

**ANDRÉS EDUARDO MARCHETTI COLOGNESI
FELIPE AUGUSTO DE MORAES MACHADO**

**USO DE PRÉ-PROCESSAMENTO NÃO-LINEAR
DE SINAIS DE EEG PARA CONTROLE DE
EXOESQUELETO DE MEMBRO SUPERIOR**

São Paulo
2020

**ANDRÉS EDUARDO MARCHETTI COLOGNESI
FELIPE AUGUSTO DE MORAES MACHADO**

**USO DE PRÉ-PROCESSAMENTO NÃO-LINEAR
DE SINAIS DE EEG PARA CONTROLE DE
EXOESQUELETO DE MEMBRO SUPERIOR**

Trabalho apresentado à Escola Politécnica
da Universidade de São Paulo para obtenção
do Título de Engenheiro Mecatrônico.

São Paulo
2020

**ANDRÉS EDUARDO MARCHETTI COLOGNESI
FELIPE AUGUSTO DE MORAES MACHADO**

**USO DE PRÉ-PROCESSAMENTO NÃO-LINEAR
DE SINAIS DE EEG PARA CONTROLE DE
EXOESQUELETO DE MEMBRO SUPERIOR**

Trabalho apresentado à Escola Politécnica
da Universidade de São Paulo para obtenção
do Título de Engenheiro Mecatrônico.

Área de Concentração:
Engenharia Biomecatrônica

Orientador:
Arturo Forner Cordero

São Paulo
2020

AGRADECIMENTOS

Como este trabalho foi desenvolvido em dupla, decidimos separar aqui um espaço para cada um fazer sua dedicatória. Apesar de termos trilhado boa parte dessa jornada juntos, cada um tem seus agradecimentos particulares a fazer.

Agradecimentos do Andrés:

Este trabalho marca o fim de uma longa e árdua jornada. Foram 5 anos extremamente intensos e de muito aprendizado. O importante para realmente fazer valer a pena todo o trabalho e empenho investidos nesses 5 anos é olhar com gratidão tudo aquilo de bom que veio dessa experiência. Já dizia a velha expressão: "não adianta chorar pelo leite derramado", então vamos ao que realmente importa.

Gostaria de agradecer infinitamente à minha incrível noiva, Luiza Letícia Caccuri Rolim, que foi minha base de sustentação durante todo este processo. Obrigado por sempre estar ao meu lado e me apoiar, por ser meu porto seguro nos momentos de turbulência. Desculpa pelos momentos difíceis, pela constante falta de tempo, e por tudo que tive que abrir mão nesse período. Mas tenho certeza que esse investimento vai ser compensado e fico imensamente feliz de ter você ao meu lado para desfrutarmos juntos.

Agradeço especialmente à minha família (de sangue ou não). À minha mãe, Marisa Elias Marchetti, que sempre me incentivou e me apoiou com muito amor. Ao meu pai, Cássio Eduardo Ferreira Rebello Colognesi, que apesar de distante geograficamente também sempre me deu seu apoio com muito orgulho. À minha irmã, Bárbara Maria Marchetti Colognesi, por sempre acreditar em mim (não no sentido de convencimento). Aos meus sogros, Flávia Aparecida Caccuri Rolim e João Luiz Rolim, que sempre me deram todo suporte, sempre preocupados em deixar esta caminhada mais leve (ou menos pesada). Agradeço aos grandes amigos que fiz nesses 5 anos, em especial a Paulo Akira Figuti Enabe, André de Souza Reis, Caio Fabrício Deberaldini Netto, Rafael da Costa Araripe, e claro, Felipe Augusto de Moraes Machado. Sem essa amizade e toda a ajuda não teria chegado até aqui. Dentro desse mesmo escopo tiro esta frase para agradecer também a Deus, que me iluminou nesta jornada repleta de provações, e sempre me acompanha e me orienta nos caminhos da vida.

Agradecimentos do Machado:

Esses últimos 5 anos na Escola Politécnica foram repletos de aprendizagem, de muitos trabalhos a serem feitos e de muitas noites em claro. Não teria sido possível chegar até aqui sem a ajuda e o apoio de diversas pessoas dentro e fora da faculdade.

Primeiro gostaria de agradecer aos grupos de extensão de que fui membro. Foi no Grupo Turing que eu conheci uma área magnífica pela qual me apaixonei, conheci pessoas maravilhosas que me ajudaram no meu desenvolvimento. No Grupo ARGO, tive a oportunidade de me aventurar em projetos bem diferentes, conhecendo pessoas de diferentes áreas e pessoas inspiradoras. Agradeço também a todos os meus colegas que me apoiaram e que me auxiliaram em diversos momentos difíceis.

Gostaria de agradecer à minha família por todo apoio que eles me proporcionaram. Minha mãe, Regina Célia de Moraes Machado, e meu pai, Alex Martins Machado, por tudo que eles fizeram por mim em todos esses anos. Sempre pude confiar neles, e sempre tento deixá-los orgulhosos como retribuição. Agradeço ao meu pequeno irmão, Pedro Luiz de Moraes Machado, por sempre me alegrar com suas brincadeiras. Agradeço também minha namorada, Renata Yuri Fróes Matuzaki, que esteve ao meu lado sempre que eu precisava de ajuda, quando precisava desabafar ou quando eu surtava; com quem pude compartilhar meus momentos de alegria e tristeza.

E por fim gostaria de agradecer ao grande Andrés Eduardo Marchetti Colognesi por tudo. Desde o nosso primeiro ano fizemos trabalhos juntos, sempre nos divertindo e aprendendo, com muitas caronas para o bandeirão. Foi uma das pessoas "chaves" para a minha formação, e eu tive muito prazer de ter feito esse trabalho com essa pessoa fantástica.

“It’s kind of fun to do the impossible”

-- Walt Disney

RESUMO

A deficiência é uma temática extremamente relevante para a sociedade, mesmo que esta não seja tão amplamente debatida como deveria. É papel da sociedade garantir que indivíduos com deficiência sejam integrados com naturalidade e autonomia nos mais diversos ambientes, tendo todos seus direitos garantidos. O desenvolvimento e uso de novas tecnologias é um instrumento crucial neste processo.

Dentro do espectro das deficiências, a de mobilidade, em especial dos membros superiores, é bastante limitante para a realização de tarefas cotidianas. Com isso em vista, uma tecnologia que pode auxiliar nesse cenário, tanto para trazer mais autonomia como para realização de exercícios de fisioterapia, é o uso de exoesqueletos.

Em especial, este trabalho se propõe a realizar o controle de posição de um exoesqueleto de um grau de liberdade, para membro superior, a partir de sinais de Eletroencefalografia (EEG). Estes sinais passam por um pré-processamento não-linear e um algoritmo de classificação. Dessa forma, é possível comparar esta abordagem com o uso de pré-processamento linear, amplamente utilizado na literatura. O controle deve ser robusto às variações de inércia provenientes da utilização do exoesqueleto por diferentes usuários.

Devido à pandemia de COVID-19, não foi possível implementar este projeto fisicamente, mas foi realizada sua simulação de forma integrada, fornecendo visualização dos resultados em "tempo real".

Palavras-Chave: Eletroencefalografia, Inteligência Artificial, Controle Robusto, Exoesqueleto.

ABSTRACT

Disability is a topic of extreme relevance for society, even if this is not widely brought up to discussion as it should. It's a duty for society to ensure that disabled people are integrated naturally and with autonomy in the most diverse scenarios, having their rights guaranteed. Developing and implementing new technologies is crucial for this process.

In the spectrum of disability, mobility impairment brings a lot of limitations to perform daily tasks, specially mobility impairment of the upper limb. With this in mind, one technology that can help in this scenario is the use of exoskeletons. This can bring more autonomy to people and also help executing physical therapy exercises.

This current work proposes to control the position of an exoskeleton with one degree of freedom for the upper limb, using electroencephalography (EEG) signals. These signals go through a non-linear pre-processing and a classification algorithm. This approach will be compared with the linear pre-processing approach that is widely used in literature. The designed controller must be robust to the inertia variation from the different human arms that will use the system.

Due to the COVID-19 pandemic, it wasn't possible to physically implement the current project. Instead, an integrated simulation of the system was performed, plotting the results in "real time".

Keywords: Electroencephalography, Artificial Intelligence, Robust Control, Exoskeleton.

LISTA DE FIGURAS

1	Fluxograma de alto nível do projeto, representando os principais blocos e sua sequência de atividades.	24
2	Fluxograma da parte de pré-processamento e classificação.	32
3	Diagrama de blocos canônico, representando a malha de controle robusto que foi projetada no presente trabalho.	33
4	Diagrama de Máquina de Estados simplificado para a lógica do controle de alto nível.	34
5	Foto da composição da estrutura da bancada de testes.	40
6	Foto da bancada de testes.	41
7	Esquema simplificado do braço robótico.	42
8	Diagrama de Corpo Livre do braço robótico.	42
9	Circuito elétrico representando o modelo dinâmico simplificado do motor CC <i>brushless</i>	44
10	Esquema representando o modelo deformável, por parâmetros concentrados, para o exoesqueleto.	47
11	Esquema representando o Diagrama de Corpo Livre de um elemento intermediário de inércia, para o modelo deformável, por parâmetros concentrados, do exoesqueleto.	48
12	<i>Templates</i> para os valores de frequência contidos no vetor de frequências. O <i>template</i> vermelho representa a frequência angular de 0.1 rad/s; o verde, de 1 rad/s; o azul, de 4.2 rad/s; e o laranja, de 40 rad/s.	54
13	Diagrama de Bode para a planta nominal.	55
14	Barreiras de robustez da estabilidade no plano de Nichols para $M = 3$. O esquema de cores e frequências segue o padrão explicado na legenda da Figura 12.	56

15	Barreiras de robustez do desempenho (acompanhamento do sinal de referência e compatibilização da malha fechada com o pré-filtro) para tolerância de 0.05. O esquema de cores e frequências segue o padrão explicado na legenda da Figura 12.	57
16	Resposta em frequência da função de transferência de malha ($G(s)K(s)$) representada no plano de Nichols juntamente com as barreiras de robustez da estabilidade e robustez do desempenho; para $K(s) = 1$	58
17	Resposta em frequência da função de transferência de malha ($G(s)K(s)$) representada no plano de Nichols juntamente com as barreiras de robustez da estabilidade e robustez do desempenho; para o compensador projetado (eq. (8.7)).	59
18	Variância total em função do número de componentes principais para o Sujeito 15.	61
19	Representação dy-dx do Sujeito 15. Esse gráfico relaciona as distâncias entre os pontos no espaço original (eixo y) com as distâncias no espaço de dimensão reduzida (eixo x).	62
20	Valor máximo de δ_r para cada frequência ω , simulando para um conjunto de plantas reais.	64
21	Valor máximo de δ_F para cada frequência ω , simulando para um conjunto de plantas reais.	64
22	Resposta do pré-filtro à entrada degrau unitário.	65
23	Valores obtidos para o tempo de estabilização 2% do sistema de acordo com as variações paramétricas das plantas reais simuladas.	66
24	Resposta do sistema de controle à entrada degrau de 1.0472 rad.	67
25	Velocidade angular do sistema a partir da derivação da resposta à entrada degrau de 1.0472 rad.	68
26	Aceleração angular do sistema a partir de duas derivações da resposta à entrada degrau de 1.0472 rad. A curva vermelha representa o este sinal filtrado por um filtro digital de fase zero com frequência de corte em 5 Hz.	68
27	Esforço de controle do sistema para a resposta à entrada degrau de 1.0472 rad.	69

28	Print da saída do modelo (probabilidade da detecção do movimento), em tempo real, gerado pelo Script.	71
29	Simulação da evolução da referência de controle comparada ao valor esperado da evolução desta variável. Foi utilizada tolerância de 1° e referência de entrada de 60°.	74
30	Simulação da evolução do esforço de controle comparada ao valor esperado da evolução desta variável. Foi utilizada tolerância de 1° e referência de entrada de 60°.	74
31	Simulação da evolução da saída da planta comparada ao valor esperado da evolução desta variável. Foi utilizada tolerância de 1° e referência de entrada de 60°.	75
32	Simulação da evolução da saída da planta após ajuste da tolerância. Foi utilizada tolerância de 2° e referência de entrada de 60°.	76
33	Evolução temporal da posição angular do exoesqueleto em movimento de flexão, comparado à referência de controle recebida.	84
34	Evolução temporal da posição angular do exoesqueleto em movimento de extensão, comparado à referência de controle recebida.	84
35	Evolução temporal do esforço de controle para o exoesqueleto em movimento de flexão.	85
36	Evolução temporal do esforço de controle para o exoesqueleto em movimento de extensão.	85
37	Evolução temporal do erro de controle para o exoesqueleto em movimento de flexão.	86
38	Evolução temporal do erro de controle para o exoesqueleto em movimento de extensão.	86
39	Comparação entre os resultados obtidos simulando o sistema como um todo e os dados experimentais de evolução do ângulo interno do cotovelo. Simulação e dados do sujeito 15, utilizando ICA como pré-processamento. .	87
40	Comparação do MSE entre o SOM clássico e seu variante, SOM-DL.	101
41	Sinal Original e Sinal reconstruído com SOM clássico.	102
42	Sinal Original e Sinal reconstruído com SOM-DL.	102

43	Erro multiplicativo para o modelo dinâmico do exoesqueleto que considera deformações, para uma discretização de $n = 1000$ elementos.	105
44	Erro multiplicativo para o modelo dinâmico do motor BLDC que considera o efeito da indutância.	107
45	Simulação da evolução da saída da planta para diferentes aproximações numéricas, comparadas ao valor esperado da evolução desta variável. Foi utilizada tolerância de 1° e referência de entrada de 60°	110

LISTA DE TABELAS

1	Requisitos parametrizados de Controle.	30
2	Informações da resposta do pré-filtro ao degrau unitário.	65
3	Informações da resposta do sistema de controle ao degrau de 1.0472 rad. .	66
4	Número de componentes principais encontrado, AUC-ROC, Sensibilidade e Especificidade para o algoritmo PCA.	81
5	AUC-ROC, Sensibilidade e Especificidade para o algoritmo ICA.	82
6	Dimensão final encontrada, AUC-ROC, Sensibilidade e Especificidade para o algoritmo CCA. Os casos marcados com asterisco apresentaram descontinuidade no espaço dos dados.	83

LISTA DE VARIÁVEIS E PARÂMETROS

Pré-processamento e Classificação:

- $S(t)$ → Sinal de EEG;
- $P(t)$ → Posição angular do braço;
- f_i → Frequência mínima de corte;
- f_a → Frequência máxima de corte;
- M → Número de vetores para quantização;
- σ → Parâmetro para atualização dos vetores;
- λ → Parâmetro para pré-processamento;
- α → Taxa de aprendizagem;
- C → Taxa de regularização;

