez que permite a identificação de erros relativos à utilização de sistemas embarcados tais como alocação de memória, ruídos devido a conversão Analógica/Digital (A/D), dentre outros. Ao término desta etapa espera-se que o código possa ser embarcado no dispositivo de controle, ou seja, uma ECU que será utilizada no veículo.

#### 2.4.4 Hardware-In-the-Loop (HIL)

Nesta etapa, o modo utilizado para verificação e validação é o *Hardware-In-the-Loop* (HIL) em que o software do controlador é então embarcado no dispositivo final, por exemplo, uma ECU para aplicações automotivas. A planta do sistema que até então era simulada em um modelo funcional (diagrama de blocos) passa a ser simulada em dispositivos físicos, geralmente computadores em tempo real, que reproduzem o sistema da forma mais próxima possível ao modelo físico. Desta forma, o HIL é utilizado para validação do controlador e principalmente na detecção de erros. A Figura 2.12 mostra uma representação da utilização do HIL.

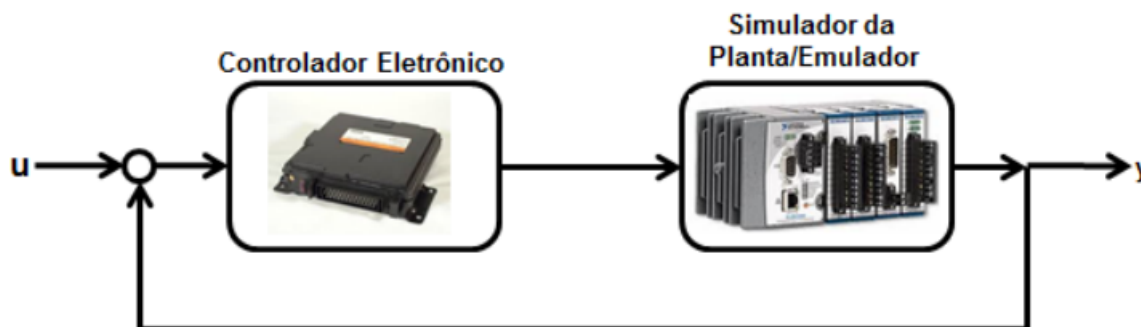


Figura 2.12: Método de verificação: HIL

O principal objetivo do HIL é verificar a integração entre os sistemas de hardwares e os softwares desenvolvidos. De fato o HIL é o teste mais próximo a realidade, porém é o que demanda de mais recursos quando comparados aos outros métodos

Ao término deste ciclo de V&V, espera-se que o controlador alcance um elevado nível de reprodutividade, pois as plataformas de testes utilizadas nesta etapa são capazes de reproduzir com alto grau de confiabilidade o sistema real do veículo.

## 2.5 DESENVOLVIMENTO TRADICIONAL VS DESENVOLVIMENTO BASEADO EM MODELOS (MBD)

No desenvolvimento de software automotivo tradicional as etapas do processo seguem um fluxo sequencial e são executadas em ambientes e plataformas diferentes. Além disso, há de se destacar que diversos passos são realizados de forma manual. No desenvolvimento tradicional, os requisitos do sistema são descritos geralmente utilizando planilhas ou ferramentas de edição de texto. Estes requisitos são traduzidos em códigos de forma manual, num processo que muitas vezes pode resultar em erros. Nas etapas de implementação, geralmente, são utilizados protótipos físicos e os códigos costumam ser implementados no hardware final, elevando assim o custo do projeto bem como o tempo de desenvolvimento. Os testes, geralmente, são realizados após a implementação do sistema já no veículo ou subsistema e os erros encontrados nesta etapa possuem um custo ainda maior para os desenvolvedores. A Figura 2.13 apresenta os processos e os principais desafios envolvidos no desenvolvimento tradicional de software.



Figura 2.13: Etapas tradicionais para o desenvolvimento de software automotivo. Fonte: (STELLA, 2015)

No Desenvolvimento Baseado em Modelos as etapas seguem basicamente o mesmo fluxo, todavia, os métodos utilizados diferem e mantêm o modelo do sistema no centro do processo durante toda fase de desenvolvimentos. Na primeira etapa, os requisitos são utilizados, não apenas como função de documentação, mas para o desenvolvimento de um modelo funcional e executável das respectivas funções. Com um modelo desenvolvido, já é possível realizar testes iniciais a fim de avaliar o sistema frente aos requisitos estabelecidos. Com o modelo desenvolvido e avaliado é possível gerar o código automaticamente e se necessário refazer os testes utilizados não mais os modelos funcionais, mas o código como uma representação do sistema.

Utilizando este método o processo é realizado no mesmo ambiente de desenvolvimento, fazendo com que o tempo de desenvolvimento seja reduzido significativamente além de padronizar as etapas o que facilita ainda mais para os envolvidos no processo. Neste método, testes podem

ser realizados desde as fases iniciais até a fase de implementação. Desta forma, os erros podem ser encontrados e corrigidos antes, se comparado ao desenvolvimento tradicional, reduzindo assim o tempo de desenvolvimento e a utilização de recursos. A Figura 2.14 mostra as etapas no desenvolvimento baseado em modelos.

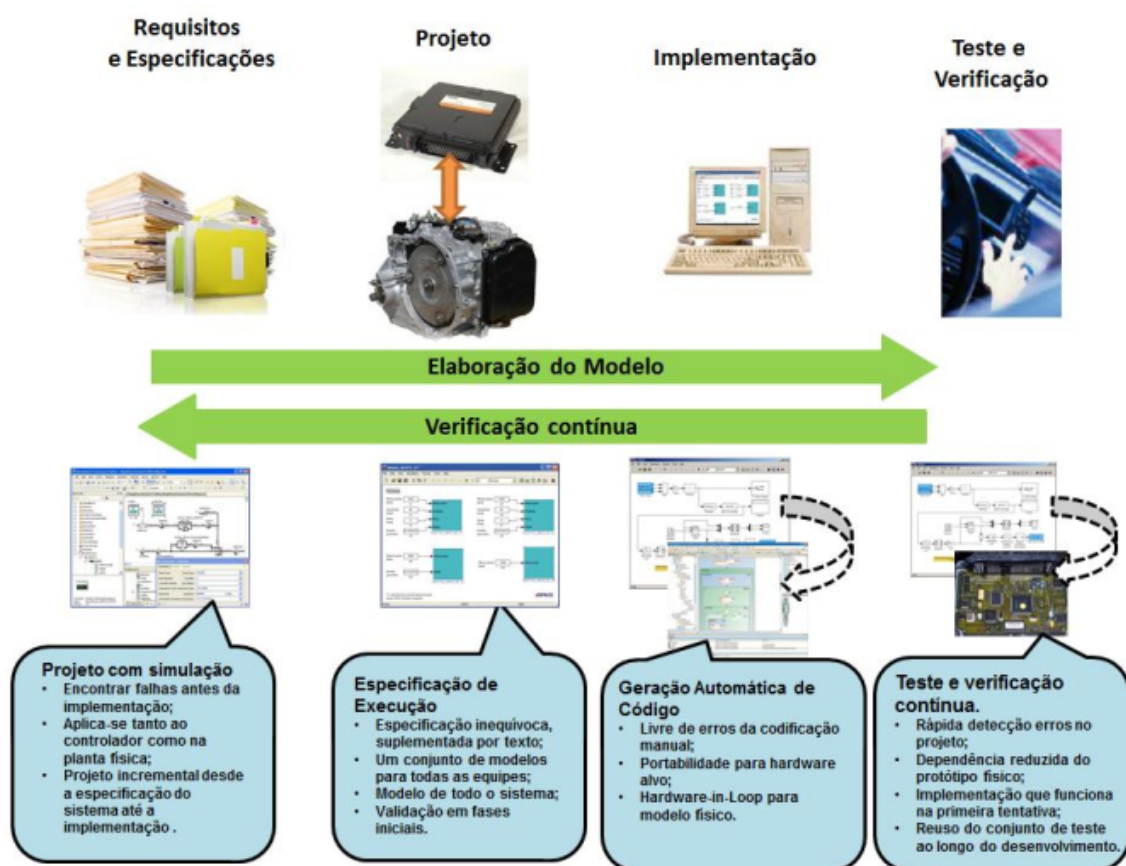


Figura 2.14: Etapas no desenvolvimento baseado em modelos. Fonte: (STELLA, 2015)

Neste sentido, o MBD pode apresentar uma série de benefícios frente aos métodos tradicionais. Em (BROY et al., 2013) um estudo é feito com diversas empresas que utilizam o MBD, integral ou parcialmente, em suas operações. Um dos resultados compara a intensidade da utilização do teste Model-In-the-Loop (MIL) com a quantidade de erros encontrados durante o teste da unidade. De fato, empresas que adotaram o MIL de forma mais intensa em seus desenvolvimentos tiveram uma redução significativa na quantidade de erros encontrados no teste do módulo final. A figura 2.15 mostra um gráfico com o impacto da utilização do MIL como método de teste. O fato de companhias que não utilizaram também apresentarem redução no número erros resulta da possibilidade de utilização de algum dos outros métodos de verificação e testes.

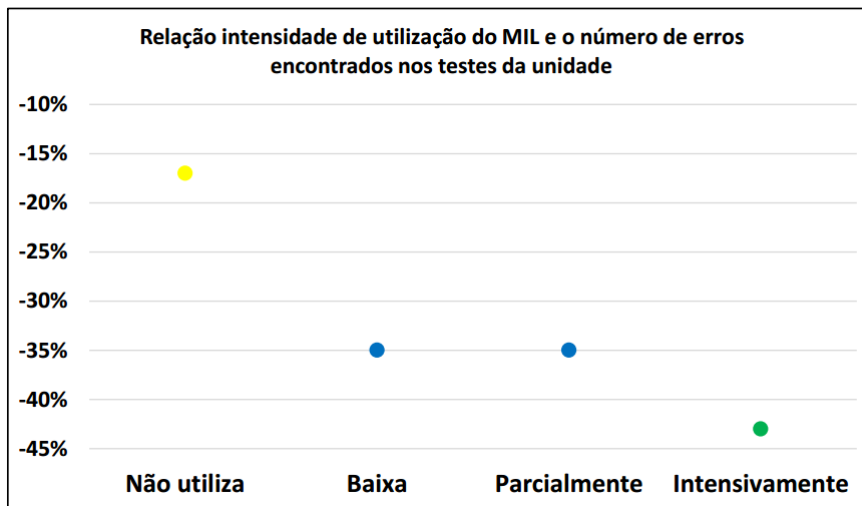


Figura 2.15: Relação entre o grau de utilização do MIL e a quantidade de erros encontradas no código. Adaptado de (BROY et al., 2013).

## 3 SISTEMA EPAS E ESTRATÉGIAS DE CONTROLE

Os capítulos anteriores abordaram a metodologia MBD aplicada ao desenvolvimento do software de controle para um sistema EPAS. Para execução das etapas do modelo V bem como do MBD é necessário, portanto, o modelo dinâmico da planta EPAS bem como a estratégia de controle que irá controlar o sistema. Neste capítulo, serão abordados o modelo dinâmico do sistema EPAS utilizado neste trabalho e as estratégias de controle propostas. Inicialmente, os componentes e a dinâmica do sistema serão discutidos. Em seguida, as equações dinâmicas serão desenvolvidas e por fim o modelo será representado no formato de espaços de estados. No que tange as estratégias de controle, será projetado, inicialmente, um controlador LQG o qual é composto por um controlador LQR e um filtro de Kalman. Em seguida será projetado o controlador MPC com definições, formulações e as técnicas de solução. Além disso, o método de parametrização exponencial que é aplicado ao MPC a fim de reduzir o esforço computacional também será discutido.

### 3.1 MODELO DINÂMICO DO EPAS

O modelo da EPAS a ser utilizado neste trabalho é composto por uma coluna de direção, um sensor de torque, uma ECU, um motor elétrico DC, um sistema de engrenagens e o conjunto pinhão-cremalheira. Este modelo foi extraído do trabalho de Marouf et al. (2012) e é apresentado na figura 3.1.

O sistema de direção elétrica apresentado funciona da seguinte maneira: No instante em que o condutor aplica um movimento no volante é produzido um torque  $T_d$  (*Torque driver*) o qual é medido por um sensor de torque localizado na coluna de direção. Este torque além de atuar diretamente no esterçamento das rodas, de forma mecânica, é utilizado também como um dos parâmetros para o cálculo do Torque de Referência ( $T_{ref}$ ). O cálculo de  $T_{ref}$  é realizado a partir de um mapa que associa o torque  $T_d$  com a velocidade do veículo. Um exemplo de mapa de torque é mostrado na figura 3.2.

De acordo com o mapa, o torque de referência  $T_{ref}$  é calculado com base no torque  $T_d$  e na velocidade do veículo. Em outras palavras,  $T_{ref}$  é alto para baixas velocidades, porém com o aumento da velocidade  $T_{ref}$  diminui gradativamente. Após calculado, o torque  $T_{ref}$  é utilizado então como a referência para o torque  $T_a$  que corresponde ao torque de assistência fornecido pelo motor elétrico. Em outras palavras, o torque  $T_a$  realizado pelo motor elétrico deve seguir o torque de referência  $T_{ref}$ .

Todavia, esta passagem de referência é realizada via corrente do motor, de modo que o torque  $T_{ref}$  é multiplicado por um ganho  $K_t$  gerando assim uma corrente de referência  $I_{ref}$ . Portanto, a



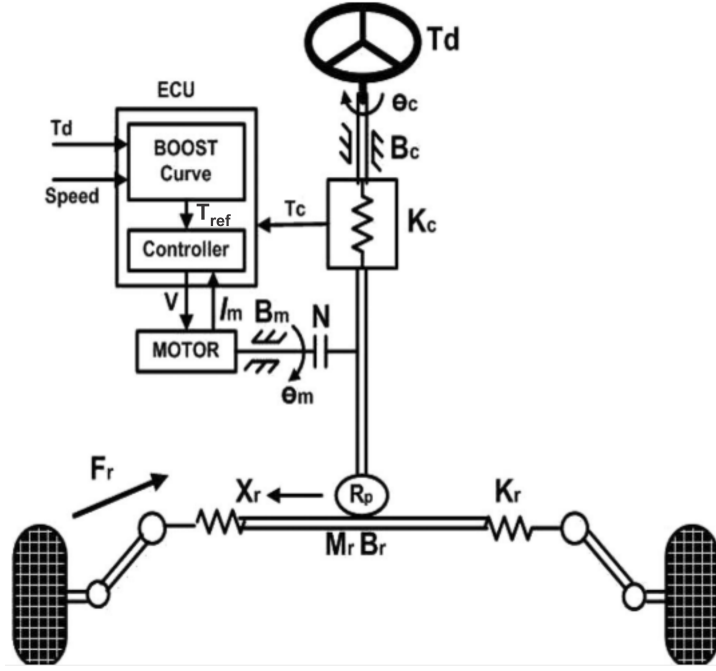


Figura 3.1: Modelo da EPAS. Adaptado de Marouf et al. (2012).

corrente,  $I_m$ , do motor é regulada de modo a seguir a corrente de referência  $I_{ref}$ .

A sensação da rigidez da direção é outra característica importante dos sistemas EPAS e deve ser levada em conta no projeto do controlador (Sugitani, 1997). De maneira geral, esta sensação se refere ao quanto de informação é transmitida através sistema de direção até volante do veículo. Grande parte da rigidez aparente da direção é ocasionada devido ao contato entre o pneu e o solo. Portanto, para um correto funcionamento do EPAS é necessário que o sistema forneça ao condutor a sensação de rigidez apenas baseada no mapa de torque e seja portanto insensível a perturbações e flutuações causadas pelo contato pneu-solo.

Com base no modelo apresentado, as variáveis  $\theta_c$ ,  $\theta_m$  e  $x_r$  correspondem, respectivamente, a posição angular da coluna de direção, posição angular do motor e a posição da barra de direção. Partindo das dinâmicas da coluna de direção, do conjunto pinhão-cremalheira e da dinâmica do motor elétrico, e de acordo com as leis de Newton, as equações de movimento do sistema dinâmico EPAS são definidas como:

$$J_c \ddot{\theta}_c = -K_c \theta_c - B_c \dot{\theta}_c + K_c \frac{\theta_m}{N} - F_c \text{sign}(\dot{\theta}_c) + T_d \quad (3.1)$$

$$J_{eq} \ddot{\theta}_m = K_c \frac{\theta_c}{N} - \left( \frac{K_c}{N^2} + \frac{K_r R_p^2}{N^2} \right) \theta_m - B_{eq} \dot{\theta}_m + K_t I_m - F_m \text{sign}(\dot{\theta}_m) - \frac{R_p}{N} F_r \quad (3.2)$$

em que  $J_{eq} = J_m + (R_p^2/N^2)M_r$ ,  $B_{eq} = B_m + (R_p^2/N^2)B_r$  e  $\theta_m = Nx_r/R_p$ .

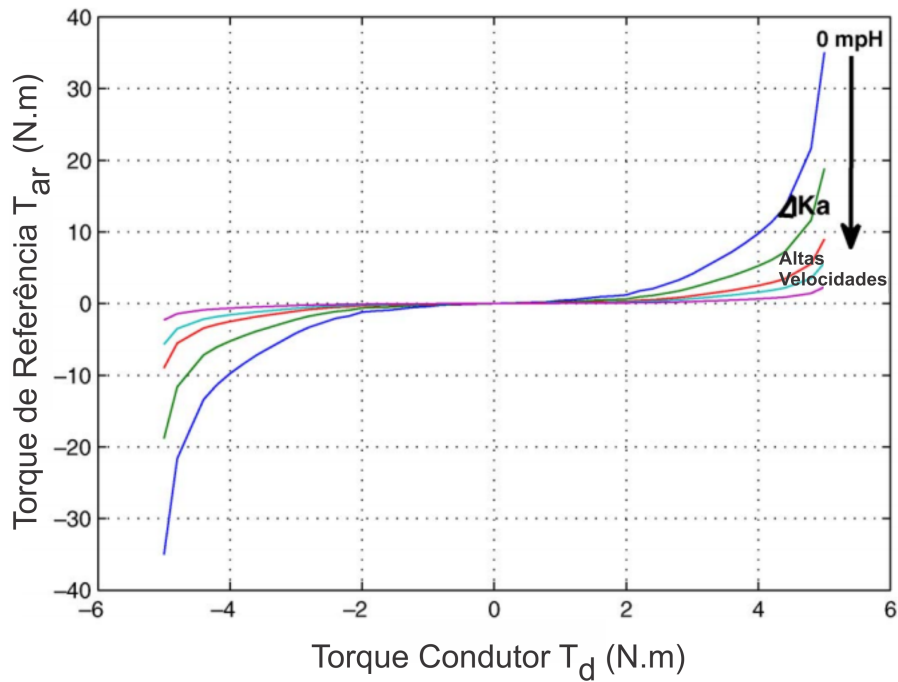


Figura 3.2: Mapa de torque. Adaptado de Marouf et al. (2012).

O motor elétrico é acoplado à coluna de direção por meio de um sistema de engrenagens. Este componente tem a função de amplificar o torque do motor e transmiti-lo à coluna de direção resultando portanto no torque de assistência  $T_a$ . Sendo assim, a equação que descreve a dinâmica do motor elétrico é apresentada na eq. 3.4 em que  $U$  corresponde a tensão e  $I_m$  a corrente no motor. Os demais parâmetros utilizados nas equações são definidos na Tabela 3.1:

$$T_a = I_m K_t N \quad (3.3)$$

$$L_m \dot{I}_m = -R_m I_m - K_t \dot{\theta}_m + U \quad (3.4)$$

As equações apresentadas acima mostram que o sistema EPAS contém uma série de entradas e saídas sendo caracterizado dessa forma como um sistema do tipo Múltiplas-Entradas e Múltiplas-saídas (MIMO). Uma maneira de representação de sistemas do tipo MIMO é forma de espaço de estados. Nesta forma de representação, um sistema de ordem  $n$  é descrito por meio de  $n$  equações de primeira ordem que são definidas em função das variáveis de estado (ROWELL, 2002). Os principais benefícios da representação de um sistema em espaço de estados são: pode ser utilizado para sistemas lineares quanto para não lineares, poder ser aplicado para sistemas MIMO e sistemas lineares invariantes no tempo (LTI).

Considerando um sistema LTI de ordem  $n$  contendo  $r$  entradas ou comandos, a representação em espaço de estados é definida da seguinte forma:

Tabela 3.1: Parâmetros do modelo do EPAS

Parâmetro	Descrição
$J_c$	Momento de inércia da coluna de direção
$B_c$	Amortecimento viscoso da coluna de direção
$K_c$	Rigidez da coluna de direção
$F_c$	Atrito na coluna de direção
$M_c$	Massa da cremalheira
$B_r$	Amortecimento viscoso da cremalheira
$R_p$	Raio do pinhão
$K_r$	Taxa de amortecimento do pneu
$J_m$	Momento de inércia do motor elétrico
$B_m$	Amortecimento viscoso no eixo do motor
$F_m$	Atrito do motor
$K_t$	Constante torque - corrente
$L_m$	Indutância do motor
$R_m$	Resistência do motor
$N$	Relação de engrenagem

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + w(t) \\ y(t) &= Cx(t) + Du(t) + v(t) \end{aligned} \quad (3.5)$$

onde a  $A$  é a matriz de estado de dimensão  $n \times n$ ,  $B$  é a matriz de entradas de dimensão  $n \times r$ ,  $x(t)$  é vetor os estados contendo as variáveis de estado que descrevem o comportamento do sistema e  $u$  são as entradas ou comandos. A segunda equação corresponde a equação das saídas, nesta equação são definido quais dos estados do sistema serão utilizados como saída. Dessa forma,  $y(t)$  corresponde a um vetor de colunas contendo as saídas desejadas,  $C$  é a matriz de seleção das saídas de dimensões  $m \times n$  onde  $m$  é o número de saídas definidas e  $D$  corresponde a matriz de transmissão direta com dimensões  $m \times r$ . Os termos  $w(t)$  e  $v(t)$  correspondem aos ruídos de medida que afetam os estados e as saídas do sistema respectivamente. Estes sinais são, geralmente, definidos como um processo gaussiano e correspondem, no caso das saídas, aos ruídos nos sensores que realizam a medição e para os estados correspondem a perturbações na planta física.

Especificamente para o caso do EPAS apresentado neste capítulo, a representação em espaço de estados é definida da seguinte forma: O vetor dos estados é  $x = [\theta_c \dot{\theta}_c \theta_m \dot{\theta}_m I_m]^T$ , enquanto o vetor das entradas é dado  $u = [U T_d T_r]^T$ , sendo  $U$  a tensão no terminal do motor DC,  $T_d$  é o torque realizado pelo condutor e  $T_r$  é o torque de reação ocasionado pelo contato pneu-pista. Por fim,  $y = [\theta_c \theta_m]^T$  corresponde ao vetor das saídas do sistema que são diretamente medidas através dos sensores. Portanto, as matrizes que definem o sistema em espaço de estado são:

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ -\frac{K_c}{J_c} & -\frac{B_c}{J_c} & \frac{K_c}{J_c N} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \frac{K_c}{J_{eq} N} & 0 & -\left(\frac{R_p^2 K_r + K_c}{J_{eq} N^2}\right) & \frac{-B_{eq}}{J_{eq}} & \frac{K_t}{J_{eq}} \\ 0 & 0 & 0 & \frac{-K_t}{L_m} & \frac{-R_m}{L_m} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{1}{J_c} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \frac{-1}{N J_{eq}} \\ \frac{1}{L_m} & 0 & 0 \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Uma análise que deve ser realizada a fim de verificar se o sistema pode ser devidamente controlado e se os estados podem ser observados são as análises de controlabilidade e observabilidade. Um sistema é controlável se existe um vetor de entradas  $(t)$  que leva o sistema de um estado inicial  $x(t_0)$  para qualquer outro estado em um tempo finito. No tocante a observabilidade, um sistema é dito observável no tempo  $t_0$  se o sistema no estado  $x(t_0)$  é possível determinar este estado através da observação das saídas em um intervalo de tempo finito.

