



**Universidade de Brasília  
Faculdade de Tecnologia  
Departamento de Engenharia Mecânica**

**Desenvolvimento de um Algoritmo de  
Detecção de Deslizamento Baseado em Dados  
para uma Garra Robótica Implementado em  
SoC-FPGA**

Jose Alfredo Mendoza Peñaloza

**DISSERTAÇÃO DE MESTRADO  
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS MECATRÔNICOS**

Brasília  
2024

**Universidade de Brasília  
Faculdade de Tecnologia  
Departamento de Engenharia Mecânica**

**Desenvolvimento de um Algoritmo de  
Detecção de Deslizamento Baseado em Dados  
para uma Garra Robótica Implementado em  
SoC-FPGA**

Jose Alfredo Mendoza Peñaloza

Dissertação de Mestrado submetida ao Departamento de Engenharia Mecânica da Universidade Brasília como parte dos requisitos necessários para a obtenção do grau de Mestre

Orientador: Prof. Dr. Daniel Muñoz Arboleda

Brasília  
2024

Mendoza Peñaloza, Jose Alfredo.  
M769d      Desenvolvimento de um Algoritmo de Detecção de Deslizamento Baseado em Dados para uma Garra Robótica Implementado em SoC-FPGA / Jose Alfredo Mendoza Peñaloza; orientador Daniel Muñoz Arboleda. -- Brasília, 2024.  
81 p.

Dissertação de Mestrado (Programa de Pós-Graduação em Sistemas Mecatrônicos) -- Universidade de Brasília, 2024.

1. Controle de impedância. 2. SoC-FPGA. 3. Detecção de deslizamentos. 4. Garra robótica. I. Arboleda, Daniel Muñoz, orient. II. Título

**Universidade de Brasília  
Faculdade de Tecnologia  
Departamento de Engenharia Mecânica**

**Desenvolvimento de um Algoritmo de Detecção de  
Deslizamento Baseado em Dados para uma Garra  
Robótica Implementado em SoC-FPGA**

Jose Alfredo Mendoza Peñaloza

Dissertação de Mestrado submetida ao Departamento de Engenharia Mecânica da Universidade Brasília como parte dos requisitos necessários para a obtenção do grau de Mestre

Trabalho aprovado. Brasília, 22 de julho de 2024:

---

**Prof. Dr. Daniel Muñoz Arboleda,**  
**UnB/FT/ENM**  
Orientador

---

**Prof. Dr. Jones Yudi Mori Alves da Silva,**  
**UnB/FT/ENM**  
Examinador interno

---

**Prof. Dr. Juan Sebastian Toquica Arenas,**  
**UFC/CIJA**  
Examinador externo

Brasília  
2024

*Este trabalho é dedicado às pessoas que necessitam do uso de próteses robóticas, pois todos devemos nos empenhar para melhorar a qualidade de vida de nós mesmos e de nossos semelhantes.*

# Agradecimentos

Agradecimentos a Deus, à minha mãe Luz Dary Peñaloza, ao meu pai Dimas Mendoza, ao meu irmão Alberto José Mendoza, aos meus amigos Mario Pastrana e Brayan Figueroa, e um agradecimento especial ao meu orientador Daniel Muñoz Arboleda, agradecimentos a todos pelo apoio incondicional.

# Resumo

No campo de estudo, desenvolvimento e controle de mãos robóticas, diversos desafios surgem atualmente, tais como o controle preciso do movimento da mão, a força aplicada para agarrar objetos e a adaptabilidade a diferentes situações de trabalho. Entretanto, um dos pontos mais críticos sobre os quais são realizados estudos é a detecção e controle de deslizamentos de objetos. Esta dinâmica não linear altera drasticamente o comportamento dos controladores de mãos robóticas e impede o desenvolvimento eficiente das tarefas.

De acordo com o exposto, este trabalho descreve o desenvolvimento de um algoritmo de detecção de deslizamentos de objetos para mãos robóticas. Trata-se de um algoritmo de baixa complexidade computacional, baseado em dados de deslizamento coletados por meio de sensores de força ao agarrar diferentes objetos. Esses dados são processados por meio de um filtro de média móvel, a seguir destacam-se suas características a partir do uso da primeira derivada e, por fim, utilizando um algoritmo de detecção de picos, encontram-se os deslizamentos ocorridos.

O algoritmo de detecção de deslizamentos foi desenvolvido como parte do projeto da mão robótica UnB-Hand, o qual foi implementado em um SoC FPGA Zynq 7020 usando a metodologia de co-projecto Hardware-Software. O algoritmo foi implementado no bloco de programação lógica (PL) através da linguagem VHDL (Hardware Description Language). Finalmente, o algoritmo foi implementado em um microcontrolador para tarefas de preensão de objetos utilizando uma garra robótica, a fim de demonstrar seu funcionamento. Os resultados mostraram que o algoritmo de detecção de deslizamentos possui uma precisão de 93,33% em sua implementação em hardware. Alcançou-se uma utilização total de recursos de Look Up Tables de 2,44% para a implementação dos módulos do filtro de média móvel, da derivada de primeira ordem e do algoritmo de detecção de picos, com um consumo energético total de 1,708 W. Para a análise dos experimentos de deslizamentos realizados, é apresentada uma matriz de confusão que mostra 11 verdadeiros positivos, 1 falso negativo e 2 falsos positivos, resultando em uma precisão de 84,62%, recall de 91,67% e F1 Score de 88%.

**Palavras-chave:** Controle de impedância. SoC-FPGA. Detecção de deslizamentos. Garra robótica.

# Abstract

In the field of study, development, and control of robotic hands, several challenges currently arise, including the precise control of hand movement, the applied force required to grasp objects, and adaptability to various working conditions. However, one of the most critical aspects under investigation is the detection and control of object slippage. This nonlinear dynamic significantly alters the behavior of robotic hand controllers and hinders the efficient execution of tasks.

Accordingly, this work describes the development of an object slippage detection algorithm for robotic hands. The algorithm is designed to be computationally efficient, utilizing slippage data collected through force sensors during the grasping of various objects. This data is processed using a moving average filter; subsequently, key characteristics are highlighted through the use of the first derivative, and finally, a peak detection algorithm is employed to identify occurrences of slippage.

The slippage detection algorithm was developed as part of the UnB-Hand robotic hand project, which was implemented on a Zynq 7020 SoC FPGA using the Hardware-Software co-design methodology. The algorithm was implemented in the programmable logic (PL) block using VHDL (Hardware Description Language). Finally, the algorithm was integrated into a microcontroller for object-grasping tasks utilizing a robotic gripper to demonstrate its functionality. The results showed that the slippage detection algorithm achieved an accuracy of 93.33% in its hardware implementation. The total utilization of Look-Up Table resources reached 2.44% for the implementation of the moving average filter, first derivative, and peak detection algorithm modules, with a total power consumption of 1.708 W. For the analysis of the slippage experiments conducted, a confusion matrix is presented, showing 11 true positives, 1 false negative, and 2 false positives, resulting in an accuracy of 84.62%, a recall of 91.67%, and an F1 score of 88%.

**Keywords:** Impedance control. SoC-FPGA. Slip detection. Robotic gripper.

# Lista de ilustrações

Figura 1 – Projeto mecânico da mão robótica UnB-Hand no software SolidWorks (PERTUZ; LLANOS, C. H.; MUÑOZ, D. M., 2021). . . . .	14
Figura 2 – Projeto de mão robótica UnB-Hand. a) Modelo em impressão 3D da UnB-Hand; b) Aplicação de agarre de objetos com a UnB-Hand (PERTUZ; GHERARDINI et al., 2022). . . . .	14
Figura 3 – Sistema de massa, mola e amortecedor. . . . .	23
Figura 4 – Modelo de controlador PID. . . . .	23
Figura 5 – Chip Zynq 7020 dual-core SoC FPGA (AMD-XILINX, 2024). . . . .	34
Figura 6 – Arquitetura interna FPGA. . . . .	35
Figura 7 – Sensor piezorresistivo FSR 400 da Interlink Electronics (ELECTRONICS, 2024). . . . .	41
Figura 8 – Sensor rotativo Murata SV01 (MURATA, 2024). . . . .	42
Figura 9 – High Speed Servo Actuator MG995 (SERVOMOTOR., 2024). . . . .	42
Figura 10 – Garra robótica de alumínio (SARAVATI, 2024). . . . .	42
Figura 11 – Dados de calibração do sensor de força FSR400 sem fita de silicone. . . . .	44
Figura 12 – Dados de calibração do sensor de força FSR400 com fita de silicone. . . . .	45
Figura 13 – Arquitetura geral do co-projeto Hardware/Software do sistema de detecção de deslizamentos. . . . .	49
Figura 14 – Representação gráfica da arquitetura em hardware do módulo filtro de média móvel. . . . .	49
Figura 15 – Representação gráfica da arquitetura em hardware do módulo derivado de primeira ordem. . . . .	50
Figura 16 – Representação gráfica da arquitetura em hardware do módulo detector de picos. . . . .	50
Figura 17 – Diagrama de blocos do co-projeto de Hardware/Software. . . . .	51
Figura 18 – Modelo interno do controle de impedância. . . . .	51
Figura 19 – Modelo de controle de impedância Matlab/Simulink. . . . .	54
Figura 20 – Circuito eletrônico da garra robótica. . . . .	54
Figura 21 – Design de garra robótica no software CAD SolidWorks (GRABCAD-COMMUNITY, 2024). . . . .	55
Figura 22 – Design 3D do acoplamento do sensor de força. . . . .	55
Figura 23 – Conjunto completo da garra robótica com sensores e servomotor. . . . .	56
Figura 24 – Corpo de prova para experimentos na garra robótica. . . . .	56
Figura 25 – Objetos para a base de dados experimental. a) Garrafa de água; b) Laranja; c) Telefone móvel; d) Livro. . . . .	58
Figura 26 – Diagrama geral do sistema robótico. . . . .	59

Figura 27 – FPGA-layout dos módulos de detecção de deslizamento. Azul: <i>Filtro de média móvel</i> . Roxo: <i>Derivada de primeira ordem</i> . Amarelo: <i>Detector de picos</i> .	60
Figura 28 – Consumo de energia da implementação em FPGA do sistema de detecção de deslizamentos.	61
Figura 29 – Zoom da simulação comportamental do sistema de detecção de deslizamentos realizada no software Vivado.	62
Figura 30 – Resultado da implementação hardware do sistema de detecção de deslizamentos. a) Comparação entre o módulo em hardware de filtro de média móvel em cor azul e modelo de referência em vermelho; b) Comparação entre o módulo em hardware da derivada de primeira ordem em cor azul e o modelo de referência de vermelho. Os pontos de detecção de deslizamento são representados com linhas verticais laranjas.	63
Figura 31 – Respostas do controlador de impedância: <b>31a)</b> <i>setpoint</i> de 300 grf; <b>31b)</b> <i>setpoint</i> de 500 grf; <b>31c)</b> <i>setpoint</i> de 650 grf; <b>31d)</b> <i>setpoint</i> de 750 grf.	64
Figura 32 – Resultados do experimento de deslizamento Nº 1. <b>32a)</b> Filtro de média móvel; <b>32b)</b> Derivada de primeira ordem com os deslizamentos detectados pelo modelo em tempo real apresentados em azul; <b>32c)</b> Saída do controlador de impedância PWM (%).	65
Figura 33 – Resultados do experimento de deslizamento Nº 2. <b>33a)</b> Filtro de média móvel; <b>33b)</b> Derivada de primeira ordem com os deslizamentos detectados pelo modelo em tempo real apresentados em azul; <b>33c)</b> Saída do controlador de impedância PWM (%).	66
Figura 34 – Resultados do experimento de deslizamento Nº 3: <b>34a)</b> Filtro de média móvel; <b>34b)</b> Derivada de primeira ordem com os deslizamentos detectados pelo modelo em tempo real apresentados em azul; <b>34c)</b> Saída do controlador de impedância PWM (%).	67
Figura 35 – Resultados do experimento de deslizamento Nº 4. <b>35a)</b> Filtro de média móvel; <b>35b)</b> Derivada de primeira ordem com os deslizamentos detectados pelo modelo de Matlab em azul; <b>35c)</b> Saída do controlador de impedância PWM (%).	68

# Lista de tabelas

Tabela 1 – Características do projeto UnB-Hand. . . . .	16
Tabela 2 – Características projeto sistema de detecção de deslizamento. . . . .	17
Tabela 3 – Trabalhos relacionados que realizam controle de força em mãos robóticas.	26
Tabela 4 – Trabalhos relacionados com detecção de deslizamentos em mãos robóticas.	33
Tabela 5 – Tabela geral de componentes de um SoC FPGA. . . . .	35
Tabela 6 – Tabela de trabalhos relacionados que utilizam FPGA para acelerar controladores de mãos robóticas. . . . .	36
Tabela 7 – Tabela de calibração do sensor FSR 400. . . . .	43
Tabela 8 – Utilização de recursos (chip xc7z020clg484). . . . .	61
Tabela 9 – Tabela da análise do erro do algoritmo de detecção de deslizamentos. (SW: software; HW: hardware). . . . .	63
Tabela 10 – Tabela de resultados dos experimentos de detecção de deslizamentos do modelo do sistema de controle da garra robótica, comparando o modelo de referência e o modelo em tempo real. . . . .	69
Tabela 11 – Gráfico de confusão para o modelo em tempo real. . . . .	70
Tabela 12 – Trabalhos correlatos que reportam precisão na detecção de deslizamentos.	70

# Glossário

- $C$  Contador de posição de derivada. 51
- $F_i$   $i$ -ésima medição de força. 50
- $LimitH$  Ponto de derivada máxima no deslizamento. 51
- $LimitL$  Ponto de derivada mínima no deslizamento. 51
- $ThresF$  Limite mínimo de força de deslizamentos. 51
- $f$  Filtro na ação derivativa. 28
- $firstslip$  Início de um possível deslizamento. 54
- $i$  Número de partículas. 28
- $j$  Número de dimensões. 28
- $n$  Número de amostras. 48
- $preAvg$  Valor do filtro de média móvel anterior. 53
- $range$  Limiar máximo de amostras da derivada. 51
- $s$  Variância. 48
- $v$  Velocidade da partícula. 28
- $w$  Fator de inércia. 28
- $x$  Posição. 26
- $K_d$  Constante derivativa. 27
- $K_i$  Constante integral. 27
- $K_p$  Constante proporcional. 27
- $K_B$  Constante do amortecedor. 26
- $K_K$  Constante da mola. 26
- $K_M$  Constante de massa. 26
- $U_1 U_2$  Valor de distribuição uniforme das partículas. 28

$\Delta k$  Tempo discreto de amostragem. 51

$\bar{F}(k)$  Força filtrada por média móvel em tempo discreto. 50

$\bar{x}$  Média de um conjunto de números. 48

$\bar{y}$  Média do conjunto de sensores FSR. 50

$\ddot{x}$  Aceleração. 26

$\dot{F}(k)$  Derivada da força em tempo discreto. 51

$\dot{F}_H$  Valor da derivada no final do deslizamento. 51

$\dot{F}_L$  Valor da derivada no início do deslizamento. 51

$\dot{x}$  Velocidade. 26

$c_1$  Coeficiente cognitivo individual. 28

$c_2$  Coeficiente cognitivo social. 28

$e_{ss}$  Erro em regime permanente. 57

$last\_x$  Último aceleração estimada. 56

$last\_v$  Último velocidade estimada. 56

$last\_p$  Último posição estimada. 56

$pos_H$  Posição final do deslizamento. 51

$pos_L$  Posição inicial do deslizamento. 51

$t_r$  Tempo de resposta. 57

$x_i$  Posição da partícula atual. 28

$y_i$  Melhor posição da partícula atual. 28

$y_s$  Melhor posição entre todas as partículas. 28

**ADC** Analog to Digital Converter. 19

**ARM** Advanced RISC Machine. 18

**AXI** Advanced eXtensible Interface. 38

**BP** Bandpass filters. 36

**BRAM** Block RAM. 64

**CAD** Computer-Aided Design. 59

**CLB** Configurable logical blocks. 38

**CNN** Convolutional Neural network. 34

**CPU** Central processing unit. 38

**CSP** Critical sliding point. 35

**DC** Direct Current. 18

**DDR** Double Data Rate memory. 52

**DESC** Discrete Event-driven Sensory Feedback Control. 39

**DNN** Deep Neural Network. 34

**DoF** Degree of Freedom. 17

**DSP** Digital Signal Processing. 19

**DT** Decision trees. 36

**FF** Flip-flops. 19

**FFT** Fast Fourier transform. 36

**FIFO**, *regF* First in-first out memory. 50

**FPGA** Field Programmable Gate Array. 18

**FSR** Force Sensing Resistor. 20

**FT** Faculdade de Tecnologia. 17

**GPU** Graphics processing unit. 34

**Grf** Gramas-Força. 47

**HP** Hodrick–Prescott filters. 36

**HW** Hardware. 66

**IMU** Inertial Measurement Unit. 30

**IP** Intellectual Property. 19

**KNN** K-nearest neighbors. 36

**LP** Lowpass filters. [36](#)

**LSTM** Long short-term memory. [30](#)

**LUT** Look up Tables. [19](#)

**MCU** Micro Controller Unit. [33](#)

**MPI** Monotonic Proportional and integral control. [31](#)

**MRD** Maximum square root difference. [35](#)

**MSE** Mean Squared Error. [45](#)

**NN** Neural networks. [35](#)

**PD** Proportional and differential control. [29](#)

**PID** Proportional, integral, and differential control. [23](#)

**PL** Programmable Logic. [18](#)

**PLA** Polylactic Acid. [20](#)

**PS** Processing System. [18](#)

**PSO** Particle Swarm Optimization. [28](#)

**PSSP** Proactive Synergistic Slip Prevention. [32](#)

**PWM** Pulse Width Modulation. [19](#)

**RAM** Random Access Memory. [38](#)

**RF** Random forest. [36](#)

**RNA** Redes Neurais Artificiais. [31](#)

**RNN** Recurrent neural networks. [34](#)

**ROS** Robot Operating System. [41](#)

**SNR** Signal-to-Noise Ratio. [72](#)

**SoC** System on Chip. [18](#)

**SVM** Support vector machines. [36](#)

**SW** Software. [66](#)

**XADC** Xilinx Analog to Digital Converter. [19](#)

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
<b>1.1</b>	<b>Descrição do Problema</b>	<b>13</b>
<b>1.2</b>	<b>Perguntas de Pesquisa</b>	<b>17</b>
<b>1.3</b>	<b>Justificativa</b>	<b>17</b>
<b>1.4</b>	<b>Objetivos</b>	<b>18</b>
1.4.1	Objetivo geral	18
1.4.2	Objetivo específicos	18
<b>1.5</b>	<b>Contribuições do Trabalho</b>	<b>18</b>
<b>1.6</b>	<b>Organização do Documento</b>	<b>20</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>21</b>
<b>2.1</b>	<b>Controle de Força de Mãos Robóticas</b>	<b>21</b>
2.1.1	Controle de impedância	22
2.1.2	Controlador PID	23
2.1.3	Algoritmo PSO	24
2.1.4	Trabalhos correlatos sobre controle de impedância	25
<b>2.2</b>	<b>Técnicas de Detecção de Deslizamentos</b>	<b>26</b>
<b>2.3</b>	<b>Hardware Reconfigurável</b>	<b>34</b>
<b>2.4</b>	<b>Conclusões do Capítulo</b>	<b>37</b>
<b>3</b>	<b>MATERIAIS E MÉTODOS</b>	<b>39</b>
<b>3.1</b>	<b>Aspectos Gerais</b>	<b>39</b>
3.1.1	Requisitos	39
3.1.2	Especificações	40
<b>3.2</b>	<b>Setup Experimental</b>	<b>41</b>
3.2.1	Componentes	41
3.2.2	Calibração dos sensores	42
<b>3.3</b>	<b>Algoritmo de Detecção de Deslizamento</b>	<b>45</b>
3.3.1	Cálculo da média para cada conjunto de dados de dois sensores	46
3.3.2	Filtro de média móvel para dados	46
3.3.3	Cálculo da derivada de primeira ordem	47
3.3.4	Detector de deslizamento	47
<b>3.4</b>	<b>Co-projeto HW/SW para Sistema de Detecção de Deslizamentos</b>	<b>48</b>
3.4.1	Arquitetura em hardware do filtro de média móvel	49
3.4.2	Arquitetura em hardware da derivada de primeira ordem	49
3.4.3	Arquitetura em hardware do módulo detector de picos	50

3.4.4	Co-projeto de hardware/software do sistema de detecção de deslizamentos	51
<b>3.5</b>	<b>Desenvolvimento do Controle de Impedância</b>	<b>51</b>
3.5.1	Controle de impedância	52
3.5.2	Controle PID	53
3.5.3	Otimização do sistema de controle	53
<b>3.6</b>	<b>Desenvolvimento da Garra Robótica</b>	<b>54</b>
3.6.1	Circuito eletrônico da garra robótica	54
3.6.2	Implementação física	55
3.6.3	Modelos do sistema robótico	56
<b>3.7</b>	<b>Proposta de Validação</b>	<b>57</b>
<b>3.8</b>	<b>Resumo do Capítulo</b>	<b>58</b>
<b>4</b>	<b>RESULTADOS</b>	<b>60</b>
<b>4.1</b>	<b>Modelo de Detecção de Deslizamentos</b>	<b>60</b>
<b>4.2</b>	<b>Controlador de Força e Detecção de Deslizamentos</b>	<b>64</b>
4.2.1	Experimento 1	65
4.2.2	Experimento 2	66
4.2.3	Experimento 3	67
4.2.4	Experimento 4	68
4.2.5	Resumo dos resultados	69
<b>5</b>	<b>CONCLUSÕES</b>	<b>71</b>
<b>5.1</b>	<b>Trabalhos futuros</b>	<b>73</b>
	<b>REFERÊNCIAS</b>	<b>75</b>

# 1 Introdução

## 1.1 Descrição do Problema

No campo da robótica, um dos pontos fundamentais dos estudos é o desenvolvimento de mãos robóticas, essenciais para o avanço tecnológico em diversas áreas, como a medicina, na forma de próteses, a química, para manipulação de elementos radioativos, e a indústria aeroespacial, para manipulação de objetos à distância, entre outras.

As próteses robóticas estão principalmente voltadas para a área da saúde, visando substituir membros perdidos ou auxiliar na reabilitação de pacientes. Nesse contexto, busca-se promover que as mãos robóticas tenham um controle robusto sobre a força aplicada nas tarefas e movimentos de precisão, especialmente no agarre de objetos. Portanto, as mãos robóticas devem possuir um controle específico de força combinado com um controle de posição. Vários estudos propõem o desenvolvimento de controladores baseados em redes neurais e controle de impedância, entre outros. Além disso, é importante destacar a relevância dos sensores utilizados no desenvolvimento de mãos robóticas, que devem fornecer informações adequadas sobre posição, peso, densidade ou composição dos objetos.

Como representação de projetos de desenvolvimento de mãos robóticas, apresenta-se a UnB-Hand, desenvolvida no laboratório LEIA/GRACO da Faculdade de Tecnologia (FT) da Universidade de Brasília. A UnB-Hand foi projetada por meio de um modelo 3D (vide Figura 1) e implementada usando impressão 3D para aplicações de agarre preciso de objetos (vide Figura 2). Cada dedo da mão robótica foi desenvolvido de forma individual, utilizando equações de mecânica analítica, onde os parâmetros do tamanho das falanges, posições angulares máximas, entre outros, foram determinados mediante o uso de algoritmos bioinspirados (PERTUZ; LLANOS, C. H.; MUÑOZ, D. M., 2021).

O projeto da UnB-Hand é composto por motores de corrente contínua e um mecanismo articular que conecta as falanges de cada dedo por meio de uma junta de alumínio. Dessa forma, define-se que os dedos da mão têm um grau de liberdade (DoF - *Degree of Freedom*), possibilitando realizar os movimentos de extensão e flexão. Os dedos indicador e o polegar possuem dois graus de liberdade (DoF), pois também têm movimento de adução e abdução. Assim, o projeto da mão robótica UnB-Hand possui um total de sete graus de liberdade (7-DoF) (PERTUZ; LLANOS, C. H.; MUÑOZ, D. M., 2021).

A UnB-Hand utiliza um controlador de força denominado controle de impedância, no qual o torque alcançado pelos motores é proporcional à força aplicada sobre o objeto. Esse controlador permite realizar simultaneamente o controle de torque e de posição, graças à natureza do controle de impedância. Para controlar o sistema, a mão robótica está

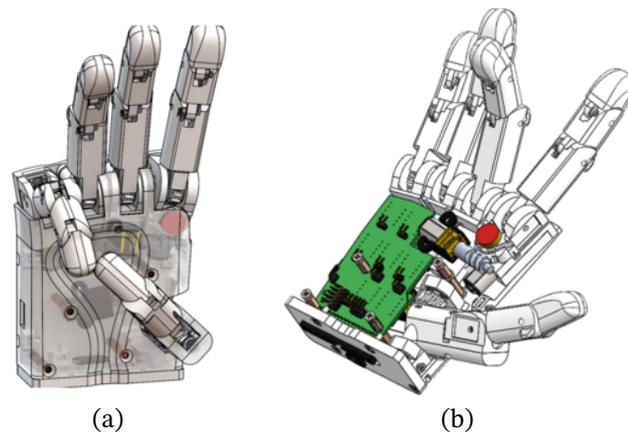


Figura 1 – Projeto mecânico da mão robótica UnB-Hand no software SolidWorks (PERTUZ; LLANOS, C. H.; MUÑOZ, D. M., 2021).

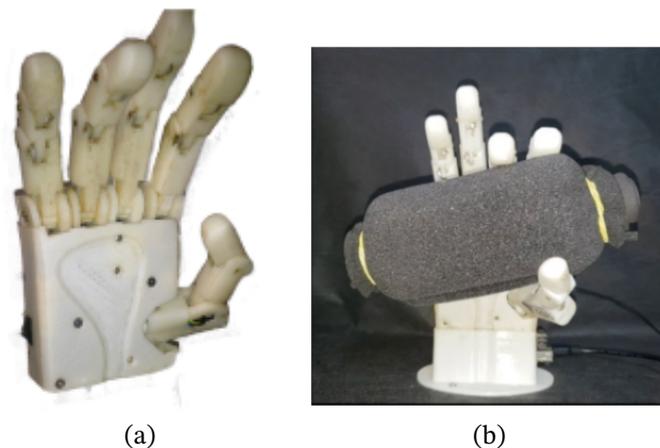


Figura 2 – Projeto de mão robótica UnB-Hand. a) Modelo em impressão 3D da UnB-Hand; b) Aplicação de agarre de objetos com a UnB-Hand (PERTUZ; GHERARDINI et al., 2022).

equipada com sensores de posição rotacionais, localizados na base dos dedos, para medir o deslocamento angular. Além disso, possui sensores de corrente para medir o consumo de corrente dos motores DC, uma vez que, quanto maior o esforço dos motores, por exemplo, para agarrar um objeto, maior será o consumo de corrente. Dessa forma, é possível calcular e estimar o torque dos motores com base no consumo de corrente (PERTUZ; LLANOS, C.; PEÑA, C. A. et al., 2018) (PERTUZ; LLANOS, C.; PEÑA, C. et al., 2021).

Para a implementação do sistema de controle da UnB-Hand, foi realizado um co-projeto Hardware/Software utilizando um SoC (*System on Chip*) FPGA (*Field Programmable Gate Array*) ZYNQ-7020 que inclui um processador ARM Cortex-A9 e um FPGA da família Artix 7 da AMD-Xilinx. O projeto é dividido em duas partes: a parte PS (*Processing System*) e a parte PL (*Programmable Logic*) (PERTUZ; LLANOS, C.; PEÑA, C. A. et al., 2018) (PERTUZ; LLANOS, C. H.; MUÑOZ, D. M., 2021).

A parte PS contém uma interface de comunicação com a lógica programável e recebe os dados estimados de posição angular  $\theta$  e corrente elétrica  $i$ . Em seguida, realiza-se o

cálculo do controle de impedância, onde um computador envia o valor de *set point* de força ao sistema, enquanto os valores dos sensores de posição angular  $\theta$  e da corrente do motor  $i$  são lidos pela FPGA por meio dos pinos XADC. Dessa forma, são calculados os erros em relação ao *set point*, iniciando o cálculo de impedância que resulta em uma sinal de controle PWM (Pulse Width Modulation) para o movimento dos motores (PERTUZ; LLANOS, C.; PEÑA, C. A. et al., 2018) (PERTUZ; LLANOS, C. H.; MUÑOZ, D. M., 2021).