Controle:

- $r_1(t)$ → Referência de sinal vinda do controle de alto nível (através do pré-processamento de EEG);
- $r(t)$ → Referência de controle, vinda do pré-filtro;
- $u(t)$ → Esforço de controle;
- $y(t)$ → Saída da planta (variável a ser controlada);
- $e(t)$ → Erro da malha de controle;
- $d(t)$ → Perturbações externas retratadas na saída da planta;
- $n(t)$ → Erro/ruído de medida;
- ω_c → Frequência de corte da planta;
- ω_r → Frequência máxima para acompanhamento do sinal de referência;
- M_p → Sobressinal;

- $t_s^{2\%} \rightarrow$ Tempo de estabilização 2%;
- $\delta_F \rightarrow$ tolerância para compatibilização do pré-filtro com a malha fechada;
- $\delta_r \rightarrow$ tolerância para acompanhamento do sinal de referência;
- $f_s \rightarrow$ Frequência de amostragem do sensor;
- $k_t \rightarrow$ Constante de torque do motor;
- $k_e \rightarrow$ Constante elétrica do motor;
- $\eta \rightarrow$ Rendimento do motor;
- $r \rightarrow$ Redução do motor;
- $J \rightarrow$ Inércia do braço robótico + braço humano;
- $J_r \rightarrow$ Inércia do braço robótico;
- $J_h \rightarrow$ Inércia do braço humano;
- $R \rightarrow$ Resistência elétrica do motor;
- $L \rightarrow$ Indutância do motor;
- $K(j\omega) \rightarrow$ Função de transferência do controlador;
- $F(j\omega) \rightarrow$ Função de transferência do pré-filtro;

SUMÁRIO

Parte I: INTRODUÇÃO, OBJETIVOS E DESAFIOS	18
1 Introdução	19
2 Estado da Arte	21
3 Objetivos	23
4 Desafios Científico-Tecnológicos	25
4.1 Desafios de Pré-Processamento e Classificação	25
4.2 Desafios de Controle	25
4.3 Desafios de Integração das Partes	26
Parte II: REQUISITOS E METODOLOGIA	27
5 Requisitos do Projeto	28
5.1 Requisitos para o Processamento	28
5.2 Requisitos de Controle	28
6 Metodologia	31
6.1 Metodologia para Pré-Processamento e Classificação	31
6.2 Metodologia para Controle	32
6.3 Metodologia para Integração dos Subsistemas	34
Parte III: DESENVOLVIMENTO	35
7 Algoritmos e Modelagem	36
7.1 Algoritmos para o Pré-Processamento	36

7.1.1	PCA	36
7.1.2	ICA	36
7.1.3	CCA	37
7.1.4	SOM	37
7.1.4.1	SOM-DL	38
7.2	Algoritmo para Classificação	39
7.2.1	SVM	39
7.3	Modelos para o Controle	40
7.3.1	Modelo Nominal	41
7.3.1.1	Modelo Nominal do Braço	42
7.3.1.2	Modelo Nominal do Motor	43
7.3.1.3	Unindo os Modelos Nominais	45
7.3.1.4	Função de Transferência da Planta	45
7.3.2	Incluindo Dinâmicas não Modeladas	46
7.3.2.1	Modelo Real do Braço	47
7.3.2.2	Modelo Real do Motor	49
7.3.3	Formulação das Incertezas	50
8	Estruturação dos Algoritmos e Projeto da Malha de Controle	52
8.1	Estruturação dos Algoritmos	52
8.2	Projeto da Malha de Controle	52
8.2.1	Projeto do Pré-Filtro	53
8.2.2	Projeto do Compensador	53
9	Simulações dos Sub-Sistemas	60
9.1	Simulações para Pré-Processamento	60
9.1.1	Pré-processamento Linear	60
9.1.1.1	PCA	60

9.1.1.2	ICA	61
9.1.2	Pré-processamento Não-Linear	61
9.1.2.1	SOM	61
9.1.2.2	CCA	62
9.2	Simulações para Controle	63
10	Integração das Partes - Simulações	70
10.1	Simulação Integrada - Pré-Processamento	70
10.2	Simulação Integrada - Controle	71
10.2.1	Classe para Controle de Baixo Nível - <i>LLC</i>	72
10.2.2	Classe para Controle de Alto Nível - <i>HLC</i>	76
10.3	Simulação Integrada - Sistema Completo	77
Parte IV	RESULTADOS E DISCUSSÃO	79
11	Resultados	80
11.1	Resultados do Pré-Processamento	80
11.2	Resultados do Controle	84
11.3	Resultados da Integração dos Sistemas	87
12	Discussão	88
12.1	Discussão para o Pré-Processamento	88
12.2	Discussão para o Controle	89
12.3	Discussão para a Integração dos Sistemas	90
Parte V	CONCLUSÕES E CONSIDERAÇÕES FINAIS	91
13	Conclusões	92
14	Considerações Finais	93

Referências	94
Apêndice A – Revisão Bibliográfica	97
Apêndice B – Simulações do Algoritmo SOM	100
B.1 Simulações de MSE	101
Apêndice C – Simulações dos Modelos 'Reais'	103
C.1 Simulação para o Braço	104
C.2 Simulação para o Motor	106
Apêndice D – Validações do Sistema de Controle	108
Apêndice E – LLC - Dissipação Numérica	109
Anexo A – Desenho Mecânico do Exoesqueleto	111

PARTE I

INTRODUÇÃO, OBJETIVOS E DESAFIOS

1 INTRODUÇÃO

A deficiência é um tema extremamente relevante para a sociedade. De acordo com a Cartilha do Censo 2010 [1], na qual investigou-se as deficiências visual, auditiva, motora e mental ou intelectual; 23,9% da população residente no Brasil em 2010 possuía pelo menos uma destas deficiências. Isto representa um total de 45.606.048 pessoas. Dentre estas, a deficiência motora é a de segunda maior ocorrência, sendo observada em 7% da população (o que representa 13.357.420 pessoas). De acordo com a própria Cartilha [1] (p. 4):

A deficiência é um tema de direitos humanos e como tal obedece ao princípio de que todo ser humano tem o direito de desfrutar de todas as condições necessárias para o desenvolvimento de seus talentos e aspirações, sem ser submetido a qualquer tipo de discriminação.

Tendo isto em vista, fica evidente a importância da engenharia neste contexto e o papel que deve desempenhar. Cabe à engenharia trabalhar para conceber e implementar soluções que promovam a inserção destes indivíduos na sociedade, garantindo assim não só uma melhor qualidade de vida, mas também sua autonomia e dignidade. O desenvolvimento e uso de novas tecnologias é um instrumento crucial neste processo.

Pensando na deficiência de mobilidade como um todo, a limitação de mobilidade nos membros superiores parece ser mais comprometedora para o desempenho da maioria das atividades do cotidiano. Portanto, selecionou-se esta como o problema a ser tratado no presente trabalho.

Norteando o trabalho para a garantia de autonomia destes indivíduos com deficiência de mobilidade nos braços, encontrou-se como solução a utilização de um exoesqueleto controlado por sinais cerebrais do usuário. Dentre as formas de medir a atividade do cérebro (imagem por ressonância magnética funcional, fMRI; magnetoencefalograma, MEG; espectroscopia de infravermelho próximo funcional, fNIRS; eletroencefalograma, EEG), o EEG foi escolhido devido a sua alta resolução temporal, facilidade de utilização e baixo custo [2]. Desta forma, o operador do exoesqueleto é o próprio usuário, que comanda o

movimento do braço com o seu cérebro (o que deve ser semelhante a controlar o próprio braço em condições ideais). Com esta abordagem, será garantida mais autonomia ao usuário, que poderá movimentar seus braços de forma independente.

Este trabalho é o primeiro passo na direção deste objetivo. Portanto, optou-se por controlar um exoesqueleto para braço, com apenas um grau de liberdade. Garantindo um bom funcionamento do projeto nessa estrutura mais simples, é possível trabalhar para generalizar a aplicação para exoesqueletos com mais graus de liberdade. A abordagem de controle que será adotada é a de controle robusto. Isso porque a utilização deste tipo de controlador é mais resistente às variações externas do meio real e aos erros de modelagem inerentes ao desenvolvimento do projeto.

Dessa forma, define-se como objetivo deste trabalho controlar a posição angular de um exoesqueleto de um grau de liberdade para membro superior, de forma robusta, a partir de sinais de EEG, utilizando pré-processamento não-linear. O trabalho também compara os resultados obtidos pelo pré-processamento não linear com os do linear, sendo este o método utilizado na literatura.

Além deste objetivo principal, o uso do sistema proposto também pode auxiliar no processo de reabilitação, para recuperação do movimento dos braços. Neste caso, as atividades para reabilitação poderiam também ser realizadas pelo paciente, com total autonomia.

Os detalhes do processo de pesquisa das referências realizado ao longo deste trabalho estão expostos no Apêndice A.

2 ESTADO DA ARTE

Em sistemas BCI, o sinal EEG é o mais utilizado pelas suas características [2] para detectar a intenção do usuário. Visto que os sinais elétricos registrados sob a pele são o resultado da superposição não linear da atividade elétrica de múltiplos neurônios que atravessam diferentes tecidos [3], é necessário processá-los. Recentemente, têm sido empregadas técnicas baseadas em Inteligência Artificial para tal finalidade [4]. Para o pré-processamento deste tipo de sinal, as 3 grandes abordagens mais utilizadas são o *Feature Select* (FS), *Feature Extract* (FE) e *Feature Transform* (FT); sendo que esta última apresenta os melhores resultados [5].

Os algoritmos não supervisionados mais utilizados no FT são lineares, como o *Principal Components Analysis* (PCA) [6] e o *Independent Components Analysis* (ICA) [7], enquanto que os não lineares são pouco abordados neste tipo de aplicação. A utilização de componentes não lineares no pré-processamento do sinal de EEG pode ser uma alternativa para melhorar o desempenho, visto que, dos modelos de regressão ou classificação utilizados nessa área [4], aqueles com comportamento não lineares apresentam resultados melhores quando comparado aos lineares [8].

O uso de exoesqueletos para movimentação de membros em humanos é bastante vasto. Atualmente existem diversas configurações mecânicas de exoesqueletos, sendo adaptadas para as mais variadas aplicações [9–12]. Em particular, pode ser ressaltado que o uso de exoesqueletos robóticos para recuperação de movimento é algo bastante estudado nas últimas duas décadas, apresentando resultados bastante satisfatórios (visto o objetivo particular de cada aplicação) [9, 10].

Revisões na literatura abordando configurações mecânicas e aplicações de exoesqueletos são facilmente encontradas. Entretanto, revisões sobre os sistemas de controle aplicados a esses exoesqueletos não são tão comuns [12].

Os sistemas de controle comumente utilizados baseiam-se em quatro grandes aspectos: modelo; hierarquia; parâmetros físicos e uso. A construção do modelo é dada pela

composição de duas partes: o modelo dinâmico do exoesqueleto e o modelo dos músculos do operador [9]. A obtenção deste primeiro modelo dinâmico pode ser feita de forma analítica, por modelagem matemática; por meio de técnicas de identificação de sistemas; ou mesmo utilizando Inteligência Artificial, sendo esta uma técnica mais moderna. Por outro lado, o intuito de modelar também o músculo é prever o movimento realizado pelo mesmo dentro da dinâmica do sistema como um todo [13].

O controle costuma ser estruturado em diversos níveis de hierarquia, variando de acordo com a aplicação. Normalmente, o nível mais baixo de controle lida diretamente com as grandezas físicas de interesse [9]. Os parâmetros físicos mais comuns nestas aplicações são a posição, a velocidade e a força/torque. Os níveis mais altos de controle, por outro lado, vão gradativamente colocando "inteligência" na aplicação. Estes são responsáveis, por exemplo, por comandar o baixo nível de acordo com alguma lógica; ou mesmo em prever determinadas configurações do sistema e agir de forma corretiva.

Voltando o olhar para o baixo nível, controlar força/torque é extremamente relevante para adequar a operação do exoesqueleto sem causar dano ao usuário. Uma técnica interessante que realiza o controle de força/torque juntamente com o controle de posição é o controle de impedância [14]. A partir desta, é possível controlar a interação entre o usuário e o exoesqueleto de forma direta. Tal técnica possibilita a obtenção de resultados bastante satisfatórios para aplicações reais [15]. Entretanto, como este trabalho é uma primeira abordagem para a aplicação proposta, não há o objetivo de monitorar tais grandezas de interação. É realizado apenas o controle de posição angular. Portanto, o uso do controle de impedância se torna desnecessário para o presente projeto, mas é uma alternativa muito interessante para futuros desenvolvimentos.

O intuito deste projeto é controlar a posição angular de um braço robótico a partir de referências de sinal de EEG. Como o controle projetado deve ser robusto a variações de inércia proveniente dos diferentes braços humanos que vão operar o exoesqueleto (tendo este apenas 1 grau de liberdade), a abordagem de controle robusto com modelagem analítica se mostra suficiente. Desta forma, haverá apenas uma malha de controle de baixo nível, que será comandada por uma lógica de controle de alto nível. Portanto, dois níveis de hierarquia serão suficientes para atingir os objetivos de controle desejados. Vale ressaltar que o projeto de um controlador robusto leva em consideração faixas de operação nas quais os requisitos são atendidos, sendo assim controladores mais resistentes a perturbações externas e a erros inerentes à modelagem. Um exemplo de uso deste tipo de controlador aplicado ao controle de exoesqueleto foi demonstrado por [16].

3 OBJETIVOS

O objetivo primário do presente trabalho é controlar a posição angular de um exoesqueleto com um grau de liberdade para membro superior, a partir de sinais de EEG, utilizando um algoritmo não-linear de pré-processamento. O algoritmo não-linear escolhido foi o CCA devido a sua boa performance em diferentes problemas [17]. O resultado desta abordagem é comparado com o dos algoritmos lineares mais utilizados na literatura, apresentados anteriormente. O controlador deve ser robusto a variações de inércia provenientes dos diferentes braços humanos que vão operar o exoesqueleto. O processamento do sinal EEG deve ser capaz de detectar a intenção de movimento do usuário, tarefa conhecida como *timing* coincidente [18]. O objetivo desta abordagem é garantir a autonomia de movimento dos membros superiores do usuário em um cenário mais próximo do real, sendo o usuário alvo uma pessoa com deficiência de mobilidade nos membros superiores.

Além deste objetivo principal, de garantia de autonomia, o uso do sistema proposto também pode auxiliar no processo de reabilitação, para recuperação do movimento dos braços. Neste caso, as atividades para reabilitação poderiam também ser realizadas pelo paciente, com total autonomia. Esta aplicação exigiria, entretanto, ajustes no controlador que foi implementado para o objetivo principal do projeto.