Para o modelo do EPAS apresentado na matriz acima, a análise de controlabilidade e estabilidade foi realizada com o software Matlab/Simulink<sup>®</sup> utilizando os pares de matrizes  $(A, B)$  para verificar a controlabilidade e os pares  $(A, C)$  para verificar a observabilidade. Verificou-se que o sistema é totalmente controlável e observável, portanto as estratégias de controle propostas e o observador de estado por filtro de Kalman podem ser aplicadas ao modelo.

Com o modelo do EPAS devidamente definido e representado em espaço de estados e as verificações de controlabilidade e observabilidade realizadas, o passo seguinte consiste em projetar os controladores que serão utilizados no modelo. As próximas seções abordam, respectivamente, os controladores LQG e MPC para controle do EPAS.

### 3.2 CONTROLADOR LINEAR QUADRÁTICO (LQR)

O projeto de um sistema de controle para atuar em determinada planta é desenvolvido, geralmente, segundo alguns objetivos qualitativos tais como: estabilização da planta, obtenção de determinada resposta em regime transiente, rejeição de ruídos de medidas, redução do erro no estado estacionário e robustez a variação de parâmetros. O projeto de controladores para sistemas com apenas uma entrada e uma saída (SISO) é geralmente obtido por soluções analíticas através de diversos métodos conhecidos: alocação de polos, avanço e atraso de fase, controladores proporcionais, integrais e derivativos. O problema se torna mais complexo quando a planta do sistema é do

tipo MIMO, em que as técnicas de controle clássicas podem não serem aplicáveis e/ou apresentar limitações (DELATORE, 2011).

Com o surgimento das teorias de controle moderno, o projeto de controladores se estendeu para sistemas MIMO que podem representar por completo a dinâmica de um sistema através de um conjunto de equações simplificadas. Dentre as classes de controladores, geralmente, aplicadas em sistemas MIMOs destacam-se os controladores ótimos. O controlador ótimo define um sinal de controle de modo a fazer com que o sistema satisfaça algumas restrições físicas e, ao mesmo tempo, minimize (ou maximize) alguns critérios de desempenho (CHACHUAT, 2007).

O LQR é um controlador ótimo que é definido como a solução para um problema de otimização definido através de uma função custo a ser minimizada. Este controle pode apresentar estabilidade em malha fechada, níveis garantidos de robustez, além de ser relativamente simples de utilizar (LEVINE, 1996).

Neste trabalho, os controladores projetados serão implementados em *hardware*. Desta forma, para aplicações digitais é necessário a discretização das matrizes no tempo. Sendo assim, o formato de espaço de estados contínuo é convertido em discreto e assume o seguinte formato (CHITU et al., 2013)

$$\begin{aligned}x(k+1) &= A_d x(k) + B_d u(k) + w(k) \\y(k) &= C_d x(k) + v(k)\end{aligned}\tag{3.6}$$

onde as matrizes  $A_d$ ,  $B_d$ ,  $C_d$  são as matrizes constantes do sistema discretizadas no tempo e  $k$  é o período de amostragem. Considerando o sistema no formato de espaço de estados apresentado, o vetor de estados é dado por  $x(k) \in \mathbb{R}^n$ , o vetor de entradas dado por  $u(k) \in \mathbb{R}^m$  e o vetor de saídas  $y(k) \in \mathbb{R}^l$ , a lei de controle para realimentação dos estados é dada por (MOUDGALYA, 2007):

$$u(k) = -Kx(k)\tag{3.7}$$

Sendo  $K \in \mathbb{R}^{m \times n}$  a matriz de realimentação de estados. O principal objetivo do LQR é encontrar uma matriz de ganho  $K$  de forma que uma função custo é minimizada (BILL; HILL, 2004). A função custo a ser minimizada é dada pelo seguinte formato:

$$J = \sum_0^{\infty} (x(k)Qx(k) + u(k)Ru(k))\tag{3.8}$$

Em que as matrizes  $Q \in \mathbb{R}^{(n \times n)}$  e  $R \in \mathbb{R}^{(m \times m)}$  são matrizes de ponderação que determinam a resposta do sistema em malha fechada. A matriz  $Q$  consiste na matriz de ponderação dos estados e  $R$  é a matriz de ponderação dos sinais de entrada. A escolha das matrizes  $Q$  e  $R$  permite que se

obtenha uma relação entre o esforço de controle e tempo de resposta.  $Q$  deve ser definida como uma matriz simétrica semi-definida positiva e  $R$  como uma matriz simétrica definida positiva (BILL; HILL, 2004).

Após a escolha adequada das matrizes  $Q$  e  $R$  a matriz de ganho  $K$  pode ser obtida por:

$$K = (R + B_d^T S(k+1) B_d)^{-1} B_d^T S(k+1) A_d \quad (3.9)$$

Em que  $S(k)$  pode ser obtido, analiticamente, através da solução da equação de Ricatti:

$$S(k) = A_d^T [S(k+1) - S(k+1) B_d R^{-1} B_d^T S(k+1)] A_d + Q \quad (3.10)$$

De forma que, se o sistema é estável, existe apenas uma matriz definida positiva  $S(k)$  que satisfaça essa equação (OGATA, 2010).

Uma particularidade do LQR é o fato de os estados serem regulados em zero no regime permanente. Todavia, para utilização deste controlador no sistema EPAS é desejado que os estados sejam rastreados segundo uma referência. Para o caso do EPAS, a corrente do motor auxiliar deve ser rastreada de acordo com a corrente de referência, portanto é necessário definir tantos os estados  $x^d$  desejados quanto a variável de comando  $u^d$  desejada, de forma a atingir a saída  $y^d$  desejada.

Os valores desejados em regime permanente de  $x^d$ ,  $u^d$  e  $y^d$  estão relacionados através das seguintes equações (ALAMIR, 2013)).

$$x^d = A_d x^d + B_d u^d \quad (3.11)$$

$$y^d = C x^d \quad (3.12)$$

Considerando  $I$  como a matriz identidade, as equações 3.11 e 3.12 podem ser reescritas na forma:

$$\begin{pmatrix} I - A_d & -B_d \\ C & 0 \end{pmatrix} \begin{pmatrix} x^d \\ u^d \end{pmatrix} = \begin{pmatrix} 0 \\ y^d \end{pmatrix} \quad (3.13)$$

Dessa forma, uma vez que a saída  $y^d$  é definida, para se obterem os valores de  $x^d$  e  $u^d$  basta pré-multiplicar a equação 3.13 pela inversa da primeira matriz do lado esquerdo, haja visto que as matrizes  $A_d$ ,  $B_d$  e  $C$  são conhecidas. A utilização da matriz  $C$  é importante para que a solução da equação 3.13 seja única, caso contrário seriam possíveis infinitos valores de  $x^d$  e  $u^d$ .

Por fim,  $x^d$  é utilizado como um vetor de referência para os estados e  $u^d$  como referência para o comando. Portanto, o comando  $u(k)$  utilizado para que os estados sigam a trajetória de

referência,  $y^d$ , desejada é dado na equação 3.14, em que  $K$  é o ganho do LQR.

$$u(k) = u^d - K(x - x^d) \quad (3.14)$$

### 3.3 FILTRO DE KALMAN

Para o projeto do controlador LQG é necessário a realimentação total dos estados do sistema. Todavia, em algumas aplicações nem todos os estados do sistema podem ser medidos diretamente através de sensores. Nestas situações, observadores de estados podem ser utilizados a fim de estimar os estados não medidos diretamente e assim fornecer a realimentação total dos estados à estratégia de controle.

O Filtro de Kalman é um estimador de estados que satisfaz dois critérios: primeiro busca manter a média dos valores dos estados estimados igual a média dos valores dos estados reais e segundo busca que os estados estimados varie menos possível em relação aos estados medidos, não apenas a média é um estimador que garante a menor variação possível dos estados estimados, em outras palavras garantir uma variância pequena do erro entre os valores reais e os estimados.

Para implementação do filtro de Kalman, algumas condições a respeito dos ruídos que afetam o sistema devem ser satisfeitas. Relembrando da equação do espaço de estados (equação 3.5),  $w(k)$  é o ruído que afeta a planta e  $v(k)$  é o ruído que afeta as medições. Para o filtro de Kalman é necessário assumir que o valor da média de ambos os sinais é zero. Além disso, é necessário assumir que não há qualquer correlação entre  $w(k)$  e  $v(k)$ , ou seja, para qualquer instante de tempo  $k$  os sinais são independentes (SIMON, 2001). Desta forma, Simon (2001) apresenta as matrizes de covariância dos ruídos e a formulação do filtro de Kalman como:

Covariância para o ruído no processo:

$$S_w = E(w_k w_k^T) \quad (3.15)$$

Covariância para o ruído de medida:

$$S_z = E(v_k v_k^T) \quad (3.16)$$

onde  $w_k^T$  e  $v_k^T$  correspondem, respectivamente, a transposta dos vetores dos ruídos de processo e de medida e  $E()$  se refere ao valor esperado ou esperança.

Com as devidas considerações sendo satisfeitas, a equação do estimador do filtro de Kalman assume o seguinte formato:

$$\hat{x}(k+1) = A_d \hat{x}(k) + B_d u(k) + K_0 (y(k) - C \hat{x}(k)) \quad (3.17)$$

onde  $\hat{x}$  corresponde aos estados estimados e a matriz  $C$  seleciona os estados que podem ser medidos diretamente através de sensores e  $y(k)$  é a equação das saídas que é definida na equação 3.5. O termo  $K_0$  é a matriz de ganho do observador que pode ser obtida através da equação 3.18.

$$K_0 = A_d P_k C^T (C P_k C^T + S_z)^{-1} \quad (3.18)$$

onde  $P_k$  é denominada de matriz de estimação da covariância do erro e pode ser obtida pela seguinte equação:

$$P_k + 1 = A_d P_k A^T + S_w - A P_k C^T S_z^{-1} C P_k A_d^T \quad (3.19)$$

Examinando a equação 3.18 nota-se que se o ruído de medida é muito alto o termo  $S_z$  também será alto, de forma que  $K_0$  será baixo e portanto os estados medidos  $y(k)$  não terão muito impacto ao calcular o próximo  $\hat{x}$ . Por outro lado, se o ruído de medida é baixo,  $S_z$  será baixo e conseqüentemente os termos medidos diretamente terão grande impacto sobre o cálculo do o próximo  $\hat{x}$ .

### 3.4 REGULADOR LINEAR QUADRÁTICO GAUSSIANO (LQG)

De fato, os procedimentos de cálculo para obtenção do ganho do controlador LQR quanto do ganho do Filtro de Kalman são similares, uma vez que, os ganhos são obtidos através da minimização de uma função custo (KURUCSó et al., 2015). Quando é feita a combinação de um estimador ótimo com um regulador ótimo o resultado é um compensador denominado Regulador Linear Quadrático Gaussiano (LQG) (ZOLOTAS et al., 2007). Este procedimento reduz o problema em dois subproblemas: Inicialmente determina a lei de controle ótima para realimentação de estado (LQR) e em seguida obtêm a estimação dos estados através do filtro de Kalman. A figura 3.3 mostra a estrutura do controlador LQG composto pelo LQR e por um filtro de Kalman, os sinais mostrados na figura são definidos nas seções acima.

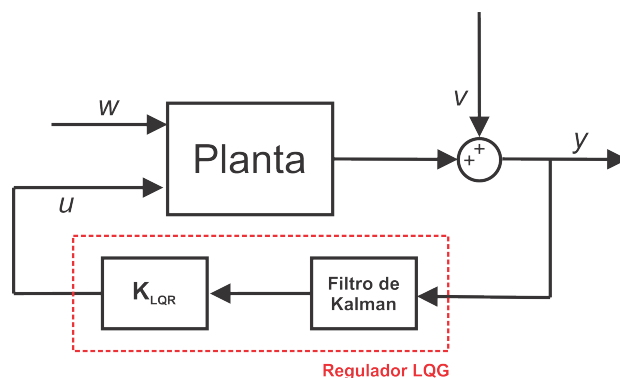


Figura 3.3: Estrutura do compensador LQG. Adaptado de (KURUCSó et al., 2015).



Nesta seção foi apresentada a estratégia de controle LQG que será utilizada, neste trabalho, para controle do sistema EPAS. De fato, o controlador LQR é uma estratégia que apresenta diversos benefícios tais como: estabilidade e robustez e combinado com o Filtro de Kalman constituem uma estratégia completa capaz de observar os estados e realizar controle mesmo na presença de incertezas e/ou ruídos que afetam o sistema. Todavia, uma limitação do LQG é o fato de não considerar restrições, principalmente, no sinal de controle ou nas saídas da planta. Esta funcionalidade é de extrema importância, uma vez que, os atuadores possuem limitações físicas inerentes ao dispositivo como: limites de torque, tensão e corrente.

Uma estratégia de controle que leva em consideração essas restrições é o Controle Preditivo Baseado em Modelos (MPC). A seção seguinte apresenta esta estratégia de controle, de forma que, será abordado o funcionamento, equacionamento e métodos de solução.

### 3.5 CONTROLE PREDITIVO BASEADO EM MODELOS (MPC)

O nome "Controle Preditivo Baseado em Modelos (MPC)" surge devido a forma com a qual a lei de controle é calculada (GARCIA; PRETT; MORARI, 1989). A estratégia de Controle MPC consiste em obter uma sequência ótima de ações de controle de forma *on-line*, a cada período de amostragem, através da minimização de uma função custo que expressa os objetivos do controle (MAYNE et al., 2000). Após a obtenção da sequência de controle ou comandos, o primeiro comando da sequência é aplicado à planta e o processo é novamente repetido a cada período de amostragem (MURILO, 2009).

De acordo com Morari e Lee (1999) as primeiras publicações relativas à aplicações puras do MPC remontam a década de 80 com os trabalhos de Richalet et al. (1978) que propõem o algoritmo *Identification and Command* (IDCOM), Cutler e Ramaker (1980) cuja a estratégia foi denominada *Dynamic Matrix Control* (DMC) e Clarke, Mohtadi e Tuffs (1987) com o *Generalized Predictive Control* (GPC).

Devido ao fato de que no MPC a lei de controle é obtida através de algoritmo de otimização, sendo calculada a cada iteração, as primeiras aplicações foram desenvolvidas para a indústria farmacêutica e petroquímica cujo os processos são lentos o suficiente a ponto de garantir que todo o esforço computacional realizado no cálculo das soluções ótimas possa ser empregado no período de amostragem definido. Este fato, representava um ponto crítico quando se trata de sistemas rápidos, onde ao ser utilizado o MPC, o tempo de processamento do controle pode inviabilizar utilização destes sistemas em tempo-real (ABU-AYYAD; DUBAY, 2007)

No que diz respeito ao desenvolvimento teórico, algumas características foram incorporadas ao MPC tais como: algoritmos que melhorem a estabilidade, robustez e técnicas para o MPC não linear. No tocante às aplicações industriais, outros segmentos como a indústria de papel, alimentação e mineração passaram a adotar a estratégia (LEE, 2011).

Nos anos seguintes, grande parte dos avanços acadêmicos e industriais do MPC se concen-

traram em desenvolver o processo de otimização a fim de reduzir a carga computacional e conseqüentemente diminuir o tempo de cálculo. Alinhado a este fato, a evolução dos hardwares permitiram que as aplicações MPC se estendessem a sistemas de dinâmicas rápidas. Algumas aplicações do MPC relatadas neste tipo de sistemas incluem: controle de tração do veículo, suspensão, sistema de propulsão automotiva e mais recentemente na integração de controle de estabilidade e controle de tração para veículos elétricos (NGUYEN et al., 2016), (HSIEH; LIU, 2016) e (BORDONS et al., 2010).

Devido às melhorias realizadas nos métodos de cálculo do controlador MPC e o avanço nos hardwares de processamento, as aplicações se expandiram pra diversas outras áreas. A figura 3.4 mostra as principais aplicações do MPC. O fato de grande parte das aplicações serem na indústria automotiva comprova que a questão de tempo de execução sofreu grande evolução uma vez que neste segmento o tempo de execução é, geralmente, um fator crítico.

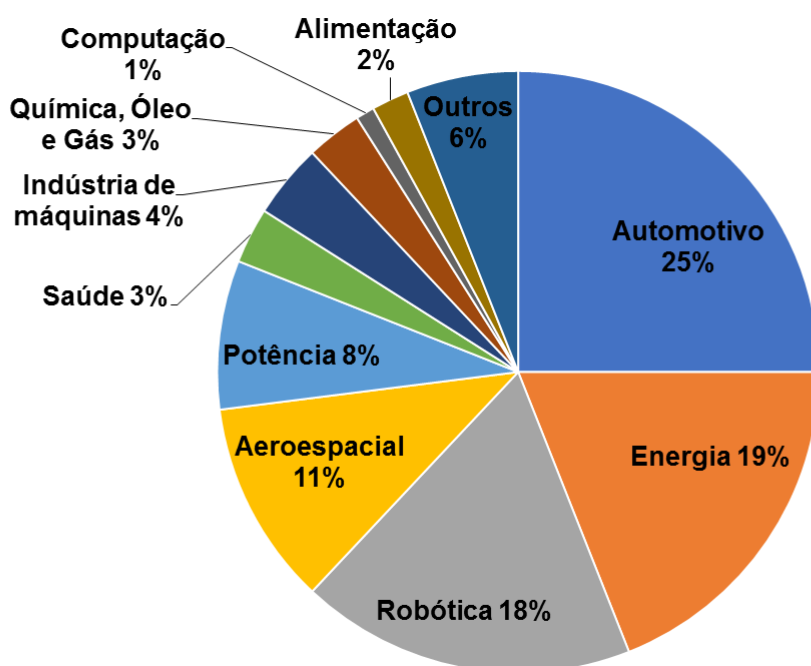


Figura 3.4: Principais aplicações do MPC. Adaptado de Domahidi (2016)

A figura 3.5 ilustra o conceito realizado no MPC, em que num determinado instante  $k$  o comportamento do sistema sobre o horizonte  $N$ , levando em conta as restrições impostas, é predito. Em seguida, utilizando um algoritmo de otimização uma sequência de comandos futuros que melhor aproxima o estado predito à referência futura é calculada. Dessa sequência de comandos calculada apenas o primeiro é aplicado levando o sistema ao próximo instante  $k + 1$ .

O fato de se utilizar apenas o primeiro comando da sequência de comandos calculada se explica pelo fato de, durante o processo o sistema pode sofrer perturbações inesperadas de forma que aqueles valores contidos em  $\tilde{u}(k)$  podem se tornar incorretos, pois não estão adaptados aos novos parâmetros .

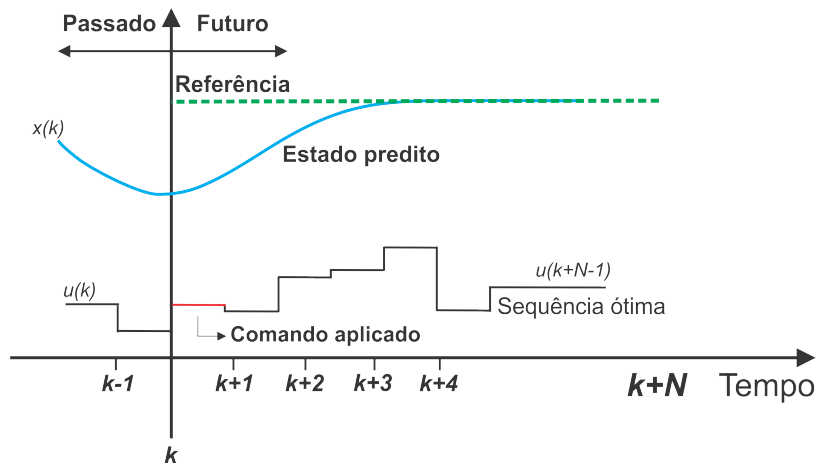


Figura 3.5: Mapa de previsão do MPC. Adaptado de Alamir (2013).

Uma das principais características do MPC é o fato de a estratégia considerar restrições tanto no sinal de comando quanto nos estados do sistema. De fato, a imposição de restrições é uma funcionalidade extremamente eficaz uma vez que os atuadores são dispositivos naturalmente limitados em suas operações. Por outro lado, os estados também podem conter limitações naturais ao processo, tais como temperaturas, velocidades e pressão que não podem atingir certos limites. Portanto, considerar restrições nos sinais presentes no processo é uma forma eficaz de respeitar os limites físicos reais dos dispositivos envolvidos. Esta característica faz com que o MPC seja a única estratégia de controle disponível para manipulação direta de restrições (MURILO, 2009).

Outro aspecto importante do MPC é o fato de ser um processo muito bem estabelecido para sistemas lineares. Todavia, sabe-se que grande parte das aplicações reais apresentam características não-lineares cuja dinâmica do processo não pode ser bem representada através da linearização do sistema. Para estes casos, a estratégia de Controle Preditivo Não-Linear (NMPC) pode ser utilizada, sendo esta uma estratégia que mantém as mesmas funcionalidades do MPC adicionando a capacidade de abordar sistemas não-lineares (MURILO, 2009).

A robustez do controlador é outro desafio no MPC. De fato, em sistemas reais incerteza nos parâmetros do modelo bem como ruídos no processo são comumente encontrados, portanto a estratégia de controle deve ser robusta o suficiente para tratar esses desvios e garantir o pleno funcionamento do sistema (BEMPORAD; MORARI, 1999). Apesar de haver aplicações sólidas para utilização de robustez em controles lineares, o mesmo não acontece para sistemas lineares com restrições como o MPC. Estas aplicações em MPC necessitam de uma aplicação ainda maior para tratar os desafios inerentes as fontes de incerteza e ruídos. Em Goodwin et al. (2014) são apresentadas diferentes abordagens do MPC de forma a apresentar características de robustez devido à fontes de incerteza.

### 3.5.1 Estrutura do Controlador

O algoritmo do MPC pode ser separado em duas partes principais: um preditor e um otimizador (ABU-AYYAD; DUBAY, 2007). O preditor tem a função de gerar um vetor de saídas previstas  $\hat{y}$  em um horizonte de previsão  $N$ . Em seguida o otimizador calcula o presente e os futuros sinais de comando  $u(k), \dots, u(k+N-1)$  de forma que as saídas previstas  $\hat{y}$  se aproximem da trajetória de referência. Os sinais de comando definidos pelo otimizador são definidos de forma a minimizar uma função custo além de respeitar as restrições impostas. A figura 3.6 mostra a estrutura geral do controlador MPC.