A parte PL, por sua vez, contém um conversor ADC para conversão dos sinais dos sensores, e implementa IPs (Intellectual Property) que realizam um processo de filtragem, composto por dois filtros digitais de corrente e posição angular. O primeiro é um filtro de Kalman, que estima a posição e a velocidade do sensor, e o segundo é um filtro passa-baixa de segunda ordem para o sensor de corrente. Além disso, a parte PL também possui um gerador de sinais PWM para os drivers dos motores. O protocolo de comunicação on chip *AXI4-Lite* é utilizado para comunicação entre o PL e o PS. A implementação dos módulos do PL foi desenvolvida utilizando aritmética de ponto flutuante de 27-bits (PERTUZ; LLANOS, C.; PEÑA, C. A. et al., 2018) (PERTUZ; LLANOS, C. H.; MUÑOZ, D. M., 2021).

Para a estimação de intenções de movimento do usuário, o projeto UnB-Hand adotou uma interface miocinética que permite estimar o deslocamento de ímãs implantados em músculos residuais de um usuário a fim de serem traduzidos em *setpoints* de posição angular de cada dedo robótico (PERTUZ; GHERARDINI et al., 2022). Para tal, em (PERTUZ; GHERARDINI et al., 2022) e (PERTUZ; MENDES et al., 2024) foram implementados modelos de regressão baseados em dados, incluindo regressores lineares e baseados em redes neurais artificiais.

Devido ao grande número de sensores utilizados em uma mão robótica, da complexidade computacional dos modelos de filtragem e de estimação paramétrica, assim como dos modelos de regressão para estimar o deslocamento dos ímãs, o consumo de recursos de hardware é elevado. Atualmente, o projeto ocupa aproximadamente 96% das *Look-up Tables* (LUTs), 35% de flip-flops (FFs) e 60% dos blocos DSPs (*Digital Signal Processing*) disponíveis no SoC FPGA utilizado no projeto UnB-Hand.

Uma questão fundamental no desenvolvimento de mãos robóticas é a detecção e o controle dos deslizamentos de objetos durante o processo de agarre. Um dos principais desafios nesse contexto é a detecção de deslizamentos, não existem muitos sensores capazes de detectar diretamente o movimento de um objeto em sua superfície quando ele é agarrado por uma mão robótica. Por isso, é importante desenvolver uma técnica de baixo custo para a detecção de movimento utilizando as informações fornecidas pelos sensores. É importante destacar que o projeto da UnB-Hand não conta com sensores de força nas superfícies de contato e apenas estima a força de agarre a partir da medição de corrente dos motores. Outro ponto importante, é que a integração de um sistema de controle de força com um sistema de detecção de deslizamentos de objetos, requer de uma metodologia que permita a operação

eficaz e coordenada desses dois sistemas.

As características do projeto UnB-Hand são as seguintes: em primeiro lugar, o projeto utiliza um SoC FPGA com um chip Zynq 7020; em segundo lugar, a mão pode segurar objetos com peso máximo de 1 quilograma; em terceiro lugar, o controle de impedância é baseado na medição de posição e de corrente dos motores para estimar o torque exercido sobre o objeto; em quarto lugar, a mão robótica foi fabricada utilizando impressão 3D com material PLA, como apresentado na Tabela 1. A estimativa de torque a partir das medições de corrente é uma limitação do projeto atual, requerendo do ajuste de modelos que dependem da dinâmica do agarre e das características dos objetos.

<b>Característica</b>	<b>Valor</b>
Plataforma	SoC FPGA Zynq 7020
Peso máximo do objeto	1 Kg
Controle de força	Controlador de impedância
Controle de posição	Controlador PID
Fabricação	Impressão 3D em material PLA
Estimativa de torque	Sensores de corrente
Estimativa de posição	Sensores rotativos SV01
Frequência de operação	100 MHz

Tabela 1 – Características do projeto UnB-Hand.

Como restrição importante relacionada à detecção de deslizamento de objetos com a UnB-Hand é que os modelos utilizados devem apresentar um baixo custo computacional, devendo consumir menos de 3% dos recursos disponíveis na plataforma Zynq 7020 da AMD-Xilinx. Outra limitação do projeto UnB-Hand são as falhas mecânicas, uma vez que os dedos robóticos foram impressos com material PLA, e com o tempo e o desgaste pelo uso, tende a ocasionar folgas mecânicas.

As características do projeto de sistema de detecção de deslizamentos são definidas da seguinte forma: primeiro, implementação do algoritmo de detecção de deslizamentos em um SoC FPGA com chip Zynq 7020; segundo, o algoritmo de detecção de deslizamentos deve ocupar no máximo 4% das LUTs totais do chip Zynq 7020; terceiro, o algoritmo de detecção de deslizamentos deve ser de baixa complexidade; quarto, a frequência de operação deve ser de 100 MHz na implementação em FPGA; quinto, implementação do controle de impedância com sensores de força; sexto, implementação do controle de impedância e do algoritmo de detecção de deslizamentos em uma garra robótica de alumínio utilizando um microcontrolador; sétimo, os sensores utilizados são sensores de força FSR 400 e sensores de posição rotativos SV01; oitavo, o peso máximo dos objetos para a implementação é de até 700 gramas, como apresentado na Tabela 2.

<b>Característica</b>	<b>Valor</b>
Plataforma	Placa SoC FPGA Zynq 7020
Ocupação de LUTs	4% do chip
Frequência de operação	100 MHz
Controle de Força	Controle de impedância
Plataforma Robótica	Garra robótica
Sensores de força	FSR 400
Sensores de posição	Murata SV01
Peso máximo de objetos	700 gramas
Tempo de detecção de deslizamentos	40 ms

Tabela 2 – Características projeto sistema de detecção de deslizamento.

## 1.2 Perguntas de Pesquisa

De acordo com as ideias apresentadas sobre o desenvolvimento de mãos robóticas que implementam sistemas de detecção de deslizamentos de objetos, surgem as seguintes questões: Como implementar um sistema de detecção de deslizamentos de baixo custo computacional? É possível a integração de um sistema de controle de força com um sistema de detecção de deslizamentos?

## 1.3 Justificativa

Para o desenvolvimento deste trabalho, optou-se pela utilização de uma garra robótica para facilitar a elaboração de um algoritmo de detecção de deslizamentos. Na garra robótica, existem apenas dois pontos de contato com o objeto, o que permite uma redução da complexidade do problema. Por outro lado, no caso da mão robótica há múltiplos pontos de contato com os objetos, onde os deslizamentos podem ser analisados.

Nesse contexto, a instrumentação em um ambiente controlado para uma garra robótica é mais simples e acelera o desenvolvimento da proposta de um sistema de detecção de deslizamentos, em comparação com o processo de instrumentação de uma mão robótica.

De acordo com as ideias apresentadas, considera-se implementar o sistema de detecção de deslizamentos baseado em dados (*data-driven*) usando a mesma placa de desenvolvimento SoC FPGA usada no projeto UnB-Hand. Tendo em vista as limitações de espaço e recursos na implementação do projeto UnB-Hand, este trabalho busca desenvolver um modelo de detecção de deslizamentos em tempo real que consuma poucos recursos da FPGA e que seja parametrizável conforme as necessidades do usuário.

## 1.4 Objetivos

### 1.4.1 Objetivo geral

Como objetivo geral deste trabalho, propõe-se desenvolver um controle de força para o agarre de objetos, com um sistema de detecção de deslizamento de objetos, implementado em uma garra robótica comercial usando sensores de força e posição.

### 1.4.2 Objetivo específicos

Com este trabalho espera-se alcançar os seguintes objetivos:

- Desenvolvimento de um modelo de detecção de deslizamentos de objetos baseado em dados e de baixo custo computacional;
- Implementação em SoC FPGA de um sistema de detecção de deslizamentos de objetos;
- Desenvolvimento de controle de força e controle de posição para uma garra robótica;
- Desenvolvimento em microcontrolador de uma garra robótica multissensorial que implementa o controle de força integrado ao sistema de detecção de deslizamentos.

## 1.5 Contribuições do Trabalho

A principal contribuição deste trabalho é o desenvolvimento de um sistema de detecção de deslizamento de objetos de baixo custo computacional. Esse algoritmo de detecção foi construído com base nos dados experimentais obtidos de experimentos de deslizamento de objetos utilizando sensores de força do tipo FSR (Force Sensing Resistors).

Outra contribuição deste trabalho é a implementação do sistema de detecção de deslizamento de objetos em uma placa de desenvolvimento SoC FPGA. Planeja-se integrar esse sistema desenvolvido ao projeto UnB-Hand, razão pela qual o sistema de detecção deve ter um baixo consumo computacional, suficiente para ser integrado ao projeto UnB-Hand, que atualmente não possui um sistema de detecção de movimento de objetos.

Finalmente, foi desenvolvido uma garra robótica multisensorial com a integração a nível de microcontrolador dos sistemas de controle de força para o agarre de objetos e o sistema de detecção de deslizamento de objetos. O objetivo é demonstrar que a proposta é funcional e pode ser testada na mão robótica UnB-Hand.

Foi realizada uma publicação sobre o desenvolvimento e implementação em FPGA de um algoritmo de detecção de deslizamentos de objetos na base de dados *IEEE-Xplore* com o nome de *Hardware Implementation of a Sliding Detection Algorithm for Robotic Hands Using Force Sensors*, o qual foi apresentado no *Symposium on Integrated Circuits and Systems*

*Design* (SBCCI 2023) (MENDOZA-PEÑALOZA; MUÑOZ, D. M., 2023). Destaca-se ainda que o trabalho foi finalista no prêmio do melhor paper da conferência e foi convidado para submeter uma versão estendida na revista *IEEE Design & Test*.

Durante a realização desta dissertação de mestrado, o autor participou das seguintes pesquisas relacionadas com sintonização de controladores usando algoritmos bioinspirados aplicados em robótica móvel educacional:

- M. A. Pastrana; L. H. Oliveira; D. A. Mendes; Douglas. L. Silva; Jose Mendoza-Peñaloza; Daniel M. Muñoz, *Implementation of a PID Controller using Online Tuning Applied to a Mobile Robot Obstacle Following/Avoidance*, 2024 Argentine Conference on Electronics (CAE), [10.1109/CAE59785.2024.10487152](https://doi.org/10.1109/CAE59785.2024.10487152) (PASTRANA; OLIVEIRA; MENDES et al., 2024).
- M. A. Pastrana; L. H. Oliveira; M. S. Santana; V. C. Oliveira; Jose Mendoza-Peñaloza; Daniel M. Muñoz, *Hardware Implementation of a GMDH Controller for Mobile Robot Obstacle Following/Avoidance*, 2023 Latin American Robotics Symposium (LARS), 2023 Brazilian Symposium on Robotics (SBR), [10.1109/LARS/SBR/WRE59448.2023.10332957](https://doi.org/10.1109/LARS/SBR/WRE59448.2023.10332957) (PASTRANA; OLIVEIRA; SANTANA et al., 2023).
- M. A. Pastrana; J. Bautista; Jose Mendoza-Peñaloza; L. H. Oliveira; D. A. Mendes; Daniel M. Muñoz, *Teaching Control Theory using Mobile Robot Obstacle Following/Avoidance with CoppeliaSim and MFO Algorithm*, 2023 Workshop on Robotics in Education (WRE), [10.1109/LARS/SBR/WRE59448.2023.10333042](https://doi.org/10.1109/LARS/SBR/WRE59448.2023.10333042) (PASTRANA; BAUTISTA et al., 2023).
- R. B. Costa; G. Itacaramby; J. Cabral; M. A. Pastrana; Jose Mendoza-Peñaloza; Daniel M. Muñoz, *Design and Construction of a Low-Cost Combat robot*, Brazilian Symposium on Robotics (SBR), 2024 (COSTA et al., 2024).
- M. A. Pastrana; W. H. C. Sanchez; Jose Mendoza-Peñaloza; M. S. Santana; Daniel M. Muñoz, *Improving Control Theory Education Through Face-to-Face Implementation of a Mobile Robotics Methodology*, Workshop on Robotics in Education (WRE), 2024 (PASTRANA; SÁNCHEZ et al., 2024).
- S. D. L. Matheus; M. A. Pastrana; S. D. Lukas; P. R. A. N. Campagnucci; Jose Mendoza-Peñaloza; R. R. L. Benevides; Daniel M. Muñoz, *Comparison of Bioinspired MLP Controllers for Mobile Robots Obstacle Following/Avoidance*, Latin America Conference on Computational Intelligence (LA-CCI), 2024 (SOUSA LUIZ et al., 2024).
- M. A. Pastrana; L. H. Oliveira; J. Bautista; Jesus Pinto-Loper; Jose Mendoza-Peñaloza; Daniel M. Muñoz, *Sub-optimal Perceptron Controller Using WOA Algorithm for Farming Watering Mobile Robot*, Latin America Conference on Computational Intelligence (LA-CCI), 2024 (PASTRANA; OLIVEIRA; BAUTISTA et al., 2024).

## 1.6 Organização do Documento

O documento está composto por outros quatro capítulos. No Capítulo 2, é apresentada a fundamentação teórica e o levantamento do estado da arte dos trabalhos relacionados aos temas específicos de controle de força utilizando mãos robóticas, detecção de deslizamento com mãos robóticas e, por fim, trabalhos que implementam FPGA para o controle de mãos robóticas. No Capítulo 3 apresentam-se materiais e ferramentas utilizados nesta pesquisa, detalhando-se a metodologia de desenvolvimento do algoritmo de detecção de deslizamento, a implementação do sistema de detecção no SoC FPGA e a construção da garra robótica com o controle de força integrado com a detecção de deslizamentos. No Capítulo 4 apresentam-se os resultados da detecção de deslizamentos, os erros calculados na implementação em hardware comparados à implementação em software, o consumo de recursos do sistema e a análise da detecção de deslizamento e controle de força na garra robótica. Finalmente, no Capítulo 5 são apresentadas as conclusões do trabalho, onde se discutem os resultados e possíveis trabalhos futuros.

## 2 Fundamentação Teórica

Este capítulo descreve os conceitos relacionados ao controle de mãos robóticas, dividindo a apresentação em quatro partes: (2.1) implementação de controle de força para mãos robóticas, (2.2) detecção e controle de deslizamento de objetos aplicados a mãos robóticas, (2.3) trabalhos que utilizam FPGAs para o desenvolvimento de algoritmos e aceleração de cálculos de controladores de mãos robóticas, e, finalmente, (2.4) uma seção de conclusões.

### 2.1 Controle de Força de Mãos Robóticas

Nos sistemas de controle de força para mãos robóticas, existem diversas estratégias, como o controle PID, o controle de força baseado em sinergia, o controle de força baseado na rigidez dos objetos e o controle de impedância. A seguir, são descritos diversos trabalhos que empregam diferentes metodologias para o controle de força em mãos robóticas.

No Projeto KITECH-Hand (LEE et al., 2017), desenvolvido pelo Instituto de Tecnologia Industrial de Coreia do Sul, O torque do motor é controlado conforme uma faixa de força potencial aplicada a uma superfície. Nesta metodologia, é implementado um controle PID clássico, responsável por manter a força de contato dentro da faixa especificada, dependendo do *setpoint*. Para o controle da força de contato, a mão robótica é fixada em uma base vertical, onde a força de contato desejada é convertida em movimento articular, enquanto o valor da força de contato é registrado. A força de contato para um dedo foi estabelecida entre 0 e 4 Newton.

No controle de força por meio da sinergia do movimento robótico da mão, a força de contato desejada é definida como a média produzida pelo contato de três dedos (polegar, indicador e médio), que define as posições ótimas das articulações de cada dedo (ORTENZI et al., 2019). Esta estratégia de controle de força é subdividida em duas etapas: Na primeira etapa se realiza o movimento em conjunto dos dedos da mão em direção à configuração de preensão desejada por meio da sinergia de trajetórias. As sinergias de trajetória são definidas como as posições articulares ideais calculadas para realizar um movimento em que os dedos alcançam a posição especificada de maneira eficiente. O movimento é feito até que cada dedo entre em contato com o objeto, aumentando gradualmente a força enquanto se modifica a trajetória sinérgica de forma a alcançar a força média desejada. É fundamental destacar que a medição da força de contato de cada dedo também depende da estratégia de controle sobre a posição relativa da mão e do objeto, pois durante a preensão, o objeto pode se deslocar entre os dedos de acordo com a distribuição da intensidade da força de contato. Na segunda etapa ocorre o controle da força de contato com os dedos desacoplados, permitindo que uma força

de contato seja atribuída a cada extremidade do dedo para obter uma melhor adaptabilidade da mão aos objetos. Quando os dedos entram em contato com o objeto, o controlador passa a mover cada dedo de maneira independente para alcançar a força de contato desejada para cada um.

Na metodologia de controle de força baseadas na rigidez do objeto (YANG et al., 2021), o controlador detecta a rigidez do objeto para implementar a estratégia de preensão utilizando uma câmera. Esta metodologia é composta por três estágios: estágio de contato, estágio de medição e estágio de preensão. No estágio de contato, os dedos se aproximam do objeto até que ocorra o contato. Para isto, a força é estimada na ponta do dedo, e as articulações movem-se para aumentar a força normal até atingir um limite pré-estabelecido, indicando que o dedo fez contato com o objeto. Se a força normal na ponta do dedo permanecer constante, a articulação do dedo é movida até que um ângulo satisfatório seja alcançado. No estágio de medição, a rigidez do objeto é estimada através do movimento da junta, estendendo as molas da mão, o que resulta no aumento da força de contato. Como consequência, se o objeto for deformável, o movimento da mola aumentará a força na ponta do dedo, permitindo estimar a rigidez do objeto. No estágio de preensão, é considerado qual dos dedos deve girar mais para completar a preensão, para determinar qual dos dedos deve se mover, é realizada uma análise utilizando a câmera. O dedo que necessita realizar a maior rotação é identificado, e as articulações de cada dedo movem-se de acordo. Em seguida, estabelece-se uma relação entre a rotação do motor e a rigidez do objeto para definir os parâmetros que permitam realizar uma preensão eficiente.

### 2.1.1 Controle de impedância

Na maioria dos estudos correlatos, é empregado um tipo específico de controle de força para mãos robóticas, conhecido como controle de impedância. Em termos gerais, este método pode ser definido como um controle de força fundamentado nas equações que descrevem um sistema massa, mola e amortecedor (vide Figura 3). Nesse sentido, a força de entrada é responsável por alterar o estado de cada componente do sistema, de modo que a força estabilizadora do sistema corresponda à força desejada pelo controlador. Para tanto, são estabelecidas as constantes de comportamento dos componentes do sistema, onde  $K_M$  representa a massa do sistema,  $K_B$  a constante do amortecedor e  $K_K$  a constante da mola.

A aceleração é representada como  $\ddot{x}$ , a velocidade como  $\dot{x}$ , a distância como  $x$ , sendo  $F$  a força resultante (vide a Equação 2.1).

$$K_M \ddot{x}(t) + K_B \dot{x}(t) + K_K x(t) = F(t) \quad (2.1)$$

O controlador de impedância é composto por dois controladores: o primeiro é um controle de força baseado nas equações de um sistema massa-mola-amortecedor, e o segundo

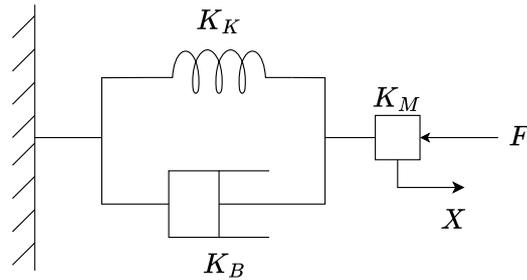


Figura 3 – Sistema de massa, mola e amortecedor.

é um controlador de posição, que geralmente é um controlador PID.

### 2.1.2 Controlador PID

O controle PID é definido como o controlador clássico responsável por alcançar o valor de uma sinal especificada como *setpoint* (vide Figura 4). Isso é realizado mediante o cálculo do erro de entrada e saída do controlador, funcionando como um sistema retroalimentado. O controle PID é composto por três constantes que ajustam a saída do controlador para alcançar o *setpoint*. Primeiramente, a constante  $K_p$  é um valor que multiplica a sinal de erro com o objetivo de aumentar ou diminuir seu valor conforme necessário. Em segundo lugar, temos a constante  $K_i$ , que representa a ganho integral da função e se encarrega de reduzir o erro abaixo da curva em relação ao *setpoint*, ou seja, reduzir o erro em estado estacionário. Por fim, a constante  $K_d$  representa o fator derivativo do sistema, responsável por reduzir a variação do sinal em relação aos valores anteriores. A Equação do controlador PID para controle de sistemas é apresentada em 2.2.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{d}{dt} e(t) \quad (2.2)$$

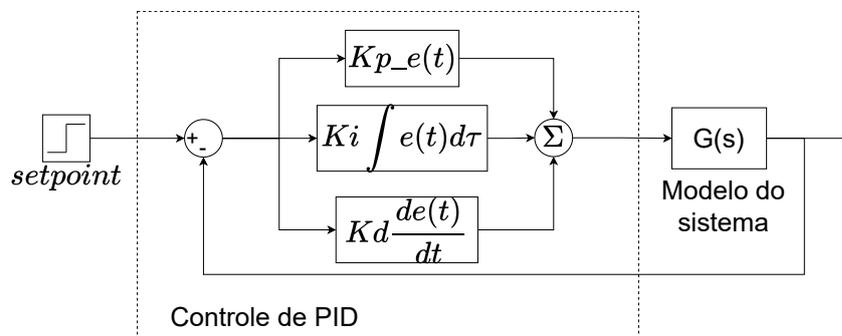


Figura 4 – Modelo de controlador PID.

A representação da Equação 2.2 no domínio da frequência não é causal. Para solucionar esse esse problema e buscando uma implementação numérica, costuma-se usar um

filtro  $f$  na ação derivativa, representado pela Equação 2.3 (PRIYAMBADA; KUMAR, D. M.; KUMAR, M. P., 2018) (KNOSPE, 2006) (WITTENMARK; ÅSTRÖM, 2001).

$$f(t) = (1 - \alpha)f(t - 1) + \alpha e(t) \quad (2.3)$$

$$\frac{de(t)}{dt} = \frac{f(t) - f(t - 1)}{\Delta t} \quad (2.4)$$

### 2.1.3 Algoritmo PSO

O algoritmo PSO (*Particle Swarm Optimization*) é um algoritmo bioinspirado baseado no comportamento natural de partículas em enxame, onde cada partícula representa um ser vivo que pertence a um grupo dentro de um ecossistema. Todas as partículas trabalham em conjunto para solucionar um problema.

---

#### Algorithm 1 Pseudocódigo do algoritmo PSO.

---

```

1: function PSO( $S, N, c_1, c_2, Max_{iter}, thres, X_{min}, X_{max}$ )
2:   Inicializar o enxame;
3:    $iter = 1$ 
4:   repeat
5:     for  $i$  do 1  $S$ 
6:       if  $f(x) \leq f(y_i)$  then
7:          $y_i \leftarrow x$ ;
8:       end if
9:     end for
10:    calcular  $y_s$  utilizando os valores otimizados  $S, f(y_i)$ 
11:    for  $i$  do 1  $S$ 
12:      for  $j$  do 1  $N$ 
13:         $v_{ij}^{(t+1)} \leftarrow w v_{ij}^{(t)} + c_1 U_{1j} (y_{ij}^{(t)} - x_{ij}^{(t)}) + c_2 U_{2j} (y_{sj}^{(t)} - x_{ij}^{(t)})$ 
14:         $x_{ij}^{(t+1)} \leftarrow x_{ij}^{(t)} + v_{ij}^{(t+1)}$ 
15:      end for
16:      if  $x_{ij}^{(t+1)} > X_{max}$  then
17:         $x_{ij}^{(t+1)} \leftarrow X_{max} - (x_{max} - x_{min}) * rand$ 
18:      end if
19:      if  $x_{ij}^{(t+1)} < X_{min}$  then
20:         $x_{ij}^{(t+1)} \leftarrow X_{min} + (x_{max} - x_{min}) * rand$ 
21:      end if
22:    end for
23:     $iter = iter + 1$ 
24:  until ( $f(y_s) < threshold$ ) || ( $Iter \leq Max_{iter}$ )
25:  return melhor partícula  $y_s$  e seu valor otimizado  $f(y_s)$ 
26: end function

```

Melhor global  
 Atualizar as partículas

---

De acordo com as Equações 2.5 e 2.6, o algoritmo PSO é composto por um número de partículas  $i$  e um número de dimensões  $j$ . A velocidade da partícula no espaço de busca é representada por  $v$ , enquanto  $w$  é o fator de inércia de deslocamento da partícula. O coeficiente cognitivo individual da partícula é denotado por  $c_1$ , e o coeficiente cognitivo social é representado por  $c_2$ . Os valores  $U_1$  e  $U_2$  representam uma distribuição uniforme das partículas dentro do sistema. Por fim,  $y_i$  representa a melhor posição da partícula atual e  $y_s$  corresponde à melhor posição entre todas as partículas, enquanto  $x_i$  é a posição da partícula atual que será deslocada. O Algoritmo 1 mostra o pseudo-código do PSO.

$$v_{ij}^{(t+1)} = wv_{ij}^{(t)} + c_1 \overbrace{U_{1j}(y_{ij}^{(t)} - x_{ij}^{(t)})}^{\text{componente cognitivo}} + c_2 \overbrace{U_{2j}(y_{sj}^{(t)} - x_{ij}^{(t)})}^{\text{componente social}} \quad (2.5)$$

$$x_{ij}^{(t+1)} = x_{ij}^{(t)} + v_{ij}^{(t+1)} \quad (2.6)$$

#### 2.1.4 Trabalhos correlatos sobre controle de impedância

Estudos relacionados empregam o controle de impedância para manipular uma mão robótica com dois dedos na tarefa de agarrar objetos. Estes estudos descrevem o controle de impedância como um controlador composto por um loop interno, o qual inicialmente realiza o controle de posição e, em seguida, um loop de controle de impedância externo que calcula a posição de acordo com a força desejada. Essa estrutura do controlador pode ser visualizada na Figura 18.

Estudo que descreve o agarre firme é obtido controlando a posição dos dedos da mão robótica. Após o cálculo interno do controle de impedância, obtém-se uma soma vetorial das posições dos dedos. Esse dado passa por um filtro de correção que finalmente designa a posição dos dedos de acordo com uma dinâmica em tempo real planejada. A etapa seguinte utiliza os dados de posição obtidos, entregando-os como *setpoint* a um controlador de posição PD, para que os dedos possam alcançar as posições desejadas. Em seguida, iniciam-se os cálculos para reduzir os erros de posição em relação às posições calculadas, o que, por sua vez, permitirá alcançar a força de preensão desejada. Dessa forma, as características dinâmicas desejadas são obtidas nos dois dedos de preensão do robô. É importante destacar que a eficácia do controlador está intrinsecamente relacionada à precisão do controlador de posição e aos parâmetros do filtro de impedância (WANG, X. et al., 2011).

Outros estudos empregam mãos robóticas em tarefas de serviço humano, como a realização de massagens, o que traz consigo o desafio de controlar a força enquanto se mantém a mesma posição na ponta do dedo. Para este caso, também é adotado o controle de impedância, sendo este definido como um controlador híbrido no qual o controle de força e o controle de posição operam simultaneamente na mão robótica. Desta forma, quando a ponta do dedo encontra um obstáculo, como um osso, ela desliza suavemente contra ele, evitando causar danos e funcionando como um tipo de movimento de mola amortecida. Neste caso, o controle de impedância é fundamentado nas mesmas equações que descrevem um sistema de massa, mola e amortecedor, com o objetivo de controlar a posição, velocidade e aceleração da ponta do dedo do robô (TERASHIMA et al., 2008).

Em estudos relacionados que empregam o controle de impedância em mãos robóticas, destaca-se uma pesquisa envolvendo uma mão robótica com controle bilateral (KOBAYASHI et al., 2011). Nesse estudo, é utilizado um método para sincronizar a intenção de força a partir de dois dispositivos, sendo o controle de intenção atuando como mestre e o controle de força

como escravo. As características da mão robótica permitem que a sensação de apreensão seja facilmente transmitida ao operador. O sistema de medição é realizado por meio de sensores de pressão, de forma que a pressão exercida por uma carga é convertida em um valor de tensão, cujo aumento está diretamente relacionado ao tamanho da carga medida pelo sensor.