Um fluxograma de alto nível do projeto, representando os principais blocos e o seu fluxo de atividade, está colocado na Figura 1.

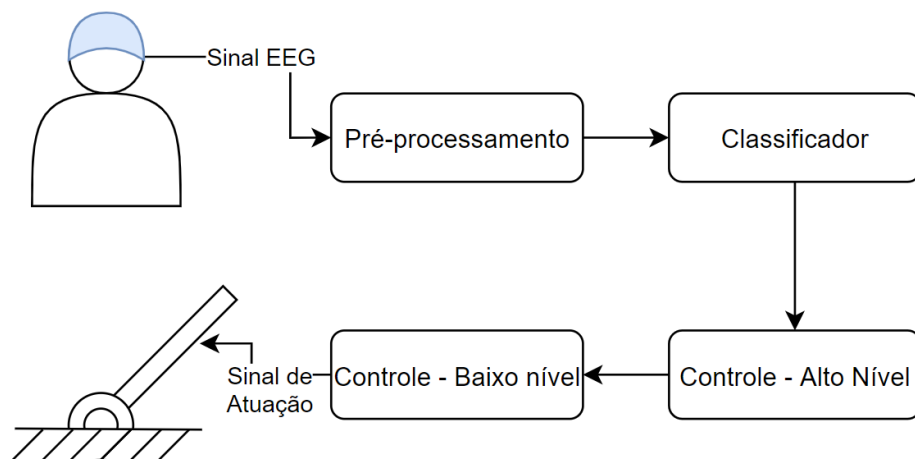


Figura 1: Fluxograma de alto nível do projeto, representando os principais blocos e sua sequência de atividades.

4 DESAFIOS CIENTÍFICO-TECNOLÓGICOS

O projeto consiste na integração de diversos subsistemas, como ilustrado na Figura 1. Estes estão desenvolvidos com base em teorias de diferentes áreas de conhecimento. Portanto, os desafios de cada parte serão abordados separadamente nas seções seguintes.

4.1 Desafios de Pré-Processamento e Classificação

Para parte de pré-processamento, o grande desafio foi conseguir gerar uma nova representação dos dados com melhor qualidade após a aplicação dos algoritmos propostos, visto que o ruído é significativo. Há também, para esse tema, o problema de desbalanceamento das classes, que afeta a aprendizagem do modelo de classificação. Além disso, os algoritmos mais complexos não são acessíveis em qualquer linguagem de programação, e desenvolvê-los de forma precisa e eficiente se tornou outro grande desafio.

Devido à pandemia da COVID-19, não foi possível realizar os experimentos propostos inicialmente. Assim, a disponibilidade de dados se sustentou exclusivamente de repositórios públicos, com um desafio adicional de encontrar um cuja problemática e a tarefa fossem próximas às deste trabalho. Isso motivou uma busca de bases de dados com experimentos que pudessem ser utilizados.

4.2 Desafios de Controle

Para a parte de controle, o maior desafio foi realizar o projeto mais simples possível de controlador, mas que fosse capaz de suportar variações paramétricas. Além disso, os requisitos de controle devem ser todos respeitados, uma vez que representam limitações de interação com o corpo humano, sendo assim cruciais para garantir a segurança do operador.

Devido à pandemia do COVID-19, que impossibilita o acesso ao laboratório, a modelagem da planta foi toda realizada de forma analítica. A elaboração analítica do modelo da planta incluindo dinâmicas não modeladas, a fim de ser representativo do problema prático, foi um desafio relevante. Além disso, devido à inviabilidade de implementação física do controlador na bancada, o sistema foi validado por meio de simulações. Estas se sustentam no modelo analítico da planta, desenvolvido para o presente projeto.

4.3 Desafios de Integração das Partes

Por fim, o maior de todos os desafios deste trabalho foi a integração do pré-processamento dos sinais de EEG e classificação do movimento, com o controle do braço robótico. Tal integração foi desenvolvida de forma a garantir a operação segura e em tempo real do exoesqueleto, atendendo todos os requisitos estabelecidos para o projeto.

Novamente devido à pandemia de COVID-19, surgiu um novo desafio: testar o funcionamento completo do projeto, de forma integrada, por meio de simulações. Para tanto, foi necessário criar uma estrutura de códigos que representassem bem o problema real que é tratado pelo presente projeto.

PARTE II

REQUISITOS E METODOLOGIA

5 REQUISITOS DO PROJETO

O presente projeto possui duas partes: uma primeira parte de processamento dos sinais de EEG para identificação da intenção de movimento do usuário e uma segunda parte de controle do exoesqueleto. Para maior clareza, os requisitos de ambas as partes serão tratados separadamente e respectivamente nas seções 5.1 e 5.2.

5.1 Requisitos para o Processamento

Para o pré-processamento, há um requisito de tempo de execução, a fim de que seja possível a aplicação dos algoritmos em tempo real. Considerando um equipamento de EEG de frequência 100Hz, ou seja, é recebido um novo sinal a cada 10ms, o requisito do pré-processamento é ter um tempo de execução de até 5ms. Como parte do objetivo do trabalho é comparar resultados de pré-processamento diferentes, não há requisitos para a parte de classificação dos dados.

5.2 Requisitos de Controle

Como a bancada de testes já está implementada [19], possuindo também um encaixe para ligar o exoesqueleto ao braço humano, os requisitos de controle são voltados aos aspectos funcionais do controle propriamente dito.

A fim de evitar que o usuário humano sofra algum dano ao usar o exoesqueleto, este tem seu ângulo de ação limitado a uma faixa máxima de 60° , definida no sentido de flexão da articulação do cotovelo. O ângulo interno do cotovelo, em repouso, deve ser de 120° , e do braço flexionado, de 60° . A velocidade angular máxima deve ser $240^\circ/s$ (velocidade utilizada para exercícios de aquecimento da articulação do cotovelo em [20]).

Em [21], limita-se a aceleração a 2 m/s^2 na extremidade do exoesqueleto, a fim de garantir o movimento suave do mesmo. Aplica-se esta aceleração ao centro de massa do

conjunto exoesqueleto e braço (do presente trabalho), o qual foi assumido estar no centro da estrutura circular da extremidade mais externa do exoesqueleto da bancada (vide Anexo A para maior clareza). Dessa forma, é obtida uma aceleração angular máxima de 20 rad/s^2 . A fim de verificar se o motor tem torque suficiente para este movimento, aplicou-se essa aceleração máxima juntamente com o valor máximo de inércia para o qual o controle foi projetado (tais valores serão abordados com mais detalhe na seção 8.2.2).

Dessa forma é obtido um torque de 8.164 Nm. Este valor está bem abaixo do máximo torque que o motor pode fornecer ao exoesqueleto (utiliza-se como base de comparação o valor do *stall torque* do motor BLDC da bancada, passando pela redução, o que resulta em um torque máximo de 494 Nm). Portanto, limitar a aceleração angular máxima a 20 rad/s^2 está completamente dentro da capacidade da bancada física.

Os dados experimentais que foram utilizados para o pré-processamento do presente trabalho [22] (apresentados na seção 6.1) mostram que o movimento de flexão ou extensão do cotovelo dura entre 1 e 2 segundos. Portanto, o tempo de estabilização 2% do controle deve estar contido nessa faixa de 1 a 2 segundos. Além disso, o sobressinal deve ser nulo para evitar trancos sobre o braço humano.

A partir da velocidade angular máxima permitida para o presente sistema, é obtida a frequência máxima de atuação do mesmo: $\omega_r = 4\pi / 3 \text{ [rad/s]}$. Dentro desta faixa, o controle deve garantir o acompanhamento de sinal de referência com tolerância de 5% ($\delta_r = 0.05$). Tal valor foi arbitrado, a fim de garantir o bom funcionamento do sistema. Vale ressaltar que uma tolerância mais apertada poderia ser atendida, uma vez que o motor possui uma redução harmônica de 1:100 e *encoder* de 12-bits de resolução (4096 contagens por volta) [19]. Dessa forma, existe uma grande margem para o correto funcionamento do sistema com a tolerância exigida.

Seguindo a mesma linha de raciocínio, é tolerado um erro máximo de 5% para a compatibilização da malha fechada com o pré-filtro projetado ($\delta_F = 0.05$). Esta tolerância deve ser respeitada também até a frequência de atuação máxima permitida ω_r . Novamente, este valor foi arbitrado, mantendo em vista a garantia de movimento suave, sem trancos.

Além disso, o controle deve ser robusto à variação de inércia, ocasionada pelas diferentes inércias de diferentes braços humanos que possam utilizar o sistema.

Vale ressaltar que o sistema é capaz de atender aos requisitos propostos sem grandes dificuldades. Os requisitos de controle são todos de baixa frequência, com tolerância de 5%. Portanto, pensando no Diagrama de Bode, a curva de resposta em frequência da

planta controlada deverá ficar acima da barreira de 26dB (aproximadamente), que se estende até a frequência de $4\pi/3$ [rad/s]. Como não há requisitos a respeito do erro de medida, pois os sensores da planta têm bom funcionamento até frequências muito mais altas, não há barreira de alta frequência (e se houvesse, a mesma estaria muitas décadas a frente).

Portanto, a resposta do sistema consegue ser tranquilamente projetada de forma a ter inclinação próxima a -20 dB/dec no cruzamento com a reta de 0 dB. Tal condição, de acordo com o Teorema de Bode, garante que o sistema controlado não instabilize. Note que a análise aqui realizada é simplificada, uma vez que considera-se a barreira nominal, e não a barreira robusta. Mas, mesmo com a barreira robusta, o mesmo raciocínio se aplica.

Para facilitar a organização de todos os requisitos parametrizáveis tratados nesta seção, a Tabela 1 apresenta todos esses parâmetros com sua descrição e seus valores.

Tabela 1: Requisitos parametrizados de Controle.

Descrição	Parâmetro	Faixa de Valores
Ângulo interno de repouso	θ_0	120°
Ângulo interno de flexão	θ_f	60°
Máxima velocidade angular	ω_{max}	$\frac{4\pi}{3} \text{ rad/s}$
Máxima aceleração angular	α_{max}	20 rad/s^2
Tempo de estabilização 2%	$t_s^{2\%}$	entre 1 e 2 s
<i>Overshoot</i>	M_p	0
Tolerância do acompanhamento de referência	δ_r	0.05
Limite de frequência de acompanhamento	ω_r	$\frac{4\pi}{3} \text{ rad/s}$
Tolerância da compatibilização do pré-filtro	δ_F	0.05
Limite de frequência de compatibilização	ω_b	$\frac{4\pi}{3} \text{ rad/s}$

6 METODOLOGIA

A fim de melhor compreender as metodologias empregadas para o desenvolvimento do presente trabalho, optou-se por dividir as metodologias em três áreas: pré-processamento dos sinais de EEG e classificação do sinal processado; controle do exoesqueleto; e integração dos subsistemas. Estas são abordadas, respectivamente, nas seções 6.1, 6.2 e 6.3.

6.1 Metodologia para Pré-Processamento e Classificação

Para a primeira parte, de pré-processamento de sinais, partiu-se do sinal original (S) coletado pelos eletrodos do equipamento. Este foi filtrado utilizando um filtro passa-banda *butterworth* de ordem 5 com frequência mínima f_i e máxima f_a . Para a transformação não-linear, o algoritmo SOM (Self-Organizing Maps [23]) foi utilizado para reduzir o número de pontos para M pontos, que foram consumidos pelo algoritmo CCA (Curvilinear Components Analysis [24]), diminuindo a dimensão. Para a análise linear, os algoritmos PCA e ICA foram utilizados no espaço original.

A classificação foi realizada utilizando o algoritmo SVM [25] com parâmetro de regularização C . Para definir o *threshold* de classificação do modelo, a metodologia escolhida foi atribuir um custo referente à Taxa de Falso Positivo (FPR) e à Taxa de Falso Negativo (FNR). Em seguida, foram criados diversos valores de limiares diferentes, espaçados de 0.01, entre o intervalo de 0 a 1. Para cada limiar, foi calculado o custo total, e foi selecionado o *threshold* com o menor custo total.

Um esquema ilustrando as abordagens de pré-processamento e classificação que foram implementados está colocado na Figura 2.

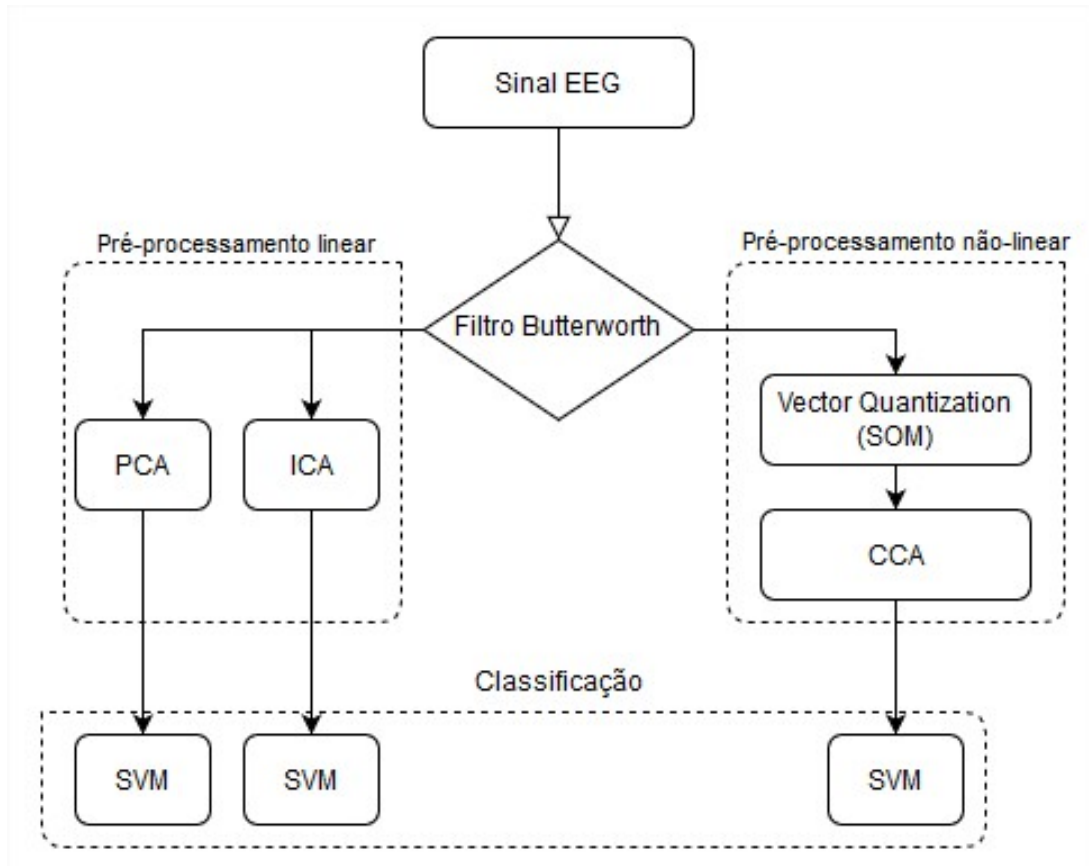


Figura 2: Fluxograma da parte de pré-processamento e classificação.