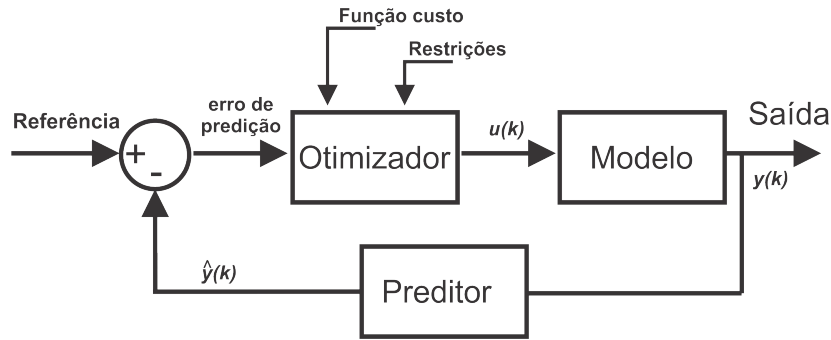


Figura 3.6: Estrutura do controlador MPC. Adaptado de (ABU-AYYAD; DUBAY, 2007)

Dessa forma, a sequência de passos realizadas na estratégia MPC é a seguinte:

1. No instante  $k$  mede os estados;
2. Calcula a sequência de ações futuras que minimizem uma função custo;
3. Aplica a primeira ação de controle  $u(k)$  da sequência;
4. No instante  $k + 1$ , mede os estados e reinicia o processo.

### 3.5.2 Formulação do controle preditivo para sistemas lineares

A formulação MPC utilizada neste trabalho é baseada no trabalho de (ALAMIR, 2013). Considerando o sistema definido no formato de espaço de estados em que  $x(k) \in \mathbb{R}^n$  corresponde ao vetor dos estados do sistema sendo,  $n$  é a quantidade de estados da planta,  $u(k) \in \mathbb{R}^{n_u}$  consiste no vetor de comando, sendo  $n_u$  a quantidade de entradas. Baseado na otimização de uma função objetiva, geralmente definida com uma função quadrática que leva em consideração a minimização do erro bem como as restrições desejadas, calcula-se uma sequência de  $N$  ações futuras no formato da equação 3.20.

$$\tilde{u}(k) := (u(k) \quad u(k+1) \cdots u(k+N-1))^T \in \mathbb{R}^{N \cdot n_u} \quad (3.20)$$

Dessa forma, dada a sequência de ações futuras e conhecendo o estado inicial  $x(k)$  o algoritmo de previsão é capaz de prever  $N$  saídas do sistema de acordo com a equação 3.21.

$$\tilde{x} := (x(k+1) \ x(k+2) \dots x(k+N-1))^T \in \mathbb{R}^{N.n} \quad (3.21)$$

A sequencia  $\tilde{u}(k)$  é calculada através da resolução de uma função custo, a qual é modelada de acordo com um objetivo de controle. As seções abaixo apresentam a formulação do MPC apresentada por (ALAMIR, 2013).

### 3.5.2.1 Horizonte de predição

Considerando que um sistema LTI seja representado em formato de espaço de estados, um mapa de predição é utilizado para prever os estados futuros  $x(k+i) \forall i \in 1, \dots, N$ . De forma que, dado o estado atual e uma sequencia de comandos  $\hat{u}(k)$  já calculada, o primeiro passo predito é dado por:

$$x(k+1) = Ax(k) + Bu(k) \quad (3.22)$$

Novamente, aplicando o mapa de predição para os seguintes passos, tem-se o seguinte:

$$\begin{aligned} x(k+2) &= Ax(k+1) + Bu(k+1) = A^2x(k) + ABu(k) + Bu(k+1) \\ &\vdots \\ x(k+i) &= A^i x(k) + [A^{i-1}B, \dots, AB, B] \begin{bmatrix} u(k) \\ \vdots \\ u(k+i-2) \\ u(k+i-1) \end{bmatrix} \end{aligned} \quad (3.23)$$

De maneira geral, para  $i \in 1, \dots, N$  a formulação pode ser escrita na forma compacta:

$$x(k+1) = \Phi_i x(k) + \Psi_i \hat{u}(k) \quad (3.24)$$

Sendo as matrizes  $\Psi_i$  e  $\Phi_i$  com  $i \in 1 \dots N$  definidas por:

$$\Phi_i = A^i \quad (3.25)$$

$$\Psi_i := [A^{i-1}B \dots AB, B] \begin{pmatrix} \Pi_1^{n_u, N} \\ \vdots \\ \Pi_{i-1}^{(n_u, N)} \\ \Pi_i^{(n_u, N)} \end{pmatrix}$$

onde  $\Pi_i^{(n_1, N)}$  é a matriz de seleção que seleciona o  $i$ -ésimo vetor de dimensão  $n_1$ , sendo este um vetor que está contido em um vetor maior que é composto por  $N$  vetores de dimensão  $n_1$ . A matriz de seleção assume o seguinte formato:

$$\Pi_i^{(n, N)} = \left( \underbrace{\mathbb{O}_{n \times n} \cdots \mathbb{O}_{n \times n}}_{(i-1) \text{ termos}} \quad \mathbb{I}_{n \times n} \quad \underbrace{\mathbb{O}_{n \times n} \cdots \mathbb{O}_{n \times n}}_{(N-i) \text{ termos}} \right) \quad (3.26)$$

Em que  $\mathbb{O}_{n \times n}$  corresponde a uma matriz de zeros e  $\mathbb{I}_{n \times n}$  a uma matriz identidade com dimensões correspondentes.

### 3.5.2.2 Função Custo

O objetivo central do controle é fazer com que a saída do sistema seja rastreada a uma referência. A saída do sistema pode ser definida como um estado ou a combinação de estados e é dada pela seguinte forma.

$$y_r = C_r x(k)$$

em que  $y(k) \in \mathbb{R}^{n_r}$  corresponde às saídas reguladas do sistema que são selecionadas de acordo com a matriz  $C_r \in \mathbb{R}^{n_r \times n_r}$ .

Dessa forma, a função custo para uma sequência de comandos  $\tilde{u}$  é definida pela equação 3.27.

$$J(\tilde{u}|x(k), \tilde{y}_r^d(k), u^d) := \sum_{i=1}^N \|y_r(k+i) - \tilde{y}_r^d(k+i)\|_{Q_y}^2 + \sum_{i=1}^N \|\Pi_i^{(n_u, N)} \tilde{u} - u^d\|_{Q_u}^2 \quad (3.27)$$

onde  $\tilde{y}_r^d(k)$  corresponde a referência a ser rastreada pela saída do sistema e  $u^d$  é o valor desejado para as entradas do sistema. Dessa maneira, a primeira parte da equação trata-se da penalização do erro entre os valores atuais das saídas do sistema e a saídas desejadas. O segundo termo penaliza o comando em relação a um ponto de equilíbrio.  $Q_y \in \mathbb{R}^{n_r \times n_r}$  consiste numa matriz quadrada utilizada para penalizar os estados utilizados como saída do sistema e  $Q_u \in \mathbb{R}^{n_u \times n_u}$  é uma matriz simétrica definida positiva utilizada para penalização das entradas, em que  $n_u$  é a quantidade de entradas do sistema.

A equação 3.27 pode ser colocado na forma de uma função quadrática padrão, em que os parâmetros da função são definidos em seguida:

$$J(\tilde{u}|x(k), \tilde{y}_r^d(k), u^d) := \frac{1}{2} H \tilde{u} + [F_1 x(k) + F_2 \tilde{y}_r^d + F_3 u^d]^T \tilde{u} + Cte \quad (3.28)$$

onde  $H$  é uma matriz quadrada denominada de Hessiana da função quadrática.  $F_1$ ,  $F_2$  e  $F_3$  são respectivamente

$$H := 2 \sum_{i=1}^N [\Psi_i^T C_r^T Q_y C_r \Psi_i + (\Pi_i^{(nu,N)})^T Q_u \Pi_i^{(nu,N)}] \quad (3.29)$$

$$F_1 := 2 \sum_{i=1}^N [\Psi_i^T C_r^T Q_y C_r \Phi_i] \quad (3.30)$$

$$F_2 := -2 \sum_{i=1}^N [\Psi_i^T C_r^T Q_y \Pi_i^{(nu,N)}] \quad (3.31)$$

$$F_3 := 2 \sum_{i=1}^N [(\Pi_i^{(nu,N)})^T Q_u] \quad (3.32)$$

### 3.5.2.3 Restrições

Uma das principais vantagens do MPC consiste no fato de que ao calcular uma sequencia ótima para os sinais de controle são levadas em consideração restrições que podem ser nos estados, na variável de controle e ainda na taxa de variação da variável de controle (DAMIRAN; FU; TAN, 2016). De fato, as restrições podem ser de diversas naturezas, por exemplo: restrições nos estados tais como temperaturas limites, velocidades, ângulos e acelerações que não podem ultrapassar certos limites. Restrições nas variáveis de comando, ou seja, atuadores possuem restrições que limitam sua faixa de operação. Desta forma, a fim de respeitar essas restrições, desigualdades matriciais devem ser consideradas nas variáveis de estados e de controle ao resolver a função custo.

As saídas reguladas de um sistema LTI, que estão sujeitas às restrições, são definidas pela matriz de restrição  $C_c \in \mathbb{R}^{(n_c, n_u)}$ , sendo  $n_c$  o número de saídas reguladas. A equação 3.33 apresenta a formulação para as saídas do sistema, as quais devem respeitar as seguintes desigualdades  $y_c^{min} \leq y_c \leq y_c^{max}$ .

$$y_c = C_c x \quad (3.33)$$

Esta desigualdade deve ser respeitada em todo horizonte de predição  $k + i$  para  $i \in 1 \dots N$ , portanto pode ser escrita da seguinte forma:

$$y_c^{min} \leq y_c(k + i) = C_c x(k + i) + D_c u(k + i - 1) \leq y_c^{max} \quad (3.34)$$

Em que fazendo as devidas manipulações pode ser representado da forma:

$$\underbrace{\begin{pmatrix} +C_c\Psi_1 + D_c\Pi_1^{(nu,N)} \\ \vdots \\ +C_c\Psi_N + D_c\Pi_N^{(nu,N)} \\ -C_c\Psi_1 - D_c\Pi_1^{(nu,N)} \\ \vdots \\ -C_c\Psi_N - D_c\Pi_N^{(nu,N)} \end{pmatrix}}_{A_{ineq}(1)} \tilde{u} \leq \underbrace{\begin{pmatrix} -C_c\Phi_1 \\ \vdots \\ -C_c\Phi_N \\ +C_c\Phi_1 \\ \vdots \\ +C_c\Phi_N \end{pmatrix}}_{G1(1)} x(k) + \underbrace{\begin{pmatrix} +y_c^{max} \\ \vdots \\ +y_c^{max} \\ -y_c^{min} \\ \vdots \\ -y_c^{min} \end{pmatrix}}_{G3(1)}$$

Outra restrição que pode ser adotada no MPC consiste na taxa de variação da variável de controle, representada por  $\delta$ . Esta restrição é um fator importante uma vez que os atuadores possuem certo tempo para alterar a magnitude do comando que imprime ao sistema. A expressão que representa essa restrição é apresentada pela seguinte desigualdade sendo  $\forall i \in 1 \dots N$ :

$$\delta^{min} \leq u(k+i) - u(k+i-1) \leq \delta^{max} \quad (3.35)$$

que pode ser representado na seguinte forma matricial:

$$\underbrace{\begin{pmatrix} +\mathbb{I} & \mathbb{O} & \mathbb{O} & \dots & \mathbb{O} & \mathbb{O} \\ -\mathbb{I} & +\mathbb{I} & \mathbb{O} & \dots & \mathbb{O} & \mathbb{O} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ +\mathbb{O} & \mathbb{O} & \mathbb{O} & \dots & -\mathbb{I} & +\mathbb{I} \\ -\mathbb{I} & \mathbb{O} & \mathbb{O} & \dots & \mathbb{O} & \mathbb{O} \\ +\mathbb{I} & -\mathbb{I} & \mathbb{O} & \dots & \mathbb{O} & \mathbb{O} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathbb{O} & \mathbb{O} & \mathbb{O} & \dots & +\mathbb{I} & -\mathbb{I} \end{pmatrix}}_{A_{ineq}(2)} \tilde{u} \leq \underbrace{\begin{pmatrix} +\mathbb{I} \\ \mathbb{O} \\ \vdots \\ \mathbb{O} \\ -\mathbb{I} \\ \mathbb{O} \\ \vdots \\ \mathbb{O} \end{pmatrix}}_{G2(2)} u(k-1) + \underbrace{\begin{pmatrix} +\delta^{max} \\ +\delta^{max} \\ \vdots \\ +\delta^{max} \\ -\delta^{min} \\ -\delta^{min} \\ \vdots \\ -\delta^{min} \end{pmatrix}}_{G3(2)}$$

A última restrição a ser levada em consideração é a magnitude do comando, uma vez que os atuadores possuem também limites máximos e mínimos de operação. A restrição da magnitude do sinal de controle é dada pela desigualdade  $\forall i \in 1 \dots N$ :

$$u^{min} \leq u(k+i-1) \leq u^{max} \quad (3.36)$$

e na forma vetorial

$$\begin{pmatrix} u^{min} \\ \vdots \\ u^{min} \end{pmatrix} \leq \tilde{u} \leq \begin{pmatrix} u^{max} \\ \vdots \\ u^{max} \end{pmatrix} \in \mathbb{R}^{(N,nu)}$$



A definição do MPC baseado no modelo de predição apresentado na equação 3.24 e na função custo 3.27, pode portanto ser acrescida das restrições descritas acima. O problema de otimização resultante é, portanto, um problema convexo desde que a matriz  $H$  seja definida positiva. Dessa forma o problema de otimização quadrático pode ser escrito da seguinte forma:

$$\begin{aligned} \tilde{u}^{opt}(k) \triangleq \operatorname{argmin} : & \left[ \frac{1}{2} \tilde{u}^T H \tilde{u} + F^T(K) \tilde{u} \right] \\ \text{sujeito a : } & A_{ineq} \tilde{u} \leq B_{ineq}(k) \\ & \tilde{u}^{min} \leq \tilde{u} \leq \tilde{u}^{max} \end{aligned} \quad (3.37)$$

onde as seguintes definições são acrescentadas:

$$F(k) = F_1 x(k) + F_2 \tilde{y}_r^d + F_3 \tilde{u} \quad (3.38)$$

$$B_{ineq} = G_1 x(k) + G_2 u(k-1) + G_3 \quad (3.39)$$

### 3.6 PARAMETRIZAÇÃO DO MPC

Em poucas palavras, a estratégia de controle MPC atua de forma a resolver um problema de otimização on line para o cálculo do sinal de comando a ser aplicado na planta do sistema (LING; MACIEJOWSKI; WU, 2008). Neste sentido o controle MPC demanda de um esforço computacional elevado e sua utilização, sem aplicação de métodos que acelerem os cálculos a serem realizados, pode inviabilizar a implementação dessa estratégia em sistemas rápidos (WANG; BOYD, 2010).

Neste sentido, diversos trabalhos apresentam estratégias que podem ser incorporadas ao MPC como forma de reduzir a carga de cálculo e em consequência reduzir o esforço computacional (TØNDEL; JOHANSEN; BEMPORAD, 2003). De acordo com Alamir (2013) a parametrização do controle é uma técnica que permite uma redução drástica no número de comandos no problema de otimização sem redução significativa da performance do controlador.

Utilizando a notação de Alamir (2013) a sequência de comandos  $\tilde{u}$ , obtida através da otimização, contém dimensão  $N \cdot n_u$  em que  $N$  corresponde ao horizonte de predição e  $n_u$  é a quantidade de atuadores (número de variáveis de comando). A sequência de comandos é dada da seguinte forma:

$$\tilde{u}(k) := \begin{pmatrix} u(k) \in \mathbb{R}^{n_u} \\ \vdots \\ u(k+N-1) \in \mathbb{R}^{n_u} \end{pmatrix}$$

Na parametrização o objetivo é a redução no número de graus de liberdade da sequência de comandos. A figura 3.7 mostra a sequência de comandos para o caso parametrizado, de forma que são definidos valores intermediários de comando ao longo da sequência. Em que  $n_r$  consiste na quantidade de termos intermediários na parametrização. Sendo assim, no caso parametrizado, o número de graus de liberdade do comando é reduzido de  $N \cdot n_u$  para  $n_r \cdot n_u$ .

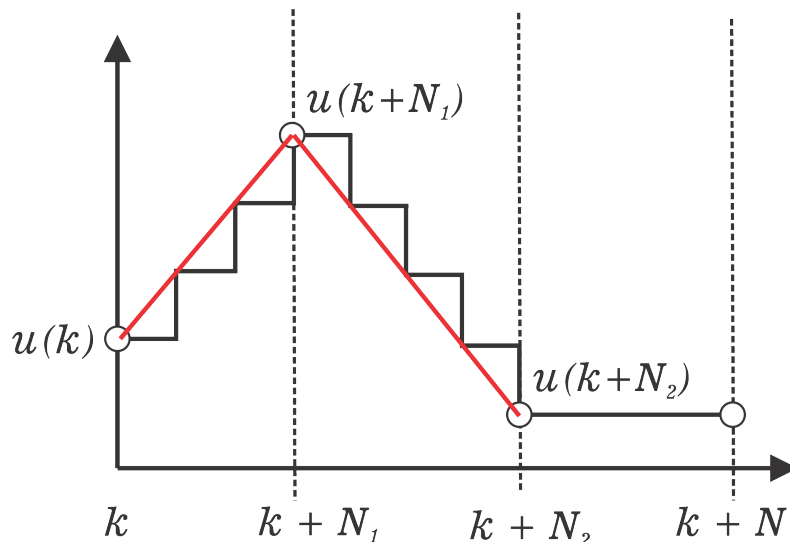


Figura 3.7: Sequência de comandos para o caso parametrizado em que  $n_r = 3$ . Adaptado de Alamir (2013).

Desta maneira, neste tipo de parametrização, os valores dos comandos nos instantes intermediários são  $u(k), u(k + N_1), \dots, u(k + N_{n_r-1})$ . E vetorialmente esta sequência assume a seguinte forma:

$$p := \begin{pmatrix} u(k) \\ \vdots \\ u(k + N_{n_r-1} - 1) \end{pmatrix}$$

Um outro método de parametrização é o exponencial. Considerando a sequência de comandos original na forma  $\tilde{u}(k)$ , utilizando a parametrização exponencial pode se obter a sequência original em termos da nova sequência de comando reduzida  $\mathbf{p}$ , através da pré-multiplicação pela matriz exponencial  $\Pi_e$  assumindo a seguinte forma:

$$\tilde{u}(k) := \begin{pmatrix} u(k) \in \mathbb{R}^{n_u} \\ \vdots \\ u(k + N - 1) \in \mathbb{R}^{n_u} \end{pmatrix} = \Pi_e \cdot p(k)$$

Alamir (2013) mostra que na parametrização exponencial o perfil de controle pode ser definido da seguinte forma:

$$u_j(k+i) := \sum_{l=1}^{n_e^{(j)}} \underbrace{\left[ e^{\frac{-\lambda(i\tau)}{(l-1)\alpha+1}} \right]}_{m_{j,l}(i)} \cdot p_l^{(j)}; \alpha > 1 \quad (3.40)$$

em que  $n_e^{(j)}$  é o número de exponenciais utilizadas para cada atuador,  $\tau$  é o período de amostragem,  $i$  é o  $i$ -ésimo comando na sequência de comandos de dimensão  $N$ ,

Esta equação pode ser reescrita na forma de produto de matrizes:

$$u_j(k+i) := [M_j(i)] \cdot p^{(j)}; p^{(j)} \in \mathbb{R}^{n_e^{(j)}} \quad (3.41)$$

Sendo que a matriz  $M_j(i) \in \mathbb{R}^{1 \text{ times } n_e^{(j)}}$  é definida por:

$$M_j(i) := (m_{j,1}(i) \cdots m_{j,n_e^{(j)}}(i)) \quad (3.42)$$

Desta forma, considerando a equação 3.41 como  $j = 1, \dots, n_u$ , leva a expressão do comando  $u(k+i)$  como:

$$\begin{aligned} u(k+i) &= \underbrace{BlockDiag \left( M_j(i)_{j=1}^{n_u} \right)}_{M(i)} \begin{pmatrix} p^{(1)} \\ \vdots \\ p^{(n_u)} \end{pmatrix} \\ &= [M(i)] \cdot p \end{aligned} \quad (3.43)$$

E finalmente para  $i = 0, \dots, N-1$ , a expressão da parametrização exponencial  $\Pi_e$  que retoma a sequência de comandos original é dada por:

$$\tilde{u} = \Pi_e \cdot p \quad (3.44)$$

## 4 METODOLOGIA

Neste capítulo será apresentada a metodologia utilizada para o desenvolvimento deste trabalho. Inicialmente serão discutidas as etapas do modelo V explicitando as principais plataformas e métodos utilizados. Em seguida serão apresentados os parâmetros dos modelos que foram abordados na simulação. Por fim, será mostrada a implementação da estratégia de controle utilizando a metodologia de Desenvolvimento Baseado em Modelos.

### 4.1 REQUISITOS DO SISTEMA

O primeiro passo do ciclo V é definição dos requisitos do sistema, esta etapa consiste na descrição detalhada das características do sistema a ser desenvolvido. Em termos de componentes e funcionalidades, os requisitos do sistema não visam descrever como os itens serão projetados, mas buscam restringir os itens em faixas de operação de forma a satisfazer os requisitos propostos pelo cliente. Por exemplo: tolerância de materiais, variação do nível de tensão de determinado componente elétrico, nível de interferência de rádio, protocolos, linguagens de programação e interfaces para componentes de software.

No projeto do EPAS, um dos primeiros requisitos a serem considerados é o aspecto físico do sistema, em outras palavras, para qual veículo a direção elétrica será projetada. Neste sentido, alguns dos pontos a serem considerados são: o máximo torque de assistência que pode ser realizado pelo motor, o máximo de torque no pinhão, a velocidade do veículo e dimensões dos componentes. Após esta etapa, diversos componentes podem ser definidos tais como: o motor elétrico, a relação de engrenagem e alguns fatores da estratégia de controle (HAIBO; YING, 2011)

No que diz respeito aos sistemas E/E, os principais requisitos para o sistema EPAS são apresentados em :

- O torque de assistência é definido com base na velocidade do veículo e no torque do condutor;
- O volante do veículo deve retornar a posição central quando o condutor deixar de aplicar o torque no mesmo;
- A estratégia de controle deve compensar os atritos mecânicos presentes nos dispositivos;
- Atenuação da perturbação proveniente dos pneus com o piso;
- Atenuação dos ruídos inerentes ao processo de medição;
- Comunicação CAN para obtenção do sinal de velocidade do veículo;

- Proporcionar análise de diagnósticos a fim de fornecer informações relativas a erros no sistema.
- Proteção quanto ao fornecimento de tensão acima dos limites físico do motor (*Over Voltage protection*).

Neste sentido, estes foram os principais requisitos funcionais abordados neste trabalho. E através das estratégias de controle buscou-se cumprir os requisitos estabelecidos e garantir o funcionamento do EPAS.