A Tabela 3 apresenta um resumo das técnicas de controle de força com aplicação em mãos robóticas, indicando também o tipo de sensor de força usado e o tipo de unidade de processamento. Este último aspecto será discutido posteriormente.

Autor, ano	Técnica de controle de força	Tipo de sensor	Processador
Terashima, K. (2008) (TERASHIMA et al., 2008)	Controle de impedância híbrida.	6-axis force sensor	microcontrolador, driver de servo e PC
Wang, X. (2011) (WANG, X. et al., 2011)	Algoritmo de controle de força e impedância de aderência ativa.	Sensor de força	Microcontrolador e PC
Kobayashi, D. (2011) (KOBAYASHI et al., 2011)	Controle de impedância bilateral.	Sensor de pressão	Microcontrolador e PC
Lee, D.H. (2017) (LEE et al., 2017)	Força aplicada estimada através de parâmetros estabelecidos do controlador PID.	Sensor de força de 6 eixos IFS53M31A25, Sensor de posição MURATA SV01.	Unidade microcontroladora MCU (STM32F103C8)
Ortenzi, D. (2019) (ORTENZI et al., 2019)	Controle de contato de força baseado em sinergia	Sensor de força óptica (OMD 40N Optoforce)	Microcontrolador e PC
Yang, H. (2021) (YANG et al., 2021)	Controle de aderência com detecção de rigidez e controle cinemático	Sensor rotativo (Bourns 3382G-1-103G), Sensor de força (Interlink Electronics FSRTM400).	Microcontrolador (Arduino Nano)

Tabela 3 – Trabalhos relacionados que realizam controle de força em mãos robóticas.

## 2.2 Técnicas de Detecção de Deslizamentos

No âmbito do estudo de mãos robóticas, busca-se um controle eficiente destas. Um dos principais desafios encontrados reside na detecção de deslizamento e o controle de força após a detecção. Considerando isso, diversos estudos têm apresentado propostas para mitigar o deslizamento e a instabilidade dos sistemas de controle devido ao deslocamento do objeto, conforme apresentado na Tabela 4.

Khin *et al.* (KHIN et al., 2021) desenvolveram uma mão robótica antropomórfica macia com sensores de força e um modelo de apreensão baseado em redes neurais. O controlador utiliza três redes neurais para avaliar a estabilidade do objeto: Primeiro, rede LSTM com 2 camadas e 30 neurônios cada para memória de curto prazo. Segundo, rede LSTM Siamesa com 2 camadas e 30 neurônios cada para comparar duas entradas simultaneamente. Terceiro, rede Tripla com 2 camadas de 30 neurônios e uma camada adicional para integrar três conjuntos de dados. Equipado com um acelerômetro IMU de três eixos e sensores de força, o sistema registra transições mecânicas a 100 Hz. Sensores com eletrodos e tecido piezorresistivo garantem alta sensibilidade para detectar deslizamentos. O sistema avalia a variação dos eixos IMU e classifica o movimento do objeto em: a) estável, b) inclina-se ou move-se levemente, c) escorrega do aperto.

Outras pesquisas propõem metodologias para controladores de mãos robóticas especializadas em tarefas de manipulação. Essas metodologias permitem a transferência de comportamentos simulados para o modelo real do robô. O ajuste do controlador de estabilidade de preensão é realizado paralelamente em cada dedo, enquanto os parâmetros de manipulação produzem trajetórias no espaço articular, equilibrando manipulação e estabilidade. O controle hierárquico é dividido em dois níveis: primeiro com controladores de estabilização de pegada em cada dedo; e segundo, um planejador de trajetórias de movimento dos dedos. O controle não hierárquico ocorre quando esses níveis não são seguidos. A estabilidade dos controladores é desenvolvida em cada dedo, baseando-se nos sinais dos sensores táteis. Estabilizadores acoplados evitam deslizamentos, mantendo o objeto estável com uma pegada de múltiplos dedos. O robô utiliza um preditor de deslizamentos baseado em aprendizado tátil, criando uma janela de predição que, se ultrapassar um limite, indica um possível deslizamento (VEIGA; AKROUR; PETERS, 2020).

Nos trabalhos relacionados, são encontradas metodologias para detecção de deslizamento em mãos robóticas por meio da utilização de sensores ópticos baseados na reflexão da luz (NAKAGAWA-SILVA; REDDY SUNKESULA et al., 2018). Através do método de controle MPI (Monotonic PI), o sistema é capaz de capturar eventos de deslizamento de objetos, onde foi realizada uma caracterização em resposta ao movimento translacional de um objeto dentro de um campo de visão definido. O controlador de supressão de deslizamento é implementado utilizando sensores ópticos. O controlador MPI é modificado para operar em sistemas não lineares. A vantagem deste tipo de controle reside no fato de não permitir que a força aplicada aos dedos diminua após a supressão do deslizamento, pois o controle transita de uma ação predominantemente proporcional para uma ação integral. Em geral, os controladores apresentam uma resposta proporcional à velocidade e à distância de deslizamento, mas o controle MPI fornece uma melhor resposta para deslizamento contínuo, uma vez que a componente integral aumenta ao longo do tempo durante o evento de deslizamento.

Pesquisas relacionadas à detecção de deslizamento usando técnicas de controle utilizando Redes Neurais Artificiais (RNA) para classificar o deslizamento e detectar sua direção durante a preensão de um objeto por mãos robóticas foram realizadas em (ABD et al., 2018). Nesse trabalho, a detecção da direção de deslizamento é utilizada para aprimorar os reflexos da mão robótica *Shadow Dexterous Hand* com sensores táteis *BioTac*, aumentando a eficiência do trabalho do robô em cooperação com seres humanos. A mão foi posicionada sobre uma superfície de vidro inclinada a 45 graus e em cada teste a mão foi abaixada a 10 mm/s até fazer contato com a superfície, atingindo um estado estacionário. Em seguida, um deslizamento foi induzido por um movimento linear em uma direção sul ou oeste específica por 10 segundos. A classificação de deslizamentos foi realizada em tempo real utilizando a plataforma Matlab/Simulink, na qual foi projetada uma rede neural feedforward, contendo uma camada oculta composta por 10 neurônios. A rede neural foi desenvolvida para determinar seis classes: quatro direções de deslizamento (Norte, Sul, Leste e Oeste) e os

estados de contato e sem contato. Embora a detecção de deslizamento mostrou-se viável com essa metodologia, os autores afirmam que alguns deslizamentos com diferentes coeficientes de atrito e de velocidades não foram detectados.

No trabalho de Zaidi *et al.* (ZAIDI *et al.*, 2017), foram propostos modelos para preensão de objetos 3D deformáveis com mãos robóticas multiarticuladas, focando na detecção de deslizamento. Usando uma mão robótica com três dedos rígidos, o modelo de elasticidade dos objetos é representado como um sistema não linear de massa-mola, com malha tetraédrica. A deformação é calculada monitorando a posição dos nós da malha via a equação de movimento, baseada na segunda lei de Newton.

O modelo de contato segue uma sequência de etapas, calculando a força de contato a partir da velocidade relativa entre os dedos e o objeto. A dinâmica do modelo atualiza a forma e a área de contato deformada em cada iteração, testando a estabilidade da aderência. A transição entre adesão e deslizamento é explicada por um modelo com duas variáveis,  $u$  e  $v$ , que representam o deslocamento tangencial elástico e o deslocamento deslizante, respectivamente. Esses valores voltam a zero quando há perda de contato.

Os modelos de contato consideram a força normal e os modos de força tangencial devido ao atrito. Um sistema de mola-amortecedor conecta o solo à ponta do dedo, permitindo o movimento dinâmico do ponto de contato. A plataforma robótica utilizada para validação inclui dois braços Kuka LWR4+ e uma Shadow Dexterous Hand com 19 graus de liberdade, equipada com sensores de toque BioTac que fornecem dados sobre força, textura e temperatura do contato (ZAIDI *et al.*, 2017).

Existem estudos relacionados ao objetivo de prevenir deslizamentos que descrevem o que há duas abordagens principais para prevenir deslizamentos durante a preensão de objetos com mãos robóticas: reativa e proativa. O método reativo detecta e corrige deslizamentos após sua ocorrência, enquanto o método proativo ajusta a força de preensão com base em previsões para evitar deslizamentos antes que eles ocorram. A detecção de deslizamentos geralmente utiliza sensores táteis e técnicas avançadas de processamento e controle de sinais (KENT; ENGBERG, 2017).

A Prevenção Proativa de Deslizamento Sinérgico (PSSP) é uma abordagem que coordena o movimento de múltiplas articulações com uma única entrada de controle para evitar deslizamentos. Este controlador aplica uma trajetória polinomial compartilhada por todas as articulações para melhorar a aderência sinérgica em próteses robóticas. A plataforma utilizada para esta abordagem foi a Shadow C6M Dexterous Hand, equipada com 24 articulações e 20 DOF. O modelo considera os dedos indicador, médio e polegar, com sensores de giroscópio e efeito Hall para medir a velocidade angular e as articulações, respectivamente, na base da palma da mão robótica.

A eficácia do PSSP foi testada com quatro objetos diferentes e seis configurações

de controle para compensar forças centrípetas, transversais e de coriolis. A configuração inicial determina a postura da mão sem compensação ativa, enquanto as configurações subsequentes ajustam a força de prensão para manter a aderência do objeto, considerando as forças inerciais e o efeito coriolis durante a manipulação (KENT; ENGBERG, 2017).

Trabalhos relacionados propõem um sistema de detecção de contato e deslizamento para mãos robóticas usando sensores piezoelétricos, validando o sistema quantitativamente através de experimentos e controle preliminar (ZHUO et al., 2015). Para adquirir sinais táteis, uma mão robótica agarrou três tipos de garrafas: vidro, plástico rígido e aço inoxidável. Os sinais foram registrados durante as etapas de tocar, segurar e soltar o objeto. O deslizamento foi induzido por carga externa, como puxar ou adicionar água, para capturar dados de deslizamento, enquanto a unidade de controle gerenciava as condições de partida e parada do motor com base nos sinais dos sensores. Limiares para reconhecimento de prensão e deslizamento foram definidos experimentalmente.

O sinal piezoelétrico gerado pela variação da força foi detectado por um sensor piezoelétrico, com grande pico observado ao tocar e soltar o objeto, mas sem sinal significativo para uma prensão estável. A unidade sensora, montada em um braço de garra robótica, convertia sinais de carga em tensão, aplicava filtros passa-baixo e notch para remoção de interferências, e amplificava o sinal. A coleta de dados era realizada a 100 Hz com uma placa USB-0816, e o processamento dos sinais era feito em um computador para análise e controle (ZHUO et al., 2015).

Trabalhos relacionados investigam o deslizamento durante a prensão de objetos, focando em soluções para melhorar a robustez das mãos robóticas (ABDULKAREEM S. AL-SHANOON; AHMAD; HASSAN, 2016). Foi desenvolvida uma mão robótica com sensores de pressão tátil e medidores de posição para aprimorar o controle durante eventos de deslizamento. O algoritmo de controle utiliza um sensor de deslizamento e um encoder rotativo para ajustar a força de prensão com base na distância de deslizamento, semelhante à Lei de Hooke (vide Equação 2.7).

$$F = -k\Delta x \quad (2.7)$$

O sistema calcula a força necessária para re-agarrar o objeto com base na distância de deslizamento, onde cada pulso do encoder representa 0,03268 cm de movimento. A mão robótica, projetada para agarrar objetos de diferentes pesos, possui uma pinça de dois dedos com abertura máxima de 60 mm e é movida por um servomotor DC com um grau de liberdade no eixo X, suportando até 3,18 N. A posição do motor é controlada por um MCU que ajusta os dedos automaticamente após a detecção do objeto via infravermelho.

Outros estudos focam na previsão de deslizamento em mãos robóticas usando sensores táteis e um classificador de aprendizado supervisionado (VEIGA; PETERS; HERMANS,

2018). A abordagem propõe um modelo que usa informações de pressão e deformação da ponta do dedo para prever futuros eventos de deslizamento. O classificador analisa os dados em dois estados: o estado inicial, onde os dados brutos do sensor são usados, e um estado futuro, onde a predição é feita para  $Tf \geq 0$ .

O preditor de deslizamento é integrado em um controlador reativo que ajusta a força de prensão com base na previsão de deslizamento, utilizando o modelo de atrito de Coulomb para manter a força dentro do cone de atrito. O controlador aumenta a velocidade normal ao contato para prevenir deslizamentos e opera a 100 Hz, classificando o estado do sensor como *slip* ou *-slip*. A plataforma experimental é um braço robótico Mitsubishi PA-10 com sete graus de liberdade e um sensor BioTac montado para manipulação de objetos.

Outros estudos exploram a detecção de deslizamento incipiente, uma técnica para manter a estabilidade do controlador em sistemas de aderência (WATERS; JONES; ALAZMANI; CULMER, P. R., 2022). Este método é baseado na curva do perfil de aderência, detectando movimentos diferenciais entre seções independentes de materiais deformáveis ou lubrificados. A ideia é identificar deslizamentos incipientes ao monitorar o deslocamento entre seções externas e a seção central de um material.

Um algoritmo automatizado foi desenvolvido para detectar deslizamentos incipientes durante a retração do tecido, aumentando a força de corte como no pior cenário de deslizamento. Foram realizados testes para avaliar os parâmetros operacionais e as características do sensor, variando forças de fixação, velocidades de retração e rigidez do tecido.

Outros trabalhos exploram o uso de redes neurais profundas (DNN) para detectar deslizamentos com um sensor tátil GelSight e uma câmera montados em uma pinça de um braço robótico (LI; DONG; ADELSON, 2018). A DNN combina uma CNN (Rede Neural Convolutiva) para extrair características das imagens da câmera e uma RNN (Rede Neural Recorrente), especificamente a LSTM (Long Short Term Memory), para processar informações sequenciais e prever deslizamentos, alcançando uma precisão de 88,03%.

O modelo DNN processa sequências de oito imagens capturadas por duas câmeras e dados do sensor de toque. A CNN extrai características das imagens, e a LSTM analisa as características ao longo do tempo. A rede é treinada no TensorFlow usando uma GPU NVidia GeForce Titan X. O braço robótico UR5 com uma pinça paralela WSG 50 é utilizado para testar a prensão de objetos.

Trabalhos recentes exploram abordagens bioinspiradas para a detecção e prevenção de deslizamentos durante a manipulação de objetos com mãos robóticas (NAKAGAWA-SILVA; THAKOR et al., 2019). Esses estudos propõem um sistema de controle baseado em sensores de deslizamento, que captura o movimento do objeto e ajusta a força de prensão para evitar deslizamentos.

O método de limite adaptativo dinâmico ajusta a detecção de deslizamento para

diferentes superfícies, utilizando um sistema de controle em malha fechada. O sistema é reflexivo, aumentando a força de prensão com base em eventos de deslizamento detectados pelo sensor. O sinal do sensor passa por um filtro passa-baixo e é analisado para gerar picos de sinal, que são usados pelo controlador MPI (Monotonic Proportional-Integral) para ajustar a força de prensão. A abordagem é inspirada em princípios naturais e avaliada para deslizamentos dinâmicos com variações de peso do objeto.

Hou et al. (HOU et al., 2023) apresentam um novo modelo de contato entre o objeto e a ponta do dedo para determinar o ponto de deslizamento crítico (CSP), utilizando a banda de sinal de acordo com os dados mapeados pelo sensor de fibra óptica Bragg Grating e a força normal detectada. Em seguida, é projetado um sistema de detecção de deslizamentos baseado na transformada wavelet e na diferença da raiz quadrada máxima (MRD), visando criar um sistema de previsão de deslizamentos em tempo real.

Wang et al. (WANG, Q. et al., 2024) apresentam um método de detecção de deslizamentos incipientes usando sensores táteis do tipo PapillArray, um sensor elástico que estima a força normal quando pressionado contra uma superfície. Foi desenvolvido um algoritmo de detecção de deslizamentos incipientes utilizando redes neurais (NN), cujo objetivo principal é classificar entradas em duas categorias: *deslizamento incipiente* e *outros*. O estado *outros* se refere aos estados de contato que não são deslizamentos incipientes, como grandes deslizamentos ou estados estacionários. Para melhorar a eficiência da captura de dados, foi usada uma rede neural recorrente (RNN), e a inclusão de uma base de dados histórica de deslizamentos ajudou a melhorar a detecção do algoritmo.

Sui et al. (SUI et al., 2024) desenvolvem um novo método para avaliação do grau de deslizamento incipiente e uma estratégia de controle de força de agarre com robôs paralelos. O deslizamento incipiente ocorre quando uma parte da região de contato começa a deslizar, enquanto outra parte permanece estática. Para isso, foi desenvolvido um sensor tátil que divide a área de contato em microelementos, permitindo decompor a direção da força normal e perpendicular. Utilizando a estimativa de fricção de Coulomb, é possível determinar os estados de aderência e deslizamento para os microelementos com base na relação entre a magnitude da força distribuída e o coeficiente de fricção, o que permite avaliar o grau de deslizamento incipiente.

Zhou et al. (ZHOU et al., 2024) apresentam uma nova pinça robótica com sensores táteis, que conta com um controle de força em laço fechado, permitindo controlar a força de agarre, detectar deslizamentos e compensar movimentos. As informações provenientes dos sensores são utilizadas para criar um modelo de aprendizado profundo com alta precisão, usando uma rede LSTM (Long Short-Term Memory) aplicada a uma rede neural recorrente (RNN). A rede LSTM, com 5 camadas, é utilizada para processar os dados dos sensores no processo de agarre, utilizando 12 dimensões para 12 sensores no robô, e a camada LSTM possui 128 unidades escondidas, seguida por uma camada de abandono e uma camada

totalmente conectada. A camada de saída gera 11 resultados, correspondendo a 10 categorias para os objetos usados no reconhecimento e um para o estado de agarre vazio.

Hu et al. (HU et al., 2024) apresentam um algoritmo de detecção e classificação de deslizamentos utilizando sensores táteis. Foram utilizados quatro classificadores com aprendizado de máquina para identificar os estados do agarre da pinça robótica: máquinas de vetores de suporte (SVM), floresta aleatória (RF), K-vizinhos mais próximos (KNN) e árvores de decisão (DT). Um conjunto de dados com aproximadamente 14.000 amostras de diferentes objetos foi usado para ajustar a precisão dos classificadores, obtendo como melhor resultado uma precisão de 71,69% na aplicação de retirar um livro de uma prateleira.

Romeo et al. (ROMEO; ZOLLO, 2020) realizam uma investigação do estado da arte sobre diversas metodologias de detecção de deslizamentos. O estudo analisa a percepção humana na detecção de deslizamentos de forma fisiológica, definindo estabilidade e o sistema nervoso do corpo. Em seguida, descrevem modelos de detecção baseados em análise de fricção, vibração e classificações de sensores como piezoeletricidade e sinais acústicos. O trabalho também explora técnicas de transformação de dados, como a transformada rápida de Fourier (FFT), filtros passa-banda (filtro BP), filtros Hodrick–Prescott (filtro HP) e filtros passa-baixa (filtro LP). Além disso, são descritos métodos físicos de detecção de deslizamentos, utilizando sensores ópticos, de velocidade e aceleração, entre outros, e outros métodos como diferenciação com sensores táteis e aprendizado de máquinas com redes neurais.

Autores	Deteção de deslizamento	Sensor	Controlador
Zhuo, Q. (2015) (ZHUO et al., 2015)	Deteção de deslizamento com base na deformação do objeto	Unidade de sensor piezoelétrico	Placa de aquisição de dados (USB-0816) e PC
Al-Shanoon, S. (2016) (ABDULKAREEM S. AL-SHANOON; AHMAD; HASSAN, 2016)	Algoritmo de realimentação automática para deteção de deslizamentos	Sensor de pressão de toque FSR (Tekscan, modelo A104)	Unidade microcontroladora (MCU)
Kent A. (2017) (KENT; ENGBERG, 2017)	Sinergia proativa para deteção de deslizamentos (PSSP)	Sensor de efeito Hall, Giroscópio (IDG-300, InvenSense)	Controlador de Dexterous Hand Shadow Robot
Zaidi, L. (2017) (ZAIDI et al., 2017)	Modelo de deteção de deslizamentos e controle baseado em WDLS (Weighted Damped Least-Squares)	Sensor de toque BioTac	Controlador de Dexterous Hand Shadow Robot
Veiga, F. e Peters, J. (2018) (VEIGA; PETERS; HERMANS, 2018)	Controle reativo e de estabilidade para deteção de deslizamentos com controle PD	Sensor de toque BioTac	Controlador de braço robótico Mitsubishi PA-10
Abd, M. (2018) (ABD et al., 2018)	Classificação de estados deslizantes usando redes neurais artificiais (ANN)	Sensor de toque BioTac	Controlador de Dexterous Hand Shadow Robot
Nakagawa, S. e Reddy, S. (2018) (NAKAGAWA-SILVA; REDDY SUNKESULA et al., 2018)	Supressão de eventos de deslizamento usando controle PI monotônico (MPI)	sensor óptico refrativo OPR5005	Microcontrolador STM32F103C8T6 e PC
Li, J. (2018) (LI; DONG; ADELSON, 2018)	Redes neurais profundas (DNN) para deteção de deslizamentos com visão computacional	Sensor óptico e de toque GelSight, Webcam	Placa de aquisição de dados e PC
Nakagawa, S. e Thakor, V. (2019) (NAKAGAWA-SILVA; THAKOR et al., 2019)	Limite de deteção bioinspirada para supressão de deslizamentos com control MPI	sensor de toque biométrico, Sensor óptico (OPR 5005- TT Electronics)	Driver-módulo I2C e PC
Veiga, F. e Akrou, R. (2020) (VEIGA; AKROUR; PETERS, 2020)	Controle hierárquico baseado em ANN com estabilizadores de deslizamento	Sensor de toque BioTac	Microcontrolador Allegro Hand (Wonik Robotics GmbH)
Romeo, A. (2020) (ROMEO; ZOLLO, 2020)	Metodologias de deteção de deslizamentos baseadas em: fricção, vibração, transformação de sinal, propriedades físicas e alternativas complementares	Sensores de força/torque, táteis, ópticos e de deslizamento	micro controlador y PC
Khin, M. (2021) (KHIN et al., 2021)	Classificação e deteção de deslizamentos com redes neurais de LSTM (Long Short-Term Memory)	Sensor IMU (Inertial Measurement Unit)	microcontrolador e PC
Waters, I. (2022) (WATERS; JONES; ALAZMANI; CULMER, P. R., 2022)	Razão da velocidade entre regiões de medição usando sensores táteis	Sensor de toque indutivo, Sensor de efeito hall de 3 eixos (MLX90393)	Microcontrolador (Tensy 3.6), Controlador (MyRIO, National Instruments)
Hou, Q. (2023) (HOU et al., 2023)	Algoritmo de monitoramento baseado na transformada wavelet para prevenção de pontos críticos de deslizamento	Sensor de fibra óptica com grade de Bragg em fibra de quartzo	Módulo de sinal e computador
Hu, X. (2024) (HU et al., 2024)	Aprendizagem de máquina combinada com taxa de variação da entropia para deteção e prevenção de deslizamentos	Sensor de toque Gel-sight	Plataforma do braço robótico UR5e e PC com processador Intel Core i5-10210U
Wang, Q. (2024) (WANG, Q. et al., 2024)	Desenvolvimento de algoritmo para deteção de deslizamentos incipientes utilizando redes neurais	Sensor tátil PapillArray	Controlador de garra de dois dedos RG2 OnRobot e PC
Suí, R. (2024) (SUI et al., 2024)	Novo método de avaliação do grau de deslizamentos incipientes baseado na análise de fricção	Sensor de toque	Controlador de garra robótica e computador
Zhuo, Z. (2024) (ZHOU et al., 2024)	Controle de força para agarre baseado em PID, com deteção de deslizamentos e compensação utilizando aprendizado com RNN e redes LSTM	Sensor de toque capacitivo	Placa Arduino (Mega 2560) e computador

Tabela 4 – Trabalhos relacionados com deteção de deslizamentos em mãos robóticas.

## 2.3 Hardware Reconfigurável

Um Sistema em Chip (SoC) é uma arquitetura que integra um FPGA (*Field Programmable Gate Array*) com um processador e diversos periféricos em um único chip. Dessa forma, o SoC permite a implementação de aplicações que combinam a facilidade de programação de um processador com a flexibilidade e o desempenho potencial de um FPGA. Destaca-se a utilização do chip FPGA SoC dual-core Zynq 7020 (vide Figura 5) nos projetos relacionados.

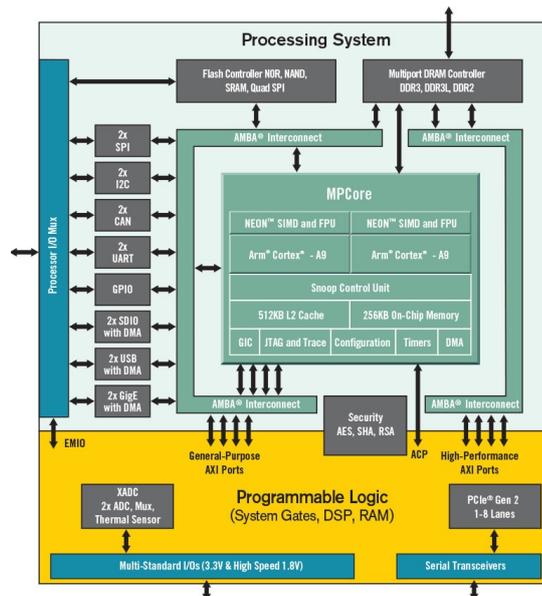


Figura 5 – Chip Zynq 7020 dual-core SoC FPGA (AMD-XILINX, 2024).

Na nomenclatura da AMD Xilinx, os SoCs Zynq 7020 são compostos por duas partes PS (*Processor System*) e PL (*Programmable Logic*). O PS inclui um ou dois cores ARM Cortex A9, controladores de memória, controladores de interrupção, periféricos e portas. Por outro lado, possuem a PL, que contém blocos lógicos configuráveis (CLBs), processadores de sinais digitais (DSP), blocos de memória RAM e blocos de entrada e saída (I/O) (vide Figura 6). Além disso, os SoCs possuem interconexões entre ambos os sistemas, como canais de interconexão de alta velocidade e protocolos de comunicação que conectam os componentes dentro do SoC, permitindo a comunicação entre a CPU, os periféricos e a lógica programável. Estes protocolos são conhecidos como AXI (*Advanced eXtensible Interface*). Uma visão geral dos componentes pode ser encontrada na Tabela 5.

Levando em consideração as potenciais aplicações dos SoCs, é possível compreender a utilidade e a aplicação deles na área de controle de mãos robóticas. Em particular, o uso de SoCs e FPGAs é frequentemente direcionado para a aceleração de processos de controle, execução de técnicas de classificação ou modelos de filtros com alta complexidade computacional. Na Tabela 6 são apresentados alguns trabalhos relacionados ao uso de SoCs e FPGAs aplicados em mãos robóticas.

Trabalhos relacionados desenvolveram um controlador de impedância conjunta para

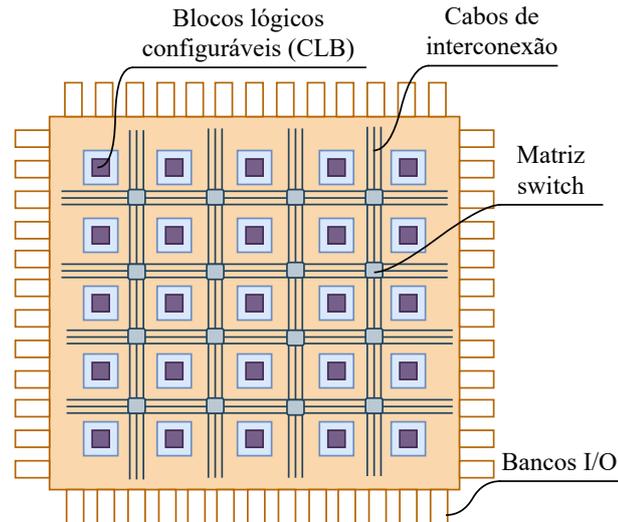


Figura 6 – Arquitetura interna FPGA.