A análise foi desenvolvida e simulada no Python 3.7 [26], a fim de encontrar os melhores parâmetros dos modelos.

Devido à quarentena (ocasionada pela pandemia de COVID-19), foi utilizado neste trabalho um banco de dados público [22] sobre movimento executado do membro superior direito. Embora esse banco de dados possua 6 classes de movimento, somente a flexão do cotovelo foi utilizada. Os dados foram extraídos de 15 indivíduos, sendo que apenas 1 é canhoto.

6.2 Metodologia para Controle

Para a segunda parte, de controle do exoesqueleto, foram desenvolvidos primeiramente os modelos nominal e real da planta. O modelo nominal deve ser um modelo simples, mas representativo do problema, considerando hipóteses simplificadoras convenientes. O modelo real é uma versão mais complexa do nominal. Este pode levar em consideração mais fatores dinâmicos que possam ser relevantes para o problema físico real ou levar em consideração incertezas nos parâmetros. A construção do modelo real varia muito de

problema para problema.

Para identificar qual abordagem de incerteza era mais adequada para o presente trabalho, foram primeiramente considerados modelos reais que incluíam dinâmicas não modeladas no modelo nominal. A partir da análise da resposta em frequência destes modelos reais, foi determinado se o efeito destas dinâmicas não modeladas seria relevante ou não para o problema tratado, dadas suas condições de funcionamento.

Partindo desta conclusão, a malha de controle foi projetada seguindo o método QFT, pois a natureza da incerteza considerada foi de incerteza estruturada (paramétrica). Como o sistema deve funcionar com braços de pessoas diferentes, é considerada a variação de inércia. Tal incerteza foi incluída como incerteza paramétrica no projeto. Também foi considerada a incerteza da eficiência do motor, por meio da qual foi inserida uma margem para incluir os efeitos dissipativos desconsiderados no modelo nominal.

A malha de controle inclui um pré-filtro ($F(j\omega)$), responsável por garantir o atendimento dos requisitos definidos no domínio do tempo (*overshooting*, tempo de estabilização, velocidade angular máxima e aceleração angular máxima). Este foi desenvolvido em conjunto com o controlador ($K(j\omega)$), que foi projetado no domínio da frequência. Desta forma, a malha de controle possui dois graus de liberdade, sendo baseada no diagrama canônico, que está representado na Figura 3. Note que os sinais $d(s)$ e $n(s)$ representam, respectivamente, perturbações tomadas na saída da planta e erros de medida dos sensores.

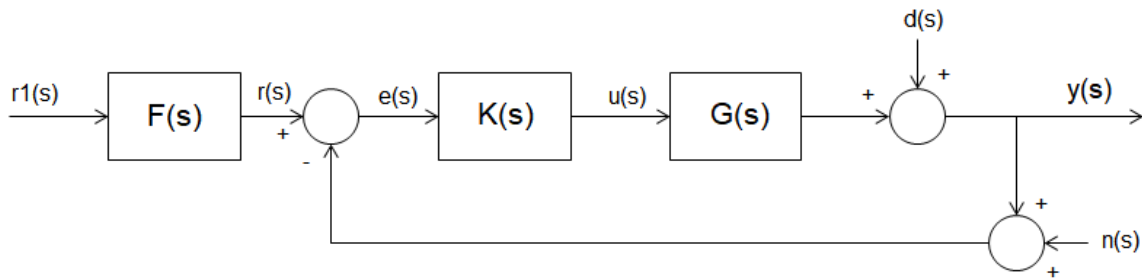


Figura 3: Diagrama de blocos canônico, representando a malha de controle robusto que foi projetada no presente trabalho.

Como foi adotado o Método QFT, o projeto do controlador foi desenvolvido com o auxílio do *software open source* SISO-QFTIT. Em seguida, as simulações da malha de controle foram realizadas no *software* MATLAB, a fim de validar o controlador e o pré-filtro, garantindo a obtenção de resultados que satisfizessem os requisitos do projeto.

Feitas as simulações e validada a malha de controle, esta foi simulada utilizando o Python. Em conjunto com a implementação do controle de alto nível, todo o sistema de controle, juntamente com a planta modelada, foi simulado de forma a mimetizar sua implementação na bancada de testes [19]. Dessa forma, a passagem do código desenvolvido em Python para um código compatível com a placa *Raspberry Pi* será facilitada para trabalhos futuros.

O controlador projetado por meio das técnicas de controle robusto é o controlador de baixo nível do projeto (vide Figura 1). O controlador de alto nível, responsável por garantir a integração do controle com a parte de pré-processamento, foi desenvolvido baseado em uma lógica de máquina de estados. Esta máquina de estados está colocada na Figura 4.

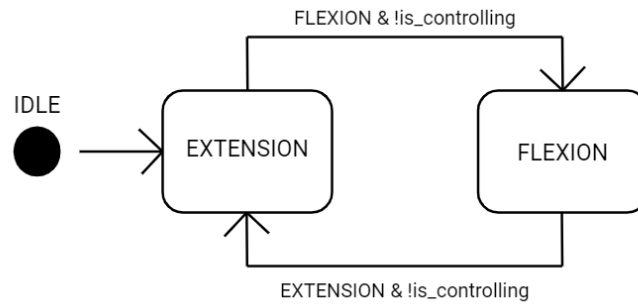


Figura 4: Diagrama de Máquina de Estados simplificado para a lógica do controle de alto nível.

6.3 Metodologia para Integração dos Subsistemas

Por fim, foi integrada a parte de pré-processamento e classificação dos sinais de EEG com a estrutura completa de controle do exoesqueleto. Devido à pandemia de COVID-19, a integração do sistema ocorreu de forma virtual, sendo simulada utilizando Python. Dessa forma, o funcionamento do sistema de forma conjunta foi testado e validado.

PARTE III

DESENVOLVIMENTO

7 ALGORITMOS E MODELAGEM

Os algoritmos e modelos utilizados e produzidos para o desenvolvimento do presente trabalho serão expostos nas seções seguintes. Note que foi feita a separação entre a parte de algoritmos de pré-processamento (seção 7.1), algoritmo de classificação (seção 7.2) e modelagens de controle (seção 7.3), para maior clareza.

7.1 Algoritmos para o Pré-Processamento

7.1.1 PCA

O PCA (*Principal Components Analysis*) [6] é um algoritmo linear que utiliza a matriz de covariância para reduzir a dimensão dos dados, criando novos eixos (Componentes Principais). Esse algoritmo realiza a mudança de eixos de acordo com a eq. (7.1). Nesta, $Y_{N \times L}$ são os dados de dimensão reduzida, $X_{N \times D}$ são os dados originais, $W_{D \times L}$ é a matriz de mudança de base, N é o número de pontos, D é a dimensão original dos dados e L é a nova dimensão ($D \geq L$).

$$Y_{N \times L} = X_{N \times D} W_{D \times L} \quad (7.1)$$

Os Componentes Principais que o PCA gera são descorrelacionados, ou seja, $E[X_1 X_2] = 0$, para quaisquer Componentes X_1 e X_2 .

7.1.2 ICA

O ICA (*Independent Component Analysis*) [7] é um algoritmo linear que busca encontrar componentes independentes para reduzir a dimensão dos dados, criando novos eixos. Esse algoritmo utiliza a mudança de eixos de acordo com a eq. (7.2). Nesta, $S_{N \times L}$ são os dados de dimensão reduzida, $X_{N \times D}$ são os dados originais, $W_{D \times L}$ é a matriz mudança de

base, N é o número de pontos, D é a dimensão original dos dados e L é a nova dimensão ($D \geq L$).

$$S_{N \times L} = X_{N \times D} W_{D \times L} \quad (7.2)$$

Os Componentes gerados por esse algoritmo são independentes, ou seja, $E[f(X_1, X_2)] = 0$ para qualquer função f e quaisquer Componentes X_1 e X_2 .

7.1.3 CCA

O CCA (*Curvilinear Components Analysis*) [24] é um algoritmo não-linear de redução de dados que utiliza como base as distâncias entre os pontos, mapeando os dados originais em um espaço de menor dimensão. A sua função de custo é dada pela eq. (7.3). Nesta, Y_{ij} é a distância euclidiana no espaço de dimensão reduzida entre os pontos i e j , X_{ij} é a distância euclidiana no espaço original entre os pontos i e j , $F(d, \lambda)$ é uma função peso e λ é um parâmetro de corte. A função F utilizada nesse trabalho está definida pela eq. (7.4).

$$E = \frac{1}{2} \sum_i \sum_{i \neq j} (X_{ij} - Y_{ij})^2 F(Y_{ij}, \lambda_y) \quad (7.3)$$

$$F(d, \lambda) = \begin{cases} 1 & \lambda \geq d \\ 0 & \lambda < d \end{cases} \quad (7.4)$$

A atualização de cada ponto é realizada de acordo com a eq. (7.5), onde y_i é o ponto i do espaço reduzido e $\alpha(t)$ é a taxa de aprendizado.

$$\Delta y_i = \alpha(t) F(Y_{ij}, \lambda) (X_{ij} - Y_{ij}) \frac{y_j - y_i}{Y_{ij}} \quad (7.5)$$

7.1.4 SOM

O *Self-Organizing Map* [23] é um algoritmo para *Vector Quantization* que utiliza aprendizagem competitiva entre os N neurônios de um mapa, que pode ser de uma ou duas dimensões. Cada neurônio m possui a mesma dimensão D dos dados de entrada e é atualizado de acordo com a eq. (7.6). Nesta, t indica o número da iteração, $h_{ci}(t)$ é uma função de adaptação da vizinhança, $x(t)$ é um ponto de entrada.

$$m_i(t+1) = m_i(t) + h_{ci}(t)[x(t) - m_i(t)] \quad (7.6)$$

A função h_{ci} utilizada neste trabalho está definida pela eq. (7.7). Nesta, r_i é a posição do neurônio i no mapa, r_c é a posição no mapa do neurônio de menor distância com o ponto x , σ é um parâmetro que determina o tamanho da vizinhança a ser modificada.

$$h_{ci}(t) = h_0 \exp\left(-\frac{\|r_i - r_c\|}{\sigma(t)^2}\right) \quad (7.7)$$

Esse algoritmo tem duas etapas de treinamento: na primeira parte, os neurônios tendem a se espalhar pelo conjunto de dados de treino; na segunda parte do treinamento, os neurônios vão se especializando no espaço que se localizam. Dessa forma, esse algoritmo consegue aprender a topologia do espaço, sendo uma ótima ferramenta para *Vector Quantization*.

7.1.4.1 SOM-DL

Dentre as variantes do algoritmo SOM, a que será utilizada neste trabalho é o *Self-Organizing Maps with Dynamic Learning* (SOM-DL) [27]. O SOM consegue discretizar bem regiões mais densas, porém falha nas regiões menos densas. O algoritmo SOM-DL consegue modificar esse comportamento modificando a eq. (7.6) como exposto pela eq. (7.8).

$$m_i(t+1) = m_i(t) + h_{ci}(t) \cdot \mu \cdot \text{sign}(x(t) - m_i(t))(x(t) - m_i(t))^2 \quad (7.8)$$

A escolha dessa variante e os comportamentos dos dois algoritmos estão descritos no Apêndice B.

7.2 Algoritmo para Classificação

7.2.1 SVM

A *Support Vector Machine (SVM)* [25] é um algoritmo classificador supervisionado, que possui comportamento variável dependendo do seu núcleo do produto interno (kernel) $K(\mathbf{x}, \mathbf{x}_i)$. Nesse estudo, será utilizado uma SVM com rede de função de base radial, sendo seu kernel definido como colocado na eq. (7.9), onde σ é a largura da função, definida pelo usuário.

$$K(\mathbf{x}, \mathbf{x}_i) = \exp\left(-\frac{1}{2\sigma^2} \|\mathbf{x} - \mathbf{x}_i\|\right) \quad (7.9)$$

Esse modelo tem como objetivo maximizar a função objetivo dada pela eq. (7.10) a partir do conjunto de dados $\{(x_i, d_i)\}_{i=1}^N$, onde $\{\alpha_i\}_i^N$ são os multiplicadores de Lagrange.

$$Q(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j d_i d_j K(\mathbf{x}_i, \mathbf{x}_j) \quad (7.10)$$

Esse modelo ainda apresenta duas características importantes, descritas em [28]:

- A dimensionalidade do espaço de características é determinada pelo número de vetores de suporte extraídos dos dados de treinamento.
- Com o *kernel* utilizado nesse trabalho, o número de funções de base radial e seus centros são determinados automaticamente pelo número de vetores de suporte e seus valores, respectivamente.

7.3 Modelos para o Controle

A planta que deve ser controlada consiste essencialmente em um braço robótico de 1 grau de liberdade, atuado por um motor *brushless* de corrente contínua (BLDC). O braço se localiza sobre uma bancada, movimentando-se na horizontal (paralelo ao chão).

Esta planta possui uma implementação física no Laboratório de Biomecatrônica da Universidade de São Paulo, tendo sido construída em um projeto de mestrado [19].

As Figuras 5 e 6 apresentam a estrutura física da planta, localizada sobre a bancada de testes.

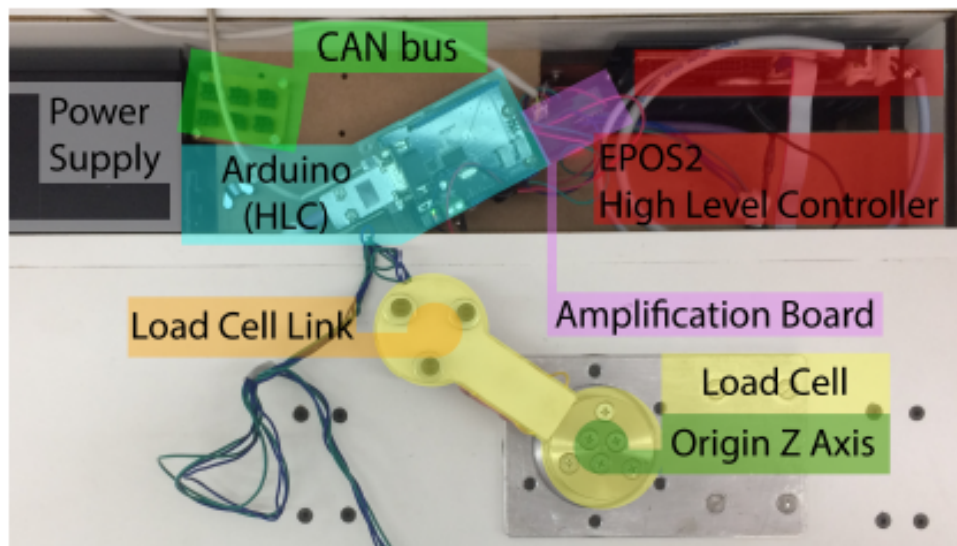


Figura 5: Foto da composição da estrutura da bancada de testes.