## 4.2 PROJETO DO SISTEMA

Na fase de projeto do sistema o objetivo principal é implementar a planta do sistema através da utilização de blocos funcionais. Neste trabalho, o modelo do EPAS foi extraído do trabalho de Marouf et al. (2012) e as equações são representadas no formato de espaço de estados. Neste fase, o *software* utilizado para esta finalidade foi o Matlab/Simulink<sup>®</sup>. Em nível de simulação, este *software* dispõe de diversos blocos funcionais que podem representar o modelo de um sistema seja através do espaço de estados ou função de transferência. Todavia, nem todos estes blocos suportam a geração automática de códigos o que torna um fator limitante quando o objetivo é a implementação em *hardware*. Neste trabalho, utilizou-se para a representação da planta do sistema o bloco *Matlab Function*, o qual permite tratar o modelo em espaço de estados através de uma função, utilizando os parâmetros do EPAS como constantes. A figura 4.1 mostra o diagrama de blocos utilizado para o projeto da planta do EPAS.

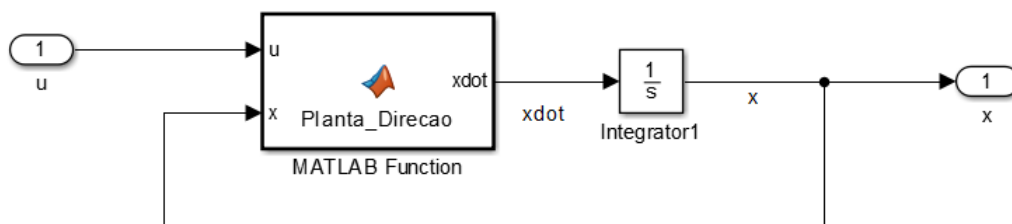


Figura 4.1: Diagrama de blocos para a planta do sistema EPAS.

## 4.3 PROJETO DOS CONTROLADORES

Nesta etapa foram projetados os controladores LQG e MPC para o sistema EPAS. O primeiro controlador projetado foi o LQG o qual é composto por um controlador LQR e um Filtro de

Kalman. Os ganhos do LQR foram calculados *off-line* utilizando as funções internas do software Matlab/Simulink<sup>®</sup>, as quais calculam os ganhos baseado nas matrizes discretas do sistema EPAS.

De forma a passar valores constantes para os estados e para o comando em regime permanente, utilizou-se a formulação apresentada na Equação 4.14. Uma vez que os valores desejados de estado e de comando estão em função da referência, é necessário que o algoritmo implementado seja executado a cada período de amostragem. Desta forma, o algoritmo deve ser acoplado ao modelo em *Simulink* e executado a cada passo. A figura 4.2 mostra a estrutura do controlador LQR projetado.

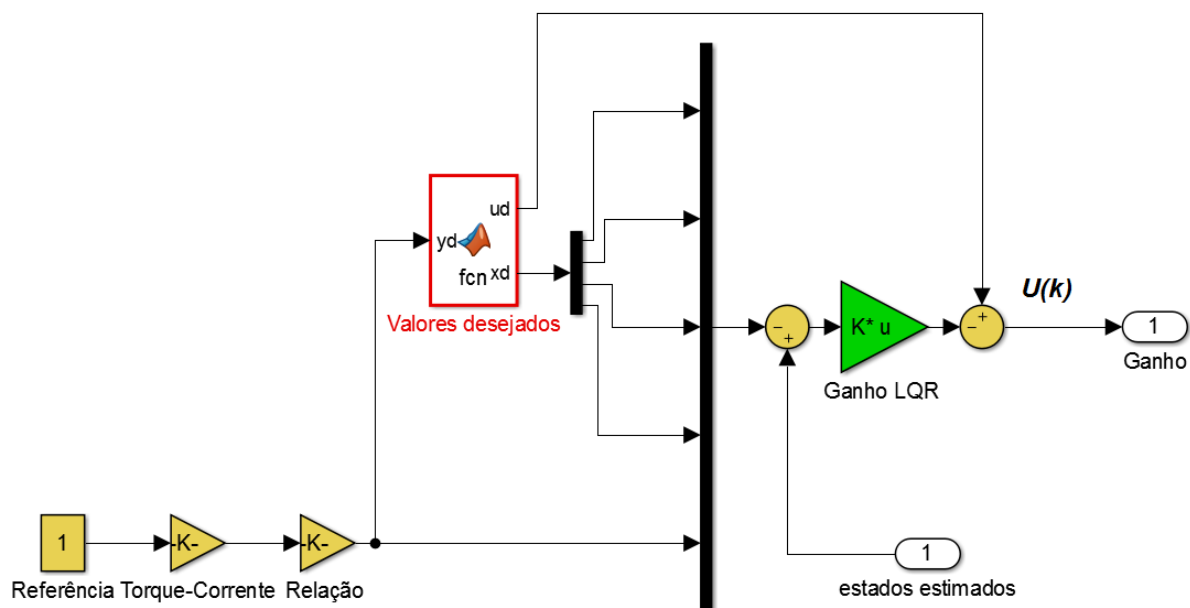


Figura 4.2: Diagrama de blocos do controlador LQR

Para estimação dos estados foi projetado um filtro de Kalman discreto. Novamente o ganho  $K_0$  do filtro de kalman foi obtido *off-line* e é utilizado como uma constante durante a simulação. A figura 4.3 mostra a estrutura para implementação do filtro de Kalman em Simulink.

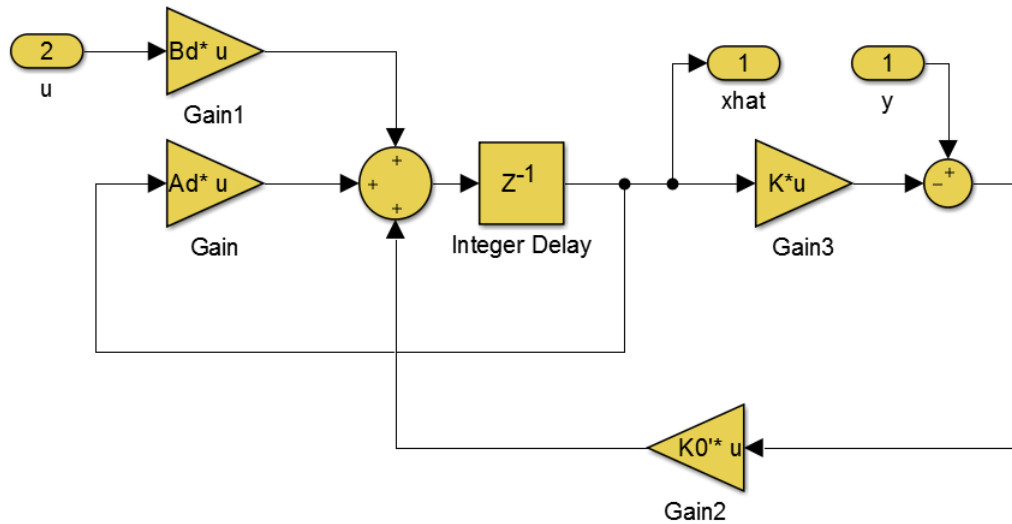


Figura 4.3: Diagrama de blocos do filtro de Kalman discreto.

Para o projeto do controlador MPC, utilizou-se a estrutura proposta por Almir (2013). As matrizes de preparação necessárias para a formulação do problema de otimização foram calculadas *off-line* e cada tempo de amostragem o algoritmo que resolve o problema de otimização quadrático sujeito a restrições é executado. A figura 4.4 mostra a estrutura em Simulink utilizada para implementação do controlador MPC.

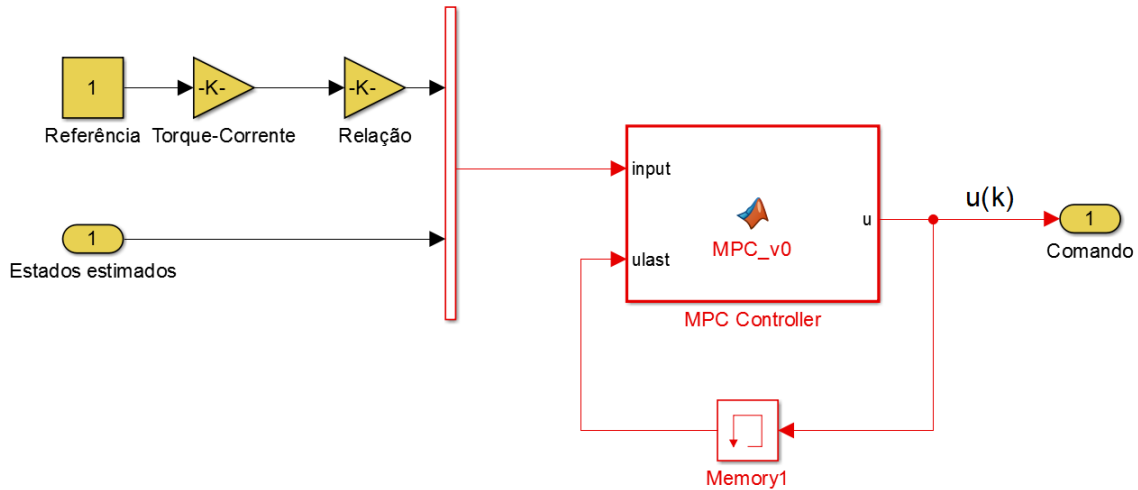


Figura 4.4: Estrutura do controlador MPC implementada em Simulink

Um dos principais desafios encontrados durante a fase de desenvolvimento do controlador foi a definição do *solver*. Neste contexto, os *solver* são algoritmos de otimização utilizados para resolver um problema quadrático sujeito a restrições. Existem, disponíveis, diversos algoritmos para estas simulações, todavia existem limitações em cada um que podem inviabilizar suas aplicações em projetos.

O primeiro *solver* utilizado nas simulações foi o *quadprog*, este é um *solver* próprio do Matlab que em nível de simulação conseguiu resolver de forma satisfatória os problemas de otimização. Todavia, este algoritmo não pode ser utilizado para geração automática de códigos, e portanto não poderiam ser implementado nos testes *Software-In-the-Loop* (SIL) e *Hardware-In-the-Loop* HIL e dessa forma não seria suficiente para aplicações neste trabalho.

A alternativa seguinte foi a utilização do *qpSOLVER*. O *qpSOLVER* é um método paramétrico para a resolução de problemas de programação quadrática (FERREAU et al., 2014). Novamente em termos de simulação o *solver* apresentou boa performance inclusive nos teste SIL em que o controlador é convertido em linguagem de programação (C ou C++). Todavia, o algoritmo do *qPOASES* é baseado em C++, e por outro lado as plataformas *dSPACE* utilizadas neste trabalho para realização do HIL possuem apenas compiladores em C. Para compilar em C++ é necessário o *Kit Integration C++* proprietário da *dSPACE*, kit este que não está disponível na plataforma utilizada, portanto a **utilização** deste *solver* não foi suficiente.

Um método baseado na expansão gradiente proposto por (ALAMIR, 2013) também foi testado como *solver*. Este algoritmo apresenta uma grande vantagem por ser uma técnica *self-contained*, ou seja, é uma rotina que pode ser completamente implementada no ambiente de simulação o que garante ao desenvolvedor mais controle sobre o processo de otimização, além disso, dispensa a dependência de bibliotecas numéricas desenvolvidas por terceiros em que se tem pouco controle sobre o algoritmo desenvolvido. Novamente, em simulação os testes MIL e SII puderam ser realizados, todavia para o problema proposta esta rotina demanda uma elevada quantidade iterações para achar os valores ótimos da otimização. Esta elevada taxa de iterações não pode ser realizada nos hardwares disponíveis, uma vez que, elevava a utilização de processamento fazendo com que a aplicação não pudesse ser realizada em tempo-real. Como a aplicação não pode ser realizada em tempo-real o próprio hardware não permite que o algoritmo seja embarcado. Dessa forma, o algoritmo proposto também não pode ser utilizado nesta aplicação.

Por fim, um algoritmo de resolução de problemas quadráticos foi implementado na versão 2015b do Matlab. O *mpcqp solver* resolve os problemas de otimização utilizando um método de conjunto ativo, o algoritmo de KWIK (SCHMID; BIEGLER, 1994). Este *solver* se apresenta como uma evolução ao *quadprog*, todavia algumas mudança nos algoritmos de preparação devem ser realizadas uma vez que a formulação e os parâmetros da função *mpcqp solver* diferem do *quadprog*. Este algoritmo além de apresentar boa performance, possui parâmetros que garantem maior controle do processo solução, como por exemplo: é possível limitar o número de iterações a serem realizadas e proporcionar o relaxamento das restrições, ou seja, definir um valor adicional no qual a restrição pode ser excedida. Além do mais, este algoritmo aceita geração automática de códigos e portanto pode ser embarcado nas plataformas *dSPACE* disponíveis.

De fato, a escolha do *solver* é um desafio no desenvolvimento do MPC, principalmente no que diz respeito a implementação em *hardware*. A tabela mostra um resumo dos *solvers* utilizados e suas principais vantagens e desvantagens para a aplicação deste trabalho.



Tabela 4.1: Comparação dos *solvers*

<i>Solver</i>	Vantagens	Desvantagens
Quadprog	Fácil implementação	Não suporta geração de códigos
qpOASES	Fácil implementação suporte	Requer compilador baseado em C++
Expansão Gradiente	Auto contido	Elevado número de iterações
mpcqsolver	Maior controle do processo Suporta geração de código	Definição dos parâmetros

#### 4.4 MODEL-IN-THE-LOOP (MIL)

Uma vez que planta do sistema EPAS e controlador foram projetados nos mesmos softwares, o primeiro teste do ciclo V&V pode ser realizado. O principal objetivo do MIL é avaliar o comportamento da planta e controlador a nível de simulação, portanto, estes diagramas de blocos são conectados e algumas informações do sistema de controle podem ser obtidas, tais como: tempo de execução do algoritmo, tempo resposta do sistema, verificação das restrições e principalmente o comportamento do sistema frente aos diversos cenários de teste. Este teste é de fundamental importância, uma vez que, permite definir e avaliar os parâmetros de sintonia dos controladores antes da implementação em *hardware*.

#### 4.5 GERAÇÃO DE CÓDIGOS E SOFTWARE-IN-THE-LOOP (SIL)

A etapa seguinte consiste em gerar o código para o controlador. A ferramenta *Simulink code generator* foi utilizada para esta finalidade. Neste sentido, o controlador passa a ser representado não por um bloco funcional, mas por uma linguagem de programação baixo nível, geralmente, C ou C++. Esse teste é importante pois permite visualizar uma série de particularidades da estratégia de controles proposta cuja estrutura não permite geração automática de códigos. No simulink é possível definir padrões nos quais o código pode ser gerado, dentre estes estão o Misra, Iso 26262. Para a realização dos testes em SIL os códigos foram gerados em linguagem C, não baseado em padrões específicos. É importante ressaltar que na geração automática de códigos a rastreabilidade e controle do sobre o código gerado é extremamente reduzida, uma vez que o código gerado é otimizado para as plataformas nas quais será embarcado. A figura 4.5 mostra a estrutura do teste SIL para o controlador LQG em que, ambos, controlador LQR e Filtro de Kalman, foram acoplados e convertidos em linguagem C por meio da geração automática de códigos.

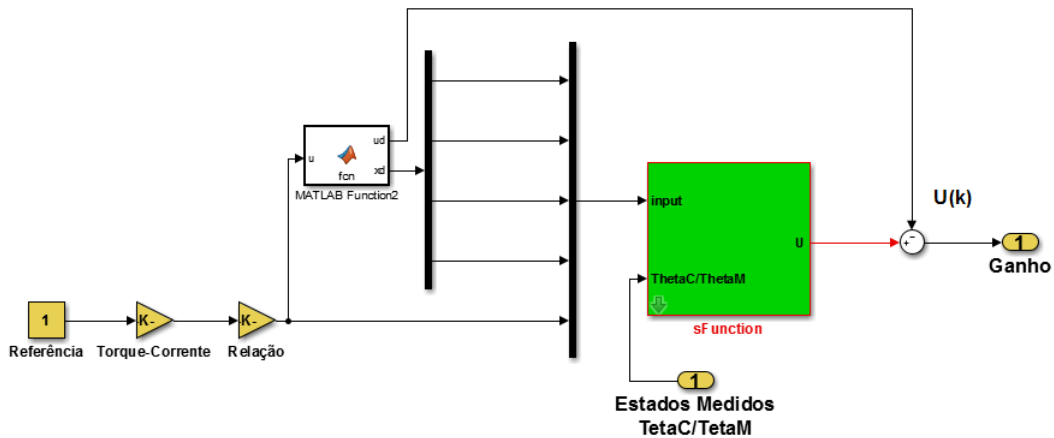


Figura 4.5: Estrutura do controlador LQR em SIL implementado no Simulink.

## 4.6 PROCESSOR-IN-THE-LOOP (PIL)

No PIL o controlador é embarcado num *hardware* e se comunica com a planta do sistema que permanece em ambiente de simulação. A comunicação entre controlador e planta, na as plataformas dSPACE é realizada geralmente por meio da *Toolbox* UDP, a qual estabelece uma comunicação *Ethernet* que permite a troca de informações entre a plataforma e o PC que roda a planta. Neste trabalho, esta etapa não foi realizada, uma vez que a *toolbox* UDP não está disponível para as plataformas utilizadas, portanto a validação foi realizada diretamente por meio do método HIL.

## 4.7 TESTE DE INTEGRAÇÃO E HARDWARE-IN-THE-LOOP (HIL)

Em seguida foram embarcados em plataformas dSPACE, combinando os softwares Matlab/Simulink com os blocos de interface I/O para os hardwares específicos dSPACE configura um ambiente de controle em tempo real. Os sinais e variáveis presentes nos modelos podem ser configurados e monitorados através do software dSPACE ControlDesk. De fato, a combinação das ferramentas Matlab/Simulink e ControlDesk formam uma poderosa ferramenta para desenvolvimento e simulação de sistemas em tempo real (CHITU et al., 2013). A figura 4.6 mostra a montagem das plataformas que caracterizam um HIL para o EPAS, em que a planta do sistema é embarcado na plataforma DS 1104 e o controlador em uma DS1401 MicroAutobox. Utilizando as ferramentas do *simulink code generation* os algoritmos, tanto do controlador quando da planta, foram gerados automaticamente a partir dos modelos funcionais implementados em *simulink*.



Figura 4.6: Plataforma HIL

O sistema HIL em tempo real utilizado neste trabalho é composto por duas plataformas da Dspace®. A primeira plataforma: MicroAutobox II® foi embarcada com o algoritmo do controlador e o do filtro de kalman enquanto que a segunda DS1104® foi embarcada com a planta do sistema EPAS, a comunicação entre as duas plataformas foi realizada através de sinal analógico. A figura 4.7 mostra o diagrama de blocos da plataforma HIL.

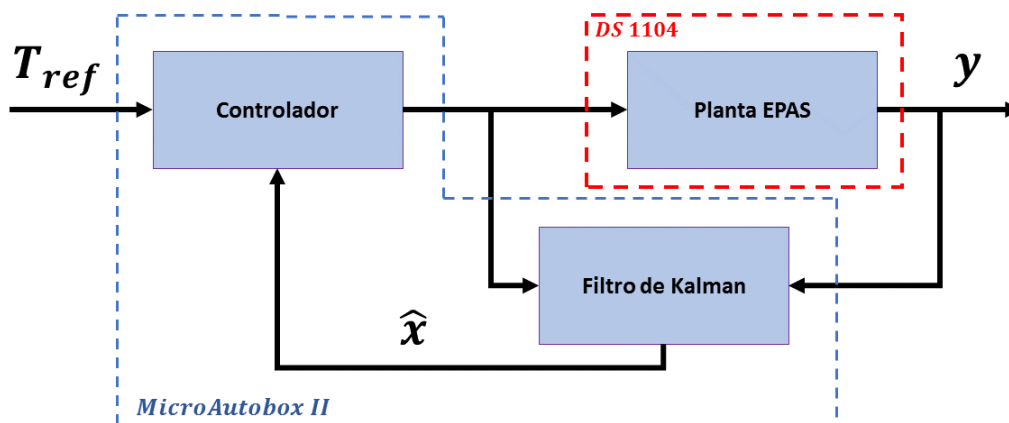


Figura 4.7: Diagrama de blocos da plataforma HIL

Abaixo são descritas os *hardwares* e *softwares* utilizados para implementação da plataforma HIL.

#### 4.7.1 MicroAutobox®

A MicroAutobox®, que pode ser vista na figura 4.8, é uma plataforma de hardware composta por um MicroAutobox Embedded PC na parte inferior, que consiste numa estação de trabalho com o Windows 7® e a MicroAutobox II (MBAX II) na parte superior que é uma controladora em tempo real desenvolvida especificamente para o segmento automotivo. A MBAX II possui uma série de entradas analógicas, digitais e também comunicação CAN.



Figura 4.8: MBAX II.

Esta plataforma quando devidamente configurada para a aplicação pode ser instalada em um veículo e então atuar como uma ECU para controle de qualquer subsistema E/E. Uma das vantagens de se ter a MBAX II e o PC na mesma plataforma é que enquanto as funções estão sendo computados em tempo real na MBAX II, aplicações adicionais tais como telemáticas, operações com câmeras e processamento de imagem podem ser realizados no PC ao mesmo tempo. A tabela 4.2 apresenta as principais especificações da plataforma MBAX II.

No ambiente Matlab/Simulink existe a *toolbox* específica para esta aplicação, onde estão os blocos referentes às interfaces analógicas, digitais, CAN e Flexray. Para o desenvolvimento de um modelo utilizando tais interfaces é necessário levar em consideração as faixas operacionais destes blocos, uma vez que possuem diferentes níveis de tensão tanto para entrada quanto para saída.

Neste trabalho foram utilizados basicamente os blocos ADC e DAC para comunicação.

#### 4.7.2 DS1104®

Para embarcar a planta do sistema EPS, utilizou-se a placa controladora DS1104 (figura 4.9)®. DS1104 é uma controladora tempo real da Dspace® que também possui uma *toolbox* em Matlab/Simulink® contendo os blocos gráficos referentes as interfaces I/O do *hardware*. A tabela 4.3 mostra as principais configurações desta plataforma.

Tabela 4.2: Configurações MBAX II

	<b>MBAX II</b>	<b>MBX Embedded PC</b>
Processador	IBM PPC 750 GL, 900 MHz	Intel Core i7 dual-core
Memória	16 MB memória Principal 6 MB exclusivo para comunicação com MBAX PC 16 MB memória flash não volátil	8 GB DDR3 Ram
Interfaces	4 canais CAN 2 RS 232 2 interfaces LIN 2 Slots FlexRay ou CAN FD	2 USB 2.0 2 USB 3.0 3 Gigabit Ethernet DVI video interface SO: Windows 7
Entradas Analógicas	16 portas de 16 bits 0 ... 5 V	
Saídas Analógicas	4 portas de 12 bits 0 ... 4.5 V	
Digital I/O	40 entradas 40 saídas Todos os canais podendo ser configurados como PWM	



Figura 4.9: DS 1104.