Processor System (PS)	
Núcleo da CPU	Processador ARM Cortex-A9 ou similares, responsável da execução do software.
Controladores de memória	Interface para memória externa e interna (DDR, SRAM).
Periféricos	UART, SPI, I2C, GPIO, USB, Ethernet, entre outros.
Controladores de interrupção	Gerencia interfaces do sistema para operação eficiente da CPU.
Programmable Logic (PL)	
Blocos lógicos configuráveis (CLBs)	Blocos básicos do FPGA usados para implementar funções lógicas e registradores.
Processadores de sinal digital (DPS)	Especializados em operações aritméticas de alto desempenho.
Blocos de memória RAM	Usados para armazenamento de dados em lógica programável.
Blocos I/O	Interfaces FPGA para conexão de dispositivos e sistemas externos.
Interconexões	
Interconexões de alta velocidade	Canais entre o PS e o PL para transferência rápida de dados.
AXI ( <i>Advanced eXtensible Interface</i> )	Implementam a comunicação entre a CPU, periféricos e o PL.

Tabela 5 – Tabela geral de componentes de um SoC FPGA.

a mão robótica DLR/HIT II, usando uma estrutura multinível baseada em DSP e FPGA para melhorar suas capacidades. A estrutura de controle possui cinco níveis, incluindo aquisição de dados e operação do motor em cada dedo, comunicação e distribuição de dados pela palma, e um controlador superior para cálculos e interface com o usuário (ZHAO et al., 2006) (HUANG et al., 2006) (LIU et al., 2008) (LAN et al., 2009).

O sistema integra sensores de torque e um sensor de posição baseados em magnetorresistência, além de um acelerômetro para medir a estabilidade do objeto. Cada dedo da mão possui um módulo com DSP para processamento de sinais e FPGA para calibração e conversão dos dados dos sensores. O DSP processa os sinais e o FPGA realiza a calibração e a transmissão dos dados para a palma da mão. A estrutura é projetada para adaptação a diferentes objetos e ambientes, utilizando técnicas de análise de elementos finitos para otimizar a prensão e o controle da mão robótica.

Outros trabalhos desenvolvem controladores para mãos robóticas usando FPGAs e o DESC (Discrete Event-driven Sensory Feedback Control) para feedback vibrotátil em tempo real. A mão robótica, equipada com sensores de força e curvatura, realiza tarefas de prensão e classifica objetos rígidos e macios com base em dados processados pelo FPGA NI

Autores	Controlador	Sensor	FPGA e processador
HIL/DHL (2006-2009) (HUANG et al., 2006) (ZHAO et al., 2006) (LIU et al., 2008) (LAN et al., 2009)	Controle de impedância baseado em posição conjunta e controle multinível baseado em DSP e FPGA	Sensor de torque e sensor de posição magnetorresistivo (GMR).	FPGA com processador NIOS II
Erbas, Í. (2020) (ERBAS; VARGAS; GÜÇLÜ, 2020)	Controle de feedback de detecção de condução de evento discreto	Sensor de força piezorresistente, Sensor flexível	FPGA NI USB-7845R OEM National Instruments
Wang, L. (2022) (WANG, L.; OHKAWA, 2022)	Controle remoto em tempo real da mão robótica	Sensor de posição	FPGA PYNQ-Z2 Zynq 7000, Xilinx KV260 Zynq Ultrascale, CPU AMD Ryzen 9 5900HS de 8-núcleos, CPU ARM Cortex-A72 de 4-núcleos AMD Ryzen 9 5900HS
Waters, I. (2022) (WATERS; JONES; ALAZMANI; CULMER, P., 2022)	Controle de força com agarre estável e algoritmo de detecção de deslizamentos	Sensor de efeito Hall	FPGA MyRIO da National Instruments e microcontrolador Teensy 3.6
Pertuz, s. (2018-2022) (PERTUZ; LLANOS, C.; PEÑA, C. A. et al., 2018) (PERTUZ; LLANOS, C.; PEÑA, C. et al., 2021) (PERTUZ; LLANOS, C. H.; MUÑOZ, D. M., 2021) (PERTUZ; GHERARDINI et al., 2022)	Controle de impedância com algoritmos de filtro de Kalman	Sensor de posição rotacional e Sensor de corrente elétrica	Plataforma SoC ZYNQ-7010c e processador ARM-CortexA9

Tabela 6 – Tabela de trabalhos relacionados que utilizam FPGA para acelerar controladores de mãos robóticas.

USB-7845R OME (ERBAS; VARGAS; GÜÇLÜ, 2020).

O sistema usa quatorze sensores de flexão e onze sensores de força, com um sensor de curvatura na articulação do dedo. Os sinais dos sensores são amplificados e convertidos para tensão, e então armazenados e enviados ao FPGA via uma placa de aquisição de dados. Esses dados são filtrados e calibrados para serem usados no FPGA. Os dados calibrados formam vetores de características para treinamento, onde 70% dos dados são usados para ajustar parâmetros de regressão logística multinomial. Esses parâmetros classificam objetos como rígidos, macios ou ausentes e identificam movimentos como estacionário, flexão, contato, liberação e extensão.

Waters et al. (WATERS; JONES; ALAZMANI; CULMER, P., 2022) desenvolvem um robô capaz de detectar deslizamentos incipientes antes que ocorra um grande deslizamento. O trabalho apresenta um sensor de conceito projetado para analisar e detectar deslizamentos incipientes em uma ampla gama de testes, com o objetivo de definir um algoritmo de detecção. O sistema completo foi utilizado para o agarre de tecido de porco, demonstrando sua eficiência em uma aplicação prática e na detecção precoce de deslizamentos ao manipular um tecido orgânico. O controlador em tempo real foi embarcado em uma placa MyRIO da National Instruments, composta por uma parte PL e outra PS, sendo utilizada para sincronizar a interface do sistema e controlar o movimento de atração e retração, além de registrar os dados com uma frequência de 100 Hz.

Trabalhos recentes exploram o uso de FPGA para melhorar o controle remoto de mãos robóticas, focando na redução da latência e integração com o ROS2. Este método acelera o processamento de dados e permite o controle independente das articulações dos dedos, sincronizando os movimentos da mão robótica com a mão do usuário (WANG, L.; OHKAWA, 2022).

O sistema coleta coordenadas da mão do usuário por um dispositivo de captura de movimento, enviando os dados e imagens para o ROS2 e FPGA. O FPGA processa essas informações para gerar sinais de controle, melhorando a eficiência e reduzindo o atraso no controle da mão robótica. O projeto utilizou um CPU AMD Ryzen 9 5900HS, um Raspberry Pi 4B e uma placa de desenvolvimento PYNQ-Z2 com FPGA Zynq-7000 para o desenvolvimento.

## 2.4 Conclusões do Capítulo

A partir da revisão dos trabalhos relacionados sobre projetos de controle em mãos robóticas apresentados neste capítulo, pode-se destacar que diversos tipos de controle de força são utilizados. No entanto, o controle de força mais comumente empregado é o controle de impedância. O controle de impedância apresenta, fundamentalmente, a vantagem de controlar simultaneamente a força e a posição de um sistema. Esse controlador possui alta adaptabilidade para qualquer sistema de força e pode ser implementado em paralelo com vários outros sistemas de controle de impedância, proporcionando alta estabilidade. Entretanto, o controle de impedância tem algumas limitações. É necessário ajustar os parâmetros  $K_M$ ,  $K_K$  e  $K_B$  para o sistema especificado, o que implica que o modelo do sistema de forças deve ser identificado previamente. Caso contrário, o sistema de controle pode se tornar instável. Além disso, este controle requer um sistema de sensoriamento constante de força e posição para seu funcionamento. Quando vários controladores de impedância são utilizados simultaneamente, o custo computacional do sistema aumenta.

Diversas pesquisas têm sido dedicadas nos últimos anos à detecção e controle de deslizamento de objetos, uma vez que essa dinâmica é crucial para o controle de mãos robóticas. Diversos trabalhos relacionados utilizam modelos de controle e detecção de deslizamentos mais complexos, como redes neurais. Isso lhes permite realizar controle e detecção de deslizamentos de forma mais variada em diferentes situações de agarre de objetos. Adicionalmente, esses trabalhos geralmente desenvolvem seus próprios sensores táteis, o que lhes proporciona um maior grau de informação sobre a posição do objeto e a direção dos deslizamentos, favorecendo a previsão dos movimentos.

No que tange ao uso de SoC FPGAs, existem vários estudos que buscam acelerar o processamento de dados para melhorar a eficiência do controle em mãos robóticas. Na implementação de sistemas de controle usando FPGA, a principal vantagem obtida é o processamento em paralelo de vários sistemas ou controladores. Além disso, há uma vantagem

no tempo de processamento das metodologias implementadas, já que dentro de uma FPGA a taxa de processamento em cada ciclo de relógio pode ser de até 100 MHz. A justificativa para o uso de FPGA em aplicações de mãos robóticas reside precisamente no processamento em paralelo, uma vez que as mãos robóticas geralmente precisam calcular os movimentos dos dedos de forma independente para o controle dos motores, assim como realizar a filtragem de múltiplos sensores simultaneamente, evitando, assim, a necessidade de usar uma matriz de microcontroladores, conforme a aplicação.

## 3 Materiais e Métodos

Este capítulo descreve os passos necessários para a construção de um modelo de garra robótica, iniciando pelas especificações do sistema e requisitos a serem cumpridos (3.1). Os requisitos e especificações foram levantados a partir da análise de limitações do projeto UnB-Hand descrito no Capítulo 1. Em seguida, é apresentada uma configuração experimental que detalha os materiais utilizados durante o desenvolvimento do projeto (3.2). O capítulo prossegue com a descrição do algoritmo de detecção de deslizamentos, abordando seu funcionamento e os componentes que o integram (3.3), bem como sua implementação em hardware (3.4). Posteriormente, é exposto o desenvolvimento do controle de impedância, abordando a composição do sistema de controle embarcado e sua aplicação (3.5). A seguir, são discutidos a construção, a simulação, os modelos e os componentes necessários para a realização do projeto da garra robótica (3.6) e também uma descrição do protocolo para validação experimental (3.7). Por fim, um resumo do capítulo (3.8).

### 3.1 Aspectos Gerais

#### 3.1.1 Requisitos

O sistema de detecção de deslizamentos usando uma garra robótica deve obedecer a parâmetros específicos para possibilitar a comunicação com outros sistemas, oferecer flexibilidade em diversos estudos de caso e utilizar os materiais adequados para este processo. Assim, são apresentados os seguintes requisitos:

- O sistema de detecção de deslizamentos para a garra robótica deve utilizar sensores de força piezo-resistivos FSR. A escolha deste sensor se justifica dada o seu baixo custo e desempenho avaliado previamente em um trabalho de iniciação científica (TEIXEIRA; MUÑOZ, D., 2023).
- O algoritmo de detecção de deslizamentos deve ser de baixa complexidade computacional, utilizando operadores simples, se forma que seja facilmente integrado ao projeto UnB-Hand (PERTUZ; LLANOS, C. H.; MUÑOZ, D. M., 2021).
- O sistema de detecção de deslizamentos deve ser integrado junto com o processador ARM, como parte de um co-projeto hardware/software.
- A latência de cálculo do algoritmo de detecção de deslizamentos deve ser inferior à latência de cálculo do controle de impedância.

- O sistema de detecção de deslizamentos deve ser projetado para facilitar a integração com o banco de filtros e sistemas de estimativa de parâmetros do projeto UnB-Hand, desenvolvido em FPGA (PERTUZ; GHERARDINI et al., 2022).
- A implementação em hardware do sistema de detecção de deslizamentos deve utilizar os IP-cores de cálculo aritmético previamente desenvolvidos pelo grupo de pesquisa (MUÑOZ, D. M. et al., 2010a) (MUÑOZ, D. M. et al., 2010b).
- Os tipos de deslizamentos a serem identificados devem ser verticais e rotacionais.
- O controlador de força em software embarcado do sistema deve utilizar a técnica de controle de impedância de forma que futuramente seja possível a sua integração com o controlador da UnB-Hand (PERTUZ; LLANOS, C.; PEÑA, C. et al., 2021) (PERTUZ; LLANOS, C.; PEÑA, C. A. et al., 2018).
- O controlador de impedância deve ser capaz de reagir aos deslizamentos detectados para evitar a queda dos objetos.
- A garra robótica deve ter uma realimentação de posição através de um sensor, para fechar o ciclo do controlador de posição interno do sistema.
- Um servomotor deve ser utilizado como atuador da garra robótica.
- Os sinais dos sensores de força e posição devem ser filtrados antes de entrar no sistema de detecção de deslizamentos e no controlador de impedância.

### 3.1.2 Especificações

Para o desenvolvimento do sistema de detecção de deslizamentos, são apresentadas especificações que devem atender os requisitos do projeto, contribuindo para o campo de estudo da robótica. As especificações técnicas são as seguintes:

- O algoritmo de detecção de deslizamentos deve ser implementado no mesmo hardware utilizando no projeto da UnB-Hand, especificamente o SoC Zynq 7020 da AMD-Xilinx.
- O funcionamento do sistema de detecção embarcado deve obedecer a um ciclo de clock mínimo de 100 MHz, de acordo com a implementação do projeto UnB-Hand, que também utiliza uma frequência de clock de 100 MHz.
- A ocupação do algoritmo de detecção de deslizamentos deve ser inferior a 5% dos recursos disponíveis do chip Zynq 7020. Considerando que, no pior cenário, as implementações do projeto UnB-Hand ocupam aproximadamente 96% das LUTs disponíveis (PERTUZ; GHERARDINI et al., 2022), é importante manter uma ocupação de LUTs abaixo de 4%.

- O peso máximo dos objetos não deve ultrapassar 700 gramas.
- A latência do sistema de detecção de deslizamentos deve ser igual ou inferior a 40 ms.
- A precisão do algoritmo de detecção de deslizamentos deve ser um valor entre 80% e 90%, adota-se essa referência porque trabalhos correlatos indicam esse como valor aceitável e, reportam taxas de precisão de até 89,96% e 91,40% respectivamente (KHIN et al., 2021) (ABD et al., 2018).
- O erro MSE (Mean Squared Error) entre a implementação em hardware e em software do algoritmo de detecção de deslizamentos não deve exceder o valor de  $1 \times 10^{-3}$ .

## 3.2 Setup Experimental

### 3.2.1 Componentes

Para a construção da garra robótica foram utilizados dois tipos de sensores para o sistema de detecção: o sensor de força FSR 400 da Interlink Electronics e o sensor de posição Rotary Position Sensor SV01 da Murata. Adicionalmente, o servo motor MG995 *High Speed Servo Actuator* foi usado para acionamento da garra.

O sensor de força FSR 400 da Interlink Electronics (vide Figura 7) consiste em um filme polimérico que apresenta diminuição da resistência à medida que a força aumenta na superfície do sensor. Este sensor é fabricado para o controle tátil de dispositivos eletrônicos em diversas áreas, como eletrônica automotiva, sistemas médicos, aplicações industriais e robótica.



Figura 7 – Sensor piezorresistivo FSR 400 da Interlink Electronics (ELECTRONICS, 2024).

O sensor de posição *Rotary Position Sensor SV01* da Murata (vide Figura 8) é composto por um potenciômetro capaz de girar  $160^\circ$ . Possui durabilidade de mais de 1 milhão de ciclos na faixa de temperatura de  $-40^\circ\text{C}$  a  $85^\circ\text{C}$ .

O High Speed Servo Actuator MG995 (vide Figura 9) é um servomotor de alta velocidade, com peso de 55 gramas e torque total de 10 kgf·cm operando a 6 volts, com rotação de  $120^\circ$  e tensão de operação variando de 4,8 a 7,2 volts.

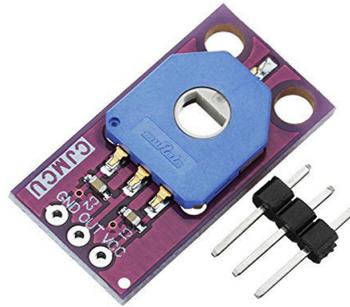


Figura 8 – Sensor rotativo Murata SV01 (MURATA, 2024).



Figura 9 – High Speed Servo Actuator MG995 (SERVOMOTOR., 2024).

Para este trabalho, foi adaptada uma garra robótica de baixo custo (vide Figura 10). A garra é confeccionada em alumínio e composta por dois elos com pontas em formato de dentes de serra. A garra robótica possui um grau de liberdade (DoF, do inglês *Degree of Freedom*), pesa 65 gramas e mede 122 milímetros de largura, 80 milímetros de comprimento e 20 milímetros de altura, também utiliza parafusos de 3 milímetros. Foi adicionado um suporte (vide Figura 22) para os sensores nas pontas do atuador visando melhorar a aderência na prensão de objetos. Para o movimento, foi instalado um motor no eixo esquerdo da garra, que transmite o movimento por meio de uma engrenagem adaptada às laterais dos elos, permitindo a transmissão do movimento para o lado direito da garra.



Figura 10 – Garra robótica de alumínio (SARAVATI, 2024).

### 3.2.2 Calibração dos sensores

Para calibrar o sensor de força, utilizou-se uma balança com o objetivo de padronizar as medições do sensor utilizando pesos conhecidos, variando de cinquenta gramas a um

quilograma (vide Tabela 7). Os valores discretos utilizados para a calibração do sensor de força devem-se a dois fatores. Primeiro, as medições entre 25 e 50 gramas representam uma resolução adequada para a implementação do controle desejado. Em segundo lugar, a balança digital utilizada é do modelo SF-400, o qual possui uma resolução de 1 grama e apresenta um erro de medição de aproximadamente  $\pm 1$  grama, razão pela qual as medições realizadas não consideram valores decimais.

Uma fita de silicone de 2 mm foi colocada na superfície do sensor para proteger sua integridade e garantir que a dispersão dos dados do sensor fosse menor em cada medição, conforme apresentado no trabalho relacionado (TEIXEIRA; MUÑOZ, D., 2023).

Para cada peso, foram coletadas 16 amostras, visando obter uma base estatística para a calibração do sensor. Esse número de amostras proporciona uma robustez estatística, de acordo com as validações apresentadas no trabalho (MCCLAVE; SINCICH, 2021), no qual se destaca que, a partir de 10 amostras, há uma certeza estatística sobre as medições. Em seguida, utilizando os valores médios de todas as amostras, foi realizada uma aproximação da calibração do sensor por meio de uma equação polinomial de segundo grau (vide Equação 3.1). Essa equação possibilita a conversão dos valores recebidos pelo sensor através do ADC (Conversor Analógico-Digital) da placa de aquisição, representado pela variável  $x$ , para a unidade de medida de força em Grama-Força (Grf).

$$F = 0,0025x^2 - 4,1507x + 1753 \quad (3.1)$$

Tabela 7 – Tabela de calibração do sensor FSR 400.

Gr	Média	Mediana	Desvio padrão	Mínimo	Tensão (V)
50	925,125	921	10,95977	917	4,501466
75	845,1875	845	2,993743	837	4,13001
100	819,125	819	0,885061	817	4,002933
125	768,9375	768	3,75	764	3,753666
150	767,625	768	1,586401	765	3,753666
175	736,5625	738	2,988171	731	3,607038
200	760,75	760	2,863564	758	3,714565
250	742,125	742	0,957427	741	3,626588
300	701,6875	702	2,301268	698	3,431085
350	702,0625	702	1,94829	699	3,431085
400	670	669	4,690416	667	3,269795
450	663,125	663	2,362908	660	3,240469
500	661	661	0,894427	660	3,230694
550	665,8125	667,5	5,682942	647	3,262463
600	648,9375	647,5	5,092069	644	3,164712
650	654,375	654	1,995829	652	3,196481
700	632,5	630	9,736529	622	3,079179
750	632,1875	632	1,470544	630	3,088954
800	619,9375	620	3,065262	615	3,030303
850	616	616	0,966092	614	3,010753
900	610,1875	609,5	2,587631	607	2,978983
950	603,8125	604	0,981071	602	2,952102
1000	611,875	611,5	10,55699	602	2,988759

Antes de calibrar o sensor, são realizados testes para verificar a dispersão das medições do sensor. Para isso, foram coletados dados de pesos específicos variando de cinquenta a mil

gramas, com dezesseis amostras cada, resultando no gráfico da Figura 11, que apresenta um desvio padrão total de 367,9314 grf. Observa-se que para valores entre 800 e 1000 grf o modelo não apresenta suficiente robustez, pois representam um mesmo valor de leitura de ADC. Para melhorar a linearidade do modelo, e visando reduzir o valor do desvio padrão das medições, optou-se por colocar uma fita de silicone de dois milímetros na superfície do sensor, o que também protege sua integridade. Em seguida, foi repetido o protocolo experimental obtendo-se a curva de calibração a Figura 12, que mostra uma diminuição do desvio padrão para 152,5427 grf. Observa-se ainda que a linearidade do modelo de calibração para pesos entre 800 e 1000 grf melhorou se comparado com o sensor sem uso da fita de silicone. Este fato justifica a adoção do sensor FSR na instrumentação da garra robótica.

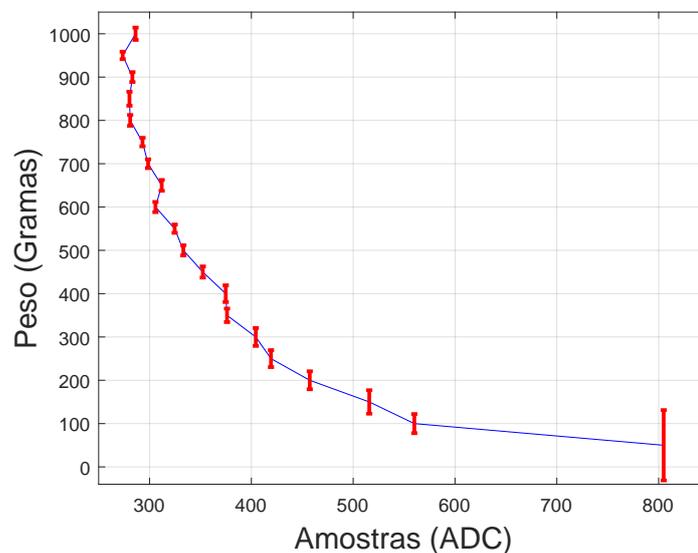


Figura 11 – Dados de calibração do sensor de força FSR400 sem fita de silicone.

O intervalo de confiança foi calculado para os dois casos, com e sem fita de silicone. Adotou-se um nível de significância de 95%. Primeiramente, para o caso com fita de silicone, utilizou-se um peso padrão de 500 gr, sendo obtida uma média de 502,4375 grf com desvio padrão de 7,7542 grf. Dessa forma, o intervalo de confiança de 95% é de  $502,4375 \pm 3,3985$  grf, conforme a Equação 3.2, onde  $\bar{x}$ ,  $s$  e  $n$  são a média, a variância e o número de amostras, respectivamente. Da mesma forma, calculou-se o intervalo de confiança para a medição do sensor sem fita de silicone, obtendo uma média de 515,8125 grf com desvio padrão de 27,0338 grf. Nesse caso, o intervalo de confiança é igual a  $515,8125 \pm 11,8482$  grf.

$$IC_{95\%} = \bar{x} \pm 1,7531 \cdot \left( \frac{s}{\sqrt{n}} \right) \quad (3.2)$$

Para a calibração do sensor de posição, foram coletadas amostras nas faixas de 0 a 180 graus de rotação do eixo interno do sensor. Depois, foi realizada uma regressão linear (vide Equação 3.3) nos dados coletados, a qual relaciona os dados do ADC da placa de aquisição em valores de ângulo em graus.

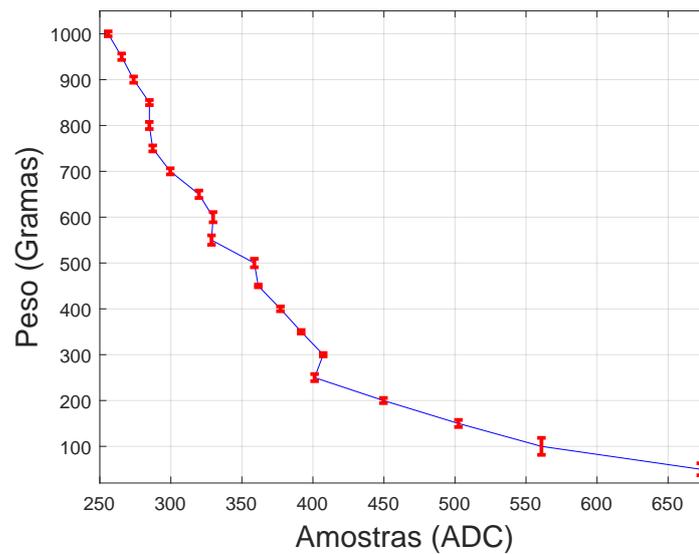


Figura 12 – Dados de calibração do sensor de força FSR400 com fita de silicone.

$$\theta = 0,1272x - 31,204 \quad (3.3)$$

### 3.3 Algoritmo de Detecção de Deslizamento

O algoritmo de detecção de deslizamento foi desenvolvido em duas etapas. A primeira etapa envolveu a criação do algoritmo, analisando dados de deslizamento de diversos objetos. Nesta fase, os sensores de força foram posicionados nas pontas do indicador e do polegar, garantindo um agarre firme nos objetos e permitindo a realização de experimentos com deslizamento uniforme.

A segunda etapa consistiu no refinamento do algoritmo de detecção de deslizamento. Esta fase envolveu a repetição de experimentos de apreensão de objetos que permitem o deslizamento devido ao seu próprio peso, com o objetivo de ajustar os parâmetros que compõem o detector de deslizamento para alcançar melhores resultados. Esta seção foca-se na descrição matemática do desenvolvimento do detector de deslizamento, utilizando o banco de dados coletado.

Foram realizados deslizamentos durante o agarre de quatro objetos diferentes: garrafa de água, laranja, telefone celular e um livro, conforme apresentado na Figura 25. A aquisição desses dados foi realizada por meio dos sensores de força FSR.

A seguir, são apresentadas as quatro etapas para a implementação do algoritmo:

- I) Cálculo da média da força para cada conjunto de dois sensores (dois sensores FSR no dedo indicador e dois sensores FSR no dedo polegar).

- II) Aplicação de um filtro de média móvel nos dados resultantes do cálculo da média entre cada par de sensores.
- III) Cálculo da derivada de primeira ordem dos dados obtidos após a filtragem.
- IV) Implementação do detector de deslizamento utilizando análise de limiares com os dados provenientes da derivada da média da força.

### 3.3.1 Cálculo da média para cada conjunto de dados de dois sensores

Após determinar a posição dos sensores de força nos elos do efetor final, o objetivo é identificar as variações na força de contato que os sensores experimentam ao segurar diferentes objetos, com o *setpoint* variando conforme o objeto. Dessa forma, ao adquirir este conjunto de dados, é possível calcular a média para cada par de sensores utilizando a Equação 3.4.

$$\bar{y} = \frac{1}{n} \sum_{i=1}^n F_i, \quad (3.4)$$

onde  $\bar{y}$  representa a média do conjunto de sensores FSR,  $n$  é o número total de elementos no conjunto, e  $F_i$  denota o  $i$ -ésima medição de força.

### 3.3.2 Filtro de média móvel para dados

O filtro de média móvel (vide Equação 3.5) é responsável por processar os dados da média dos sensores FSR, a fim de reduzir o ruído que entra no sistema de detecção. É importante salientar que a natureza dos sinais de deslizamento é de alta intensidade e alta frequência.

Neste trabalho utilizou-se o filtro de média móvel iterativa, que possui uma vantagem significativa na implementação em sistemas embarcados, calculando novos valores filtrados à medida que novos dados são recebidos a cada ciclo de *clock*, fornecendo uma resposta imediata. O filtro requer uma FIFO para implementar a janela móvel conforme a Equação 3.5.

$$\bar{F}(k) = \bar{F}(k-1) + \frac{(\bar{y} - FIFO(0))}{N}, \quad (3.5)$$

onde,  $\bar{F}(k)$  é a média móvel no tempo  $k$ ,  $\bar{F}(k-1)$  é a prévia da média móvel,  $\bar{y}$  é a média da força de cada par de sensores FSR,  $FIFO(0)$  é a amostra de força mais antiga na janela móvel, e  $N$  é o número de pontos considerados na média. Os  $N$  valores da janela móvel podem ser escolhidos como potências de dois para facilitar a implementação em hardware.

### 3.3.3 Cálculo da derivada de primeira ordem

A função derivada (vide Equação 3.6) utiliza os valores atual e o anterior de filtro de média móvel e calcula a derivada entre eles. Esse processo é realizado para facilitar a análise das características dos dados, possibilitando a identificação dos pontos nos quais ocorre o deslizamento dos objetos.

$$\dot{F}(k) = \frac{d\bar{F}(k)}{dk} = \frac{\bar{F}(k) - \bar{F}(k - 1)}{\Delta k}, \quad (3.6)$$

onde,  $d\bar{F}(k)/dk$  ou  $\dot{F}(k)$  é a derivada da força média no tempo,  $\bar{F}(k)$  é o valor atual da média móvel da força,  $\bar{F}(k - 1)$  é o valor anterior da média móvel da força, e  $\Delta k$  é o tempo de amostragem.