Figura retirada de [19].

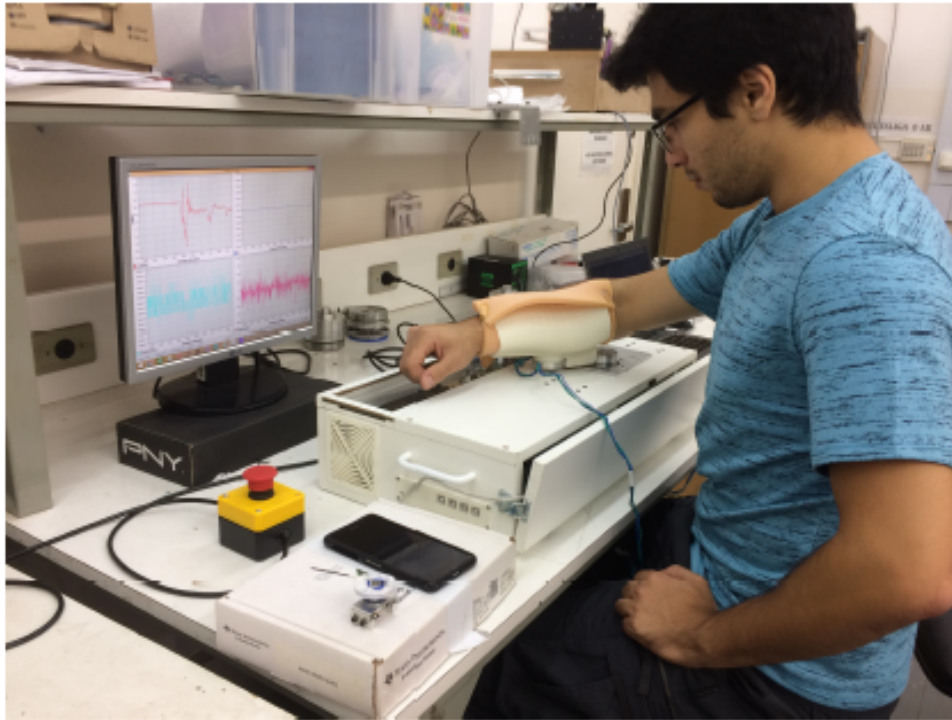


Figura 6: Foto da bancada de testes.

Figura retirada de [19]

O controle foi projetado com base na teoria de controle robusto. Com isso em vista, foi necessário o desenvolvimento de dois tipos de modelo para a planta controlada: o modelo nominal e os modelos reais. Sobre o modelo nominal foi projetado o controlador, sendo esse uma versão mais simplificada da planta. Os modelos reais foram utilizados para definir de forma quantitativa a robustez do sistema, garantindo o atendimento dos requisitos de controle no funcionamento real. Isso se deu com base na definição da formulação de incerteza que foi utilizada para o projeto do controlador.

Nas subseções seguintes estão demonstradas as modelagens realizadas sobre a planta, assim como a definição das incertezas consideradas para o projeto do controlador robusto.

7.3.1 Modelo Nominal

Para obtenção do modelo dinâmico simplificado da planta, realizou-se a modelagem analítica do braço e do motor. As equações diferenciais obtidas para cada parte foram unidas, gerando a equação diferencial da planta.

7.3.1.1 Modelo Nominal do Braço

O braço mecânico da bancada foi esquematizado de forma simplificada como exposto na Figura 7. A partir desse esquema, obteve-se o Diagrama de Corpo Livre (DCL) do braço, como ilustrado na Figura 8. O momento M é o torque transmitido do motor ao braço.

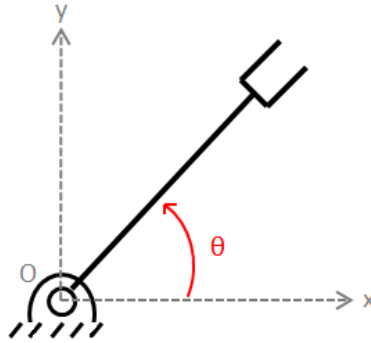


Figura 7: Esquema simplificado do braço robótico.

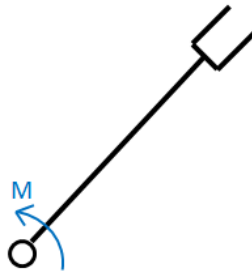


Figura 8: Diagrama de Corpo Livre do braço robótico.

Para modelar o braço, a partir do esquema simplificado apresentado (Figura 7), assumiu-se as seguintes hipóteses:

- Atrito e forças dissipativas desprezíveis (estes serão incluídos posteriormente como incerteza paramétrica);
- Efeito do peso próprio do braço é desprezível (atuando no eixo z);
- Modelagem plana (plano Oxy);

- O braço robótico é um corpo rígido;
- Eficiência da transmissão de potência entre motor e braço é 100%;
- Efeitos de alteração de inércia do braço robótico devido ao seu acoplamento ao braço humano serão considerados posteriormente como incerteza paramétrica;

Dessa forma, obteve-se a equação diferencial da dinâmica do braço utilizando o Método de Lagrange. Foi assumido θ como a coordenada generalizada e M como o esforço externo não conservativo. A equação diferencial obtida está dada pela eq. (7.11), na qual $\ddot{\theta}$ é a aceleração angular, M é o momento aplicado pelo motor ao braço e J é o momento de inércia do braço.

$$\ddot{\theta} = \frac{M}{J} \quad (7.11)$$

7.3.1.2 Modelo Nominal do Motor

O motor utilizado na bancada é o EC 90 flat, ϕ 90mm, *brushless*, 90 Watt (*port number* 323772) [19]. Este é um motor *brushless* de corrente contínua. Pelo *datasheet* do mesmo, a eficiência máxima do motor é de 84%.

Apesar de existir uma diferença estrutural entre o motor *brushless* de corrente contínua (BLDC) e o motor de corrente contínua de ímã permanente, o modelo dinâmico do primeiro (para efeitos eletromecânicos) é o mesmo do segundo [29]. Portanto, a dinâmica do motor BLDC foi modelada como a de um motor DC, devido à maior simplicidade deste. A adequação entre os dois motores foi feita a partir da relação entre os parâmetros, como estabelecido em [29]. O modelo elétrico do motor CC de ímã permanente é dado pelo circuito da Figura 9, onde R é a resistência, L é a indutância, e_a é a queda de tensão na armadura, v é a tensão elétrica e i é a corrente.

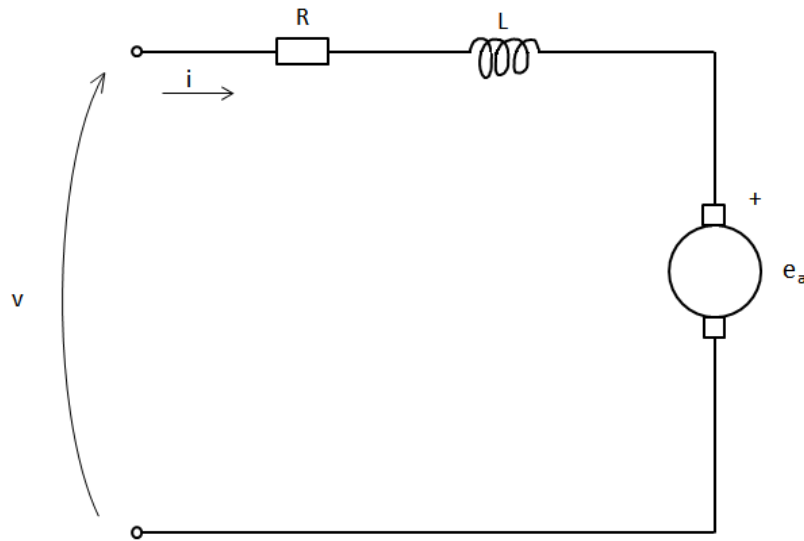


Figura 9: Circuito elétrico representando o modelo dinâmico simplificado do motor CC *brushless*.

Para modelar o motor a partir do circuito da Figura 9, foram adotadas as seguintes hipóteses:

- O modelo dinâmico do motor CC *brushless* foi aproximado ao modelo dinâmico de um motor CC com ímã permanente, devido a sua semelhança dinâmica em termos eletromecânicos. Esta aproximação pressupõe a consideração embutida do *driver* EPOS 70 juntamente com o motor BLDC, sendo esse responsável pelo controle das correntes nas fases do motor;
- Desconsiderou-se os efeitos de transição (devido à comutação) e os ruídos de corrente;
- Como a indutância está ligada a efeitos transitórios, desconsiderou-se a indutância do circuito (uma vez que nos interessa analisar o modelo em regime permanente);
- A rotação do rotor $\dot{\theta}_{rot}$ não é a mesma do exoesqueleto $\dot{\theta}$ devido à presença do redutor harmônico no motor. Vale a relação $r = 1/100$, retirada do *datasheet* do motor [19], que traz a relação: $\dot{\theta}_{rot} = \dot{\theta}/r$;
- O redutor e suas engrenagens são considerados perfeitamente rígidos;
- A eficiência da redução é representada pelo parâmetro η .

Aplicando a 2ª Lei de Kirchhoff ao circuito da Figura 9, obteve-se a eq. (7.12), na qual k_e é a constante elétrica do motor, e r é a relação de redução da redução harmônica do motor. Note que o produto desta com a velocidade angular é a queda de tensão e_a .

$$i = \frac{1}{R} \left(v - k_e \frac{\dot{\theta}}{r} \right) \quad (7.12)$$

7.3.1.3 Unindo os Modelos Nominais

Sabe-se que o torque nominal gerado pelo motor pode ser expresso pela eq. (7.13), onde k_t é a constante de torque do motor [Nm/A]. Como há um rendimento η na geração de torque do motor, pode-se escrever uma relação entre o momento aplicado no braço pelo motor e o torque nominal gerado pelo motor. Esta relação está dada pela eq. (7.14). Note que este rendimento η engloba efeitos dissipativos diversos entre o acoplamento do motor e o braço robótico.

$$T_{EM} = k_t i \quad (7.13)$$

$$M = \eta \frac{T_{EM}}{r} \quad (7.14)$$

Com as eqs. (7.13) e (7.14) na EDO do braço, eq. (7.11), é possível relacionar a equação obtida com a EDO do motor, eq. (7.12). Tal resultado representa a dinâmica da planta, e está dada pela eq. (7.15).

$$\ddot{\theta} = \frac{k_t \eta}{r J R} \left(v - k_e \frac{\dot{\theta}}{r} \right) \quad (7.15)$$

7.3.1.4 Função de Transferência da Planta

O objetivo do controle de baixo nível do projeto é controlar a posição angular do braço a partir da tensão do motor, seguindo assim sinais de referência de posição angular. Portanto, θ deve ser a saída da planta, e v a entrada.

A partir desta consideração e aplicando a transformada de Laplace na eq. (7.15), é possível obter a função de transferência da planta, $G(s)$, dada pela eq. (7.16).

$$G(s) = \frac{\theta(s)}{V(s)} = \frac{\frac{r}{k_e}}{s \left(\frac{J R r^2}{k_t k_e \eta} s + 1 \right)} \quad (7.16)$$

A partir da eq. (7.16), pode-se obter a expressão para a constante de tempo mecânica, como dado pela eq. (7.17), o que permite reescrever a função de transferência do modelo nominal como na eq. (7.18).

$$\tau_m = \frac{R J r^2}{k_t k_e \eta} \quad (7.17)$$

$$G(s) = \frac{\frac{r}{k_e}}{s (\tau_m s + 1)} \quad (7.18)$$

Vale ressaltar que, para realizar a compatibilização paramétrica entre o BLDC e o modelo DC utilizado, basta multiplicar por 3 a resistência fase a fase dada pelo *datasheet* do motor BLDC (R_{fase}), como colocado na eq. (7.19). Isso ocorre devido à natureza trifásica da estrutura elétrica interna do BLDC. Além disso, o cálculo da constante k_e está colocado na eq. (7.20), que é obtida a partir da relação de potências elétrica e mecânica do motor BLDC [29].

$$R = 3 R_{fase} \quad (7.19)$$

$$k_e = 0.0605 k_t \quad (7.20)$$

7.3.2 Incluindo Dinâmicas não Modeladas

As dinâmicas não modeladas para o modelo nominal podem ser um fator de incerteza que deve ser considerado no projeto do controlador robusto. A fim de avaliar se estas dinâmicas terão papel significativo sobre a operação da planta, foram analisados seus efeitos por meio da construção de "modelos reais". Apesar de, na teoria, existirem infinitos modelos reais, na prática selecionamos um modelo que inclua algumas dinâmicas mais relevantes para o problema. Para a obtenção deste modelo real, foram desenvolvidos modelos analíticos mais completos para o braço e para o motor.

Com a finalidade de avaliar a ação dessas dinâmicas incluídas, foram realizadas simulações para os modelos reais de forma separada. Ou seja, primeiro foi acoplado o modelo real do braço com o modelo nominal do motor, e em seguida o modelo nominal do braço com o real do motor.

As dinâmicas que foram incluídas nos modelos reais são: o efeito de elasticidade do exoesqueleto e o efeito da indutância no motor.

7.3.2.1 Modelo Real do Braço

A partir do modelo adotado para o braço robótico, foi considerada a hipótese de que o braço se comporta como corpo rígido. Para a obtenção do modelo real, foi levada em consideração a deformação deste exoesqueleto. Note que existe uma redução harmônica no motor, mas como os dentes das engrenagens são muito menores do que o braço do exoesqueleto, os engrenamentos continuarão sendo considerados como perfeitamente rígidos.

Para tanto, o braço foi modelado como um conjunto de n pequenas inércias ligadas entre si por n molas torcionais, como ilustrado na Figura 10. Nesta, θ_i representa a posição angular da i -ésima massa, T_i representa o torque provocado pela i -ésima mola torcional. Os elementos de inércia ΔJ_i possuem o mesmo valor, assim como os de mola Δk_i . Tal abordagem foi inspirada em [30]. Vale ressaltar que a inércia de número 1 foi considerada ligada a uma mola engastada, uma vez que estará fixamente acoplada ao motor. Ao mesmo tempo, aplica-se sobre esta inércia o torque M proveniente do motor.