### 4.7.3 ControlDesk

O Dspace ControlDesk<sup>®</sup> é um software utilizado para o desenvolvimento de ECUs, uma vez que permite a realização de todas as tarefas necessárias para essa finalidade desde a modelagem até validação e testes, além da vantagem de tudo ser realizado na mesma plataforma de desenvolvimento. Dentre as diversas finalidades do ControlDesk, podem ser destacadas:

- Prototipagem Rápida (RCP)
- Simulações em Hardware-in-the-Loop (HIL)
- Medições, calibração e diagnóstico de ECUs automotivas

Tabela 4.3: Configurações DS1104

DS 1104	
Processador	MPC8240, PowerPC 603e core, 250 MHz 32 kByte cache
Memória	16 MB memória Principal DRAM 8 MB memória flash
Entradas Analógicas	4 portas de 16 bits 4 portas de 12 bits $\pm 10$ V
Saídas Analógicas	8 portas de 16 bits $\pm 10$ V
Digital I/O	20-bit digital I/O
Interface Serial	Serial UART (RS232, RS485 e RS 422)

- Acesso a barramentos (CAN, CAN FD, LIN, FLeXRay)
- Validação e criação de ECUs virtuais.

A principal funcionalidade do ControlDesk<sup>®</sup> aplicada neste trabalho é a capacidade de leitura dos sinais do modelo embarcado, ou seja, ao gerar o código do modelo para ser embarcado, é gerado também um arquivo .sdf (*system description file*) o qual contém todos os blocos presentes no modelo. Desta forma, ao carregar este arquivo no ControlDesk<sup>®</sup> é possível ter acesso a todos estes blocos e, portanto, calibrá-los e visualizar a magnitude dos sinais que passam por eles.

Esta funcionalidade foi de extrema importância em diversas fases do trabalho. Inicialmente, para assegurar que os sinais que estão sendo enviados por um hardware está chegando corretamente na outra plataforma é necessário fazer a calibração dos blocos no Simulink<sup>®</sup>, uma vez que, estes blocos gráficos que representam as interfaces possuem faixas operacionais de entradas e saídas diferentes. A figura 4.10 mostra um exemplo de interface de comunicação entre as plataformas MBAXII<sup>®</sup> e DS1104<sup>®</sup> em que pode ser observado as faixas operacionais de entrada e saída para cada um dos blocos.

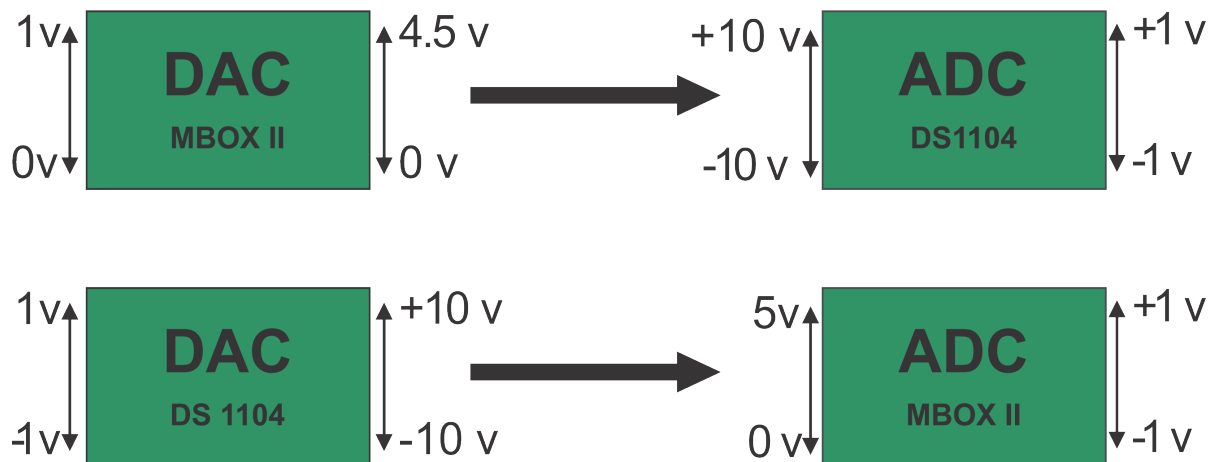


Figura 4.10: Interface de comunicação entre as plataformas MBAXII<sup>®</sup> e DS1104<sup>®</sup>.

Para garantir que o sinal enviado por uma plataforma chegue corretamente na outra é necessário fazer a compensação utilizando blocos de ganhos e somatórios afim de calibrar as diferentes faixas operacionais dos blocos de cada interface. Um ponto importante a ser considerado para calibração dos blocos é analisar a características do sinal a ser transmitido, ou seja, é necessário adequar a magnitude do sinal aos valores admitidos pelos blocos e em seguida compensar esta alteração para retornar o sinal ao valor original.

## 5 RESULTADOS E DISCUSSÃO

Neste capítulo serão apresentados os resultados das simulações dos controladores LQG e MPC. A primeira seção apresenta os resultados das simulações executadas para o LQG em *Model-In-the-Loop* (MIL), ou seja, controlador e planta estão na forma de blocos funcionais em um ambiente virtual. Os resultados das simulações em *Software-In-the-Loop* (SIL) são mostrados em paralelo, uma vez que, as respostas são semelhantes. Todavia, para o SIL o controlador é convertido em código C através da geração automática de códigos. No tópico seguinte os resultados obtidos em *Hardware-In-the-Loop* (HIL) são mostrados.

Na segunda seção são apresentados os resultados para o MPC seguindo a mesma ordem dos testes MIL, SIL e HIL. Os resultados do MPC quanto às restrições também serão abordados nesta seção. Os parâmetros da planta do EPAS utilizado nas simulações são mostrados na Tabela 5.1.

Tabela 5.1: Parâmetros do EPAS utilizados nas simulações

Nomenclatura	Descrição	Valor [Unidades]
$J_c$	Momento de inércia da coluna de direção	$0.04Kg.m^2$
$B_c$	Amortecimento viscoso da coluna de direção	$0.072N.m./(rad/s)$
$K_c$	Rigidez da coluna de direção	$115N.m./rad$
$F_c$	Atrito da coluna de direção	$0.027 N.m$
$M_c$	Massa da cremalheira	$32Kg$
$B_r$	Amortecimento viscoso da cremalheira	$3820N/(m/s)$
$R_p$	Raio do pinhão	$0.007m$
$K_r$	Taxa de amortecimento do pneu	$43000N.m/m$
$J_m$	Momento de inércia do motor	$0.0004Kg.m^2$
$B_m$	Amortecimento viscoso eixo do motor	$0.0032N.m./(rad/s)$
$F_m$	Atrito do motor	$0.056N.m$
$K_t$	Constante Torque-Corrente	$0.05N.m/A$
$L_m$	Indutância do motor	$0.0056H$
$R_m$	Resistência do motor	$0.37\Omega$
$N$	Relação engrenagem	$13.65$

### 5.1 RESULTADOS LQG

Nesta seção são apresentados os resultados para o controlador LQG nos métodos de testes MIL, SIL e HIL. A amplitude dos ruídos brancos adicionados aos sinais de processo e de medida foram de 10% da magnitude do sinal com tempo de amostragem de 1 milissegundo para ambos os casos. A covariância dos ruídos foram utilizados como os parâmetros de sintonia do Filtro de Kalman. Uma vez que a perturbação proveniente do atrito pneu-pista não pode ser medida diretamente, utilizou-se uma onda quadra com amplitude de 1 N.m como forma de emular a perturbação. O torque de referência utilizado para estas simulações foi obtido através da rede



CAN (*Controler Area Network*) de um veículo real, em que foram executados testes de campo e através da interface *Onboard Diagnostics* (OBD) do veículo obteve-se a medição do sensor de torque localizado na coluna de direção.

### 5.1.1 MIL e SIL para o controlador LQG

As primeiras simulações foram realizadas para os métodos MIL e SIL. Uma vez que para ambos os casos as simulações são realizadas em ambiente virtual, é esperado respostas semelhantes para os dois métodos. As Figuras 5.1 e 5.2 apresentam a resposta do torque de assistência para a referência fornecida.

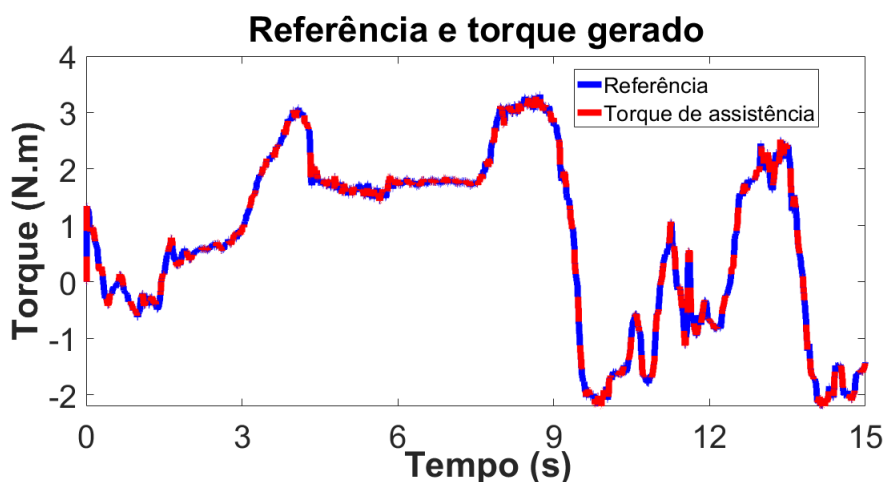


Figura 5.1: Referência e torque de assistência geral para o teste MIL

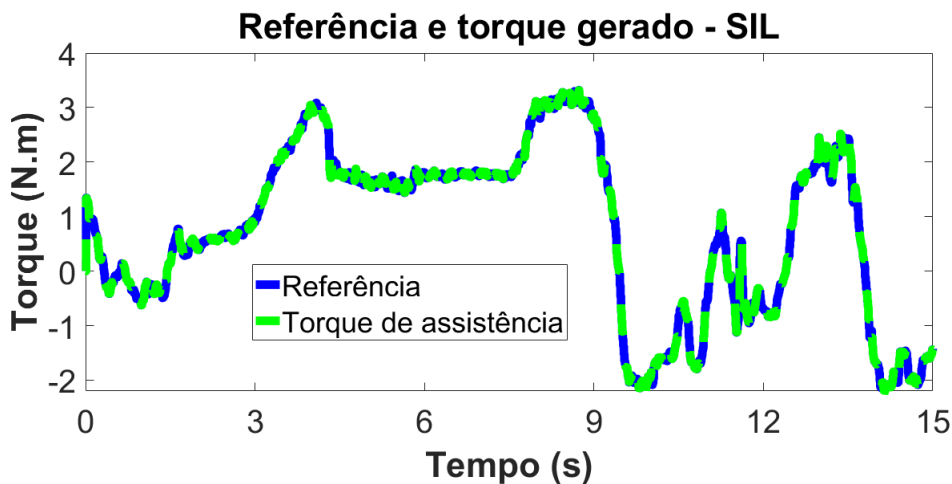


Figura 5.2: Referência e torque de assistência geral para o teste SIL

Observa-se que o torque de assistência segue a referência passada tanto para os testes MIL quanto para o SIL, apresentando boa estabilidade e performance do controlador. Os resultados obtidos com o SIL mostram que o controlador projetado pode ser convertido em códigos, uma vez

que, não houveram erros durante o processo de geração de código. A similaridade nos resultados mostra que o código gerado em linguagem C representa de forma satisfatória o modelo projetado em diagrama de blocos. As figuras 5.3, 5.4, 5.5 e 5.6 mostram a performance do filtro de Kalman para estimação dos resultados nos testes MIL e SIL.

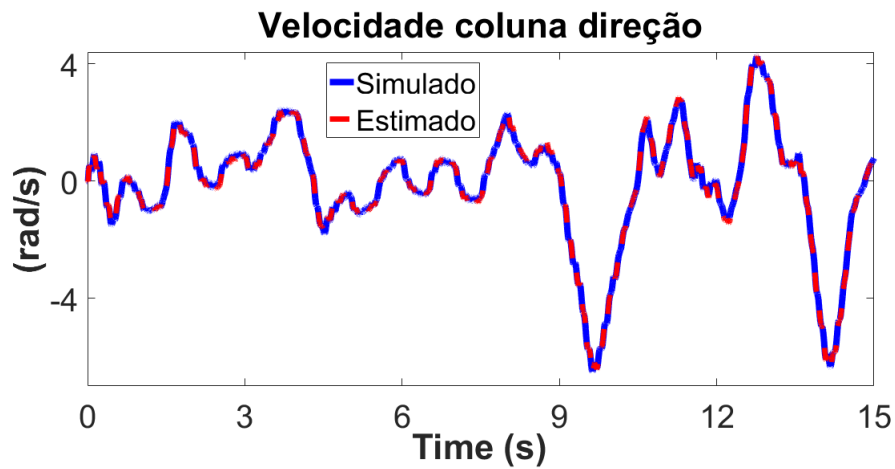


Figura 5.3: Estimação velocidade da coluna em MIL

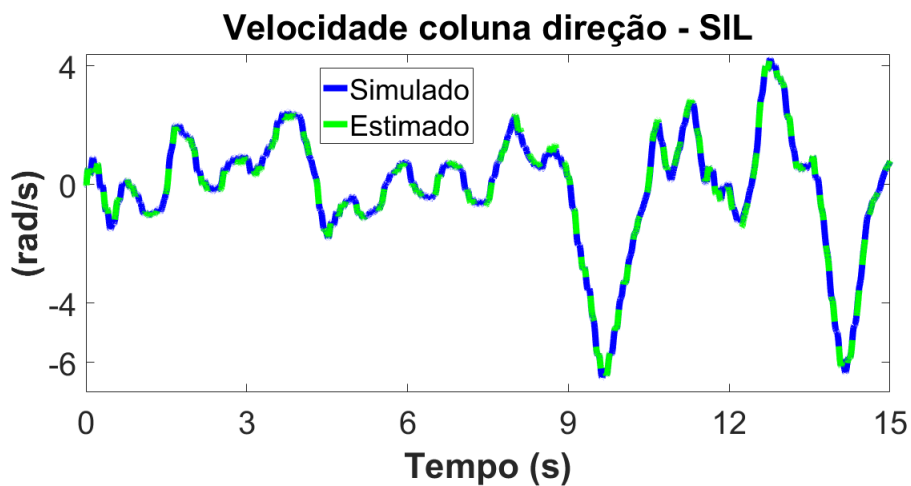


Figura 5.4: Estimação velocidade da coluna em SIL

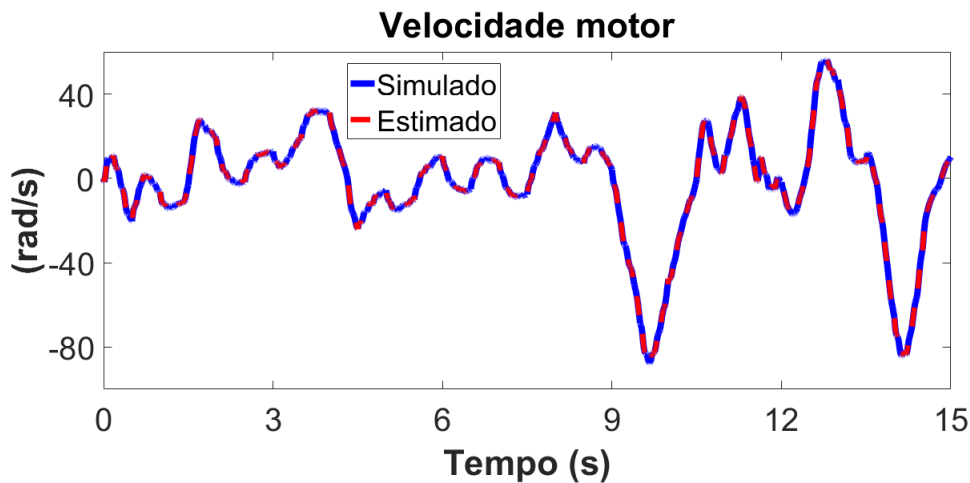


Figura 5.5: Estimação velocidade do motor em MIL

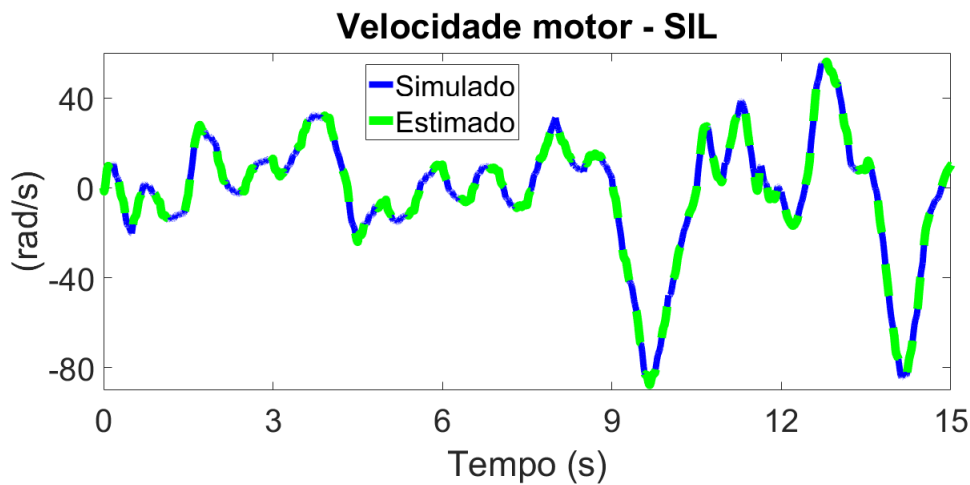


Figura 5.6: Estimação velocidade do motor em SIL

A estimação por filtro de Kalman apresentou bons resultados, de forma que os valores estimados corresponderam de forma satisfatória aos valores medidos diretamente. Portanto, a estratégia de controle apresentada como uma composição do controlador LQR e o filtro de Kalman apresentou boa performance e estabilidade para a trajetória definida.

Outra característica da estratégia de controle a ser avaliada é a robustez do controlador quanto à variação dos parâmetros da planta. Esta característica de robustez é importante uma vez que em situações práticas os parâmetros de um modelo podem sofrer variações ao longo do tempo, ou seja, componentes e peças do sistema podem ser substituídos por outros com parâmetros diferentes e até mesmo variarem as características originais devido ao tempo de utilização.

Desta forma, a fim de avaliar a característica de robustez da estratégia de controle frente à variação dos parâmetros, realizou-se uma simulação com a variação dos parâmetros ( $J_c$ ,  $B_c$ ,  $J_{eq}$ ,  $B_{eq}$ ,  $K_t$ ,  $R_m$ ,  $L_m$ ) em 10% dos valores originais que foram apresentados na tabela 5.1. As figuras 5.7 e 5.8 apresentam os resultados das simulações para o segmento da referência e estimação do filtro de Kalman

após a variação dos parâmetros.

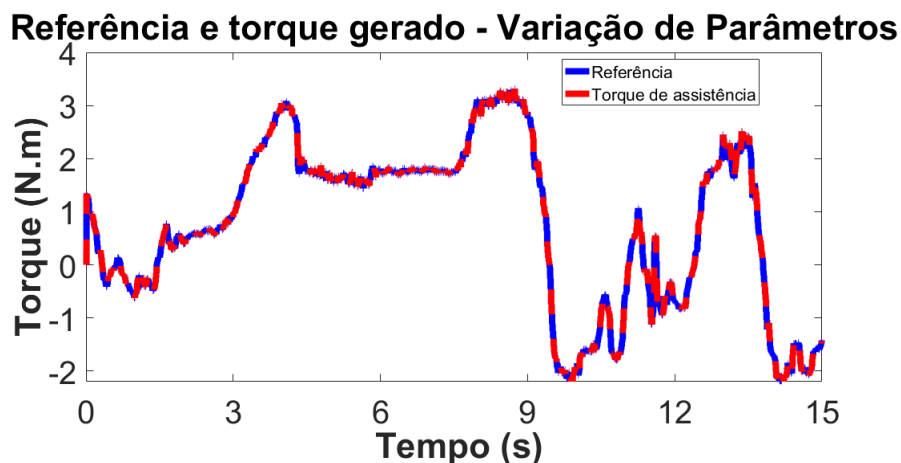


Figura 5.7: Referência e torque de assistência gerado para o teste MIL após variação dos parâmetros.

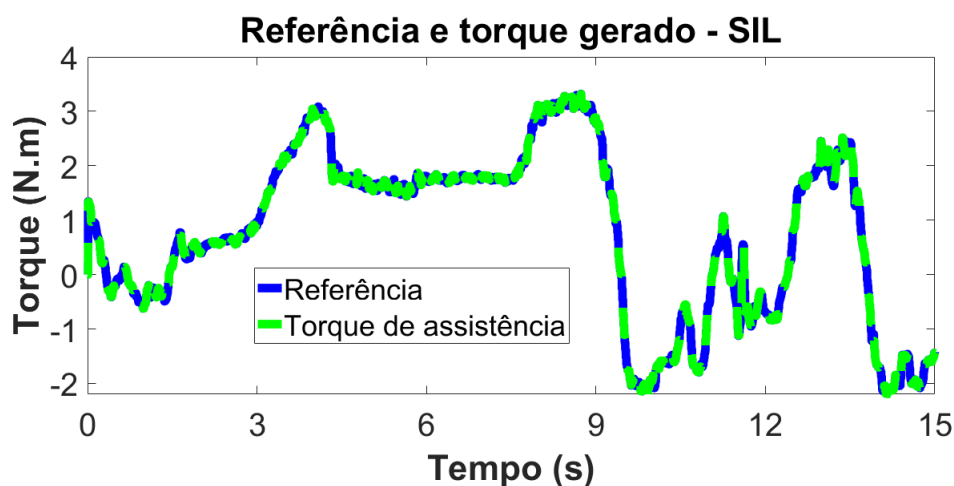


Figura 5.8: Referência e torque de assistência geral para o teste SIL após variação dos parâmetros.

Observa-se que mesmo sujeito à variação de parâmetros, o controlador apresentou boa performance quanto ao rastreamento da referência e estabilidade, conferindo uma característica de robustez à estratégia proposta. As figuras 5.9, 5.10, 5.11 e 5.12 mostram os resultados da estimação do Filtro de Kalman após a variação dos parâmetros.

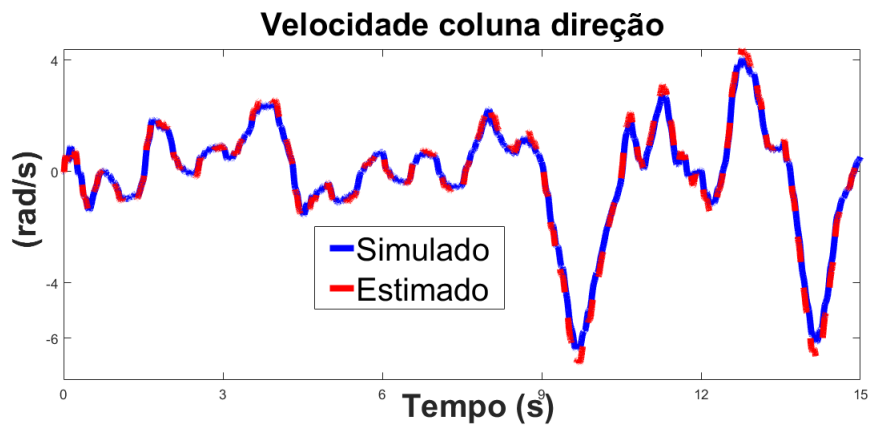


Figura 5.9: Estimação velocidade da coluna após variação de parâmetros no teste MIL.