### 3.3.4 Detector de deslizamento

A partir da análise da base de dados de deslizamento de objetos (detalhes são apresentados na Seção 3.7), foi possível estabelecer que o detector de deslizamentos é fundamentado nos seguintes parâmetros: valor inicial de deslizamento ( $\dot{F}_L$ ), valor final de deslizamento ( $\dot{F}_H$ ), posição inicial ( $pos_L$ ), posição final ( $pos_H$ ), limiar máximo de amostras da derivada (*range*) e um contador de posição de derivada ( $C$ ). Com base nisso, o detector de deslizamento também pode ser compreendido como um detector de picos.

A função de detecção de deslizamentos (Algoritmo 2) toma como entrada os dados da derivada da força ( $\dot{F}(k)$ ). Além disso, utiliza como entradas os parâmetros *LimitL*, que é o ponto de derivada mínima no qual um deslizamento pode ocorrer; *LimitH*, que é o ponto de derivada máxima no qual um deslizamento pode ocorrer;  $\bar{F}$ , que é a força média (saída do filtro de média móvel); e *ThresF*, que é o limite mínimo de força no qual ocorrem deslizamentos. É importante notar que esses parâmetros são ajustáveis e dependem de vários aspectos, como o peso, o tamanho do objeto, o coeficiente de atrito, e dos pontos de contato do objeto.

Dentro da função de detecção, todas as amostras passam por funções condicionais (*if*) que armazenam a posição do vetor e o valor da derivada nos pontos em que ela está dentro dos limites máximo (*LimitH*) e mínimo (*LimitL*) dos possíveis deslizamentos. Se os valores da derivada ( $\dot{F}(k)$ ) permanecerem entre esses pontos limites, uma variável ( $C$ ) conta quantas amostras fazem parte de um mesmo deslizamento, permitindo determinar a duração do mesmo. Uma sinal de *reset* é usado para zerar os valores caso os limites do contador sejam ultrapassados, significando que não foram encontrados deslizamentos.

Por fim, temos uma função condicional (vide *if* da linha 19) que define se as amostras armazenadas representam um deslizamento. Primeiro, é verificado se o valor da derivada atual ( $\dot{F}(k)$ ) está dentro do intervalo [*LimitL* *LimitH*]; depois, é verificado se a variável do contador ( $C$ ) é menor que a faixa de amostras definidas (*range*); e, finalmente, se a

---

**Algorithm 2** Pseudocódigo de detecção de picos.
 

---

```

1: function PEAKDETECTION( $\bar{F}, range, LimitL, LimitH, F, ThresF$ )
2:   inicialização;
3:   for  $k := 1 \rightarrow 160$  do
4:     if  $\dot{F}(k) \leq LimitL$  then
5:        $pos_L \leftarrow k$ ;
6:        $\dot{F}_L \leftarrow \dot{F}(k)$ ;
7:     end if
8:     if  $\dot{F}(k) \geq LimitH$  then
9:        $pos_H \leftarrow k$ ;
10:       $\dot{F}_H \leftarrow \dot{F}(k)$ ;
11:    end if
12:    if  $\dot{F}(k) \geq \dot{F}_L \ \& \ C \leq range$  then
13:       $C \leftarrow C + 1$ ;
14:    else
15:       $C \leftarrow 0$ ;  $\dot{F}_L \leftarrow 0$ ;
16:       $pos_H \leftarrow 0$ ;  $\dot{F}_H \leftarrow 0$ ;
17:       $pos_L \leftarrow 0$ ;
18:    end if
19:    if  $\dot{F}(k) \geq \dot{F}_L \ \& \ \dot{F}(k) \leq \dot{F}_H \ \& \ C \leq range \ \& \ F(k) \geq ThresF$  then
20:       $sliding \leftarrow sliding + 1$ ;
21:      return Retorna as posições e os valores nos quais o deslizamento foi detectado e exhibe o valor de  $sliding$ .
22:    else
23:      Nenhum deslizamento detectado,  $sliding \leftarrow 0$ ;
24:    end if
25:  end for
26: end function

```

---

força média ( $\bar{F}$ ) medida naquele ponto é maior que o limite mínimo de força para que ocorra o deslizamento ( $ThresF$ ). Se todas essas condições forem atendidas, pode-se definir que as amostras representam um deslizamento, e os valores limites são armazenados. É importante salientar que o detector de deslizamento proposto tem a capacidade de identificar deslizamentos consecutivos.

### 3.4 Co-projeto HW/SW para Sistema de Detecção de Deslizamentos

A arquitetura geral do sistema de detecção de deslizamentos (vide Figura 13) foi implementada em um SoC FPGA Zynq 7020 utilizando a metodologia de co-projeto de Hardware/Software. Nesta metodologia, o projeto é dividido em dois blocos principais. Primeiro, o bloco PS (Processor System), que contém a memória DDR onde são armazenados os dados de força média dos sensores e um processador ARM Cortex A9 responsável pela comunicação com o FPGA. Segundo, o bloco PL (Programmable Logic), composto pelo módulo de comunicação *AXI-Interconnect*, além dos módulos desenvolvidos em hardware: módulo de *filtro de média móvel*, módulo de *derivada de primeira ordem* e módulo de *detecção de picos*.

Os componentes da detecção de deslizamentos foram descritos em VHDL utilizando IPs de cálculo aritmético com representação em ponto flutuante de 27-bits previamente desenvolvidos (MUNOZ, D. M. et al., 2010a), (MUNOZ, D. M. et al., 2010b). Esses IP-Cores

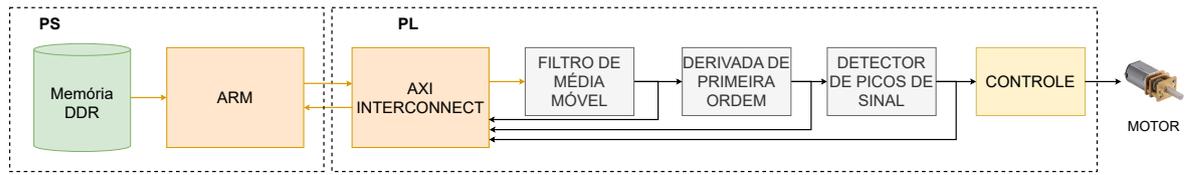


Figura 13 – Arquitetura geral do co-projeto Hardware/Software do sistema de detecção de deslizamentos.

são baseados no padrão *IEEE-754 standard*, onde a palavra de 27-bits é composta por um bit de sinal, expoente de 8-bits, e mantissa de 18-bits, ocupando apenas um bloco DSP para cada operação de multiplicação.

### 3.4.1 Arquitetura em hardware do filtro de média móvel

A arquitetura do filtro de média móvel em hardware (vide Figura 14) implementa a Equação 3.5 e recebe como entradas um sinal *start* e os dados de força média  $\bar{y}(k)$ . Esses dados são armazenados em uma FIFO de oito posições, denominado *regF*, representando a janela móvel. Os dados são agrupados de forma consecutiva e, quando o limite de oito dados é ultrapassado, o último é descartado. Em seguida, é realizada uma operação de subtração (*sub*) entre o dado atual de  $\bar{y}(k)$  e o último dado da FIFO. O resultado dessa subtração é então dividido pelo tamanho da janela móvel. Para realizar essa divisão de forma eficiente em uma implementação em FPGA, se o divisor for uma potência de dois, é realizada apenas uma operação de subtração entre o valor do expoente (*E*) e a potência de dois equivalente ao divisor desejado (3). O resultado desta operação (*div*) é somado ao valor da saída anterior, chamada de *preAvg*, resultando no sinal  $\bar{F}(k)$ , e em um sinal *ready* como saída do módulo.

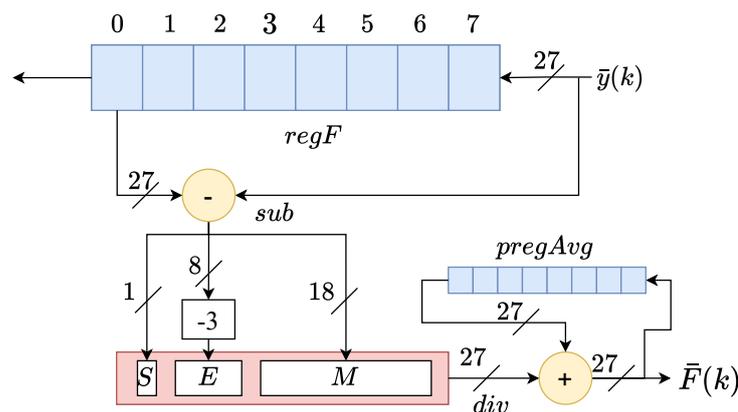


Figura 14 – Representação gráfica da arquitetura em hardware do módulo filtro de média móvel.

### 3.4.2 Arquitetura em hardware da derivada de primeira ordem

A arquitetura de hardware da derivada de primeira ordem (vide Figura 15) implementa a Equação 3.6, e recebe como entrada um sinal *start* e o valor de  $\bar{F}(k)$ . Esse valor é armazenado em um registrador  $\bar{F}(k - 1)$ , depois é realizada a operação de subtração entre

$\bar{F}(k)$  e  $\bar{F}(k-1)$ , resultando no valor *sub*. Em seguida, é realizada a operação de multiplicação entre o valor de *sub* e o inverso do tempo de amostragem de aquisição de dados. Finalmente, como saída do módulo, obtém-se um sinal *ready* e o valor de  $\dot{F}(k)$ .

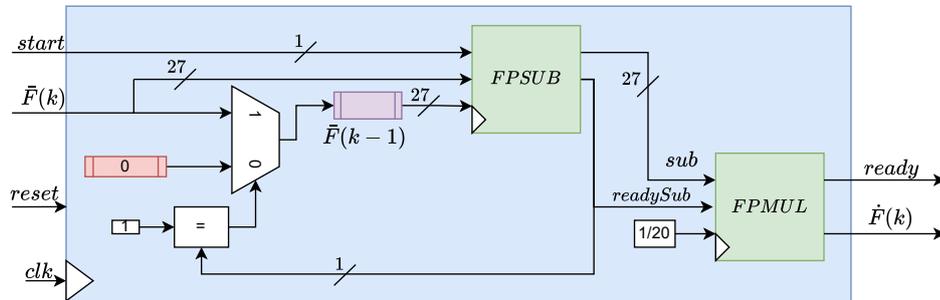


Figura 15 – Representação gráfica da arquitetura em hardware do módulo derivado de primeira ordem.

### 3.4.3 Arquitetura em hardware do módulo detector de picos

A arquitetura do detector de picos (vide Figura 16) implementa o pseudo-código do Algoritmo 2, o qual recebe como entrada o valor de  $\dot{F}(k)$  e passa por duas condicionais que verificam dois limites diferentes:  $limit_L$  e  $limit_H$ . Se o valor de  $\dot{F}(k)$  atender a qualquer uma dessas duas condições, o valor de  $\dot{F}(k)$  e sua posição são armazenados. Quando a condição *LimitHigh* é atendida, o valor de  $\dot{F}(k)$  é armazenado como  $\dot{F}_H$  e sua posição como  $pos_H$ . Da mesma forma, se a condição *LimitLow* for atendida, o valor de  $\dot{F}(k)$  é armazenado como  $\dot{F}_L$  e sua posição como  $pos_L$ , indicando pelo sinal *firstslip* o início de um possível deslizamento. O algoritmo funciona da seguinte maneira. Inicialmente, enquanto o sinal *firstslip* estiver assertado, o contador (*C*) incrementa para cada nova amostra sempre que estiver dentro da faixa de tempo pré-estabelecida (*range*). Em seguida, é realizada uma comparação do valor da força no ponto atual ( $\bar{F}(k)$ ) com o valor limite de força (*thresF*). Finalmente, quando o sistema recebe os sinais de  $\dot{F}_H$  e  $pos_H$ , o detector indica que um deslizamento foi encontrado, e o sinal *ready* é assertado representando que as saídas  $\dot{F}_H$ ,  $pos_H$ ,  $\dot{F}_L$ ,  $pos_L$  são válidas.

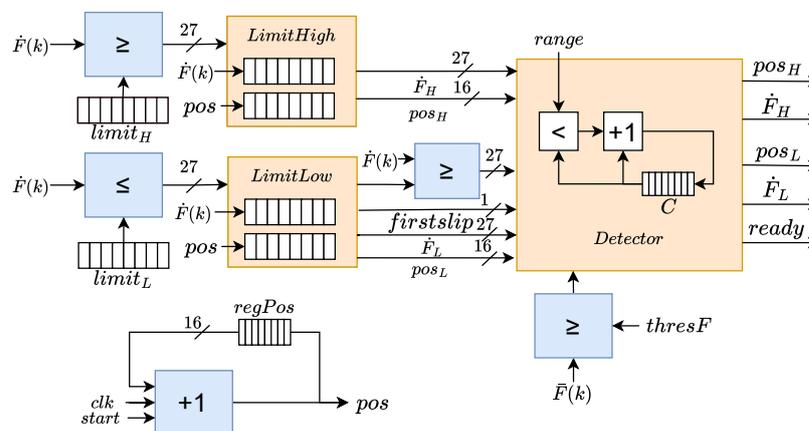


Figura 16 – Representação gráfica da arquitetura em hardware do módulo detector de picos.

### 3.4.4 Co-projeto de hardware/software do sistema de detecção de deslizamentos

O diagrama de blocos que compõe a arquitetura do sistema de comunicação e detecção de deslizamentos (vide Figura 17), é formado pelo bloco do processador ZYNQ7, que contém uma memória DDR com os dados de força média dos sensores, o bloco de comunicação AXI-Interconnect e o IP-core *slidingIP\_0*, que engloba os módulos descritos: filtro de média móvel, derivada de primeira ordem e detector de picos.

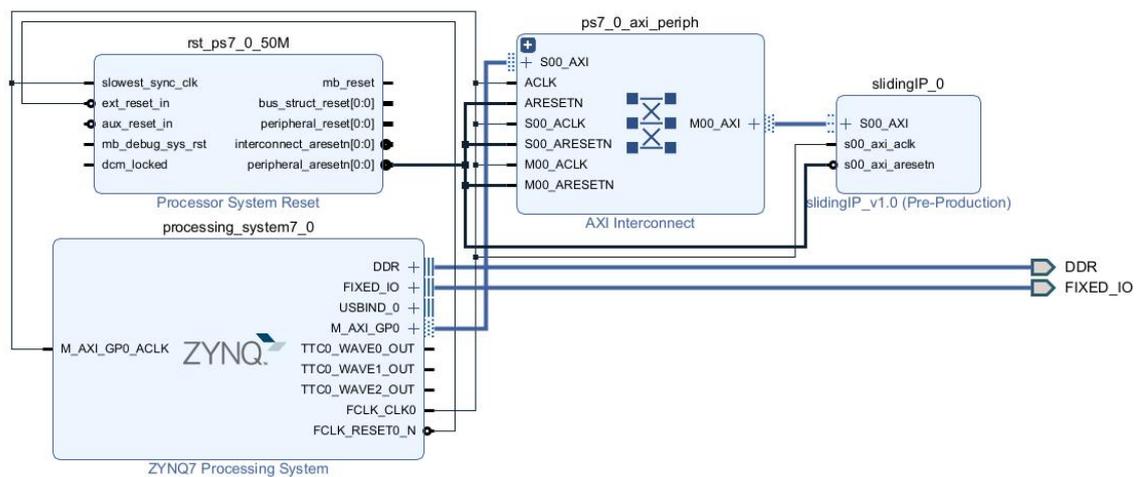


Figura 17 – Diagrama de blocos do co-projeto de Hardware/Software.

## 3.5 Desenvolvimento do Controle de Impedância

O sistema de controle é ilustrado na Figura 18. Este modelo é composto pela entrada do sistema como o *setpoint* de força desejado, que é subtraído do sinal medido pelo sensor de força. O resultado dessa operação é então utilizado como entrada para o controle de impedância, que por sua vez gera a posição necessária do atuador para alcançar a força desejada. Essa posição é subtraída do valor do sensor de posição, e o erro de posição forma a entrada do controlador de posição PID do motor do sistema, que produz a ação de controle na forma de um sinal PWM.

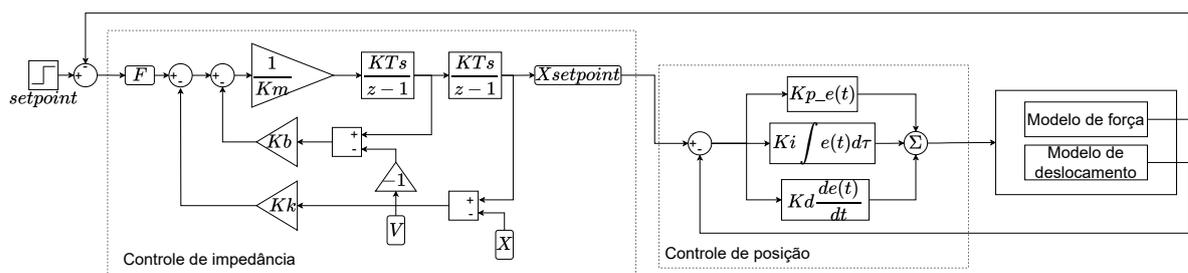


Figura 18 – Modelo interno do controle de impedância.

### 3.5.1 Controle de impedância

O controlador de impedância foi concebido utilizando as Equações 3.11 3.12 3.13, na qual o erro de força ( $Force_{error}$ ) é calculado como a diferença entre o *set point* e um dos sensores de força do sistema ( $Force_{output}$ ), como indicado na Equação 3.7. Posteriormente, o erro de posição ( $x_{error}$ ) é determinado pela diferença entre a posição desejada ( $x$ ) (determinada pelo controle de impedância) e o sinal de saída do sensor de posição ( $x_{output}$ ), como indicado na Equação 3.8. Para calcular o erro de velocidade ( $\dot{x}_{error}$ ), primeiramente é obtida a derivada temporal da medição do sensor de posição ( $\dot{x}_{output}$ ), gerando um valor de referência para comparação, como indicado na Equação 3.9. Em seguida, o erro de velocidade é calculado como a diferença entre a velocidade estimada pelo modelo de impedância ( $\dot{x}$ ) e a velocidade estimada pelo sensor ( $\dot{x}_{output}$ ), como indicado na Equação 3.10.

$$Force_{error} = (set\ point - Force_{output}) \quad (3.7)$$

$$x_{error} = (x - x_{output}) \quad (3.8)$$

$$\dot{x}_{output} = \frac{x_{output} - x_{prev}}{\Delta k} \quad (3.9)$$

$$\dot{x}_{error} = (\dot{x} - \dot{x}_{output}) \quad (3.10)$$

Prosseguindo, a Equação 3.11 calcula a aceleração do modelo de impedância ( $\ddot{x}$ ), onde  $K_M$ ,  $Force_{error}$ ,  $K_K$ ,  $x_{error}$ ,  $K_B$ , e  $\dot{x}_{error}$ , representam a constante de massa, o erro de força, a constante de mola, o erro de posição, a constante de amortecimento, e o erro de velocidade, respectivamente. A Equação 3.12, estima a velocidade do modelo de impedância ( $\dot{x}$ ), onde  $sampletime$ ,  $\ddot{x}$ ,  $last\_x$  e  $last\_x$  representam o tempo de amostragem, a aceleração estimada, a última aceleração estimada, e a última velocidade estimada, respectivamente.

$$\ddot{x} = \frac{1}{K_M} (Force_{error} - K_K \cdot x_{error} - K_B \cdot \dot{x}_{error}) \quad (3.11)$$

$$\dot{x} = (sampletime \cdot 0.5) \cdot (\ddot{x} + last\_x) + last\_x \quad (3.12)$$

$$x = (sampletime \cdot 0.5) \cdot (\dot{x} + last\_x) + last\_x \quad (3.13)$$

A Equação 3.13 calcula a posição do modelo de impedância ( $x$ ), onde  $sampletime$ ,  $\dot{x}$ , ( $last\_x$ ), ( $last\_x$ ), representam o tempo de amostragem, a velocidade estimada, a última velocidade estimada, e a última posição estimada, respectivamente. A posição  $x$  estimada pelo modelo de impedância é então utilizada como entrada para o controle PID de posição da garra.

### 3.5.2 Controle PID

O controlador PID recebe como entrada o erro de posição com um windup de -200 e 200. A parte integral do controle é calculada multiplicando-se três variáveis: o  $K_i$  pelo acumulador integral e pelo tempo de amostragem. Este resultado é limitado por um windup de -100 e 100.

Posteriormente, o filtro derivativo do controle é calculado conforme representado pela Equação 2.3. A derivada do erro é então calculada como a diferença entre o valor do filtro atual e o valor do filtro anterior, dividido pelo tempo de amostragem. A parte derivativa do controle é definida como o produto de  $K_d$  pela derivada do erro.

Finalmente, a ação de controle é calculada multiplicando-se  $K_p$  pelo erro de posição, somando-se a parte integral e a parte derivativa do controle. Este resultado também foi limitado por um windup de -100 e 100. Este valor de saída do controle de impedância é então mapeado para um valor de *PWM* entre 65 e 160, ou equivalente a uma faixa de 1,27 V a 3,14 V, que corresponde à faixa de atuação do motor integrado no sistema robótico.

### 3.5.3 Otimização do sistema de controle

A otimização do controlador foi realizada pelo algoritmo PSO (*Particle Swarm Optimization*). O PSO foi configurado para otimizar seis variáveis de decisão:  $K_K$ ,  $K_B$ ,  $K_M$ ,  $K_p$ ,  $K_i$  e  $K_d$ , as quais correspondem às constantes do controlador de impedância e do controle PID.

Os parâmetros utilizados no algoritmo PSO foram os seguintes: 250 ciclos de iteração, 45 partículas, coeficiente cognitivo e social de 1,25, e limites do espaço de busca entre -15 a 600. A função custo utilizada no processo de otimização está representada pela Equação 3.14, onde  $t_r$ ,  $e_{ss}$ , *overshoot*, e *MSE* representam, o tempo de resposta, o erro de estado estacionário, o sobre passo, e o erro quadrático médio da resposta do sistema, respectivamente.

$$\min f_{cost} = 0.4t_r + 0.1e_{ss} + 0.5overshoot + 0.4MSE \quad (3.14)$$

Durante o processo de otimização, cada partícula avalia a sua aptidão usando a função custo, para o qual é necessário realizar uma simulação do sistema robótico, a qual foi conduzida utilizando o ambiente Matlab-Simulink (vide Figura 19). A simulação no Simulink foi realizada utilizando modelos de tempo discreto para obter resultados mais confiáveis para a implementação do controlador.

Para o processo de otimização foi utilizado um *setpoint* de 600 grf (*Gramas-Força*). Com essa metodologia, busca-se encontrar um controlador ótimo para esse objeto.

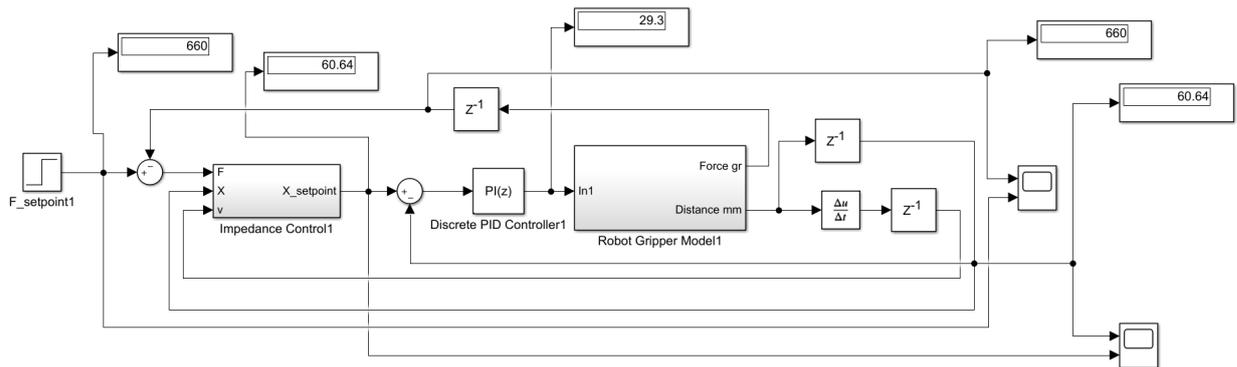


Figura 19 – Modelo de controle de impedância Matlab/Simulink.

## 3.6 Desenvolvimento da Garra Robótica

Nesse sistema, o controle de impedância foi testado juntamente com o detector de deslizamento, ambos embarcados em uma placa Arduino Uno. Salienta-se ainda, que o detector de deslizamentos também foi implementado em um SoC FPGA.

### 3.6.1 Circuito eletrônico da garra robótica

O circuito elétrico da garra robótica é composto por um microcontrolador, sensores, servomotor e fonte de energia (vide Figura 20). Utilizou-se a placa Arduino Uno R3, equipada com um microprocessador *ATmega328P*, que integra o controlador de impedância e o detector de deslizamentos para gerenciar o sistema. Foram usados quatro *FSR400*, dispostos em matrizes de dois sensores nas duas extremidades do efetor final da garra, conectados a resistores de  $10k\Omega$  para assegurar seu bom funcionamento. Adicionalmente, um sensor de posição *SV01* foi conectado ao eixo servomotor e ao eixo central da garra, associado a um capacitor de  $10\mu F$  para reduzir o ruído do sinal. O servomotor *MG995*, com engrenagens metálicas, recebe o sinal PWM do Arduino e é alimentado por uma fonte de energia externa de 6 V, permitindo que o servomotor atinja a força máxima de  $10Kgrf * cm$ .

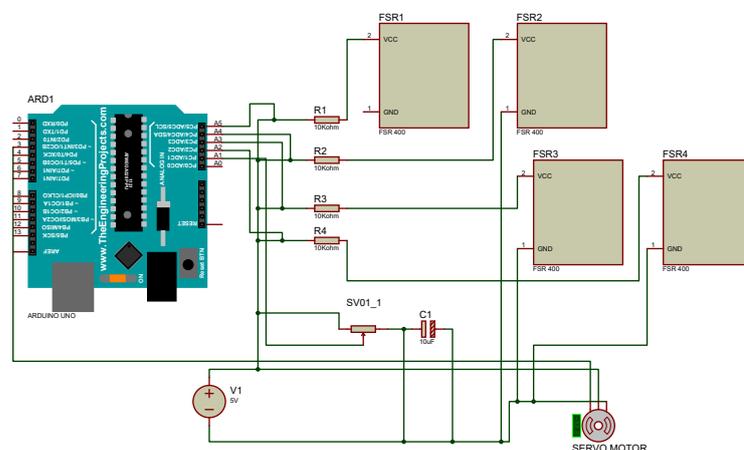


Figura 20 – Circuito eletrônico da garra robótica.

### 3.6.2 Implementação física

Neste trabalho foi utilizado o modelo da garra robótica apresentado na Figura 21. O projeto foi elaborado no SolidWorks e retirado de uma página de projeto CAD (Computer-Aided Design) de acesso gratuito *GradCAD* (GRABCAD-COMMUNITY, 2024).

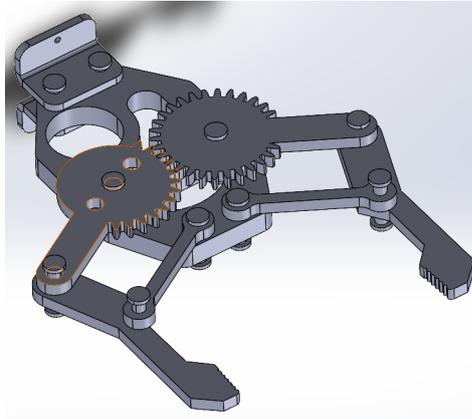


Figura 21 – Design de garra robótica no software CAD SolidWorks (GRABCAD-COMMUNITY, 2024).

Algumas adaptações mecânicas foram realizadas no intuito de acoplar o sensor de posição e os sensores de força. O sensor de posição foi instalado utilizando um parafuso de três milímetros e um eixo acoplador projetado em uma impressora 3D, passando pelo rotor do sensor para se alinhar com o movimento do servomotor. Por outro lado, os sensores de força foram instalados em acoplamentos curvos que se adaptam à forma do efetor final da garra robótica, tal como mostrado na Figura 22. Adicionalmente, a engrenagem esquerda da garra foi redesenhada para corrigir desalinhamentos provocados pela instalação do sensor de posição.