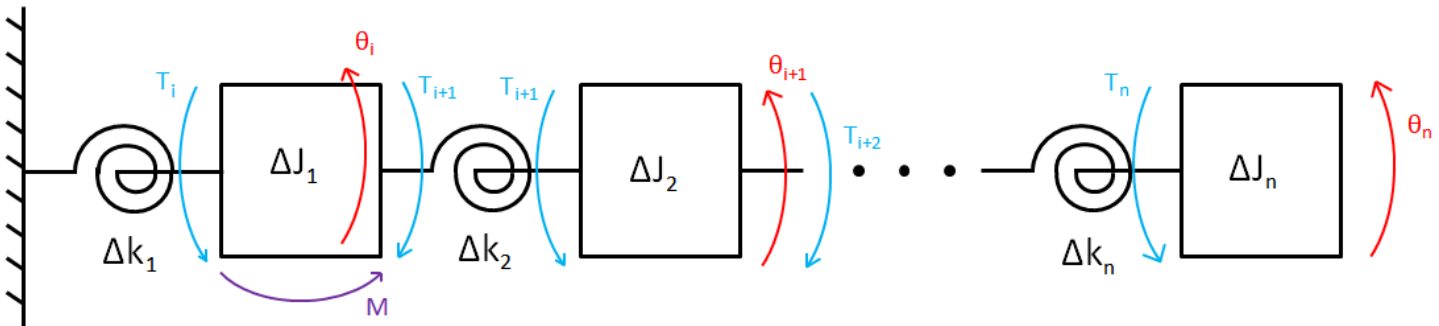


Figura 10: Esquema representando o modelo deformável, por parâmetros concentrados, para o exoesqueleto.

A partir do Diagrama de Corpo Livre de uma das inércias intermediárias qualquer, representado na Figura 11, foi obtida a equação diferencial do elemento (utilizando o Teorema do Momento Angular - TMA). Note que o torque aplicado por cada mola é dado

pela eq. (7.21). A EDO de cada inércia é dada pela eq. (7.22), na qual adotou-se como positivo o sentido anti-horário. Vale ressaltar que, como todas os elementos de inércia são iguais e os de mola também, $\Delta J_i = \Delta J$ e $\Delta k_i = \Delta k$.

$$T_i = (\theta_i - \theta_{i-1}) \Delta k_i \quad (7.21)$$

$$\Delta J \ddot{\theta}_i + \Delta k (-\theta_{i-1} + 2\theta_i - \theta_{i+1}) = 0 \quad (7.22)$$

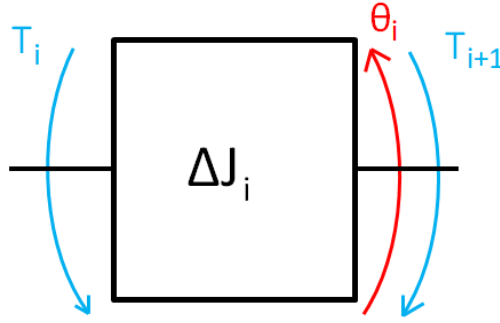


Figura 11: Esquema representando o Diagrama de Corpo Livre de um elemento intermediário de inércia, para o modelo deformável, por parâmetros concentrados, do exoesqueleto.

Para a primeira e a última inércia, a eq. (7.22) sofre algumas alterações. Para a inércia de número n , não há ação de T_{i+1} , portanto vale a eq. (7.23). Já para a primeira inércia (de número 1), há a ação do momento M aplicado pelo motor e também $\theta_{i-1} = 0$ devido ao engaste. Neste caso, vale a eq. (7.24).

$$\Delta J \ddot{\theta}_n + \Delta k (\theta_n - \theta_{n-1}) = 0 \quad (7.23)$$

$$\Delta J \ddot{\theta}_1 + \Delta k (2\theta_1 - \theta_2) + M = 0 \quad (7.24)$$

Com isso, fica evidente que a conexão com o modelo nominal do motor ocorre na primeira inércia - eq. (7.24). Utilizando as expressões obtidas no desenvolvimento do modelo nominal do motor [eqs. (7.14), (7.13) e (7.12)] é obtida a expressão para M , que substituída na eq. (7.24) resulta na eq. (7.25).

$$\Delta J \ddot{\theta}_1 + \Delta k (2\theta_1 - \theta_2) + \frac{\eta k_t}{R} \left(v - k_e \frac{\dot{\theta}_1}{r} \right) = 0 \quad (7.25)$$

Aplicando a Transformada de Laplace às eqs. (7.22), (7.23) e (7.25), é possível construir uma equação matricial representativa da dinâmica do sistema. Esta expressão é dada pela eq. (7.26). As expressões de $\alpha(s)$, $\beta(s)$ e $\gamma(s)$ são dadas, respectivamente, pelas eqs (7.27), (7.28) e (7.29).

$$\begin{bmatrix} \alpha(s) & -\Delta k & 0 & 0 & \dots & 0 \\ -\Delta k & \beta(s) & -\Delta k & 0 & \dots & 0 \\ 0 & -\Delta k & \beta(s) & -\Delta k & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & -\Delta k & \beta(s) & -\Delta k \\ 0 & \dots & 0 & 0 & -\Delta k & \gamma(s) \end{bmatrix} \cdot \begin{bmatrix} \theta_1(s) \\ \theta_2(s) \\ \theta_3(s) \\ \theta_4(s) \\ \vdots \\ \theta_n(s) \end{bmatrix} = \begin{bmatrix} \frac{\eta k_t}{r R} V(s) \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (7.26)$$

$$\alpha = s^2 \Delta J - s \frac{\eta k_t k_e}{R r^2} + 2 \Delta k \quad (7.27)$$

$$\beta = s^2 \Delta J + 2 \Delta k \quad (7.28)$$

$$\gamma = s^2 \Delta J + \Delta k \quad (7.29)$$

O que interessa para o problema é saber a posição angular da última inércia do braço θ_n . Isso porque a deformação angular será máxima na extremidade (uma vez que o sistema se comporta aproximadamente como uma barra engastada). Portanto, interessa a função de transferência dada pela eq. (7.30).

$$G_{R,braco,dinamicas}(s) = \frac{\theta_n(s)}{V(s)} \quad (7.30)$$

7.3.2.2 Modelo Real do Motor

Para obtenção do modelo real do motor, inclui-se a parcela da dinâmica referente à indutância do circuito da Figura 9. Novamente pela 2ª Lei de Kirchhoff, é obtida a

eq. (7.31).

$$R i + L \frac{di}{dt} = v - k_e \dot{\theta} \quad (7.31)$$

Aplicando a transformada de Laplace à eq. (7.31) e rearranjando termos, é obtida a eq. (7.32), representativa da dinâmica do motor BLDC (aproximado por um motor CC de ímã permanente).

$$I(s) = \frac{1}{R + sL} [V(s) - s k_e \theta(s)] \quad (7.32)$$

Utilizando as equações obtidas para o modelo nominal do braço, das eqs. (7.14) e (7.13) em (7.11), aplica-se a transformada de Laplace. Nesta, substitui-se a corrente, isolada na eq. (7.32), o que fornece a função de transferência da eq. (7.33) [29]. Note que τ_e é a constante de tempo elétrica do sistema, dada pela eq. (7.34). Além disso, a constante de tempo mecânica, τ_m , é a mesma do modelo nominal, dada pela eq. (7.17).

$$G_{R, motor, dinamicas}(s) = \frac{\frac{r}{k_e}}{\tau_m \tau_e s^2 + \tau_m s + 1} \quad (7.33)$$

$$\tau_e = \frac{L}{R} \quad (7.34)$$

7.3.3 Formulação das Incertezas

Nos Apêndices C.1 e C.2 estão colocadas as simulações dos modelos da planta que incluem efeitos de dinâmicas não modeladas. A partir destas, conclui-se que não serão consideradas incertezas dinâmicas para o desenvolvimento deste projeto. Isso porque tais incertezas não-estruturadas não têm efeito significativo dentro das condições de operação do sistema. Ou seja, para o desenvolvimento do controlador robusto, serão considerados apenas erros estruturados.

Como o controlador deve operar com diferentes braços humanos, garantindo o atendimento dos requisitos, será considerada incerteza paramétrica no valor da inércia do conjunto exoesqueleto-braço humano. Vale ressaltar que o controle do exoesqueleto não foi projetado para operar sem carga, portanto, daqui em diante será sempre considerado o valor da inércia conjunta do braço e do exoesqueleto. Tomando o valor utilizado em [19]

como valor médio deste parâmetro, foi considerada uma incerteza de $\pm 30\%$ sobre o mesmo. Esta margem foi arbitrada de modo a representar a realidade de forma satisfatória para o problema. Portanto, a inércia do conjunto exoesqueleto-braço humano J deve estar contida no intervalo $0.2198 \leq J \leq 0.4082$ [kg.m²].

Além disso, também foi incluída incerteza paramétrica no valor do rendimento do motor, uma vez que o *datasheet* do mesmo apenas fornece um valor máximo [19]. Aproveita-se para incluir neste parâmetro de rendimento a ação de efeitos dissipativos desconsiderados anteriormente. Uma vez que o valor máximo dado é de 84%, é razoável assumir um valor mínimo de 75% com certa segurança. Portanto, o rendimento η deve estar contido no intervalo: $0.75 \leq \eta \leq 0.84$.

Tratando-se, então, de um modelo que considera apenas incertezas paramétricas, o erro é representado na forma de *templates*. Dessa forma, o projeto do controlador foi baseado no método QFT.

8 ESTRUTURAÇÃO DOS ALGORITMOS E PROJETO DA MALHA DE CONTROLE

A construção dos algoritmos e o projeto da malha de controle serão abordados, respectivamente nas seções 8.1 e 8.2. Novamente, tal separação foi adotada para maior clareza.

8.1 Estruturação dos Algoritmos

Os algoritmos PCA, ICA e SVM foram importados da biblioteca Scikit-Learn [31], enquanto que os algoritmos CCA e SOM foram implementados em Python 3.7 [26], ambos presentes no *GitHub* do projeto [32]. Foram desenvolvidos ao todo três scripts para todo o processo do pré-processamento: o primeiro é responsável pela abertura do conjunto de dados original, limpando-o e criando uma coluna referente à classe para o treinamento do modelo. O segundo carrega esse novo conjunto de dados limpos e aplica cada algoritmo de pré-processamento separadamente, aplicando uma janela temporal e salvando esses dados processados. Por sua vez, o terceiro e último script abre os dados processados e treina os três modelos diferentes, comparando os valores de AUC-ROC, Sensibilidade e Especificidade.

8.2 Projeto da Malha de Controle

Como mencionado na seção 6.2, a malha de controle a ser projetada (Figura 3) possui 2 graus de liberdade: um referente ao pré-filtro $F(s)$ e um referente ao compensador $K(s)$. O projeto de cada um desses elementos da malha será abordado nas subseções seguintes, iniciando pelo projeto do pré-filtro.

8.2.1 Projeto do Pré-Filtro

O pré-filtro é responsável por garantir o atendimento dos requisitos temporais do sistema. A ideia é que a malha fechada de controle possua uma resposta temporal semelhante à imposta pelo pré-filtro. Com base na Seção 5.2, tem-se que estes requisitos se traduzem em um sobressinal nulo $M_p = 0$ e em um tempo de estabilização máximo de entre 1 e 2 segundos $1 \leq t_s^{2\%} \leq 2$. Como deseja-se um sobressinal nulo e é preferível que o sistema responda o mais rápido possível, o pré-filtro foi projetado como um sistema de segunda ordem com amortecimento crítico. Para isso, tem-se que $\zeta = 1$.

Apesar de a formulação usual para cálculo de tempo de estabilização 2% ser definida para sistemas sub-amortecidos, esta foi utilizada como base para estimativa do parâmetro ω_n . Utilizando a eq. (8.1) e partindo do valor mínimo estipulado para o tempo de estabilização, é obtido um valor de $\omega_n = 4$.

$$t_s^{2\%} = \frac{4}{\zeta \omega_n} \quad (8.1)$$

A partir dessa estimativa inicial, foi desenvolvido um código simples no Matlab para obter a resposta do pré filtro a uma entrada degrau unitário. Utilizando esta simulação, o parâmetro ω_n foi ajustado a fim de alcançar um tempo de estabilização dentro da faixa desejada.

Por fim, foi selecionado $\omega_n = 5$, o que resultou em um tempo de estabilização $t_s^{2\%} = 1.1668$ segundos, com sobressinal nulo. Dessa forma, a função de transferência do pré filtro está colocada na eq. (8.2).

$$F(s) = \frac{25}{s^2 + 10s + 25} \quad (8.2)$$

8.2.2 Projeto do Compensador

O compensador foi projetado com auxílio da ferramenta *open source* SISO-QFTIT, que permite realizar o projeto via QFT de forma visual e interativa para sistemas SISO (*Single Input, Single Output*). Vale ressaltar que o *software* utiliza a função de transferência da planta na forma de polos e zeros (e não na forma de constante de tempo, como colocado na eq. (7.16)). Para facilitar a compreensão, a eq. (8.3) representa a função de transferência da planta no formato de polos e zeros, na qual K_{QFT} representa

o ganho e P_{QFT} , o polo.

$$G(s) = \frac{\frac{k_t \eta}{r J R}}{s \left(s + \frac{k_t k_e \eta}{r^2 J R} \right)} = \frac{K_{QFT}}{s (s + P_{QFT})} \quad (8.3)$$

De acordo com os intervalos paramétricos considerados (subseção 7.3.3), tem-se que:

$$\begin{cases} 12.59 \leq K_{QFT} \leq 26.18 \\ 5.37 \leq P_{QFT} \leq 11.17 \end{cases}$$

Esta faixa de valores para o ganho da planta e para o polo real foram inseridas no *software*, que juntamente com a definição do vetor de frequência gerou os *templates* para o projeto. Assumiu-se os limites inferiores de ambos os parâmetros como o valor nominal.

O vetor de frequências deve ser uma discretização de frequências angulares que sejam representativas dentro das condições de operação do sistema. Partindo do requisito de frequência máxima de operação $4\pi/3$ (que equivale aproximadamente a 4.189) adotou-se os seguintes valores de frequências angulares, em rad/s: [0.1, 1, 4.2, 40]. Vale ressaltar que os *templates* representam todas as possíveis funções de transferência reais da planta (partindo das incertezas paramétricas). Os *templates* calculados para cada frequência do vetor de frequências estão expostos na Figura 12.