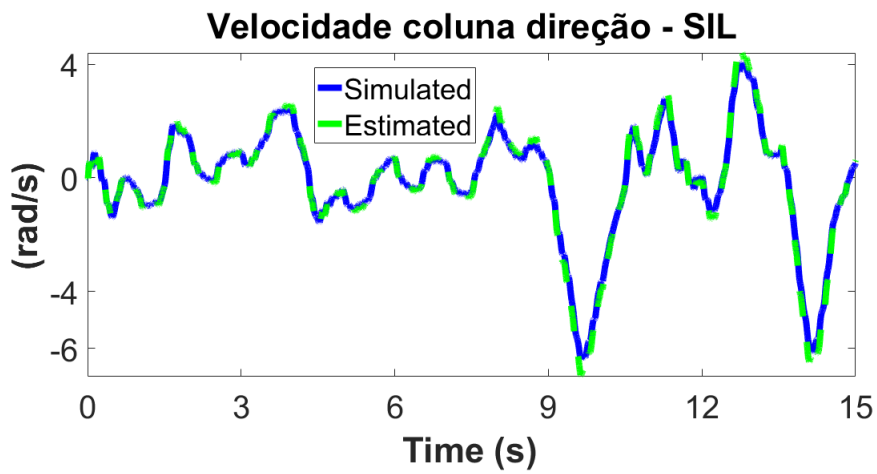


Figura 5.10: Estimação velocidade da coluna após variação de parâmetros no teste SIL.

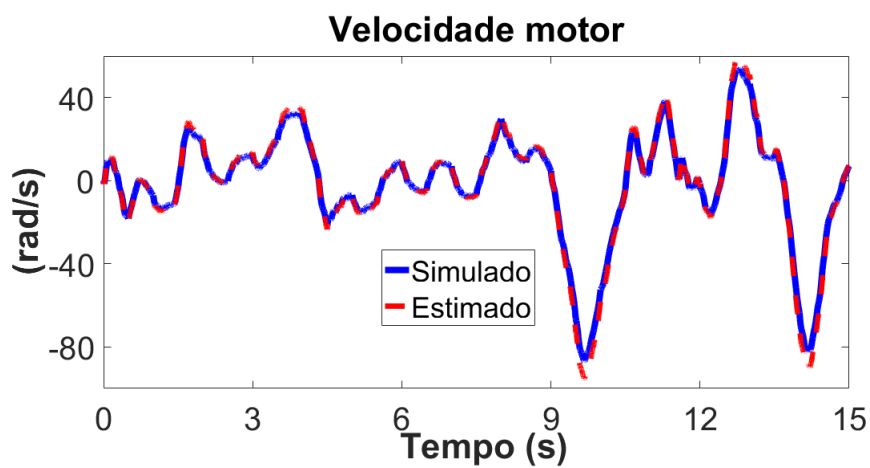


Figura 5.11: Estimação velocidade do motor após variação de parâmetros no teste MIL.

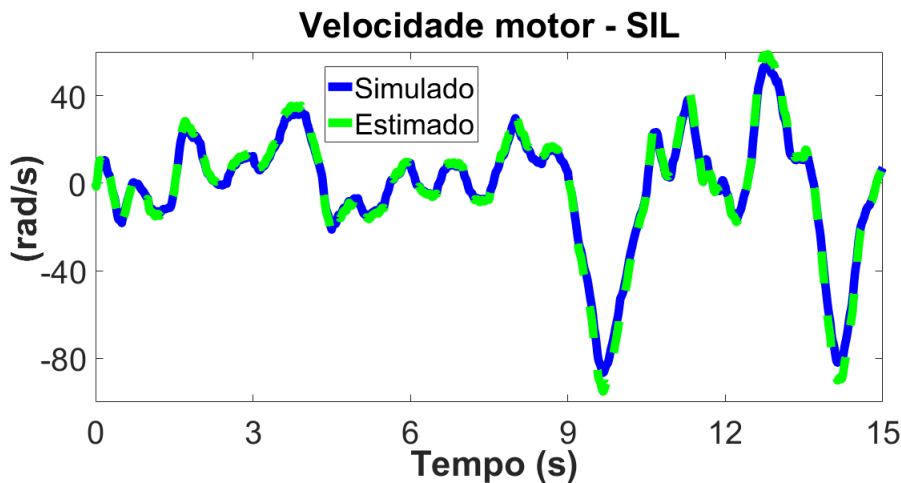


Figura 5.12: Estimação velocidade do motor após variação de parâmetros no teste SIL.

Como pode ser observado, a estratégia de controle proposta apresenta resultados satisfatórios, no tocante ao rastreamento da referência, mesmo frente a variação de parâmetros. Portanto, após os testes em ambiente de simulação mostrarem que o controlador atendeu aos critérios desejados no que diz respeito ao segmento de trajetória, estimação e robustez devido a variação dos parâmetros, o próximo teste de V&V pode ser realizado. Na seção seguinte, os resultados dos testes realizados em HIL são apresentados.

### 5.1.2 HIL para o controlador LQG

Nesta etapa, através da geração automática de códigos, os modelos do controlador e planta foram embarcados nas plataformas dSPACE®. Os parâmetros de sintonia do controlador bem como os parâmetros da planta utilizados nas simulações anteriores foram aplicados para o HIL. Todavia, a plataforma HIL desenvolvida, apresenta ruído elevado devido, principalmente, as conversões ADC e DAC. Neste sentido, a covariância dos ruídos do processo e de medida utilizados em HIL não foram os mesmos utilizados em simulação, como o ruído ocasionado pela plataforma não pode ser mensurado, para os testes em HIL os valores das covariâncias foram definidos por tentativa e erro de forma a reduzir o máximo possível erro entre os sinais medidos e estimados.

Para os testes em HIL, o modelo do controlador foi embarcado na plataforma MicroAutobox II, através da geração automática de código, com período de amostragem de 1 milissegundo. De fato, a definição do tempo de amostragem foi definida com o objetivo de garantir a performance do controlador quanto ao rastreamento da referência. Verificou-se que o tempo de amostragem definido foi suficiente para realizar as medições, calcular os valores de entrada e estados desejados em regime permanente e realizar o controle. No que tange à planta do EPAS, o modelo foi embarcado na plataforma DS1104 com o menor tempo de amostragem possível a fim de manter o modelo o mais próximo possível do contínuo. Desta forma, para a planta o tempo de amostragem definido foi de 100 microsegundos. A figura 5.13 mostra a resposta do sistema para o torque de assistência.

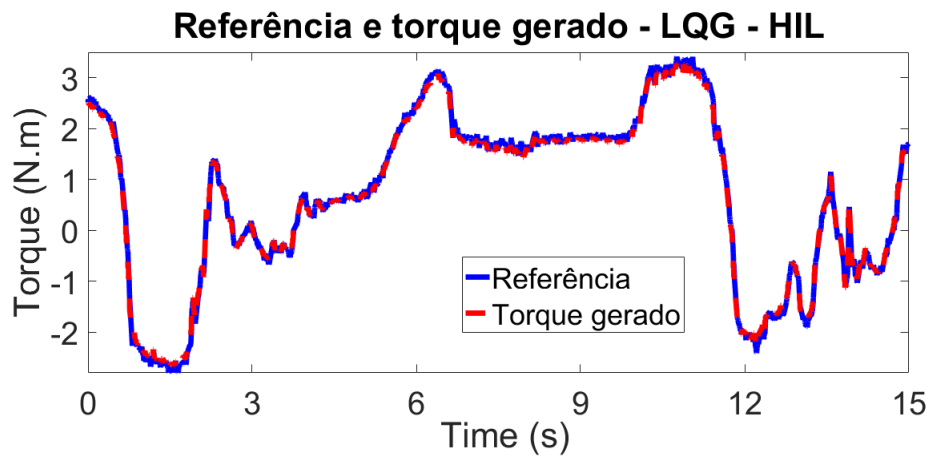


Figura 5.13: Referência e torque de assistência para o teste em HIL

Em HIL a resposta do torque de assistência manteve o mesmo desempenho dos testes em simulação. De fato, os modelos de controlador e planta embarcados na plataforma de teste através da geração automática de códigos apresentou resultados semelhantes às simulações apesar da presença dos ruídos provenientes da comunicação analógica. O desempenho do filtro de Kalman também foram analisados em HIL. As figuras 5.14 e 5.15 mostram os resultados da estimação do filtro de Kalman para a simulação em HIL.

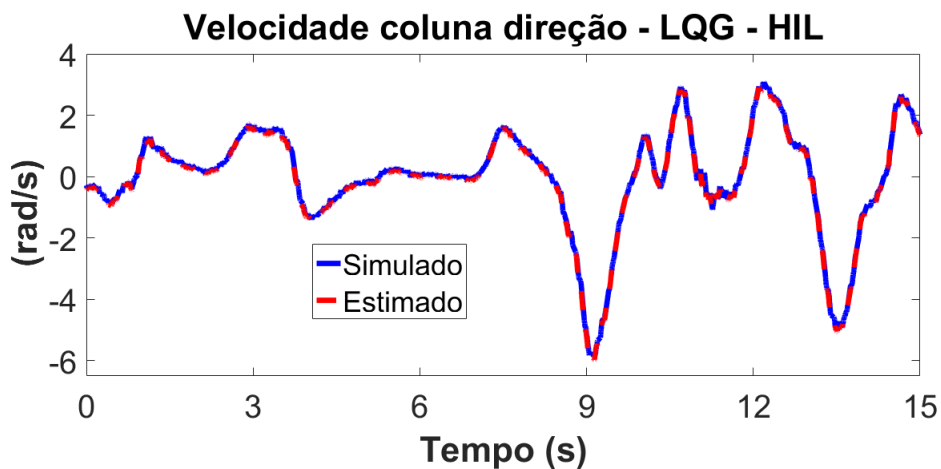


Figura 5.14: Performance do estimador de estados para a velocidade da coluna no teste HIL.

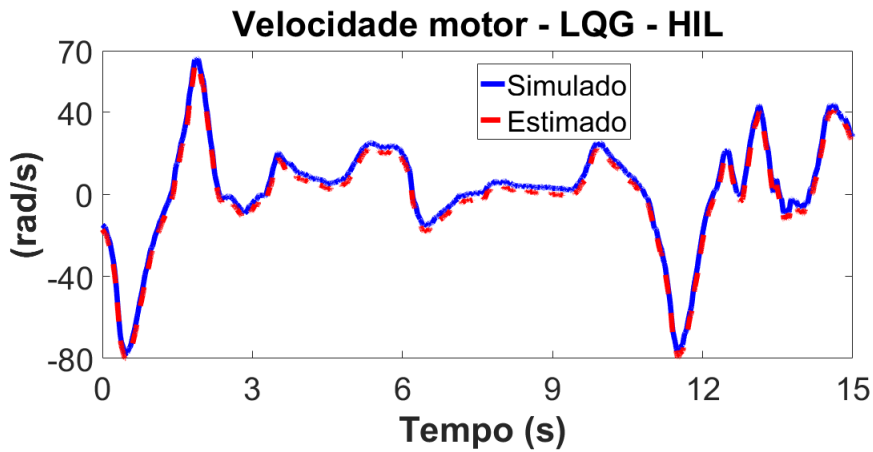


Figura 5.15: Performance do estimador de estados para a velocidade do motor no teste HIL.

Em HIL, apesar de realizar uma boa estimação, o estimador de estados apresentou performance inferior aos resultados em simulação, sendo essa divergência ocasionada, principalmente, pelo ruído que é adicionado ao sistema devido à comunicação analógica. Todavia, mesmo com este desempenho do filtro de Kalman a resposta do sistema quanto ao segmento da trajetória manteve-se satisfatória, demonstrando mais uma característica de robustez do controlador frente a variação no processo de estimação. Para melhor visualização da performance do estimador nos testes em HIL, as figuras 5.16 e 5.17 mostram o erro de estimação entre o valor medido (simulado) e o valor estimado.

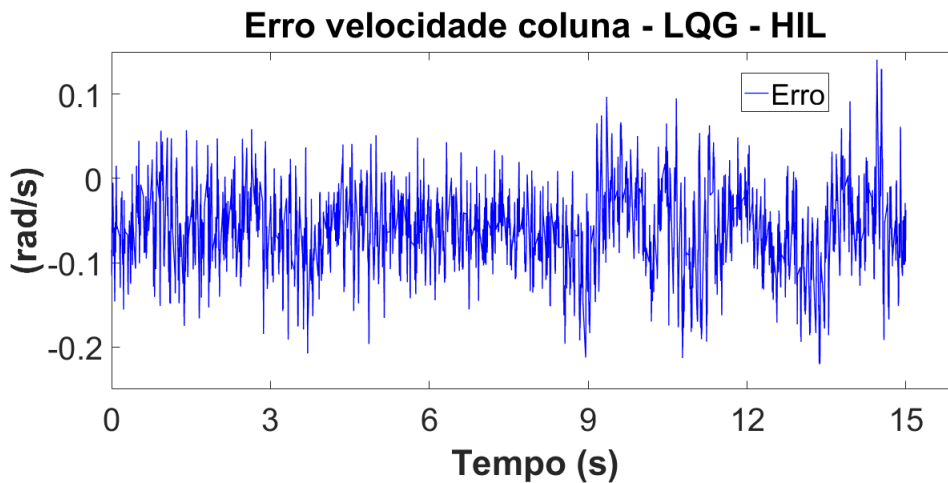


Figura 5.16: Erro entre o valor medido e o valor estimado para a velocidade da coluna.



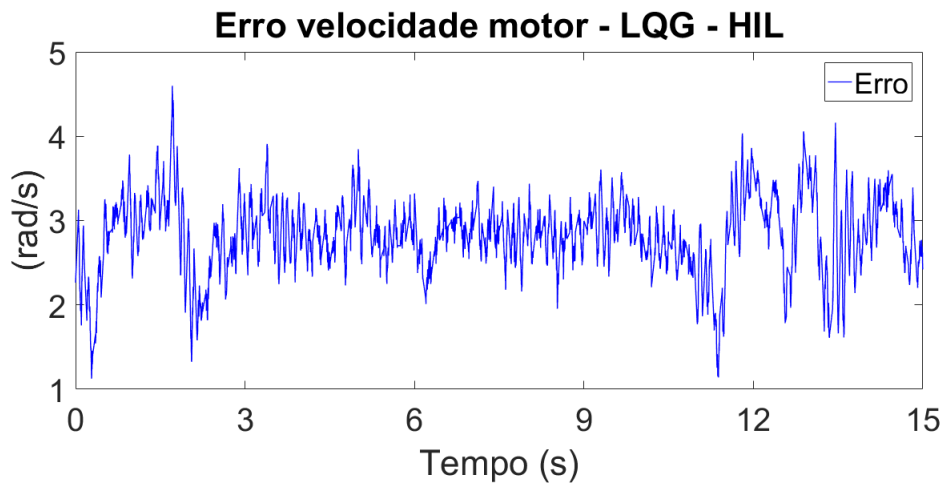


Figura 5.17: Erro entre o valor medido e o valor estimado para a velocidade do motor.

De fato, a estratégia de controle LQG apresentou bons resultados no que diz respeito ao seguimento da trajetória de referência e estimação dos estados. A seção seguinte apresenta os resultados da estratégia de controle MPC aplicada ao EPAS. Inicialmente são mostrados os resultados para as simulações em MIL e SIL, em seguida os resultados em HIL são discutidos.

### 5.1.3 MIL e SIL para o controlador MPC

Para as simulações do MPC, no que diz respeito a planta, utilizou-se os mesmos parâmetros de simulação do modelo EPAS apresentados na tabela 5.1. Os parâmetros de sintonia de controlador  $Q_y$  e  $Q_u$  utilizados para penalização dos estados e comando respectivamente, foram definidos por tentativa e erro a fim de obter a resposta adequada da estratégia de controle no que diz respeito ao rastreamento do torque de assistência frente a referência. Desta forma, os valores foram definidos como:  $Q_y = [10^5]$  e  $Q_u = [0.01]$ . No que se refere ao horizonte de predição, definiu-se horizonte de  $N = 5$ . As figuras 5.18 e 5.19 mostram os resultados das simulações para o controlador MPC nos testes MIL e SIL. A referência fornecida é a mesma para o caso do controlador LQG.

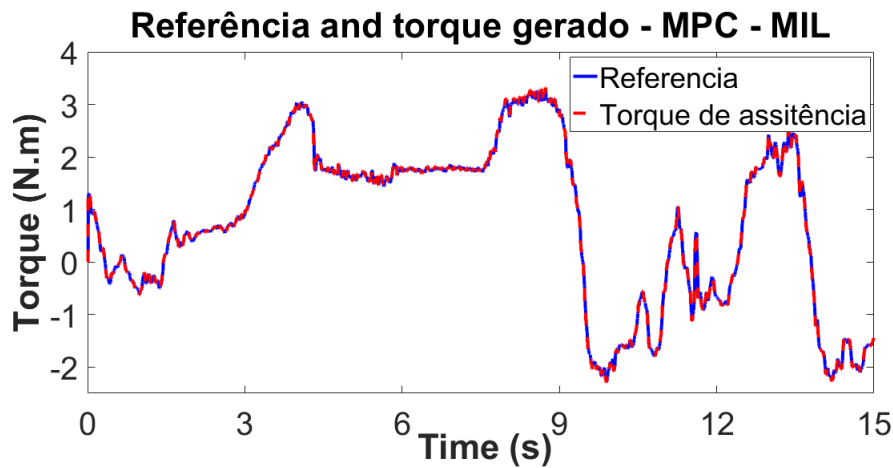


Figura 5.18: Referência e torque de assistência gerado para o teste MIL.

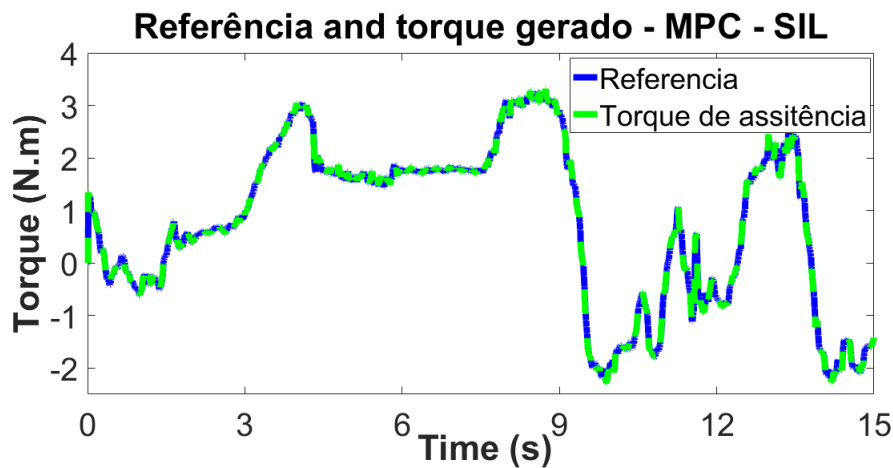


Figura 5.19: Referência e torque de assistência gerado para o teste SIL.

O controlador apresentou boa performance quanto ao rastreamento da referência. Para a simulação em SIL, os resultados mostram que o código foi corretamente gerado na linguagem C, uma vez que, as respostas são semelhantes ao teste em MIL. A seguir são mostrados os resultados do estimador de estados, os ruídos de processo e de medidas adicionados ao sistema correspondem a 10% da magnitude do sinal. O parâmetro de sintonia do filtro de Kalman foram as covariâncias dos ruídos do processo e de medidas. As figuras 5.20, 5.21 e 5.23 mostram os resultados da estimação nos testes MIL e SIL.

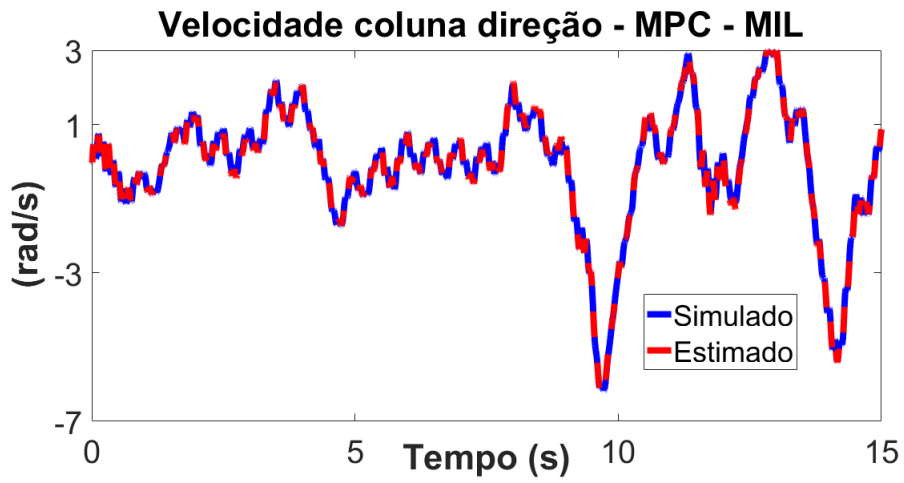


Figura 5.20: Estimação da velocidade da coluna em MIL.

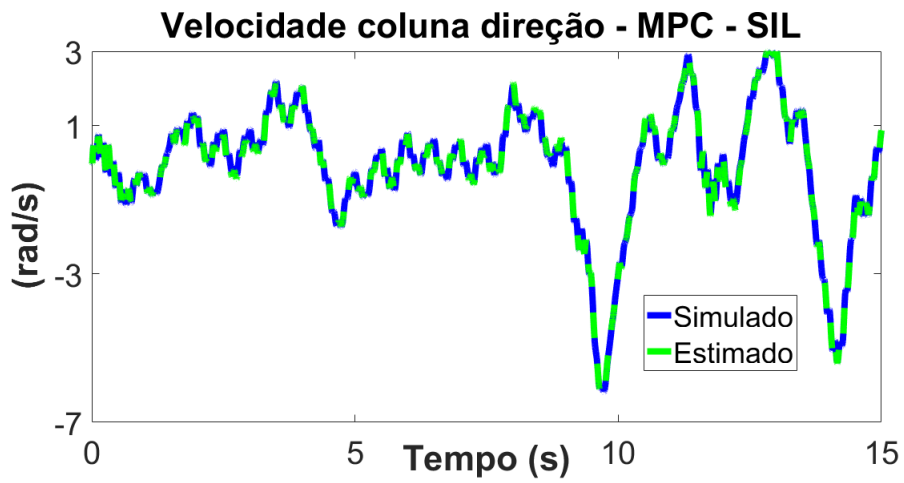


Figura 5.21: Estimação velocidade da coluna em SIL.