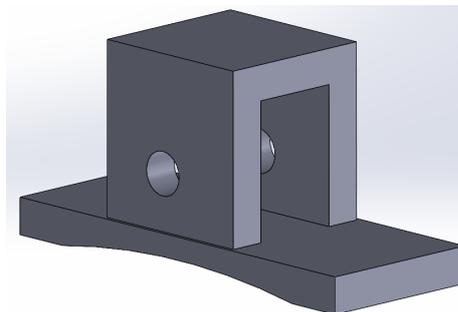


Figura 22 – Design 3D do acoplamento do sensor de força.

A implementação física do sistema é apresentada na Figura 23. Observa-se a garra robótica metálica, o servomotor *MG995* e as duas matrizes de dois sensores de força *FSR400* com fita de silicone instaladas em cada lado da garra por meio de dois acoplamentos fabricados em impressão 3D.

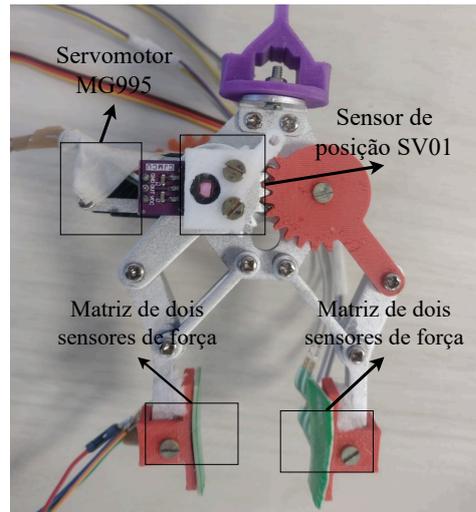


Figura 23 – Conjunto completo da garra robótica com sensores e servomotor.

### 3.6.3 Modelos do sistema robótico

Para o processo de sintonização dos controladores, a garra robótica foi representada usando um modelo matemático para a força e outro modelo para a posição. Esses dois modelos foram obtidos a partir de dados experimentais obtidos diretamente da garra robótica montada com os diferentes sensores, usando o seguinte protocolo experimental: a) foi aplicado um degrau no sinal de PWM variando de zero até dois volts, o qual é suficiente para segurar firmemente um corpo de prova de 250 gr (vide Figura 24); b) os dados de força e posição angular foram coletados no micro-controlador Arduino Uno com uma taxa de amostragem de 10 ms; c) os dados foram exportados para o Matlab onde foi utilizado o *System Identification Toolbox* para identificar os modelos.



Figura 24 – Corpo de prova para experimentos na garra robótica.

Modelo de força: a Equação 3.15 representa o comportamento da garra robótica quando o motor recebe o sinal PWM e a força é transmitida ao efector final que agarrar o corpo de prova, retornando como saída a força estimada.

$$G(z) = \frac{0.2501}{z - 0.8638} z^{(-30)} \quad (3.15)$$

Modelo de posição: a Equação 3.16 representa o comportamento da garra quando o motor é acionado pelo sinal PWM, movimentando o eixo conectado ao sensor de posição.

$$G(z) = \frac{1.211}{z - 0.3684} \quad (3.16)$$

Os modelos de força e de posição alcançaram um percentual de acerto de 83% e 85%, respectivamente.

### 3.7 Proposta de Validação

O processo de validação dos métodos propostos foi realizado em duas etapas. Na primeira etapa foi criada uma base de dados com objetos de uso cotidiano no intuito de ajustar os parâmetros do detector de deslizamentos. Na segunda etapa foi utilizada a garra robótica e um corpo de prova para validar a integração do detector de deslizamentos e o controlador de impedância.

A base de dados foi criada utilizando os sensores de força nas ponta dos dedos indicador e polegar, obtendo dados de força durante o agarre de objetos de uso cotidiano, tais como uma garrafa de água, uma laranja, um telefone, e um livro (vide Figura 25). Durante esse procedimento, foram induzidos manualmente deslizamentos verticais e rotacionais. Os primeiros foram induzidos diminuindo a força de sustentação de cada objeto, permitindo o deslizamento por gravidade.

Um ponto fundamental do algoritmo de detecção de deslizamento e seu ajuste é a possibilidade de alterar vários parâmetros para se adequar a diferentes objetos. Em particular, foram ajustados os limites mínimo (*LimitLow*) e máximo (*LimitHigh*) para detecção de picos dos valores da derivada de primeira ordem, a faixa do contador de amostras (*range*) para determinar se o objeto permanece em deslizamento constante, e o limiar de força (*thresF*) que representa a força mínima necessária para manter o objeto agarrado sem que ele caia.

Após o ajuste do detector de deslizamentos, o mesmo foi implementado no SoC FPGA Zynq 7020 e os circuitos foram caracterizados em termos do consumo de recursos, precisão numérica, e consumo de energia. No final desta etapa, foi conduzida uma validação experimental usando o telefone móvel com três usuários distintos, no intuito de avaliar o comportamento do detector de deslizamentos para diferentes usuários.

Na segunda etapa de validação, o algoritmo de detecção de deslizamentos foi ajustado usando a garra robótica e o corpo de prova de 250 gr que garante o ponto de contato com

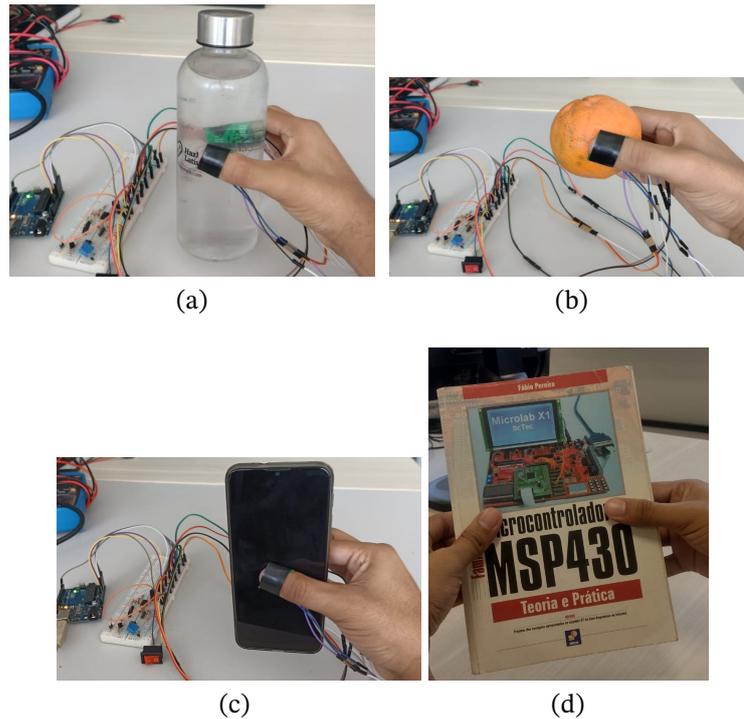


Figura 25 – Objetos para a base de dados experimental. a) Garrafa de água; b) Laranja; c) Telefone móvel; d) Livro.

os sensores FSR. Foram conduzidos dois tipos de experimentos. O primeiro visa validar o comportamento do controlador de impedância. Para tal, foram usados *setpoints* de força de 300 grf, 500 grf, 650 grf, e 750 grf sem induzir deslizamentos. Posteriormente, foram realizados experimentos para validar o controlador de força integrado ao detector de deslizamentos. Neste caso, foi utilizado o corpo de prova de 250 gr e aplicado um *setpoint* de 400 grf. Curvas de resposta do controlador de força e da ação de controle (sinal PWM) foram avaliadas após os deslizamentos verticais induzidos no corpo de prova. No final desta etapa, a acurácia do detector de deslizamentos foi validada usando uma matriz de confusão.

### 3.8 Resumo do Capítulo

Neste capítulo foram apresentados os materiais utilizados na realização do projeto da garra robótica. Foi explicada a construção do detector de deslizamento, sua implementação em hardware e software embarcado, além do controlador de impedância e sua composição. Foram também apresentados os modelos, as simulações, o contexto de montagem da garra robótica e o procedimento experimental para a identificação dos modelos de força e posição da garra robótica.

Adicionalmente, foi apresentado o protocolo para validação experimental do sistema integrado. A Figura 26 ilustra a estrutura geral do funcionamento completo do sistema robótico, abrangendo o controle de força e a detecção de deslizamentos.

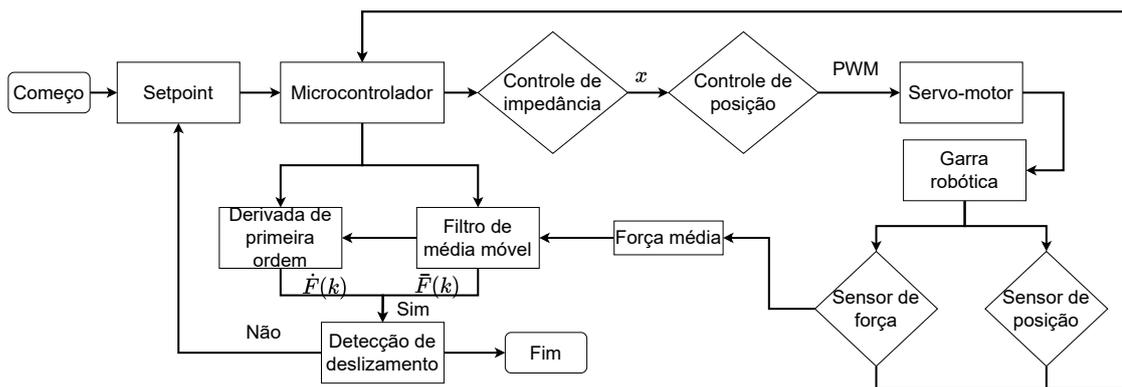


Figura 26 – Diagrama geral do sistema robótico.

## 4 Resultados

Este capítulo está dividido em duas partes: primeiramente, a apresentação dos resultados do modelo de detecção de deslizamentos implementado em hardware (4.1); em segundo lugar, os resultados relacionados ao controle de força e à detecção de deslizamentos implementados em tempo real utilizando um microcontrolador (4.2).

### 4.1 Modelo de Detecção de Deslizamentos

Os resultados a seguir apresentam a implementação em hardware do modelo de detecção de deslizamentos. A Figura 27 apresenta os resultados da ocupação no SoC FPGA. Em cor azul, está a implementação do filtro de média móvel; em cor roxo, a implementação da derivada de primeira ordem; e em cor amarelo, a implementação do detector de picos, enquanto o restante representa a ocupação do protocolo de comunicação *AXI-Lite*. Esses módulos estão representados no lado direito da imagem. No lado esquerdo da imagem, representa-se em cor laranja o processador ARM Cortex A9 e os periféricos de entrada e saída e de interconexão com o PL.

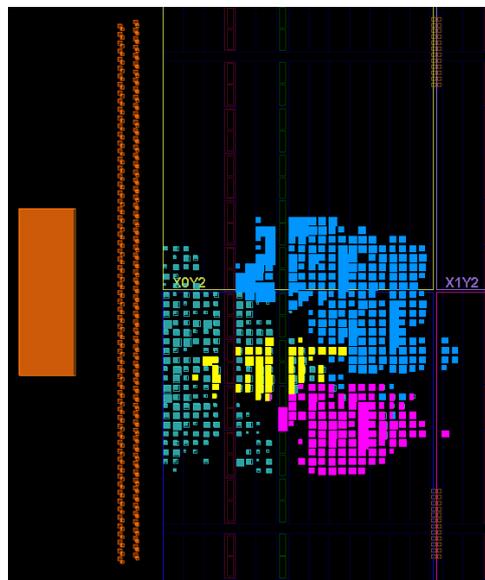


Figura 27 – FPGA-layout dos módulos de detecção de deslizamento. Azul: *Filtro de média móvel*. Roxo: *Derivada de primeira ordem*. Amarelo: *Detector de picos*.

Na Tabela 8 apresentam-se os resultados de utilização de recursos da implementação do sistema de detecção de deslizamentos, discriminando-se a ocupação de recursos de cada um dos módulos implementados. Primeiro, o filtro de média móvel ocupa 800 LUTs representando 1,5%, 378 FFs representando 0,36%, 0 DSPs (0%) e 0 BRAMs (0%). O módulo

de derivada de primeira ordem ocupa com 442 LUTs representando 0,83%, 108 FFs representando 0,10%, 1 DSP representando 0,45% e 0 BRAMs (0%). Finalmente, o módulo de detecção de picos ocupa com 57 LUTs representando 0,11%, 164 FFs representando 0,16%, 0 DSPs e 0 BRAMs. Assim, obtém-se um resultado total de recursos utilizados de 1299 LUTs (2,44%), 650 FFs (0,61%), 1 DSP (0,45%) e 0 BRAMs, sem contabilizar o consumo de recursos do módulo de comunicação *AXI-Interface*. É importante salientar que a interface AXI4-Lite já se encontra presente no projeto da UnB-Hand, portanto, a análise de ocupação pode ser feita usando apenas os módulos do sistema de detecção de deslizamentos. Dessa forma, observa-se que a ocupação de LUTs se manteve abaixo 3%, alcançando esta especificação do projeto elencada na Seção 3.1.

Tabela 8 – Utilização de recursos (chip xc7z020clg484).

Módulo	LUTs (53200)	FFs (106400)	DSPs (220)	BRAMs (140)
Filtro de média móvel	800	378	0	0
Derivada de primeira ordem	442	108	1	0
Detector de picos	57	164	0	0
Subtotal	1299	650	1	0
(%)	2,44%	0,61%	0,45%	0%
AXI Interface	1331	759	0	0
Total	2630	1409	1	0

Na Figura 28 apresenta-se o consumo energético da implementação do sistema de detecção de deslizamento. O consumo de energia está dividido em duas partes: potência dinâmica, com 92% de consumo, e potência estática, com 8% de consumo.

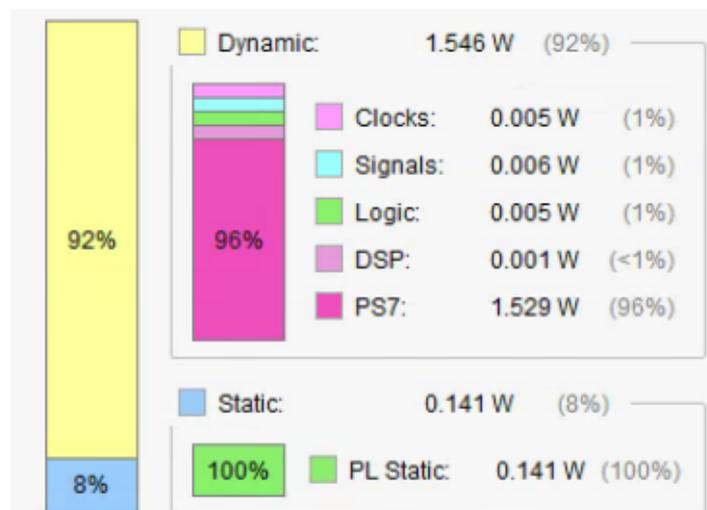


Figura 28 – Consumo de energia da implementação em FPGA do sistema de detecção de deslizamentos.

A potência dinâmica total é de 1,546 W, dos quais o processador ARM (PS7 na figura) requer 96% da energia, enquanto o detector de deslizamentos requer apenas 17 mW. Por

outro lado, a potência estática de 0,141 W, representa a energia necessária para manter o chip FPGA alimentado.

A partir de uma simulação comportamental do sistema de detecção em hardware, busca-se determinar a latência e o throughput da seguinte forma. Na Figura 29, pode-se observar a latência desde a entrada do valor de força média (média das medições dos dois sensores de força) no filtro de média móvel até a saída do valor da detecção de deslizamento (assumindo que a FIFO já está cheia), é de 12 ciclos de clock ou aproximadamente 120 ns para um clock de 100 MHz. A etapa mais lenta do pipeline é o filtro de média móvel, que define o throughput do sistema, processando dados a cada 7 ciclos de clock. Dessa forma, a taxa de processamento é de 14,3 milhões de saídas por segundo. Este valor não implica que o sistema proposto possa detectar 14,3 milhões deslizamentos por segundo, pois é importante esclarecer que a detecção de deslizamentos requer um conjunto de medições que obedece a uma determinada taxa de amostragem, além de depender da dinâmica do deslocamento do objeto enquanto está sendo agarrado.

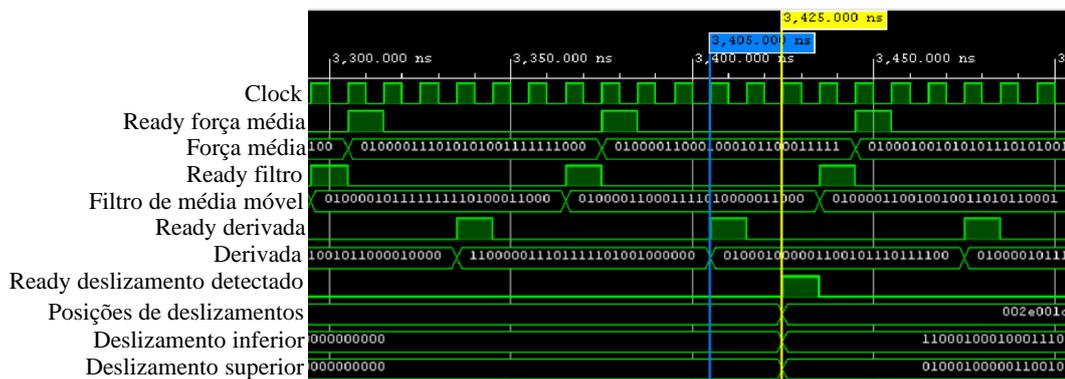


Figura 29 – Zoom da simulação comportamental do sistema de detecção de deslizamentos realizada no software Vivado.

Os resultados da implementação FPGA do sistema de detecção de deslizamentos, composto pelo filtro de média móvel, derivada de primeira ordem e detecção de picos, são apresentados na Figura 30. Destaca-se na figura a comparação entre os resultados obtidos pela implementação em hardware e os resultados em software dos modelos de referência do filtro de média móvel, o qual apresenta um erro MSE (Erro quadrático médio) de  $1,757E-5$ , e da derivada de primeira ordem, o qual apresenta um erro MSE de  $1,004E-4$ . Adicionalmente, na Figura 30b os deslizamentos detectados são indicados por linhas verticais laranjas, indicando o início e fim de cada deslizamento (na figura três deslizamentos são representados).

A Tabela 9 apresenta os resultados obtidos após a aplicação do protocolo de validação do sistema de detecção de deslizamentos, o qual é composto por vários experimentos de deslizamento de um telefone móvel e realizados por três usuários diferentes identificados como A, B e C.

O usuário A realizou três experimentos de deslizamento do objeto e as posições

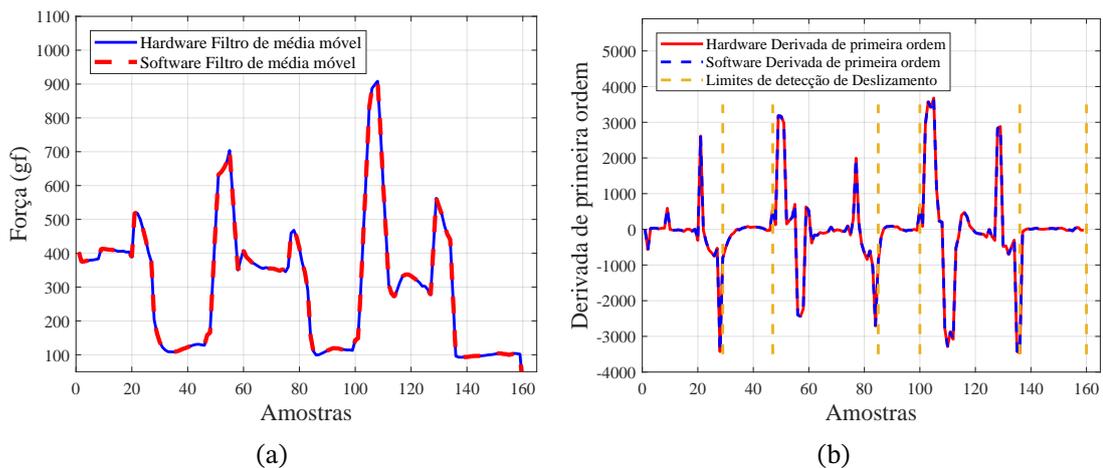


Figura 30 – Resultado da implementação hardware do sistema de detecção de deslizamentos. a) Comparação entre o módulo em hardware de filtro de média móvel em cor azul e modelo de referência em vermelho; b) Comparação entre o módulo em hardware da derivada de primeira ordem em cor azul e o modelo de referência de vermelho. Os pontos de detecção de deslizamento são representados com linhas verticais laranjas.

onde ocorre o deslizamento, chamadas  $pos_L$  e  $pos_H$ , foram comparadas nas suas respectivas implementações em hardware (HW) e em software (SW), obtendo assim o erro MSE em cada posição de deslizamento. Destaca-se que todos os deslizamentos foram detectados em ambos os modelos, representando uma precisão de 100%. O usuário B realizou quatro experimentos de deslizamento, onde todos os deslizamentos foram detectados, alcançando uma precisão de 100%, com o maior valor de MSE sendo  $1,35E-4$ . Por fim, o usuário C realizou cinco experimentos de deslizamento, onde alguns pontos não foram detectados na implementação em hardware no último experimento, resultando em uma precisão de 80% para o algoritmo de detecção de deslizamento. Além disso, neste caso, o valor mais alto de MSE foi de  $1,61E-4$ .

Tabela 9 – Tabela da análise do erro do algoritmo de detecção de deslizamentos. (SW: software; HW: hardware).

Slip	$pos_L$		$pos_H$		MSE	
	SW	HW	SW	HW	$pos_L$	$pos_H$
1A	-796,45	-796,45	613,87	613,87	$3,84E-5$	$1,21E-6$
2A	-901,41	-901,41	618,70	618,70	$1,74E-5$	$2,60E-5$
3A	-3464,79	-3464,78	1863,90	1863,90	$1,39E-4$	$3,03E-5$
1B	-3229,92	-3229,92	107,50	107,50	$3,24E-6$	$1,63E-6$
2B	-3333,46	-3333,45	211,85	211,85	$1,62E-5$	$1,23E-5$
3B	-615,50	-615,49	157,76	157,75	$1,52E-5$	$1,58E-5$
4B	-3192,16	-3192,15	110,0	110,0	$1,35E-4$	$1,29E-24$
1C	-3998,76	-3998,76	458,66	458,66	$4,90E-5$	$1,38E-6$
2C	-757,39	-757,39	387,70	387,70	$1,80E-7$	$6,37E-6$
3C	-2820,75	-2820,74	495,11	495,11	$5,06E-8$	$1,20E-5$
4C	-554	-553,98	289,80	289,80	$1,61E-4$	$6,80E-7$
5C	-574,41	N/A	116,82	N/A	N/A	N/A

## 4.2 Controlador de Força e Detecção de Deslizamentos

A Figura 31 mostra as curvas de resposta do controlador para diferentes valores de referência usando o corpo apresentado na Figura 24. São observados valores de overshoot de 2,45%, 27,62%, 6,70%, e 10,48% para os *setpoints* de 300 grf, 500 grf, 650 grf, e 750 grf, respectivamente. Por outro lado, são observados erros em regime permanente de 7,6 grf, 5,1 grf, 30,8 grf e 25,8 grf para os mesmos *setpoints*, respectivamente.

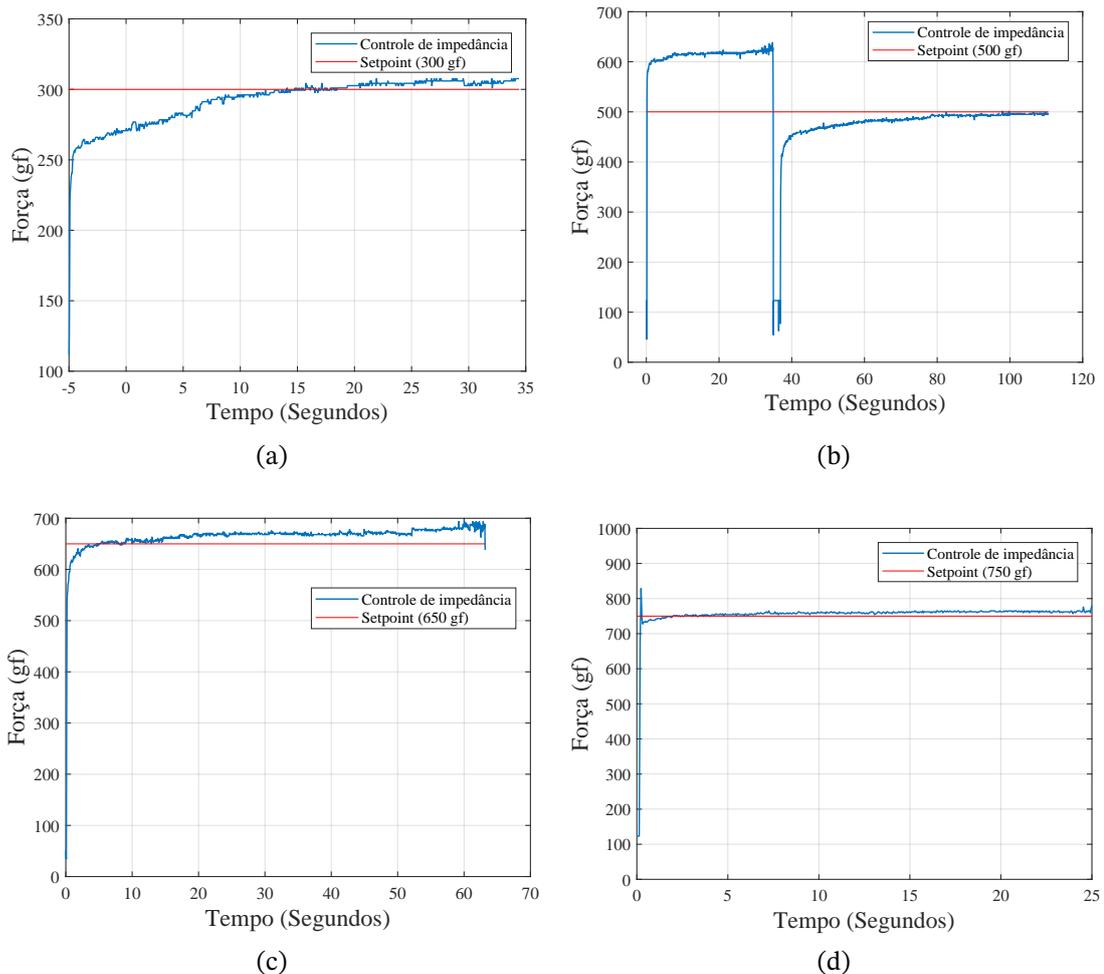


Figura 31 – Respostas do controlador de impedância: 31a) *setpoint* de 300 grf; 31b) *setpoint* de 500 grf; 31c) *setpoint* de 650 grf; 31d) *setpoint* de 750 grf.

Para a experimentação com a implementação utilizando microcontrolador e a garra robótica, foram propostos experimentos de deslizamentos verticais para verificar a eficiência do sistema de detecção de deslizamentos e sua robustez frente a erros. Foram realizados quatro experimentos, nos quais se buscou observar o comportamento do sistema de detecção de deslizamentos em diversos cenários, como deslizamentos múltiplos, deslizamentos acompanhados de perturbações que apresentam características semelhantes a deslizamentos (como agitações) e também de um deslizamento unitário. Os experimentos foram realizados com o corpo de prova e um *setpoint* de 400 grf, sendo que os deslizamentos foram induzidos

após a estabilização do controlador. A duração dos experimentos varia de 1 a 2 minutos.

#### 4.2.1 Experimento 1

No experimento 1, foram induzidos três deslizamentos: 2 a 3 segundos, 11 a 13 segundos e 31 a 33 segundos. A Figura 32b mostra que o modelo em Matlab detectou quatro deslizamentos (não representados na figura), sendo dois reais (primeiro e segundo), e dois falsos positivos os quais estão localizados em 14 a 16 segundos e 21 a 22 segundos. O modelo em tempo real detectou quatro deslizamentos, três reais e um falso positivo o qual está localizado em 15 a 20 segundos, os quais estão representados pelas barras verticais em azul na figura. Observa-se ainda que a resposta do controlador não aumenta a força de controle durante os deslizamentos; só no segundo 20 o PWM é ajustado pelo controlador, conforme mostrado na Figura 32a e 32c. O MSE do filtro de média móvel é  $9,17E-03$  e o da derivada de primeira ordem é  $8,14E-05$ .