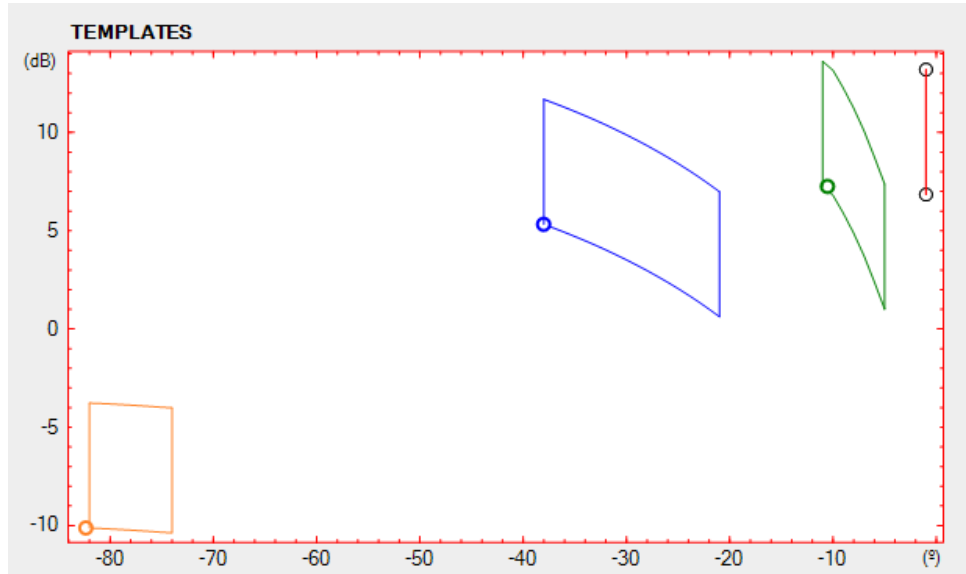


Figura 12: *Templates* para os valores de frequência contidos no vetor de frequências. O *template* vermelho representa a frequência angular de 0.1 rad/s; o verde, de 1 rad/s; o azul, de 4.2 rad/s; e o laranja, de 40 rad/s.

Para garantir a robustez da estabilidade, define-se um limiar $M > 0$, sobre o qual deve valer a condição dada pela eq. (8.4). Nesta, $T(j\omega_i)$ representa a sensibilidade complementar da malha fechada, calculada para cada valor ω_i do vetor de frequências. Ressaltando que, quanto maior o valor de M , mais próximo o sistema pode chegar ao ponto crítico.

$$|T(j\omega_i)| < M \quad (8.4)$$

A partir da eq. (8.3), percebe-se que a planta pode ser interpretada como um sistema de primeira ordem com um integrador. Portanto, não possui um pico de ressonância como sistemas subamortecidos de 2ª ordem. Tal conclusão teórica se confirma com o diagrama de Bode obtido através do Matlab e colocado aqui na Figura 13.

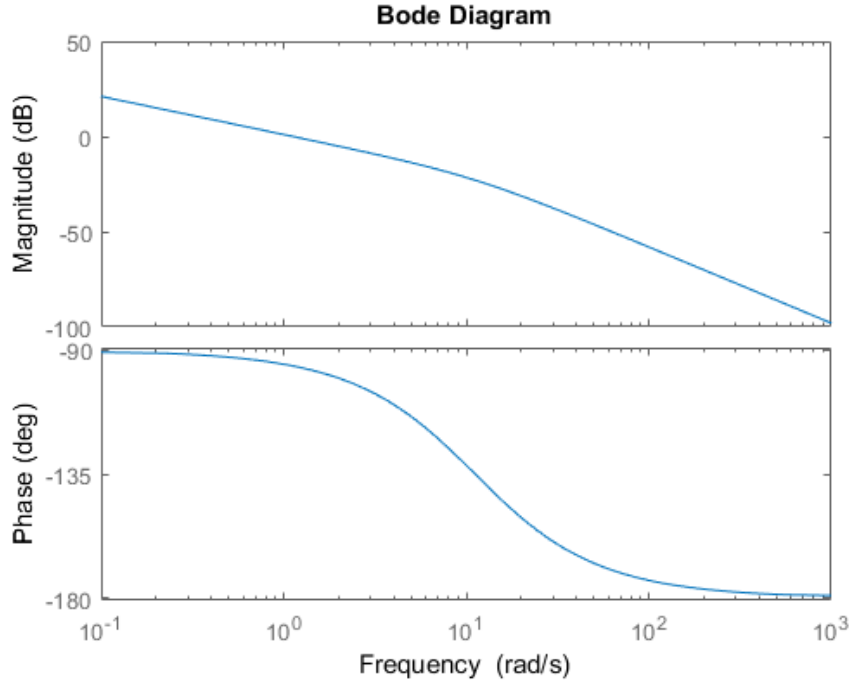


Figura 13: Diagrama de Bode para a planta nominal.

Partindo desta constatação, pode-se assumir valores mais elevados de M sem riscos ao projeto. Utilizando a ferramenta iterativa do SISO-QFT, adotou-se $M = 3$. Isto gerou as barreiras de robustez da estabilidade no plano de Nichols, de acordo com a Figura 14. Vale ressaltar que, nesse *software*, as linhas cheias representam barreiras que não podem ser ultrapassadas para baixo, e linhas tracejadas não podem ser ultrapassadas para cima.

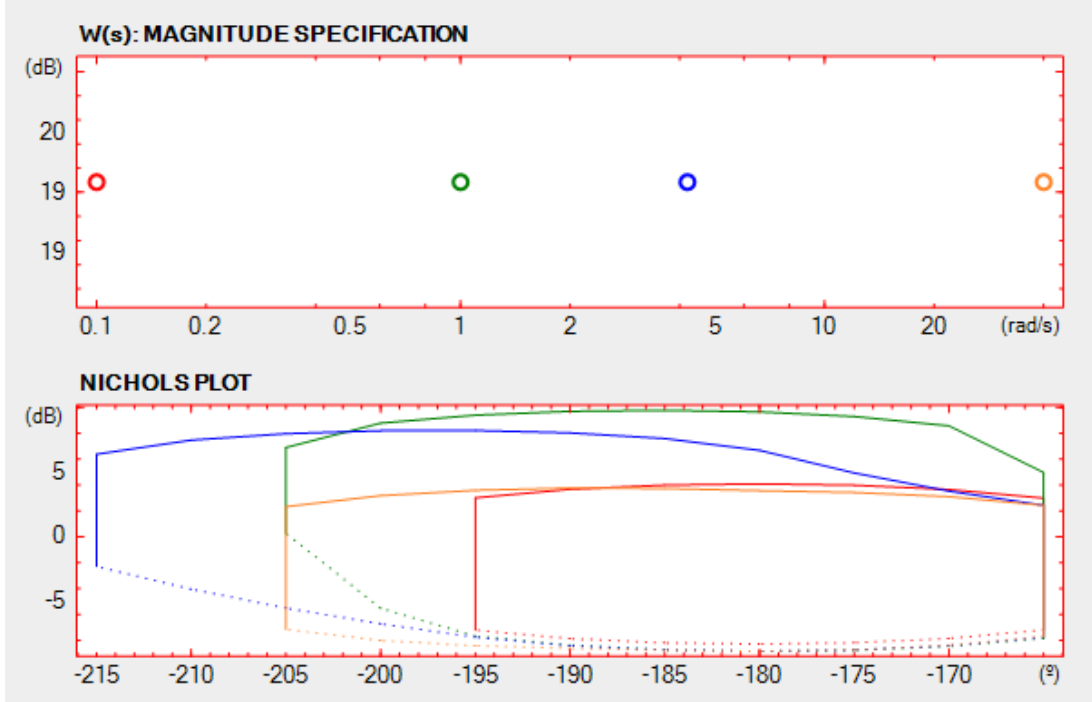


Figura 14: Barreiras de robustez da estabilidade no plano de Nichols para $M = 3$. O esquema de cores e frequências segue o padrão explicado na legenda da Figura 12.

Estabelecida a condição de robustez de estabilidade, é necessário definir os requisitos de desempenho do controle. Estes são o acompanhamento do sinal de referência e a compatibilização do pré-filtro com a malha fechada. Ambas são condições de baixa frequência, dadas respectivamente pelas eqs. (8.5) e (8.6). Vale lembrar que δ_r é a tolerância para o acompanhamento do sinal de referência (estipulada como 0.05 nos requisitos) e δ_F é a tolerância para a compatibilização do pré-filtro (também dada como 0.05 nos requisitos). Além disso, $F(j\omega)$ é a função de transferência do pré-filtro, previamente projetada e dada pela eq. (8.2).

$$\frac{1}{|1 + G_R(j\omega) K(j\omega)|} \leq \delta_r(\omega) \quad (8.5)$$

$$\frac{1}{|1 + G_R(j\omega) K(j\omega)|} \leq \frac{\delta_F(\omega)}{|F(j\omega)|} \quad (8.6)$$

No *software* SISO-QFTIT, as eqs. (8.5) e (8.6) se enquadram ambas em restrição do tipo 2. Portanto, toma-se para o projeto apenas aquela que é mais restritiva. No caso do vetor de frequências escolhido, sendo que $\delta_r = \delta_F$, a eq. (8.5) é mais restritiva, pois $|F(j\omega)|$ é ligeiramente menor do que 1 em baixas frequências. As barreiras de desempenho, que só são aplicadas até a frequência de 4.2 rad/s, estão representadas na Figura 15. Vale

ressaltar que estes valores de tolerância são inseridos em dB no *software*.

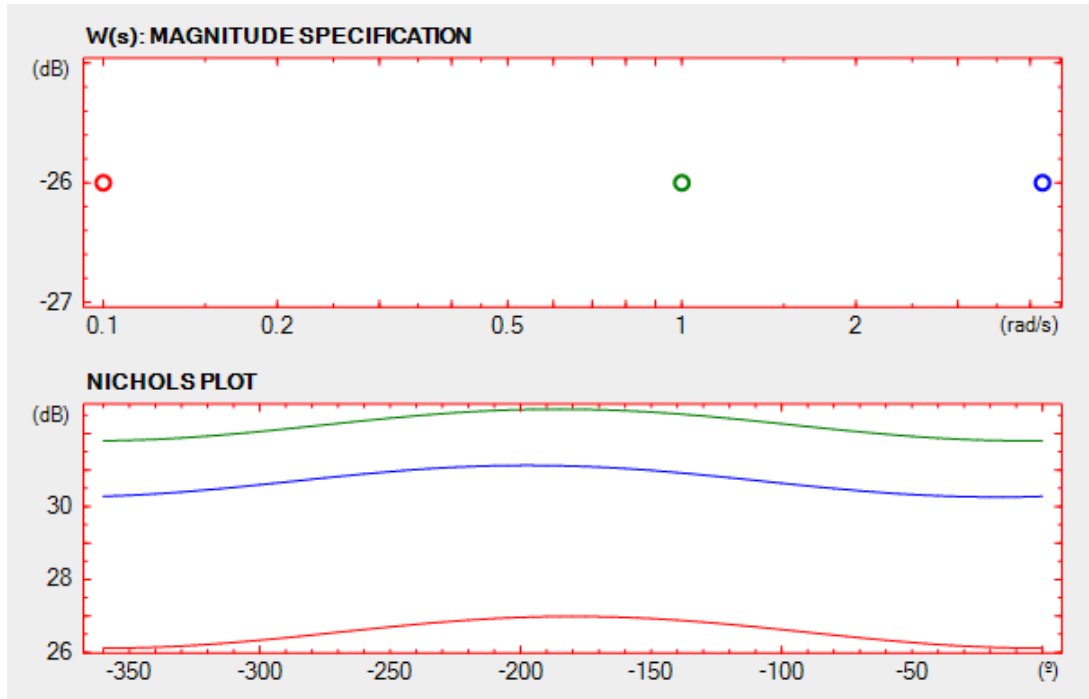


Figura 15: Barreiras de robustez do desempenho (acompanhamento do sinal de referência e compatibilização da malha fechada com o pré-filtro) para tolerância de 0.05. O esquema de cores e frequências segue o padrão explicado na legenda da Figura 12.

Finalmente, parte-se para o projeto do compensador a partir da resposta em frequência da função de transferência de malha ($G(s)K(s)$) representada no plano de Nichols juntamente com as barreiras de robustez da estabilidade e robustez do desempenho. Note que, inicialmente, tem-se que $K(s) = 1$, resultando na Figura 16.

A ideia para o projeto é construir um compensador de forma que os pontos vermelho, verde e azul fiquem acima das barreiras sólidas de mesma cor e o ponto amarelo não adentre a região amarela em torno do ponto crítico (0dB e -180°).

Para tanto, projetou-se o compensador composto por um ganho e um avançador de fase, com frequência de corte em 1 rad/s (ponto verde). A partir de cálculos e ajustes finos com o software, chegou-se à função de transferência do compensador, dada pela eq. (8.7). A resposta em frequência, no plano de Nichols, da função de transferência de malha com o compensador projetado está representada na Figura 17.

$$K(s) = \frac{8.7(2.82s + 1)}{s + 1} \quad (8.7)$$

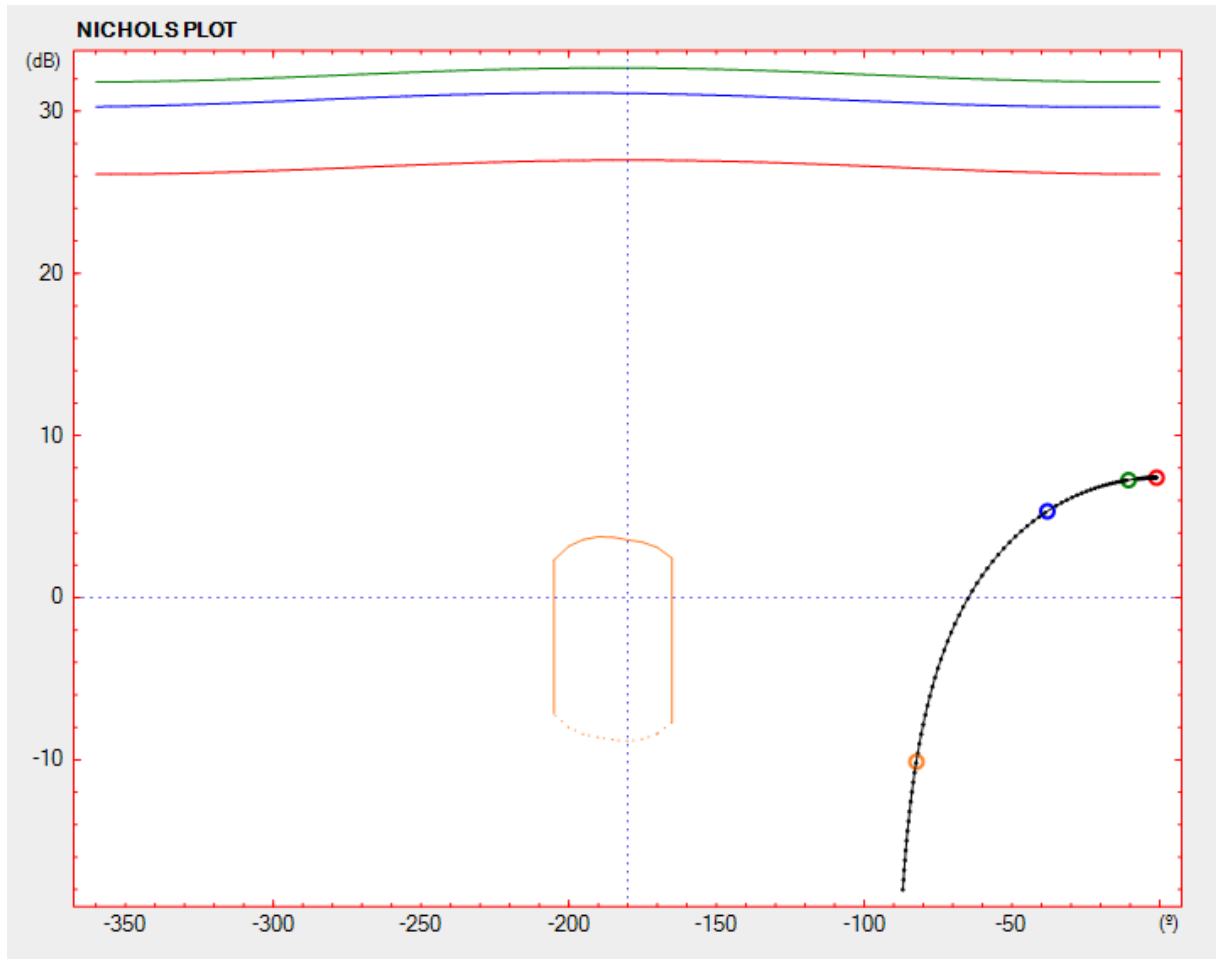


Figura 16: Resposta em frequência da função de transferência de malha ($G(s)K(s)$) representada no plano de Nichols juntamente com as barreiras de robustez da estabilidade e robustez do desempenho; para $K(s) = 1$.