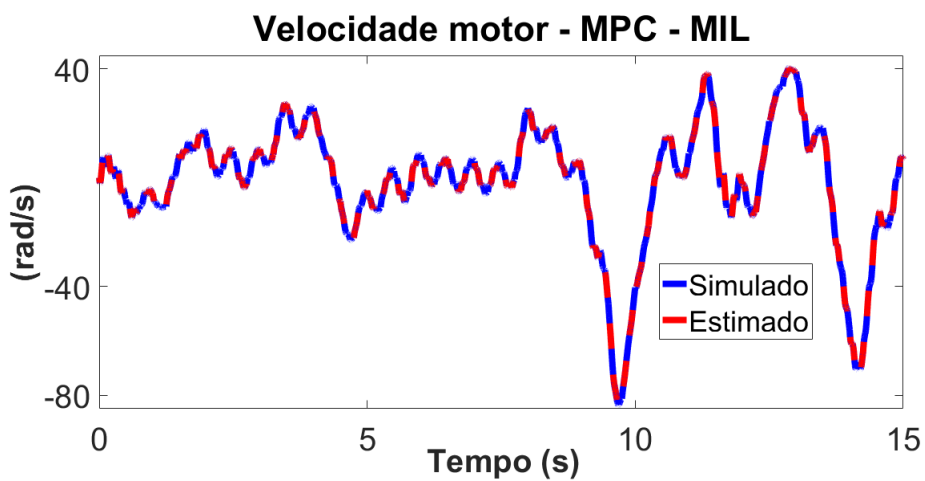


Figura 5.22: Estimação velocidade do motor em MIL.

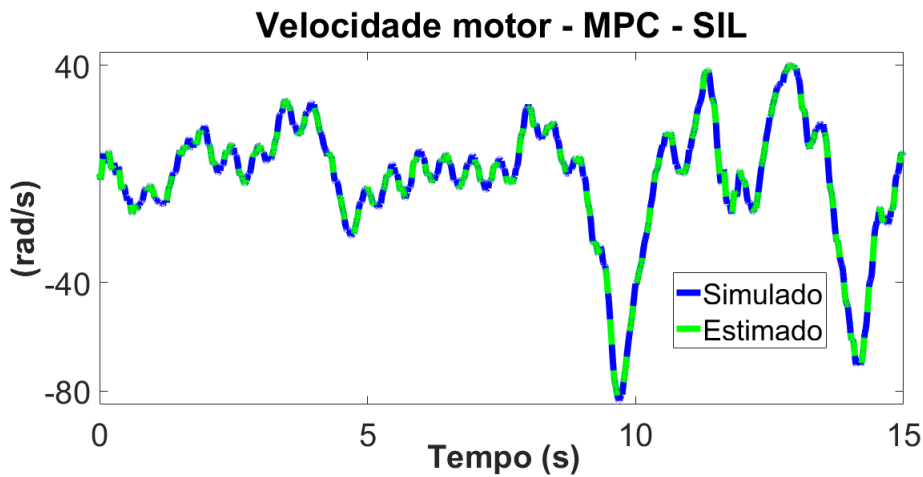


Figura 5.23: Estimação velocidade do motor em SIL.

Observa-se boa performance do estimador, uma vez que os valores estimados são bem próximos aos valores medidos diretamente. Neste sentido, a nível de simulação o controlador MPC apresentou performance e estabilidade quanto ao segmento da trajetória de referência. Os testes seguintes foram realizados em HIL, em que os modelos funcionais foram embarcados nas plataformas dSPACE®.

#### 5.1.4 Resultados das simulações em HIL para o controlador MPC

Para a simulação em HIL, a definição dos parâmetros de sintonia se tornam mais críticos, uma vez que há limitações de hardware, em outras palavras, a fim de garantir que a plataforma opere em tempo real a definição de parâmetros tais como tempo de amostragem, horizonte de predição, matrizes de ponderação da função custo e os parâmetros de sintonia do filtro de Kalman se tornam fatores críticos que devem ser definidos de forma que as plataformas operem em tempo real e garantam os objetivos de controle.

A definição do horizonte de predição e do período de amostragem deve ser definida de forma que atenda aos requisitos desejados para o desempenho do controlador, além de serem fundamentais para a implementação em *hardware*, haja visto, devido as limitações de processamento é necessário definir os parâmetros de forma a garantir que a plataforma execute suas tarefas em tempo real. A tabela 5.2 mostra as simulações que foram realizadas com diferentes tempos de amostragem  $[\tau]$  e horizonte de predição  $[N]$ .

Tabela 5.2: Simulações com diferentes tempos de amostragem e horizonte de predição.

$\tau$	N	Resultado
1ms	10	Não pode executar em tempo real
1ms	20	Não pode executar em tempo real
5ms	10	Não pode executar em tempo real
10ms	10	Executa em tempo real, mas não segue a referência
10ms	20	Executa em tempo real, mas não segue a referência

A demanda computacional exigida pelo MPC faz com que sua implementação em HIL seja limitada a capacidade do sistema de executar o código em tempo real. De fato, conforme mostra os resultados da tabela 5.2, para tempos de amostragem abaixo de 10 milissegundos não foi possível realizar as simulações em tempo real, mesmo variando o horizonte de predição. Por outro lado, aumentar o tempo de amostragem, apesar de permitir que o sistema seja executado em tempo real perde-se desempenho no controle do sistema, não sendo possível rastrear a referência. Neste sentido a utilização da parametrização exponencial foi de fundamental importância, uma vez que reduziu significativamente a demanda computacional do MPC e assim permitiu utilizar tempos de amostragem abaixo de 5 milissegundos, o que garante boa performance do controlador para o segmento da referência e respeito as restrições. Abaixo são apresentados os resultados das simulação em HIL adicionando também os resultados da imposição de restrições aos sinais do estado, comando e variação do comando, estes também para a simulação HIL.

Nestas simulações o tempo de amostragem definido foi de 5 milissegundos com horizonte de predição de 50. Inicialmente, essa configuração não era suportada pelas plataformas, porém após a implementação da parametrização a configuração acima descrita pode ser embarcada e as funcionalidades do MPC tanto de controle quanto na aplicação das restrições puderam ser utilizadas. A figura 5.24 mostra o resultado da simulação para o torque referência.

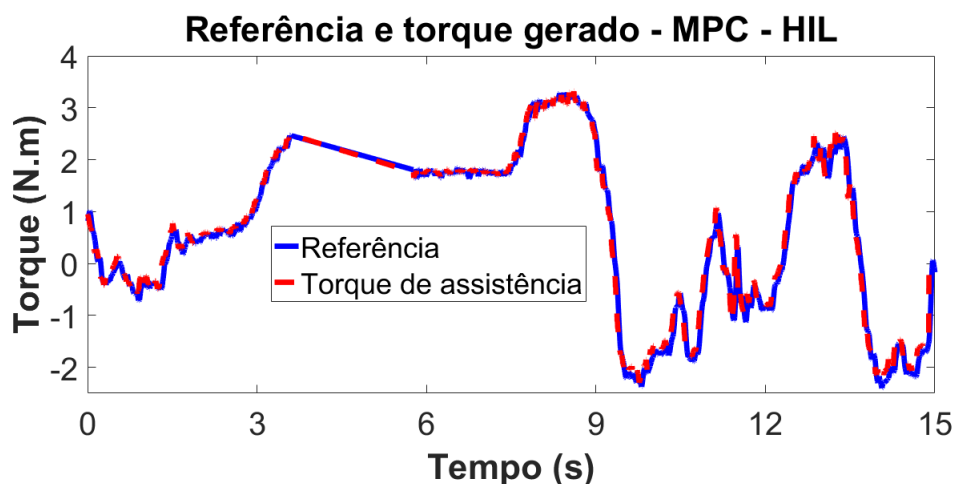


Figura 5.24: Referência e torque de assistência gerado para o teste em HIL.

Nota-se que o controlador MPC garantiu estabilidade no segmento da trajetória de referência. O filtro de Kalman também foi implementado para estimação dos estados, assim como no caso do LQG em HIL, devido aos ruídos nos parâmetros adicionados ao sistema devido a comunicação analógica, a definição dos parâmetros de sintonia do estimador não foi a mesma dos testes em MIL e SIL uma vez que a característica do ruído não pode ser conhecida. As figuras 5.25 e 5.26 mostram os resultados das estimações em HIL.

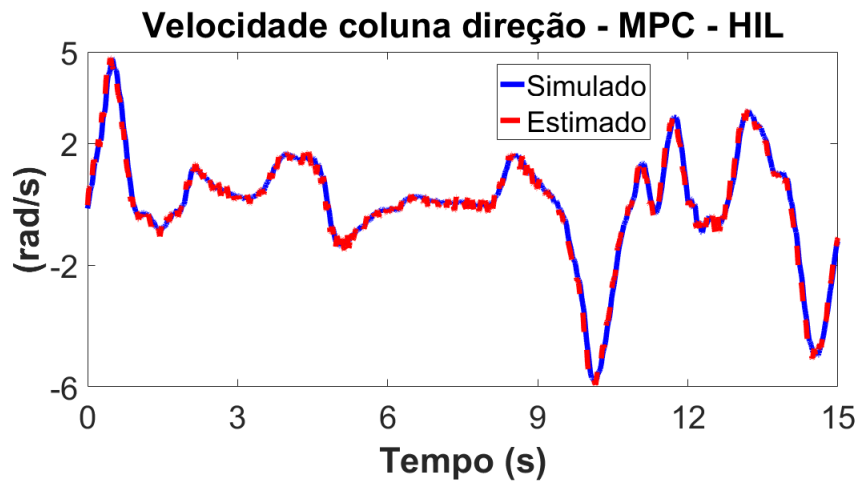


Figura 5.25: Estimação velocidade da coluna em MIL.

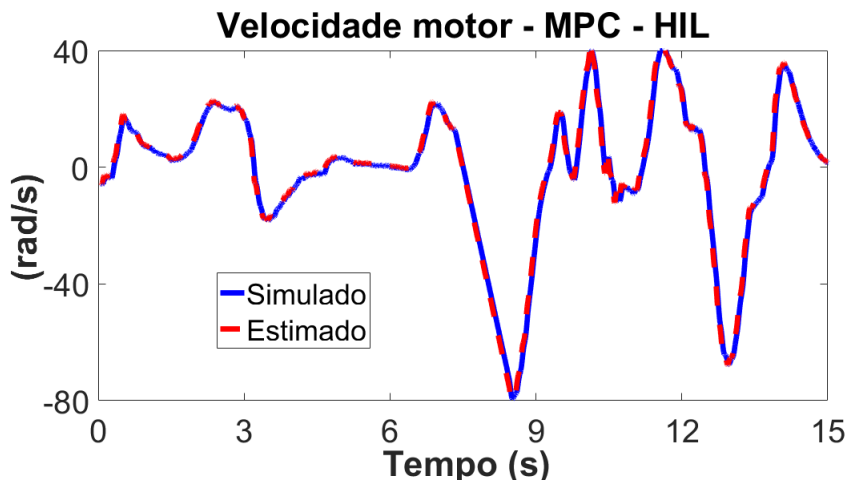


Figura 5.26: Estimação velocidade da coluna em SIL.

Observa-se que o filtro de Kalman estimou de forma satisfatória os estados não medidos diretamente. A fim avaliar a característica de robustez frente a variação dos parâmetros, uma simulação com a variação dos parâmetros ( $J_c$ ,  $B_c$ ,  $J_{eq}$ ,  $B_{eq}$ ,  $K_t$ ,  $R_m$ ,  $L_m$ ) em 10% da magnitude foi realizada. A figura 5.27 mostra o resultado da simulação em HIL para o torque de assistência quanto a variação dos parâmetros da planta do EPAS.

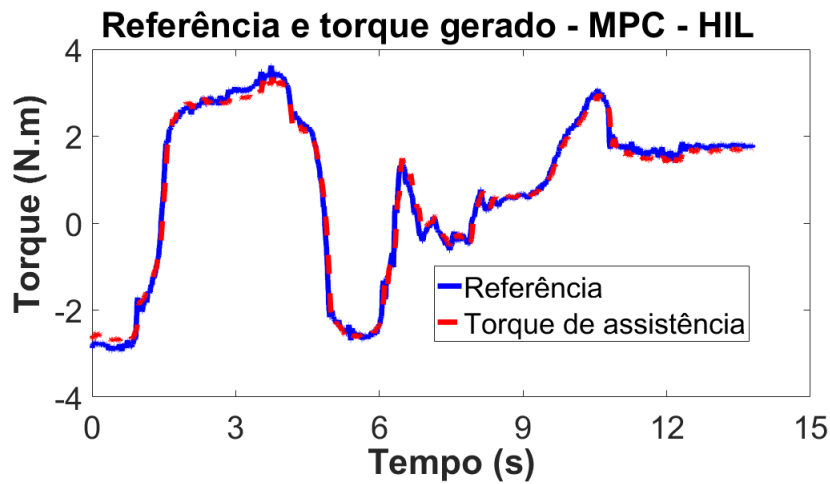


Figura 5.27: Referência e torque de assistência gerado para o teste HIL após variação dos parâmetros.

A fim de evidenciar o impacto do horizonte de predição na performance do controlador MPC, uma simulação foi para as mesmas condições de variação de parâmetros, todavia neste caso utilizou-se horizonte de predição  $N = 10$ . A figura 5.28 mostra o resultado desta simulação, em que observa-se o desempenho inferior da estratégia de controle quanto à variação dos parâmetros.

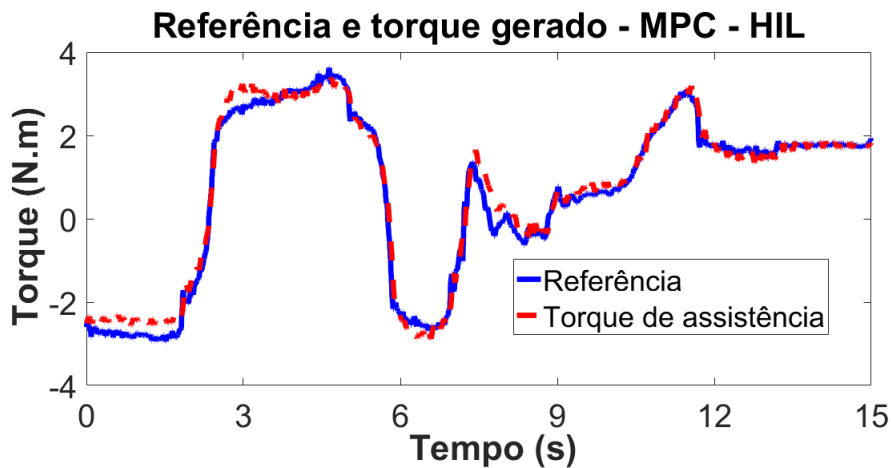


Figura 5.28: Referência e torque de assistência geral para o teste HIL após variação dos parâmetros com horizonte de predição reduzido.

Os resultados do estimador de Kalman também foram analisados após a variação dos parâmetros. As figuras 5.29 e 5.30 mostram os resultados da estimação com filtro de Kalman. Observa-se que apesar de haver redução na performance do estimador, a estratégia proposta apresentou característica de robustez após à variação dos parâmetros.

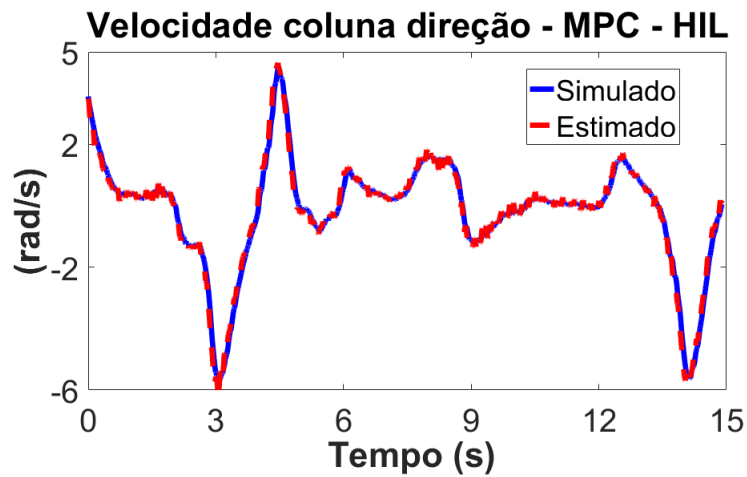


Figura 5.29: Estimação velocidade da coluna em HIL após a variação dos parâmetros

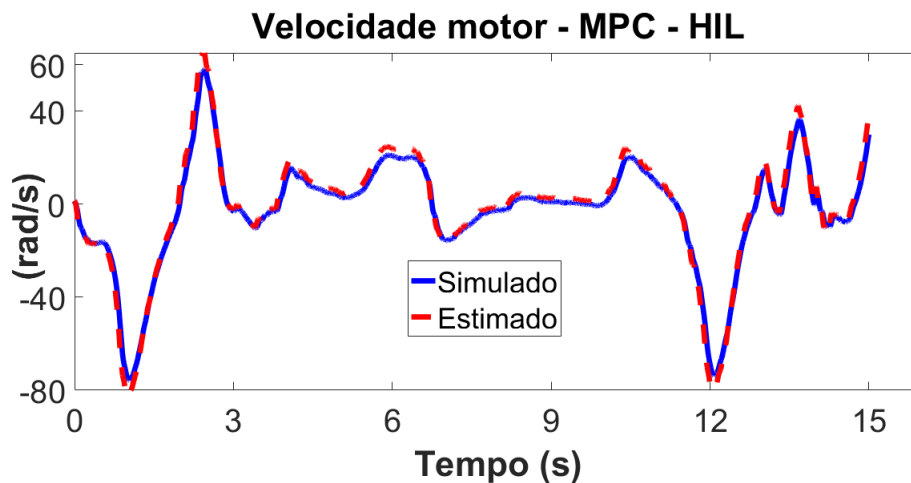


Figura 5.30: Estimação velocidade do motor em HIL após a variação dos parâmetros

A principal vantagem do controlador MPC comparado ao LQG se encontra no fato de adotar restrições que podem ser implementadas tanto nos estados, sinal de comando e taxa de variação do sinal de comando. No sistemas EPAS a aplicação das restrições são de grande importância a fim de manter o motor elétrico atuando dentro de uma faixa de operação. A figura 5.31 mostra a curva de velocidade do motor submetida à restrições  $[-40... + 40]rad/s$ , nota-se um bom desempenho da estratégia de controle uma vez que a velocidade do motor manteve-se totalmente dentro da faixa de operação e ao atingir os limites se manteve constante.



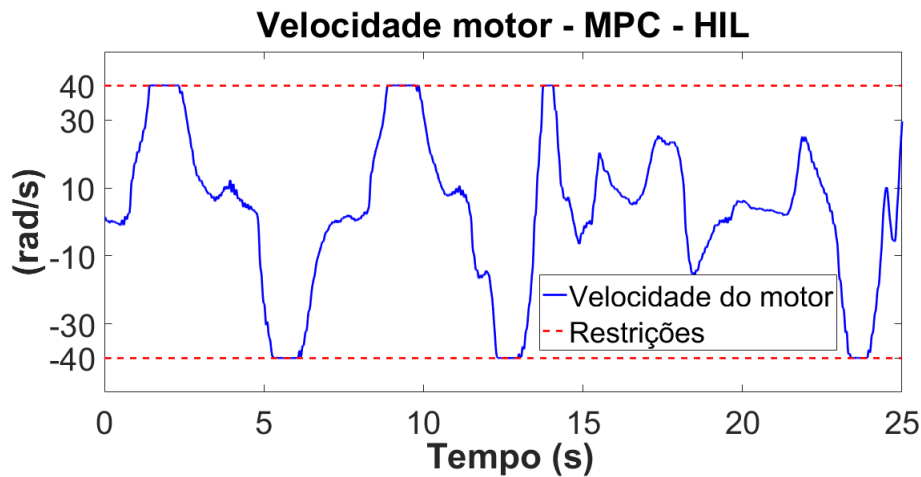


Figura 5.31: Velocidade do motor após a implementação de restrições de  $[-40... + 40]rad/s$ .

A variável de comando no caso do EPAS é a tensão aplicada no terminal do motor elétrico, o controle desta variável é de fundamental importância uma vez, esta variável ao passar dos limites operacionais do motor pode comprometer o dispositivo e em casos reais, trazer prejuízos para a segurança dos ocupantes do veículo. A figura 5.32 mostra o perfil da variável de controle para as simulações apresentadas no caso anterior, quando a velocidade do motor esteve sujeita à restrições.

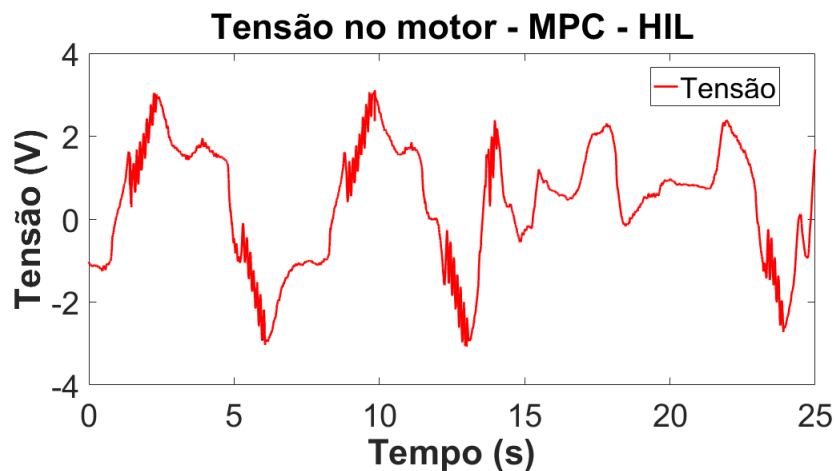


Figura 5.32: Perfil do sinal de controle para o caso de restrição na velocidade do motor.

Aplicar restrições ao comando é de fundamental importância para manter o motor elétrico dentro da faixa operacional. Neste caso, definiu-se uma faixa de operação para o sinal de controle entre  $[-2.5... + 2.5]V$ . A figura 5.33 mostra o resultado da simulação após a restrição na variável de comando.

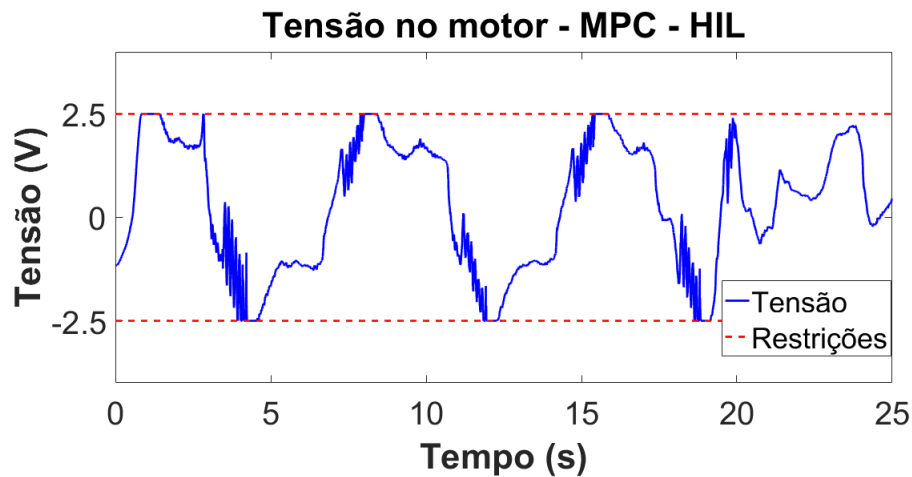


Figura 5.33: Perfil do sinal de controle sob ação de restrições em  $[-2.5... + 2.5]V$ .