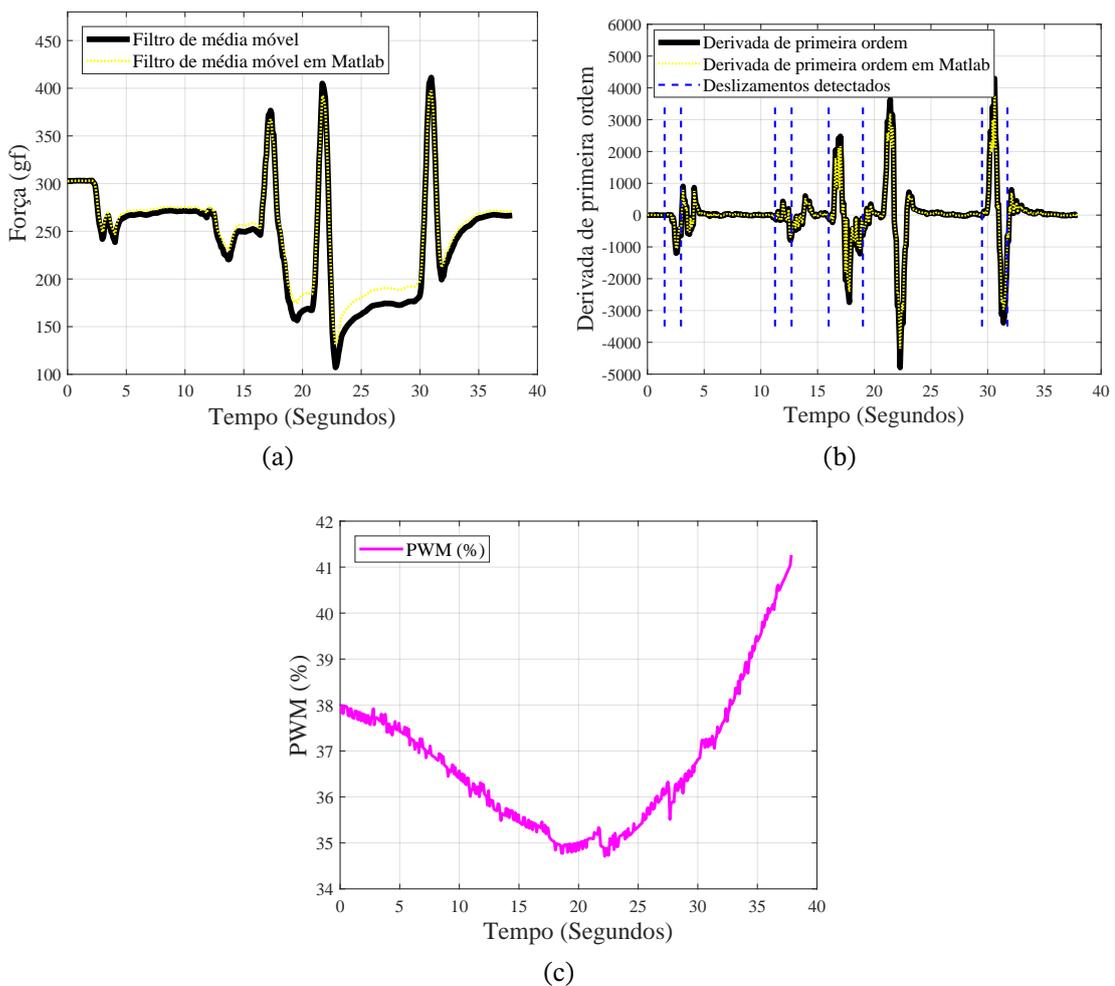


Figura 32 – Resultados do experimento de deslizamento N° 1. 32a) Filtro de média móvel; 32b) Derivada de primeira ordem com os deslizamentos detectados pelo modelo em tempo real apresentados em azul; 32c) Saída do controlador de impedância PWM (%).

## 4.2.2 Experimento 2

No experimento 2, foram induzidos seis deslizamentos, localizados da seguinte forma: o primeiro nos segundos 4 e 6, o segundo nos segundos 40 e 42, o terceiro nos segundos 113 e 115, o quarto nos segundos 116 e 118, o quinto nos segundos 119 e 124, e o sexto nos segundos 127 e 143. O modelo de Matlab detectou apenas o primeiro e o sexto deslizamentos, além de um falso positivo entre 118 e 119 s.

A Figura 33b mostra que o modelo em tempo real detecta todos os deslizamentos e um falso positivo entre 123 e 124 s. Observa-se que o controlador ajusta a força a partir do segundo 10 e conseqüentemente o PWM aumenta até o segundo 40 devido à queda de força causada pelo primeiro deslizamento (vide Figuras 33a e 33c). O MSE da derivada de primeira ordem é  $3,15E-04$  e o MSE do filtro de média móvel é  $2,39E-03$ .

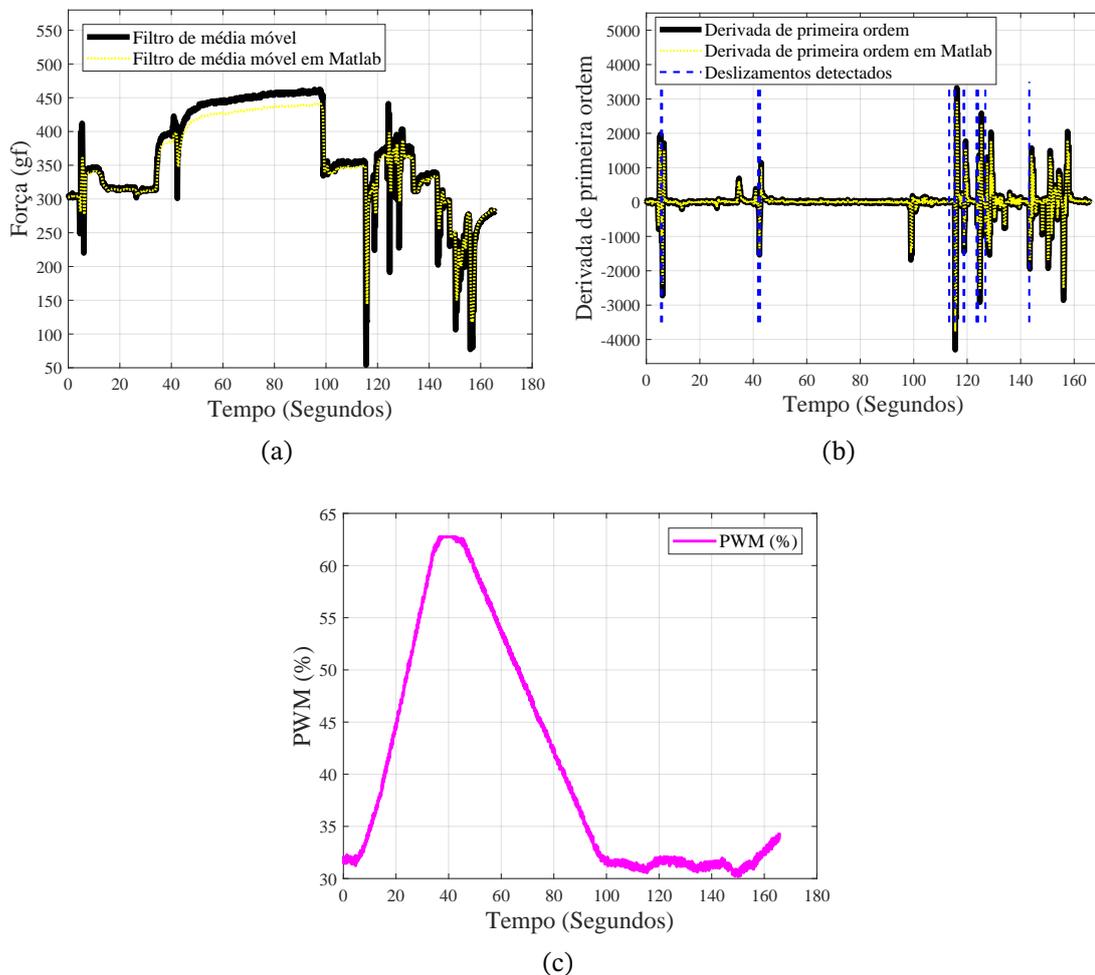


Figura 33 – Resultados do experimento de deslizamento N<sup>o</sup> 2. 33a) Filtro de média móvel; 33b) Derivada de primeira ordem com os deslizamentos detectados pelo modelo em tempo real apresentados em azul; 33c) Saída do controlador de impedância PWM (%).

### 4.2.3 Experimento 3

No experimento 3 foram induzidos dois deslizamentos, localizados entre os segundos 8 e 10, e entre os segundos 37 e 40. Tanto o modelo de Matlab quanto o modelo em tempo real detectam os deslizamentos. Estes últimos estão representados Figura 34b. Analisando o comportamento do controlador, observa-se que, entre os segundos 8 e 10, a força diminui, assim como o valor do PWM. Isso é causado pelo efeito do primeiro deslizamento no sistema. Após o segundo 10, o controlador aumenta o PWM para incrementar a força, até o segundo 30, quando o valor do PWM começa a diminuir até o segundo 40. Depois disso, a ação de controle se mantém estável até o segundo 48, quando há uma grande diminuição na força, devido à perda de contato do sensor com o objeto durante o segundo deslizamento. Em seguida, o valor do PWM aumenta e se mantém constante até o segundo 60, conforme ilustrado nas Figuras 34a e 34c. Adicionalmente, o erro MSE da derivada de primeira ordem é de  $7,17E-05$ , e o erro do filtro de média móvel entre os modelos é de  $9,20E-03$ .

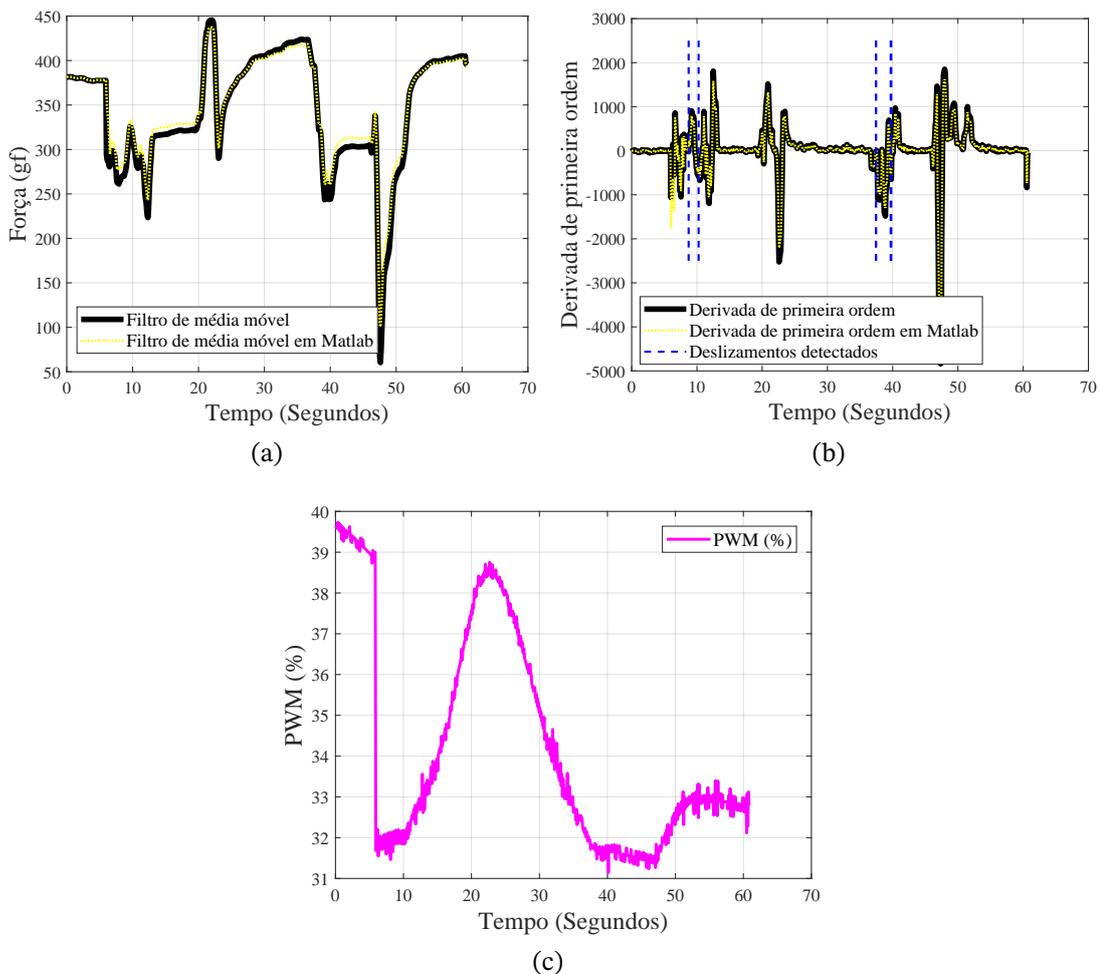


Figura 34 – Resultados do experimento de deslizamento N° 3: 34a) Filtro de média móvel; 34b) Derivada de primeira ordem com os deslizamentos detectados pelo modelo em tempo real apresentados em azul; 34c) Saída do controlador de impedância PWM (%).

#### 4.2.4 Experimento 4

No experimento 4 foi induzido apenas um único deslizamento localizado entre os segundos 17 e 19. Neste caso, o modelo em Matlab detecta o deslizamento (vide 35b), enquanto o modelo em tempo real não é capaz de detectá-lo. Analisando o sistema de controle, observa-se que, no momento em que o deslizamento ocorre, o PWM e a força caem e se mantêm constantes até o segundo 40, quando o sistema tenta novamente alcançar o valor adequado de *setpoint*. Isso pode ser observado nas Figuras 35a e 35c. Entretanto, ressalta-se que, embora o PWM tenha aumentado a partir do segundo 40, a força medida pelos sensores não acompanhou o comportamento esperado, possivelmente pela perda dos pontos de contato com o objeto. Adicionalmente, é apresentado um erro MSE de  $4,76E-04$  entre os modelos de derivada de primeira ordem, enquanto o MSE dos modelos de filtro de média móvel é de  $2,80E-03$ .

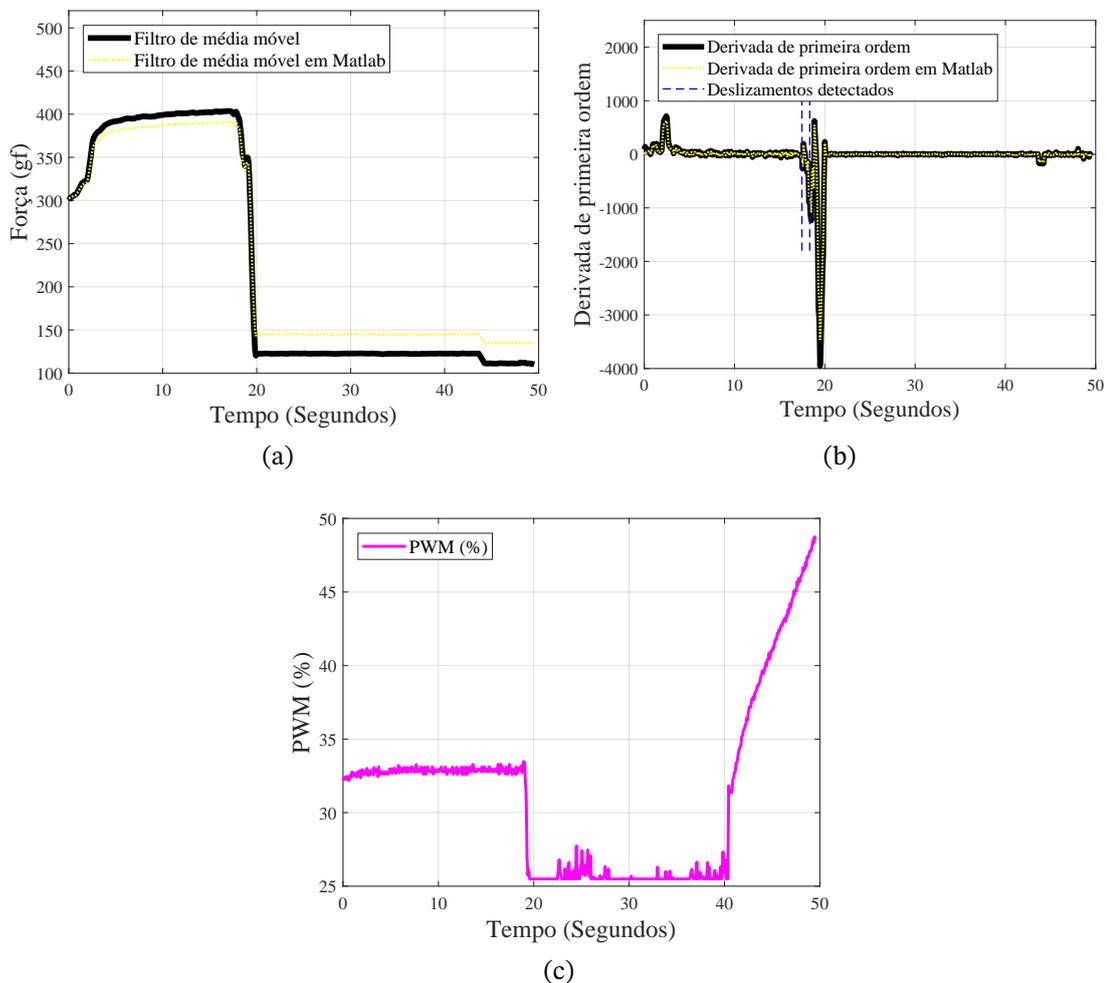


Figura 35 – Resultados do experimento de deslizamento N<sup>o</sup> 4. 35a Filtro de média móvel; 35b Derivada de primeira ordem com os deslizamentos detectados pelo modelo de Matlab em azul; 35c Saída do controlador de impedância PWM (%).

Foi realizada uma validação da resposta do filtro de média usando o indicador SNR (*Signal-to-Noise Ratio*), que atribui um valor numérico à qualidade do sinal, resultando

em um SNR de 13,55 dB. Um valor entre 10 dB e 20 dB indica presença de ruído, mas o sinal permanece utilizável. Um vídeo demonstrativo do controlador de impedância proposto reagindo a deslizamentos verticais induzidos no corpo de prova pode ser observado no seguinte [link](#) (MENDOZA, 2024).

#### 4.2.5 Resumo dos resultados

Na Tabela 10 apresenta-se uma compilação dos resultados dos experimentos de deslizamentos, classificados como A, B, C e D. O experimento A apresenta três deslizamentos correspondentes à Figura 32b, onde o sistema em tempo real detecta os três deslizamentos, enquanto o modelo de Matlab detecta apenas dois. No experimento B, são representados seis deslizamentos correspondentes ao gráfico 33b, todos detectados pelo modelo em tempo real, enquanto o modelo de Matlab detecta apenas três deslizamentos. No experimento C, representado no gráfico 34b, apresentam-se dois deslizamentos, ambos detectados pelos dois modelos, porém em posições distintas dentro do intervalo em que os deslizamentos ocorreram. Finalmente, o experimento D, representado no gráfico 35b, apresenta apenas um deslizamento, que é detectado pelo modelo de Matlab, enquanto o modelo em tempo real não conseguiu detectá-lo.

Tabela 10 – Tabela de resultados dos experimentos de detecção de deslizamentos do modelo do sistema de controle da garra robótica, comparando o modelo de referência e o modelo em tempo real.

Exp	$pos_L$		$pos_H$	
	Matlab	microcontrolador	Matlab	microcontrolador
1A	-683,38	-647,95	-8,81	-10,10
2A	-637,56	-672,12	-36,94	-42,18
3A	N/A	-635,54	N/A	-33,45
1B	-696,56	-689,97	-9,31	-13,18
2B	N/A	-643,07	N/A	-39,55
3B	-615,06	-606,38	-22,94	-24,58
4B	N/A	-668,53	N/A	-13,45
5B	N/A	-632,97	N/A	-13,38
6B	-677,06	-652,58	-40,38	-46,16
1C	-633,56	-595,90	-11,13	-22,51
2C	-680,69	-610,98	-64,99	-23,43
1D	-697,50	N/A	-22,50	-25,73

Na Tabela 11 apresenta-se a matriz de confusão para o modelo em tempo real. Do total de 12 deslizamentos induzidos no corpo de prova, aconteceram 11 verdadeiros positivos (detecções corretas), um falso negativo (deslizamento que não foi detectado), e dois falsos positivos (o sistema estimou dois deslizamentos que não foram induzidos). Dessa forma, obteve-se uma precisão de 84,62%, um *Recall* de 91,67%, e um *F1 score* de 88%.

Tabela 11 – Gráfico de confusão para o modelo em tempo real.

	Previsão: sim	Previsão: não
Realidade: sim	11 (verdadeiro positivo)	1 (falso negativo)
Realidade: não	2 (falso positivo)	N/A (verdadeiro negativo)

A matriz de confusão representa a confiabilidade do sistema de detecção de deslizamentos de objetos de acordo com o número de acertos na detecção e o número de falhas de detecção que não representam um deslizamento. Essa medida permite realizar comparações com algoritmos de detecção de deslizamentos com plataformas robóticas com elementos rígidos.

Na Tabela 12 apresenta os principais trabalhos que reportam precisão na detecção de deslizamentos. Este trabalho atinge uma precisão média de 93,33% em sua implementação de hardware usando a base de dados de deslizamentos, superando trabalhos que obtêm precisões médias de aproximadamente 91,40% (ABD et al., 2018), 89,96% (KHIN et al., 2021), além de ser comparável com os resultados reportados em (HU et al., 2024) (entre 71,7% e 95,6%), (WANG, Q. et al., 2024) (95,6%) e (WATERS; JONES; ALAZMANI; CULMER, P., 2022) (70% até 100%). No entanto, a precisão do algoritmo de detecção na implementação na garra robótica foi reduzida para 84,62% devido à presença de falsos positivos.

Tabela 12 – Trabalhos correlatos que reportam precisão na detecção de deslizamentos.

Ref., ano	Técnica de detecção	Precisão
(KHIN et al., 2021), 2021	Classificação e detecção de deslizamentos com redes neurais de LSTM	89,96%
(ABD et al., 2018), 2018	Classificação de estados deslizantes usando redes neurais artificiais	91,40%
(WANG, Q. et al., 2024), 2024	Desenvolvimento de algoritmo para detecção de deslizamentos incipientes utilizando redes neurais	95,6%
(HU et al., 2024), 2024	Aprendizagem de máquina combinada com taxa de variação da entropia para detecção e prevenção de deslizamentos	71,7% até 95,6%
(WATERS; JONES; ALAZMANI; CULMER, P., 2022), 2021	Razão da velocidade entre regiões de medição usando sensores táteis	77% até 100%
Este trabalho (MENDOZA-PEÑALOZA; MUÑOZ, D. M., 2023), 2023	Primeira derivada da força e análise de picos	84% até 93,3%

## 5 Conclusões

Neste trabalho, desenvolveu-se um sistema de detecção de deslizamentos de objetos para sistemas robóticos, o qual foi implementado em um SoC FPGA para integrar-se ao projeto UnB-Hand (PERTUZ; LLANOS, C. H.; MUÑOZ, D. M., 2021) (PERTUZ; LLANOS, C.; PEÑA, C. A. et al., 2018). Para a realização dos testes de funcionamento do projeto, foi desenvolvida uma garra robótica com a finalidade de integrar o sistema de detecção de deslizamentos a um controlador de força para o agarre de objetos. A primeira etapa do desenvolvimento do detector de deslizamentos envolveu a utilização de sensores de força resistivos (FSR) nos dedos indicador e polegar, de forma a coletar dados durante a execução de deslizamentos com diversos tipos de objetos. Em seguida, com base nessas informações, desenvolveu-se um algoritmo de detecção de baixa complexidade computacional.

O algoritmo de detecção de deslizamentos foi desenvolvido em três etapas principais. Primeiramente, implementou-se um filtro de média móvel para reduzir o ruído das medições dos sensores de força. Em seguida, realizou-se a derivada de primeira ordem da média da força. Por fim, criou-se um algoritmo para detectar picos no sinal da derivada, baseado em quatro parâmetros principais: dois limites (inferior e superior) para o valor da derivada, definidos com base nos dados coletados; um intervalo de amostras que representa a duração do deslizamento, além de um limiar de força que indica que, mesmo com certa força de agarre, o objeto continua deslizando.

Após o desenvolvimento do sistema de detecção de deslizamentos, procedeu-se à sua implementação em uma placa de desenvolvimento SoC FPGA Zynq-7020 (chip xc7z020clg484). Nessa etapa, foram mapeados em hardware os três módulos principais: o filtro de média móvel, a derivada de primeira ordem e o detector de picos no sinal da derivada. Em seguida, foi desenvolvido um co-projeto hardware/software por meio de uma interface *AXI4-Lite* para integrar o detector de deslizamento a um processador ARM Cortex A9. Para fins de validação, no processador ARM foram armazenados os dados de deslizamento usando a memória DDR e, posteriormente, enviados para o detector de deslizamento em hardware. Nas implementações de hardware foi usada uma representação aritmética em ponto flutuante de 27-bits. Em relação à ocupação total de recursos no FPGA, os circuitos propostos ocupam 2,44% dos LUTs. Esse fator é o mais relevante para a integração do modelo de detecção de deslizamento com o restante dos módulos de hardware do projeto UnB-Hand, tais como filtros Kalman, estimadores de intenção de movimento e, controladores de posição e força, os quais ocupam aproximadamente 96% dos recursos disponíveis. Apesar de a ocupação estar no limite da capacidade da FPGA, a integração de ambos os sistemas ainda é possível. Pode-se também afirmar que, se a precisão do algoritmo de detecção de deslizamentos for reduzida, o consumo de recursos da FPGA poderá ser diminuído, uma vez que seria possível

utilizar um tamanho de palavra menor. Esta também poderia ser outra alternativa para que o algoritmo de detecção de deslizamentos seja integrado ao projeto UnB-Hand.

Em termos de validação, realizou-se uma comparação entre os resultados obtidos com o modelo de referência em Matlab e os dados resultantes da implementação na SoC FPGA. Conforme apresentado na Figura 30, o erro máximo na estimação das posições de início e fim dos deslizamentos foi de  $1,61E-4$ . Finalmente, o sistema de detecção de deslizamentos implementado em FPGA foi validado para três usuários manipulando um smartphone (vide Tabela 9), alcançando uma precisão de 100% de detecção para dois usuários e de 80% para o terceiro usuário. Dessa forma foi alcançada uma precisão média de 93,33%. É importante salientar que a implementação em hardware utilizou um clock de 100 MHz. Entretanto, o clock mínimo para o sistema pode ser de até 100 kHz, já que os deslizamentos podem ser detectados em escalas de tempo de milissegundos. Nesse contexto, ressalta-se que a redução da frequência do clock pode levar a uma diminuição no consumo de energia do algoritmo de detecção de deslizamentos.

Dando continuidade ao desenvolvimento do projeto, foram identificados modelos da garra robótica baseados em dados experimentais. O primeiro modelo estima a força alcançada pelo efector final da garra robótica em função do sinal PWM injetado no motor da garra. O segundo modelo corresponde à estimação da posição angular do eixo da garra robótica em função do sinal PWM injetado no motor da garra. Esses modelos foram desenvolvidos com o objetivo de criar um sistema de otimização, baseado no algoritmo de otimização por enxame de partículas, para encontrar os valores dos parâmetros que compõem o controle de impedância.

Finalmente, procedeu-se com a integração do controle de impedância sintonizado e do sistema de detecção de deslizamentos em um microcontrolador para o gerenciamento da garra robótica. Inicialmente, foram realizados experimentos para demonstrar o funcionamento do controle de impedância diante de diferentes *setpoints* de força, apresentando um erro em regime permanente entre 7,6 grf e 30,8 grf, overshoots entre 2,4% e 27,6%, e tempos de resposta de até 20 segundos para *setpoints* de 300 grf e 500 grf, e tempos de resposta inferiores a 5 segundos para *setpoints* de 650 grf ou 750 grf. Os tempos elevados de resposta do controlador, são causados por diversas razões, tais como falhas mecânicas, desgaste dos sensores, perda do ponto de contato entre os sensores e a superfície dos objetos, ou ajustes incorretos dos parâmetros de controle de impedância. Cabe destacar que a garra robótica pode ser utilizada em aplicações de mecânica lenta, como em tarefas de *pick and place*. Em seguida, foram realizados experimentos com deslizamentos verticais induzidos para validar o desempenho do controlador integrado ao sistema de detecção, comparando os resultados obtidos com os modelos de referência de detecção desenvolvidos no Matlab.