Observa-se que o sinal de controle se manteve de forma satisfatória dentro das faixas de operações impostas, mostrando mais uma vez que a estratégia de controle MPC apresentou a performance esperada. Outra restrição que pode ser operada pelo controlador MPC é a restrição na variação do sinal de comando. A figura 5.34 mostra a variação do comando para o caso da restrição no comando apresentada na simulação anterior.

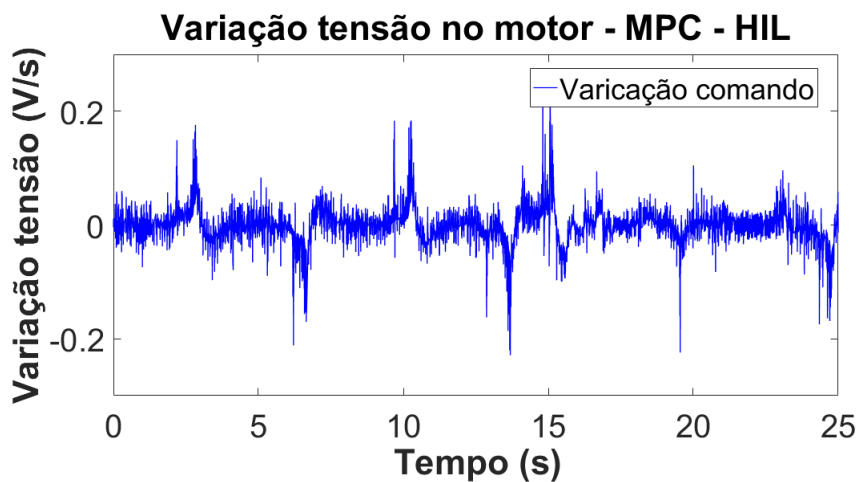


Figura 5.34: Perfil da variação do sinal controle para o caso de restrição na tensão do motor.

Desta forma, pode se definir faixas de operações, também, para a variação do sinal de controle como mostra a figura 5.35.

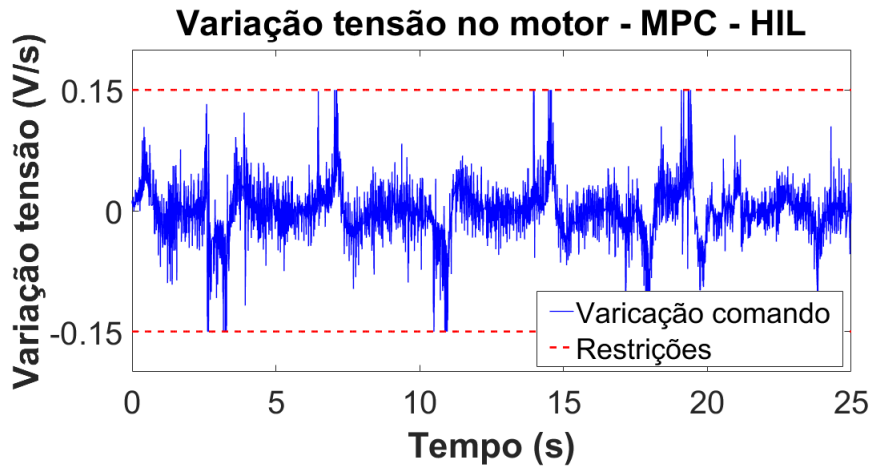


Figura 5.35: Perfil da variação de controle sob ação de restrições em  $[-0.15... + 0.15]V/s..$

A imposição de restrições em faixas de operação baixas em relação ao exigido pelo atuador pode comprometer o funcionamento do mesmo, ou seja, limitar sua operação de forma que não seja possível atingir o objetivo de controle e assim comprometer o segmento da trajetória. A figura 5.36 mostra o perfil de controle para uma simulação em que a tensão do motor é restrita em  $-3V$  e na figura 5.37 é mostrada o torque de assistência frente a referência gerada. Nota-se que o rastreamento da referência foi comprometido devido às restrições na tensão do motor, portanto pode ser necessário relaxar um pouco mais a restrição a fim de atingir o rastreamento, caso este seja o objetivo da estratégia de controle.

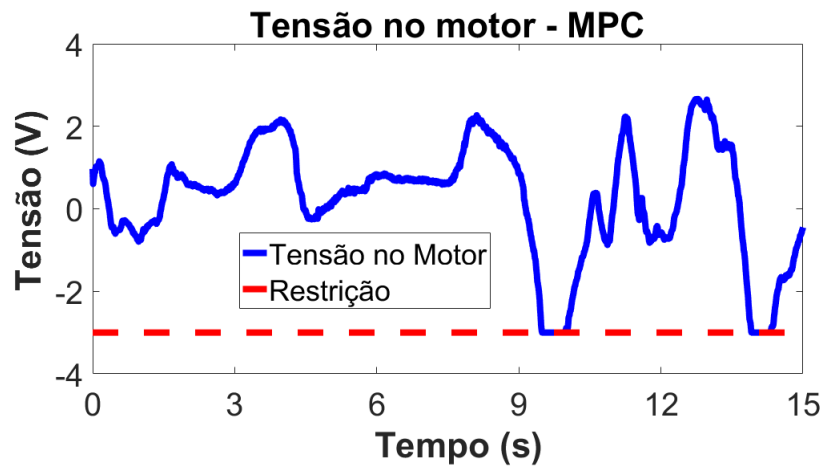


Figura 5.36: Perfil do sinal de controle sob ação de restrições em  $-3V$ .

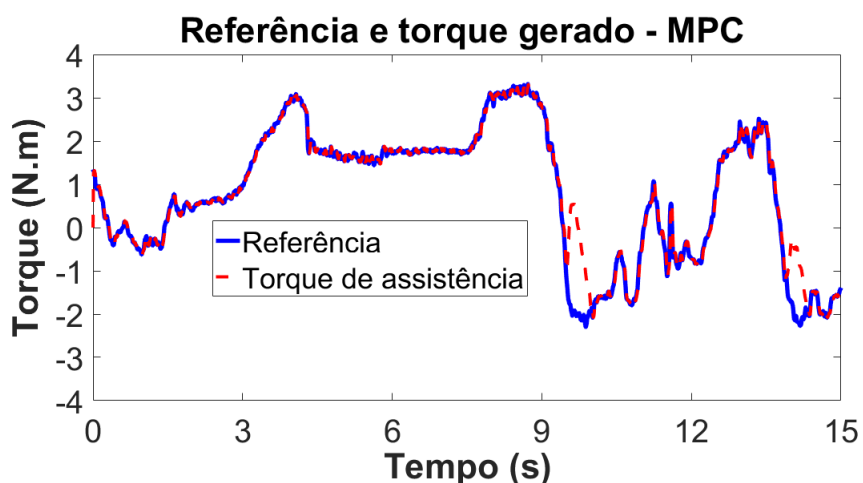


Figura 5.37: Comportamento do torque de assistência quando da tensão do motor sob ação de restrições.

De fato, o controle MPC ao ser implementado apresentou desempenho satisfatório no que diz respeito ao segmento de trajetória, mesmo na presença de perturbações e ruídos no sistema. Apresentou também características de robustez quanto à variação dos parâmetros da planta do EPAS. Por fim, as restrições nas variáveis de estado, sinal de comando e variação do sinal de comando foram implementadas de forma satisfatória, com as faixas de operação definidas nas restrições sendo respeitadas em ambos os casos.

## 6 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho buscou aplicar a metodologia de projeto baseado em modelos apoiada ao ciclo V no desenvolvimento do software de controle para um sistema EPAS. Duas estratégias de controle foram projetadas sendo controlador LQG e o controlador MPC. Ambas estratégias de controles foram implementadas através de modelos funcionais e testadas ao longo ciclo de desenvolvimento utilizando os métodos de testes MIL, SIL e HIL.

Em MIL as estratégias propostas foram testadas em ambiente de simulação onde pode se avaliar o desempenho dos controladores ainda como forma de modelos funcionais através de diagrama de blocos. O método SIL foi realizado através da geração automática de códigos em que pode verificar que os modelos funcionais puderam ser convertidos em linguagem C de forma satisfatória. Por fim a utilização do HIL permitiu testar as estratégias propostas em hardwares específicos para aplicações automotivos.

No que tange as estratégias de controle, o controlador LQG apresentou desempenho satisfatório e estabilidade quanto ao segmento da trajetória de referência. A característica de robustez do controlador quanto à variação de parâmetros também foi verificada ao variar os parâmetros da planta do sistema em 10% do valor original. O filtro de Kalman também apresentou resultados satisfatórios, uma vez que os sinais estimados foram mantidos próximos aos sinais diretamente medidos.

Para o controlador MPC, a estratégia apresentou resultados satisfatórios tanto em nível de simulação quanto em nível de hardware. Mesmo submetidos às fontes de perturbação e ruídos a estratégia manteve o desempenho e estabilidade. O MPC apresentou, também, características de robustez frente à incerteza nos parâmetros da planta do modelo EPAS.

A estratégia de parametrização do controle foi de fundamental importância para implementação do MPC em HIL, uma vez que, permitiu cenários com baixo tempo de amostragem o que garante a performance do controlador e horizonte de predição elevado, o que permitiu a utilização das restrições nos estados, variável de comando e variação da variável de comando.

A contribuição deste trabalho foi abordar de forma sequencial os passos da utilização do MBD para o desenvolvimento do software de controle de um subsistema automotivo. Em que verificou-se que os testes MIL, SIL e HIL são complementares e se projetado de forma correta, espera-se resultados semelhantes em ambos os casos. Outra contribuição foi a configuração de uma plataforma HIL para validação de sistemas de controle em tempo real, em que todos os softwares já estão devidamente configurados para receber outros tipos de sistemas e controladores.

Uma importante contribuição futura seria a implementação de um processo de RCP (*Rapid Control Prototyping*) aplicado em um sistema EPAS real, uma vez que, a plataforma MicroAuto-boxII oferece suporte a esse processo. Portanto, diferentes estratégias de controle poderiam ser aplicadas e os desempenhos comparados em uma situação prática. Outra contribuição significa-

tiva é projetar o modelo do veículo de forma que o perfil de perturbação proveniente do contato pneu-pista represente com mais fidelidade uma aplicação real.

# REFERÊNCIAS BIBLIOGRÁFICAS

- AARENSTRUP, R. *Managing Model-Based Design*. [S.l.]: The MathWorks, Inc., 2015.
- ABU-AYYAD, M.; DUBAY, R. Real-time comparison of a number of predictive controllers. *ISA transactions*, Elsevier, v. 46, n. 3, p. 411–418, 2007.
- ALAMIR, M. *A Pragmatic Story of Model Predictive Control: Self-Contained Algorithms and Case-Studies*. 2013.
- AWEDIKIAN, R.; YANNOU, B. Design of a validation test process of an automotive software. *International Journal on Interactive Design and Manufacturing (IJIDeM)*, v. 4, n. 4, p. 259–268, Nov 2010. Disponível em: <<http://dx.doi.org/10.1007/s12008-010-0108-2>>.
- BEMPORAD, A.; MORARI, M. Robust model predictive control: A survey. *Robustness in identification and control*, Springer, p. 207–226, 1999.
- BERGSTROM, D.; GORANSSON, R. *Model- and Hardware-in-the-Loop Testing in a Model-Based Design Workflow*. 2016.
- BILL, C.; HILL, R. *Control of a Tailless Fighter using Gain-Scheduling*. 2004.
- BLAKE, D.; LEADER, A. Embedded systems and vehicle innovation. *Automotive Engineering International*, SAE INTERNATIONAL, v. 113, n. 10, p. 42, 2005.
- BORDONS, C. et al. Model predictive control for power management in hybrid fuel cell vehicles. In: *2010 IEEE Vehicle Power and Propulsion Conference*. [S.l.: s.n.], 2010. p. 1–6.
- BOSCH. *Injeção eletrônica: sensores e atuadores*. [S.l.], 2014.
- BROY, M. et al. What is the benefit of a model-based design of embedded software systems in the car industry? *Software Design and Development: Concepts, Methodologies, Tools, and Applications: Concepts, Methodologies, Tools, and Applications*, IGI Global, p. 310, 2013.
- BUCANAC, C. The v-model. *IDE, University of Larlskrona/Ronneby*, 1999.
- CHABAAN, R. C.; WANG, L. Y. Control of electrical power assist systems: H infinity design, torque estimation and structural stability. *JSAE review*, Elsevier, v. 22, n. 4, p. 435–444, 2001.
- CHACHUAT, B. *Nonlinear and dynamic optimization: From theory to practice*. [S.l.], 2007.
- CHITU, C. et al. Controller design for an electric power steering system based on lqr techniques. *COMPEL-The international journal for computation and mathematics in electrical and electronic engineering*, Emerald Group Publishing Limited, v. 32, n. 3, p. 763–775, 2013.
- CICEO, S. et al. Model-based design and testing for electric vehicle energy consumption analysis. *ELECTROTEHNICĂ, ELECTRONICĂ, AUTOMATICĂ (EEA)*, print, 2015.
- CLARKE, D. W.; MOHTADI, C.; TUFFS, P. Generalized predictive control—part i. the basic algorithm. *Automatica*, Elsevier, v. 23, n. 2, p. 137–148, 1987.
- CUTLER, C. R.; RAMAKER, B. L. Dynamic matrix control?? a computer control algorithm. In: *Joint automatic control conference*. [S.l.: s.n.], 1980. p. 72.

- DAMIRAN, U.; FU, C.; TAN, W. Model predictive control of nonlinear drum boiler. In: *2016 11th International Forum on Strategic Technology (IFOST)*. [S.l.: s.n.], 2016. p. 544–548.
- DELATORE, F. *Controle multivariável em redes de trocadores de calor com bypasses*. Tese (Doutorado), 2011.
- DOMAHIDI, A. Survey of industrial application of embedded model predictive control. In: *European Control Conference*. [S.l.: s.n.], 2016.
- DONG, L. et al. Active disturbance rejection control for an electric power assist steering system. *International Journal of Intelligent Control and System*, v. 15, n. 1, p. 18–24, 2010.
- EYISI, E. et al. Model-based control design and integration of cyberphysical systems: an adaptive cruise control case study. *Journal of Control Science and Engineering*, Hindawi Publishing Corp., v. 2013, p. 1, 2013.
- FERREAU, H. J. et al. qpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, p. 327–363, 2014.
- GARCIA, C. E.; PRETT, D. M.; MORARI, M. Model predictive control: theory and practice—a survey. *Automatica*, Elsevier, v. 25, n. 3, p. 335–348, 1989.
- GOODWIN, G. C. et al. Robust model predictive control: reflections and opportunities. *Journal of Control and Decision*, Taylor & Francis, v. 1, n. 2, p. 115–148, 2014.
- GUIMARÃES, A. *Análise da norma ISO11783 e sua utilização na implementação do barramento do implemento de um monitor de semeadora*. Dissertação (Mestrado), 2003.
- GUMBRICH, S.; KOPPINGER, P. Embedded systems – here, there and everywhere. 2004.
- HAIBO, M.; YING, Y. Electrical power assisted steering for evs and hev. *EPS*, v. 6, p. 35, 2011.
- HSIEH, Y. M.; LIU, Y. C. Model predictive control strategy for plug-in hybrid electric vehicles. In: *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. [S.l.: s.n.], 2016. p. 1–6.
- JI, G. et al. Development of a model-based controller prototype for a pem fuel cell engine with ascet-md. In: *2009 IEEE International Conference on Control and Automation*. [S.l.: s.n.], 2009. p. 2100–2105. ISSN 1948-3449.
- KATTANKULATHUR, K. Systematic approach in v-model development cycle for an automotive embedded control system. 2013.
- KELEMENOVÁ, T. et al. Model based design and hil simulations. *American Journal of Mechanical Engineering*, Science and Education Publishing, v. 1, n. 7, p. 276–281, 2013.
- KIM, J.-H.; SONG, J.-B. Control logic for an electric power steering system using assist motor. *Mechatronics*, Elsevier, v. 12, n. 3, p. 447–459, 2002.
- KRÜGER, M. B. I. H.; MEISINGER, M. Automotive software-connected services in mobile networks. Springer, 2006.
- KURUCSÓ, B. et al. State space control of quadratic boost converter using lqr and lqg approaches. In: *2015 Intl Aegean Conference on Electrical Machines Power Electronics (ACEMP), 2015 Intl Conference on Optimization of Electrical Electronic Equipment (OPTIM) 2015 Intl Symposium on Advanced Electromechanical Motion Systems (ELECTROMOTION)*. [S.l.: s.n.], 2015. p. 642–648.



- LEE, D.; KIM, K. S.; KIM, S. Controller design of an electric power steering system. *IEEE Transactions on Control Systems Technology*, PP, n. 99, p. 1–8, 2017. ISSN 1063-6536.
- LEE, J. H. Model predictive control: Review of the three decades of development. *International Journal of Control, Automation and Systems*, Springer, v. 9, n. 3, p. 415–424, 2011.
- LEVINE, W. S. *The control handbook*. [S.l.]: CRC press, 1996.
- LING, K. V.; MACIEJOWSKI, J. M.; WU, B. F. Computing the cost of multiplexed mpc. In: *2008 10th International Conference on Control, Automation, Robotics and Vision*. [S.l.: s.n.], 2008. p. 582–587.
- LIU, B.; ZHANG, H.; ZHU, S. An incremental v-model process for automotive development. In: *2016 23rd Asia-Pacific Software Engineering Conference (APSEC)*. [S.l.: s.n.], 2016. p. 225–232.
- MAROUF, A. et al. A new control strategy of an electric-power-assisted steering system. *IEEE Transactions on Vehicular Technology*, IEEE, v. 61, n. 8, p. 3574–3589, 2012.
- MATHWORKS. *Why Adopt Model-Based Design for Embedded Control Software Development?* 2014. Disponível em: <<https://www.mathworks.com/programs/resource-portals/adopting-model-based-design/files/why-adopt-model-based-design-for-embedded-control-software-development.pdf>>.
- MAYNE, D. et al. Constrained model predictive control: Stability and optimality. *Automatica*, v. 36, n. 6, p. 789 – 814, 2000.
- MORARI, M.; LEE, J. H. Model predictive control: past, present and future. *Computers & Chemical Engineering*, Elsevier, v. 23, n. 4, p. 667–682, 1999.
- MOUDGALYA, K. M. Linear quadratic gaussian control. In: \_\_\_\_\_. *Digital Control*. John Wiley & Sons, Ltd, 2007. p. 463–485. ISBN 9780470517109. Disponível em: <<http://dx.doi.org/10.1002/9780470517109.ch13>>.
- MURILO, A. *Contributions à la Commande Prédicative Non Linéaire pour les Systèmes à Dynamiques Rapides*. Tese (Doutorado) — Institut National Polytechnique de Grenoble-INPG, 2009.
- NAVET, N. et al. Trends in automotive communication systems. *Proceedings of the IEEE*, v. 93, n. 6, p. 1204–1223, June 2005. ISSN 0018-9219.
- NEME, J. H.; SANTOS, M. M. D.; TEIXEIRA, E. L. S. Model based design for automobile external lighting systems. In: *SAE Technical Paper*. [S.l.]: SAE International, 2015.
- NGUYEN, M. Q. et al. A model predictive control approach for semi-active suspension control problem of a full car. In: *2016 IEEE 55th Conference on Decision and Control (CDC)*. [S.l.: s.n.], 2016. p. 721–726.
- PARVATHY, R.; DANIEL, A. E. A survey on active disturbance rejection control. In: *IEEE. Automation, Computing, Communication, Control and Compressed Sensing (iMac4s), 2013 International Multi-Conference on*. [S.l.], 2013. p. 330–335.
- PRABHU, S. M.; MOSTERMAN, P. J. Model-based design of a power window system: Modeling, simulation and validation. In: *Proceedings of IMAC-XXII: A Conference on Structural Dynamics, Society for Experimental Mechanics, Inc., Dearborn, MI*. [S.l.: s.n.], 2004.
- RICHALET, J. et al. Model predictive heuristic control: Applications to industrial processes. *Automatica*, Elsevier, v. 14, n. 5, p. 413–428, 1978.
- ROWELL, D. State-space representation of lti systems. In: \_\_\_\_\_. *2.14 Analysis and Design of Feedback Control Systems*. [S.l.: s.n.], 2002.

- SANGIOVANNI-VINCENTELLI, A.; NATALE, M. D. Embedded system design for automotive applications. *Computer*, IEEE, n. 10, p. 42–51, 2007.
- SCHMID, C.; BIEGLER, L. Quadratic programming methods for reduced hessian sqp. *Computers & Chemical Engineering*, v. 18, n. 9, p. 817 – 832, 1994. ISSN 0098-1354. An International Journal of Computer Applications in Chemical Engineering.
- SENTHILKUMAR, K.; RAMADOSS, R. Designing multicore ecu architecture in vehicle networks using autosar. In: *2011 Third International Conference on Advanced Computing*. [S.l.: s.n.], 2011. p. 270–275.
- SEO, S. H. et al. A model-based design for electronic control units based on osek/vdx. In: *2009 IEEE International Symposium on Industrial Electronics*. [S.l.: s.n.], 2009. p. 681–685. ISSN 2163-5137.
- SHAHBAKHTI, M.; LI, J.; HEDRICK, J. K. Early model-based verification of automotive control system implementation. In: *IEEE. American Control Conference (ACC), 2012*. [S.l.], 2012. p. 3587–3592.
- SHWETHA, G.; RAMESH, H.; SHANKAPAL, S. Modeling, simulation and implementation of a proportional-derivative controlled column-type eps. *International Journal of Enhanced Research in Science Technology and Engineering*, v. 2, n. 9, p. 10–19, 2013.
- SIMON, D. Kalman filtering. *Embedded systems programming*, v. 14, n. 6, p. 72–79, 2001.
- SMITH, P. F.; PRABHU, S. M.; FRIEDMAN, J. *Best practices for establishing a model-based design culture*. [S.l.], 2007.
- SONG, J. et al. Model development and control methodology of a new electric power steering system. *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, Sage Publications, v. 218, n. 9, p. 967–975, 2004.
- STELLA, G. *Aplicando a metodologia de desenvolvimnto baseado em modelos para funções de software automotivo*. Dissertação (Mestrado), 2015.
- TECHOLOGIES, I. *Electric Power Steering (EPS) with GMR-Based Angular and Linear Hall Sensor*. [S.l.], 2008.
- TØNDEL, P.; JOHANSEN, T. A.; BEMPORAD, A. An algorithm for multi-parametric quadratic programming and explicit mpc solutions. *Automatica*, v. 39, n. 3, p. 489 – 497, 2003. ISSN 0005-1098. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0005109802002509>>.
- WANG, Y.; BOYD, S. Fast model predictive control using online optimization. *IEEE Transactions on Control Systems Technology*, v. 18, n. 2, p. 267–278, March 2010. ISSN 1063-6536.
- ZANG, H.; LIU, M. Fuzzy neural network pid control for electric power steering system. In: *IEEE. 2007 IEEE International Conference on Automation and Logistics*. [S.l.], 2007. p. 643–648.
- ZHAO, W.-z. et al. H infinity control of novel active steering integrated with electric power steering function. *Journal of Central South University*, Springer, v. 20, p. 2151–2157, 2013.
- ZOLOTAS, A. C. et al. A study on lqg/ltr control for damping inter-area oscillations in power systems. *IEEE Transactions on Control Systems Technology*, v. 15, n. 1, p. 151–160, 2007.

## APÊNDICES