Ao comparar os resultados alcançados com outros estudos que também desenvolvem métodos de detecção de deslizamentos, este trabalho alcança uma precisão média de

93,33% em sua implementação em Hardware, superando trabalhos que obtêm precisões de aproximadamente 91,40% (ABD et al., 2018) e 89,96% (KHIN et al., 2021). Contudo, a precisão do algoritmo de detecção na implementação na garra robótica foi reduzida para 84,62% devido à presença de falsos positivos, ou seja, detecções de deslizamentos que, na verdade, não estavam ocorrendo. Outros trabalhos relacionados que apresentam precisão no algoritmo de detecção reportam valores de 95,6% (WANG, Q. et al., 2024), 71,7% até 95,6% (HU et al., 2024) e 77% até 100% (WATERS; JONES; ALAZMANI; CULMER, P., 2022), os quais são comparáveis aos valores de precisão apresentados neste trabalho.

No âmbito da implementação conjunta do controle de força e do sistema de detecção de deslizamentos em um microcontrolador para a garra robótica, este trabalho valida o funcionamento da integração dos modelos propostos. Entretanto, os seguintes aspectos devem ser destacados: i). Ao analisar o uso do sensor na aquisição de dados, observou-se que a aplicação dos sensores nos dedos de uma mão humana difere da utilização na garra robótica, mesmo com a utilização da fita de silicone em ambos os casos. A principal diferença reside na superfície de contato e na dureza desta. Nos dedos, a superfície é suave, permitindo um contato completo com o objeto. Para replicar essa metodologia na garra robótica, optou-se por usar um corpo de prova e um acoplamento para o elo da garra, ambos impressos em 3D, visando garantir que os sensores tenham contato total com a superfície do objeto; ii). Os parâmetros para a detecção de deslizamentos são baseados em dados e, portanto, não são universais para todos os objetos e usuários, pois não sempre se aplica a mesma força para agarrar o mesmo objeto. Por essa razão, os parâmetros do detector de deslizamentos podem ser ajustados para diferentes objetos ou usuários; iii). Foi possível verificar que, mesmo que o instante de tempo em que um deslizamento é identificado pelo modelo de referência em Matlab e o modelo em tempo real sejam distintos, ambos podem estar detectando o mesmo deslizamento, uma vez que se localizam em regiões onde se previa que o deslizamento tivesse ocorrido.

## 5.1 Trabalhos futuros

Para trabalhos futuros, são propostos os seguintes desenvolvimentos:

1. Investigar outras metodologias de controle de impedância que possam proporcionar melhores resultados ou que sejam mais simples de implementar em sistemas robóticos;
2. Incorporar metodologias de aprendizado por demonstração ao projeto UnB-Hand para permitir o agarre de objetos de diferentes tamanhos e formas, também para ajustar os parâmetros do sistema de detecção de deslizamentos;
3. Sugere-se a realização de outros trabalhos que possam melhorar o controle de força e a detecção de deslizamentos, por meio da implementação de uma função sensorial

integrada que utilize diversos sensores, como sensores de corrente, de posição, câmeras para retroalimentação visual, ou sensores táteis *PapillArray*, que são apresentados na literatura referenciada, e são empregados para a detecção de deslizamentos (WANG, Q. et al., 2024).

4. Implementar técnicas de reconfiguração dinâmica parcial para reduzir o consumo de recursos no FPGA. Isto permitiria a integração com interfaces de acionamento para a mão robótica UnB-Hand, como por exemplo, a luva eletrônica desenvolvida em (TEIXEIRA; MUÑOZ, D., 2023) ou a interface miocinética baseada em imãs implantados apresentada em (PERTUZ; GHERARDINI et al., 2022).

## Referências

- ABD, M. A.; GONZALEZ, I. J.; COLESTOCK, T. C.; KENT, B. A.; ENGERBERG, E. D. Direction of slip detection for adaptive grasp force control with a dexterous robotic hand. In: IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics, AIM. Ago. 2018. P. 21–27. DOI: [10.1109/AIM.2018.8452704](https://doi.org/10.1109/AIM.2018.8452704). Citado nas pp. 27, 33, 41, 70, 73.
- ABDULKAREEM S. AL-SHANOON, A.; AHMAD, S. A.; HASSAN, M. K. B. Re-gripping analysis based on implementation of slip-detection device for robotic hand model. In: 2016 IEEE Region 10 Symposium (TENSYP). 2016. P. 203–206. DOI: [10.1109/TENCONSpring.2016.7519405](https://doi.org/10.1109/TENCONSpring.2016.7519405). Citado nas pp. 29, 33.
- AMD-XILINX. **AMD Zynq™ 7000 SoCs with Hardware and Software Programmability**. acessado em 2024/12/07. 2024. Disponível em: <https://www.amd.com/en/products/adaptive-socs-and-fpgas/soc/zynq-7000.html#tabs-bea253336e-item-66aaed9bf5-tab>. Citado na p. 34.
- COSTA, R. B.; ITACARAMBY, G.; CABRAL, J.; PASTRANA, M. A.; PEÑALOZA, J. M.; MUÑOZ, D. M. Design and Construction of a Low-Cost Combat Robot. In: 16TH Brazilian Symposium on Robotics (SBR). 2024. Citado na p. 19.
- ELECTRONICS, I. **Sensor Interlink Electronics FSR 400**. acessado em 2024/12/07. 2024. Disponível em: <https://www.interlinkelectronics.com/fsr-400>. Citado na p. 41.
- ERBAS, Í.; VARGAS, D. A.; GÜÇLÜ, B. FPGA Implementation of Multinomial Logistic Regression For Vibrotactile Feedback In a Robotic Hand. In: 2020 E-Health and Bioengineering IEEE International Conference (EHB). 2020. DOI: [10.1109/EHB50910.2020.9280239](https://doi.org/10.1109/EHB50910.2020.9280239). Citado na p. 36.
- GRABCAD-COMMUNITY. **Bibliotecas de acesso aberto da comunidade GrabCAD**. 2024. acessado em 2024/12/07. Disponível em: <https://grabcad.com/library>. Citado na p. 55.
- HOU, Q.; FU, Y.; LUO, M.; LU, C.; LI, G.; SUN, L. FBG-Based Methods for Monitoring Critical Slip Point during Flexible Fingers Clamping. **IEEE Sensors Journal**, Institute of Electrical e Electronics Engineers Inc., v. 23, p. 28067–28074, 22 nov. 2023. ISSN 1558-1748. DOI: [10.1109/JSEN.2023.3316280](https://doi.org/10.1109/JSEN.2023.3316280). Citado nas pp. 31, 33.
- HU, X.; VENKATESH, A.; WAN, Y.; ZHENG, G.; JAWALE, N.; KAUR, N.; CHEN, X.; BIRKMEYER, P. Learning to detect slip through tactile estimation of the contact force field and its entropy properties. **Mechatronics**, Elsevier, v. 104, dez. 2024. ISSN 0957-4158. DOI: [10.1016/j.mechatronics.2024.103258](https://doi.org/10.1016/j.mechatronics.2024.103258). Citado nas pp. 32, 33, 70, 73.

- HUANG, H.; JIANG, L.; LIU, Y.; HOU, L.; CAI, H.; LIU, H. The Mechanical Design and Experiments of HIT/DLR Prosthetic Hand. In: IEEE Int. Conf. on Robotics and Biomimetics. 2006. P. 896–901. DOI: [10.1109/ROBIO.2006.340339](https://doi.org/10.1109/ROBIO.2006.340339). Citado nas pp. 35, 36.
- KENT, B. A.; ENGERBERG, E. D. Robotic Hand Acceleration Feedback to Synergistically Prevent Grasped Object Slip. **IEEE Transactions on Robotics**, Institute of Electrical e Electronics Engineers Inc., v. 33, p. 492–499, 2 abr. 2017. ISSN 1552-3098. DOI: [10.1109/TR0.2016.2633574](https://doi.org/10.1109/TR0.2016.2633574). Citado nas pp. 28, 29, 33.
- KHIN, P. M.; LOW, J. H.; ANG, M. H.; YEOW, C. H. Development and Grasp Stability Estimation of Sensorized Soft Robotic Hand. **Frontiers in Robotics and AI**, Frontiers Media S.A., v. 8, mar. 2021. ISSN 2296-9144. DOI: [10.3389/frobt.2021.619390](https://doi.org/10.3389/frobt.2021.619390). Citado nas pp. 26, 33, 41, 70, 73.
- KNOSPE, C. PID control. **IEEE Control Systems Magazine**, v. 26, n. 1, p. 30–31, 2006. DOI: [10.1109/MCS.2006.1580151](https://doi.org/10.1109/MCS.2006.1580151). Citado na p. 24.
- KOBAYASHI, D.; WATANABE, K.; KOBAYASHI, K.; KURIHARA, Y. Impedance force control of robot hand using pressure sensor. In: SICE Annual Conference 2011. 2011. P. 1981–1984. Citado nas pp. 25, 26.
- LAN, T.; LIU, Y.; JIN, M.; FAN, S.; FANG, H.; XIA, J.; LIU, H. DSP&FPGA-based joint impedance controller for DLR/HIT II dexterous robot hand. In: IEEE/ASME Int. Conf. on Advanced Intelligent Mechatronics. 2009. P. 1594–1599. DOI: [10.1109/AIM.2009.5229817](https://doi.org/10.1109/AIM.2009.5229817). Citado nas pp. 35, 36.
- LEE, D. H.; PARK, J. H.; PARK, S. W.; BAEG, M. H.; BAE, J. H. KITECH-Hand: A Highly Dexterous and Modularized Robotic Hand. **IEEE/ASME Trans. on Mechatronics**, v. 22, p. 876–887, 2017. DOI: [10.1109/TMECH.2016.2634602](https://doi.org/10.1109/TMECH.2016.2634602). Citado nas pp. 21, 26.
- LI, J.; DONG, S.; ADELSON, E. Slip Detection with Combined Tactile and Visual Information. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). 2018. P. 7772–7777. DOI: [10.1109/ICRA.2018.8460495](https://doi.org/10.1109/ICRA.2018.8460495). Citado nas pp. 30, 33.
- LIU, H.; WU, K.; MEUSEL, P.; SEITZ, N.; HIRZINGER, G.; JIN, M.; LIU, Y.; FAN, S.; LAN, T.; CHEN, Z. Multisensory five-finger dexterous hand: The DLR/HIT Hand II. In: 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems. 2008. P. 3692–3697. DOI: [10.1109/IR0S.2008.4650624](https://doi.org/10.1109/IR0S.2008.4650624). Citado nas pp. 35, 36.
- MCCLAVE, J.; SINCICH, T. Statistics. In: 13. ed.: Pearson Education, 2021. cap. 14. Citado na p. 43.
- MENDOZA, J. **Controle de força e detecção de deslizamentos de objetos com garra robótica multisensorial | LabMicro FGA**. 2024. Online. Disponível em: [https://www.youtube.com/watch?v=PuRsd6q3FkA&ab\\_channel=Laborat%C3%B3riodeMicroeletr%C3%B4nicaUnB](https://www.youtube.com/watch?v=PuRsd6q3FkA&ab_channel=Laborat%C3%B3riodeMicroeletr%C3%B4nicaUnB). Acesso em: 7 dez. 2024. Citado na p. 69.

- MENDOZA-PEÑALOZA, J.; MUÑOZ, D. M. Hardware Implementation of a Sliding Detection Algorithm for Robotic Hands Using Force Sensors. In: 2023 36th SBC/SB-Micro/IEEE/ACM Symposium on Integrated Circuits and Systems Design (SBCCI). 2023. P. 1–6. DOI: [10.1109/SBCCI60457.2023.10261972](https://doi.org/10.1109/SBCCI60457.2023.10261972). Citado nas pp. 19, 70.
- MUÑOZ, D. M.; SANCHEZ, D. F.; LLANOS, C. H.; AYALA-RINCÓN, M. FPGA based floating-point library for Cordic algorithms. In: IEEE Southern Programmable Logic Conf. 2010a. P. 55–60. DOI: [10.1109/SPL.2010.5483002](https://doi.org/10.1109/SPL.2010.5483002). Citado nas pp. 40, 48.
- MUÑOZ, D. M.; SANCHEZ, D. F.; LLANOS, C. H.; AYALA-RINCÓN, M. Tradeoff of FPGA Design of a Floating-point Library for Arithmetic Operators. **Journal of Integrated Circuits and Systems**, v. 5, p. 42–52, 2010b. ISSN 1807-1953. Citado nas pp. 40, 48.
- MURATA. **Murata Datasheet SV01A103AEA01R00 sensor**. acessado em 2024/12/07. 2024. Disponível em: [https://www.murata.com/-/media/webrenewal/products/sensor/rotaryposition/sv01/datasheet\\_sv01.ashx](https://www.murata.com/-/media/webrenewal/products/sensor/rotaryposition/sv01/datasheet_sv01.ashx). Citado na p. 42.
- NAKAGAWA-SILVA, A.; REDDY SUNKESULA, S. P.; PRACH, A.; CABIBIHAN, J.-J.; THAKOR, N. V.; SOARES, A. B. Slip suppression in prosthetic hands using a reflective optical sensor and MPI controller. In: IEEE Biomedical Circuits and Systems Conf. 2018. P. 1–4. DOI: [10.1109/BIOCAS.2018.8584711](https://doi.org/10.1109/BIOCAS.2018.8584711). Citado nas pp. 27, 33.
- NAKAGAWA-SILVA, A.; THAKOR, N. V.; CABIBIHAN, J. J.; SOARES, A. B. A Bio-Inspired Slip Detection and Reflex-Like Suppression Method for Robotic Manipulators. **IEEE Sensors Journal**, Institute of Electrical e Electronics Engineers Inc., v. 19, p. 12443–12453, 24 dez. 2019. ISSN 1558-1748. DOI: [10.1109/JSEN.2019.2939506](https://doi.org/10.1109/JSEN.2019.2939506). Citado nas pp. 30, 33.
- ORTENZI, D.; SCARCIA, U.; MEATTINI, R.; PALLI, G.; MELCHIORRI, C. Synergy-Based Control of Anthropomorphic Robotic Hands with Contact Force Sensors. **IFAC-PapersOnLine**, v. 52, n. 15, p. 340–345, 2019. 8th IFAC Symposium on Mechatronic Systems MECHATRONICS 2019. ISSN 2405-8963. DOI: <https://doi.org/10.1016/j.ifacol.2019.11.698>. Citado nas pp. 21, 26.
- PASTRANA, M. A.; OLIVEIRA, L. H.; BAUTISTA, J.; PINTO-LOPERA, J.; MENDOZA-PEÑALOZA, J.; MUÑOZ, D. M. Sub-optimal Perceptron Controller Using WOA Algorithm for Farming Watering Mobile Robot. In: 2024 IEEE Latin American Conference on Computational Intelligence (LACCI). 2024. Citado na p. 19.
- PASTRANA, M. A.; SÁNCHEZ, W. H. C.; MENDOZA-PEÑALOZA, J.; SANTANA, M. S.; MUÑOZ, D. M. Improving Control Theory Education Through Face-to-Face Implementation of a Mobile Robotics Methodology. In: 2024 Workshop on Robotics in Education (WRE). 2024. Citado na p. 19.

- PASTRANA, M. A.; BAUTISTA, J.; MENDOZA-PEÑALOZA, J.; OLIVEIRA, L. H.; MENDES, D. A.; MUÑOZ, D. M. Teaching Control Theory using Mobile Robot Obstacle Following/Avoidance with CoppeliaSim and MFO Algorithm. In: 2023 Latin American Robotics Symposium (LARS), 2023 Brazilian Symposium on Robotics (SBR), and 2023 Workshop on Robotics in Education (WRE). 2023. P. 579–584. DOI: [10.1109/LARS/SBR/WRE59448.2023.10333042](https://doi.org/10.1109/LARS/SBR/WRE59448.2023.10333042). Citado na p. 19.
- PASTRANA, M. A.; OLIVEIRA, L. H.; MENDES, D. A.; SILVA, D. L.; MENDOZA-PEÑALOZA, J.; MUÑOZ, D. M. Implementation of a PID Controller using Online Tuning Applied to a Mobile Robot Obstacle Following/Avoidance. In: 2024 Argentine Conference on Electronics (CAE). 2024. P. 10–15. DOI: [10.1109/CAE59785.2024.10487152](https://doi.org/10.1109/CAE59785.2024.10487152). Citado na p. 19.
- PASTRANA, M. A.; OLIVEIRA, L. H.; SANTANA, M. S.; OLIVEIRA, V. C.; MENDOZA-PEÑALOZA, J.; MUÑOZ, D. M. Hardware Implementation of a GMDH Controller for Mobile Robot Obstacle Following/Avoidance. In: 2023 Latin American Robotics Symposium (LARS), 2023 Brazilian Symposium on Robotics (SBR), and 2023 Workshop on Robotics in Education (WRE). 2023. P. 206–211. DOI: [10.1109/LARS/SBR/WRE59448.2023.10332957](https://doi.org/10.1109/LARS/SBR/WRE59448.2023.10332957). Citado na p. 19.
- PERTUZ, S. A.; GHERARDINI, M.; SANTOS, G. V.; MUÑOZ, D. M.; AYALA, H. V. H.; CIPRIANI, C. Data-Driven Real-Time Magnetic Tracking Applied to Myokinetic Interfaces. **IEEE Transactions on Biomedical Circuits and Systems**, v. 16, n. 2, p. 266–274, 2022. DOI: [10.1109/TBCAS.2022.3161133](https://doi.org/10.1109/TBCAS.2022.3161133). Citado nas pp. 14, 15, 36, 40, 74.
- PERTUZ, S. A.; LLANOS, C.; PEÑA, C.; MUÑOZ, D. A parallel system-on-chip approach for impedance controller for a 7-DoF robotic hand. **Analog Integrated Circuits and Signal Processing**, Springer, v. 106, p. 195–204, 1 jan. 2021. ISSN 1573-1979. DOI: [10.1007/s10470-020-01652-7](https://doi.org/10.1007/s10470-020-01652-7). Citado nas pp. 14, 36, 40.
- PERTUZ, S. A.; LLANOS, C.; PEÑA, C. A.; MUÑOZ, D. A Modular and Distributed Impedance Control Architecture on a Chip for a Robotic Hand. In: 2018 31st Symposium on Integrated Circuits and Systems Design (SBCCI). 2018. P. 1–6. DOI: [10.1109/SBCCI.2018.8533266](https://doi.org/10.1109/SBCCI.2018.8533266). Citado nas pp. 14, 15, 36, 40, 71.
- PERTUZ, S. A.; LLANOS, C. H.; MUÑOZ, D. M. Development of a Robotic Hand Using Bioinspired Optimization for Mechanical and Control Design: UnB-Hand. **IEEE Access**, v. 9, p. 61010–61023, 2021. DOI: [10.1109/ACCESS.2021.3073010](https://doi.org/10.1109/ACCESS.2021.3073010). Citado nas pp. 13–15, 36, 39, 71.
- PERTUZ, S. A.; MENDES, D. A.; GHERARDINI, M.; MUÑOZ, D. M.; AYALA, H. V. H.; CIPRIANI, C. Dynamic Reconfiguration for Multi-Magnet Tracking in Myokinetic Prosthetic Interfaces. **IEEE Transactions on Medical Robotics and Bionics**, v. 6, n. 4, p. 1678–1687, 2024. DOI: [10.1109/TMRB.2024.3464093](https://doi.org/10.1109/TMRB.2024.3464093). Citado na p. 15.

- PRIYAMBADA, S.; KUMAR, D. M.; KUMAR, M. P. Design of PD-PID Controller with Double Derivative Filter for Frequency Regulation. In: 2ND IEEE International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES-2018). IEEE, 2018. P. 1142–1147. DOI: [10.1109/ICPEICES.2018.8897276](https://doi.org/10.1109/ICPEICES.2018.8897276). Citado na p. 24.
- ROMEO, R. A.; ZOLLO, L. Methods and Sensors for Slip Detection in Robotics: A Survey. **IEEE Access**, v. 8, p. 73027–73050, 2020. DOI: [10.1109/ACCESS.2020.2987849](https://doi.org/10.1109/ACCESS.2020.2987849). Citado nas pp. 32, 33.
- SARAVATI, L. **Loja de componentes eletrônicos Saravati**. acessado em 2024/12/07. 2024. Disponível em: <https://www.saravati.com.br/garra-robotica-articulada-metal-2dof.html>. Citado na p. 42.
- SERVOMOTOR., M. P. J. **High Speed Servo Actuator MG995 datasheet**. acessado em 2024/12/07. 2024. Disponível em: <https://www.mpja.com/download/31150mp.pdf>. Citado na p. 42.
- SOUSA LUIZ, M. de; PASTRANA, M. A.; SILVA, L. A. D.; CAMPAGNUCCI, P. R. A. N.; MENDOZA-PENALOZA, J.; BENEVIDES, R. R. L.; MUÑOZ, D. M. Comparison of Bioinspired MLP Controllers for Mobile Robots Obstacle Following/Avoidance. In: 2024 IEEE Latin American Conference on Computational Intelligence (LACCI). 2024. Citado na p. 19.
- SUI, R.; ZHANG, L.; HUANG, Q.; LI, T.; JIANG, Y. A Novel Incipient Slip Degree Evaluation Method and Its Application in Adaptive Control of Grasping Force. **IEEE Transactions on Automation Science and Engineering**, IEEE, v. 21, p. 2454–2468, 3 2024. ISSN 1558-3783. DOI: [10.1109/TASE.2023.3261403](https://doi.org/10.1109/TASE.2023.3261403). Citado nas pp. 31, 33.
- TEIXEIRA, N. S.; MUÑOZ, D. **Desenvolvimento de Luva Eletrônica para Controle de Mão Robótica, Trabalho PIBIC-UnB**. Acessado em 2024/12/07. 2023. Disponível em: <https://www.youtube.com/watch?v=RC7TtpnNwPc>. Citado nas pp. 39, 43, 74.
- TERASHIMA, K.; MOURI, K.; MINYONG, P.; KITAGAWA, H.; MIYOSHI, T. Hybrid Impedance Control of Human Skin Muscle by Multi-fingered Robot Hand. **IFAC Proceedings Volumes**, v. 41, n. 2, p. 15742–15749, 2008. 17th IFAC World Congress. ISSN 1474-6670. DOI: <https://doi.org/10.3182/20080706-5-KR-1001.02662>. Citado nas pp. 25, 26.
- VEIGA, F.; AKROUR, R.; PETERS, J. Hierarchical Tactile-Based Control Decomposition of Dexterous In-Hand Manipulation Tasks. **Frontiers in Robotics and AI**, v. 7, article no. 521448, 2020. DOI: [10.3389/frobt.2020.521448](https://doi.org/10.3389/frobt.2020.521448). Citado nas pp. 27, 33.
- VEIGA, F.; PETERS, J.; HERMANS, T. Grip Stabilization of Novel Objects Using Slip Prediction. **IEEE Transactions on Haptics**, IEEE, v. 11, p. 531–542, 4 out. 2018. ISSN 2329-4051. DOI: [10.1109/TOH.2018.2837744](https://doi.org/10.1109/TOH.2018.2837744). Citado nas pp. 29, 33.

- WANG, L.; OHKAWA, T. FPGA-Based Control of Remote Robotic Hands and ROS2 Integration. In: 2022 Tenth International Symposium on Computing and Networking Workshops (CANDARW). 2022. P. 438–440. DOI: [10.1109/CANDARW57323.2022.00090](https://doi.org/10.1109/CANDARW57323.2022.00090). Citado nas pp. 36, 37.
- WANG, Q.; ULLOA, P. M.; BURKE, R.; BULENS, D. C.; REDMOND, S. J. Robust Learning-Based Incipient Slip Detection Using the PapillArray Optical Tactile Sensor for Improved Robotic Gripping. **IEEE Robotics and Automation Letters**, IEEE, v. 9, p. 1827–1834, 2 fev. 2024. ISSN 2377-3766. DOI: [10.1109/LRA.2023.3347141](https://doi.org/10.1109/LRA.2023.3347141). Citado nas pp. 31, 33, 70, 73, 74.
- WANG, X.; ZHAO, Y.; FAN, X.; WU, H. Active gripping impedance force control with dual fingers hand. In: PROCEEDINGS of 2011 International Conference on Electronic and Mechanical Engineering and Information Technology, EMEIT 2011. 2011. v. 9, p. 4531–4534. DOI: [10.1109/EMEIT.2011.6024037](https://doi.org/10.1109/EMEIT.2011.6024037). Citado nas pp. 25, 26.
- WATERS, I.; JONES, D.; ALAZMANI, A.; CULMER, P. Encouraging and Detecting Preferential Incipient Slip for Use in Slip Prevention in Robot-Assisted Surgery. **Sensors**, MDPI, v. 22, 20 out. 2022. ISSN 1424-8220. DOI: [10.3390/s22207956](https://doi.org/10.3390/s22207956). Citado nas pp. 36, 70, 73.
- WATERS, I.; JONES, D.; ALAZMANI, A.; CULMER, P. R. Utilising Incipient Slip for Grasping Automation in Robot Assisted Surgery. **IEEE Robotics and Automation Letters**, IEEE, v. 7, p. 1071–1078, 2 abr. 2022. ISSN 2377-3766. DOI: [10.1109/LRA.2021.3137554](https://doi.org/10.1109/LRA.2021.3137554). Citado nas pp. 30, 33.
- WITTENMARK, B.; ÅSTRÖM, K. J. **Computer Control: An Overview. IFAC PROFESSIONAL BRIEF**. acessado em 2024/12/07. 2001. Disponível em: [https://www.kth.se/social/upload/4f311a97f2765461cf00000d/IFAC\\_control.pdf](https://www.kth.se/social/upload/4f311a97f2765461cf00000d/IFAC_control.pdf). Citado na p. 24.
- YANG, H.; WEI, G.; REN, L.; QIAN, Z.; WANG, K.; XIU, H.; LIANG, W. A low-cost linkage-spring-tendon-integrated compliant anthropomorphic robotic hand: MCR-Hand III. **Mechanism and Machine Theory**, v. 158, p. 104210, abr. 2021. DOI: [10.1016/j.mechmachtheory.2020.104210](https://doi.org/10.1016/j.mechmachtheory.2020.104210). Citado nas pp. 22, 26.
- ZAIDI, L.; CORRALES, J. A.; BOUZGARROU, B. C.; MEZOUAR, Y.; SABOURIN, L. Model-based strategy for grasping 3D deformable objects using a multi-fingered robotic hand. **Robotics and Autonomous Systems**, v. 95, p. 196–206, 2017. ISSN 0921-8890. DOI: <https://doi.org/10.1016/j.robot.2017.06.011>. Citado nas pp. 28, 33.
- ZHAO, D.; JIANG, L.; HUANG, H.; JIN, M.; CAI, H.; LIU, H. Development of a Multi-DOF Anthropomorphic Prosthetic Hand. In: 2006 IEEE International Conference on Robotics and Biomimetics. 2006. P. 878–883. DOI: [10.1109/ROBIO.2006.340336](https://doi.org/10.1109/ROBIO.2006.340336). Citado nas pp. 35, 36.

- ZHOU, Z.; ZUO, R.; YING, B.; ZHU, J.; WANG, Y.; WANG, X.; LIU, X. A Sensory Soft Robotic Gripper Capable of Learning-Based Object Recognition and Force-Controlled Grasping. **IEEE Transactions on Automation Science and Engineering**, IEEE, v. 21, p. 844–854, 1 jan. 2024. ISSN 1558-3783. DOI: [10.1109/TASE.2022.3228255](https://doi.org/10.1109/TASE.2022.3228255). Citado nas pp. 31, 33.
- ZHUO, Q.; TIAN, L.; FANG, P.; LI, G.; ZHANG, X. A piezoelectret-based approach for touching and slipping detection in robotic hands. In: 2015 IEEE International Conference on Cyber Technology in Automation, Control and Intelligent Systems, IEEE-CYBER 2015. Out. 2015. P. 918–921. ISBN 9781479987290. DOI: [10.1109/CYBER.2015.7288067](https://doi.org/10.1109/CYBER.2015.7288067). Citado nas pp. 29, 33.