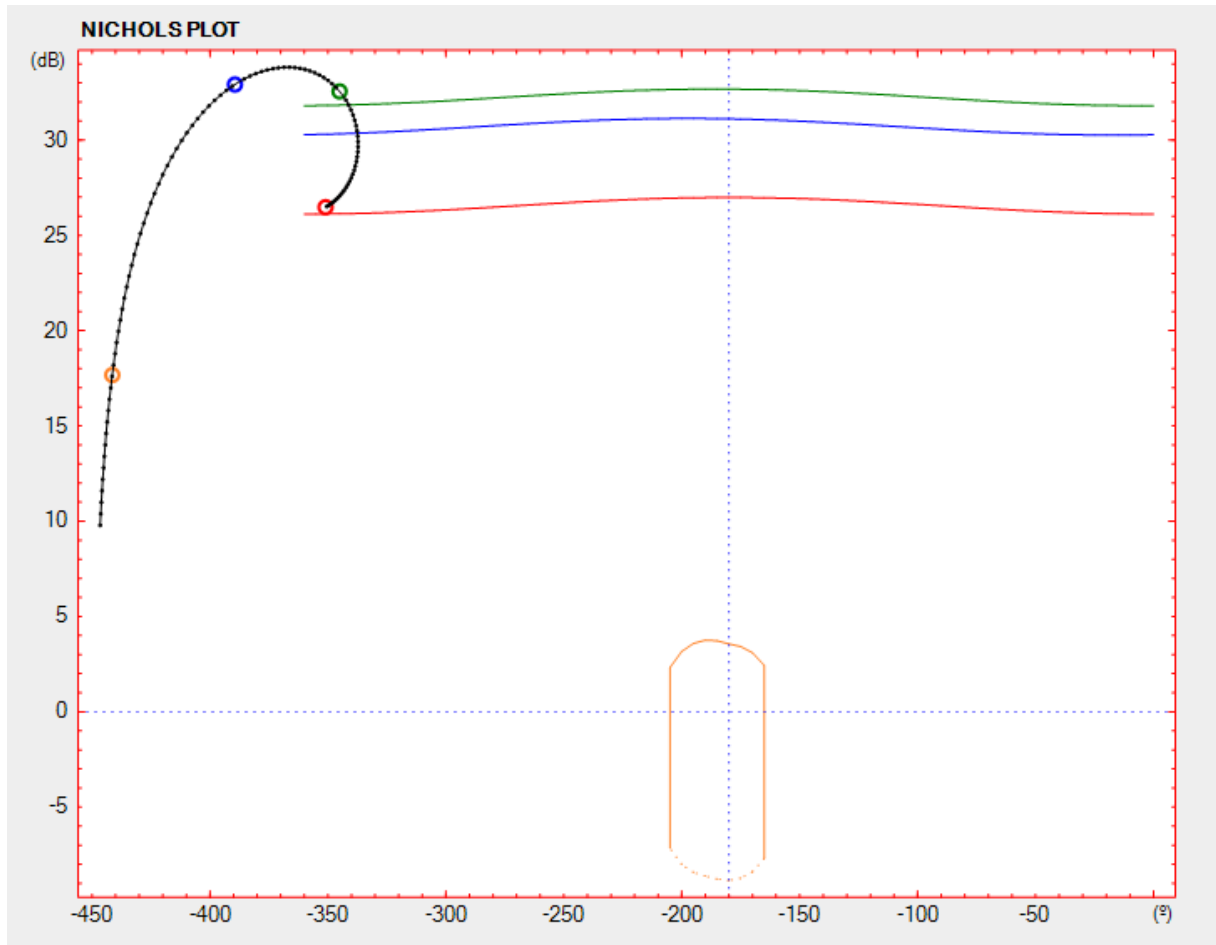


Figura 17: Resposta em frequência da função de transferência de malha ($G(s)K(s)$) representada no plano de Nichols juntamente com as barreiras de robustez da estabilidade e robustez do desempenho; para o compensador projetado (eq. (8.7)).

9 SIMULAÇÕES DOS SUB-SISTEMAS

Nas seções seguintes serão expostas as simulações computacionais realizadas para o desenvolvimento das duas partes que compõe o presente trabalho, de forma separada. Para tanto, foi novamente feita a separação entre a parte de pré-processamento e classificação (Seção 9.1) e de controle (Seção 9.2) para maior clareza.

9.1 Simulações para Pré-Processamento

9.1.1 Pré-processamento Linear

9.1.1.1 PCA

Para a redução dimensional utilizando o algoritmo PCA, foram selecionados os componentes principais até alcançar ao menos 99% de variância contida nos dados. Para cada participante do estudo, foi gerado um gráfico contendo a variância total pelo número de componentes principais. Um desses gráficos, para o Sujeito 15, está indicado na Figura 18. Nesse exemplo, um mínimo de 19 componentes são necessários para explicar 99% da variância.

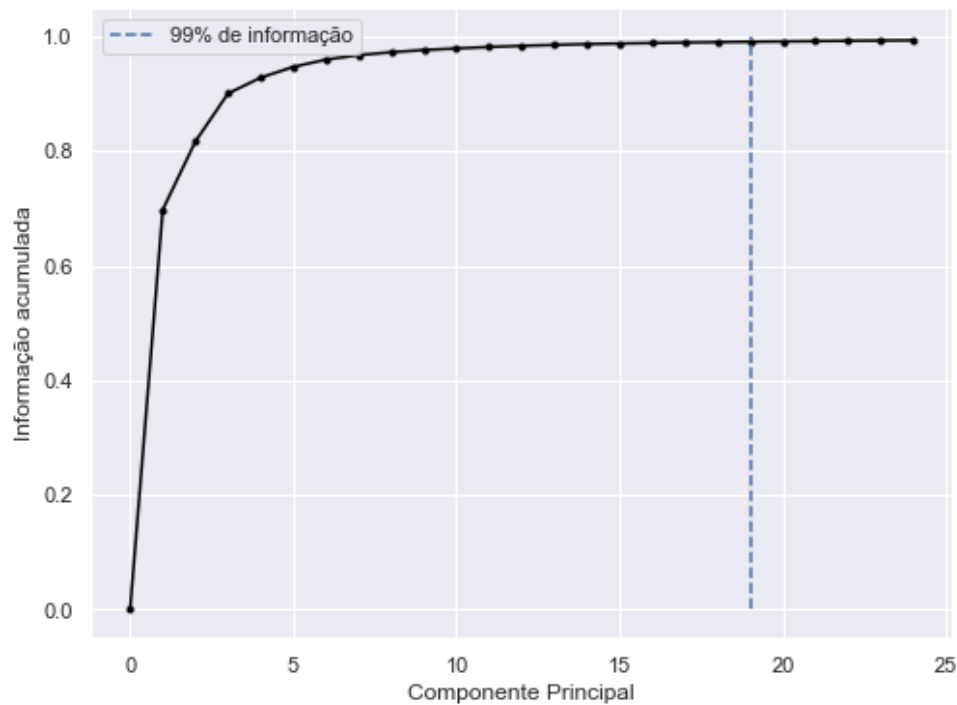


Figura 18: Variância total em função do número de componentes principais para o Sujeito 15.

9.1.1.2 ICA

No caso do ICA, tentou-se reduzir a dimensão dos dados usando o número de componentes encontrados pelo algoritmo PCA. No entanto, para a maioria dos dados, o ICA não conseguia convergir com essa nova dimensionalidade. Visto esse comportamento, foi decidido não utilizar o ICA para redução dimensional, e sim para realizar uma transformação linear no conjunto de dados. Essa transformação realizada busca encontrar sinais independentes, característica que faz esse algoritmo muito utilizado em dados de EEG.

9.1.2 Pré-processamento Não-Linear

9.1.2.1 SOM

Para a tarefa de *Vector Quantization*, foi utilizado o algoritmo SOM-DL com um mapa de 40 por 40 neurônios, totalizando 1600. Seguindo a recomendação do autor, o número de iterações de treinamento foi de 801000 (1000 para organização do mapa no espaço e 500×1600 para especialização de cada neurônio). Após o treinamento, os centros foram salvos para serem utilizado no CCA e para aplicar nos outros conjuntos de dados.

9.1.2.2 CCA

O algoritmo CCA foi treinado com os neurônios do SOM, reduzindo a dimensão desses centros. Para determinar os parâmetros necessários para a redução, foi seguido o procedimento descrito em [33]: inicia com uma dimensão elevada e reduz aos poucos. Se os pontos na representação dy-dx [34] (relaciona as distâncias dos pontos no espaço original com o espaço reduzido) ficarem na diagonal principal, significa que a dimensão escolhida é superior à intrínseca. Quando os pontos saírem da diagonal principal, significa que a dimensão atual é a correta. Para estimar o valor de λ , utiliza-se novamente a representação dy-dx. Dessa vez, escolhe-se um valor de lambda que apresente pontos abaixo da diagonal principal. Um exemplo do resultado final desse processo pode ser visualizado na Figura 19.

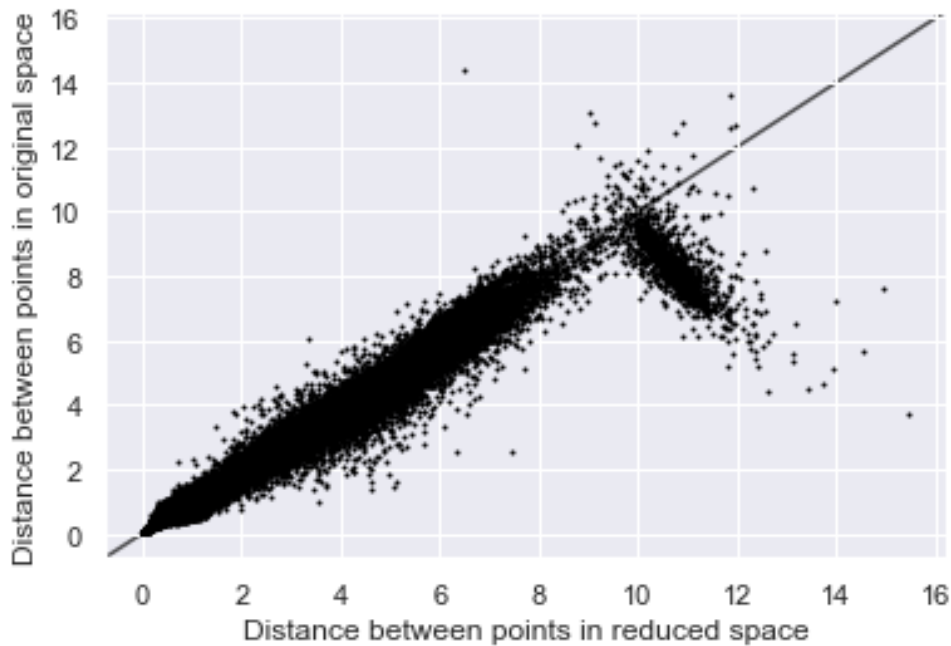


Figura 19: Representação dy-dx do Sujeito 15. Esse gráfico relaciona as distâncias entre os pontos no espaço original (eixo y) com as distâncias no espaço de dimensão reduzida (eixo x).

9.2 Simulações para Controle

A fim de validar o projeto do pré-filtro (subseção 8.2.1) e do compensador (subseção 8.2.2), foi desenvolvido um código em Matlab para realizar as validações do sistema de controle tanto em frequência como no tempo. Este código está colocado no *GitHub* do projeto [32]. Para essas validações, foram simuladas várias possíveis plantas reais, a fim de garantir a verificação da robustez do projeto. O algoritmo está descrito em detalhe no Apêndice D.

Para a validação em frequência, realizou-se uma varredura de frequências dentro do intervalo de atuação (definido nos requisitos do projeto - seção 5.2). Para cada valor de frequência angular, o sistema foi excitado com uma senoide. A partir disso, foi verificado se as relações de sinais dadas pelas eqs. (9.1) e (9.2) estavam respeitando as tolerâncias definidas na seção 5.2. Vale relembrar que os sinais seguem a nomenclatura da Figura 3.

Mapeando o máximo valor de delta obtido por frequência (uma vez que, para cada frequência, foram simuladas várias plantas), foram obtidos os resultados expostos nas Figuras 20 e 21. Note que, em ambas as figuras, o valor calculado não ultrapassa 0.05, que foi a tolerância estipulada nos requisitos (tanto para o acompanhamento do sinal de referência como para a compatibilização do pré-filtro com a malha fechada). A partir do algoritmo é obtido que o máximo valor de δ_r é 0.042803, que ocorre em $\omega = 4.2$ rad/s para $K_{QFT} = 12.588$ e $P_{QFT} = 11.168$; enquanto o valor máximo de δ_F é 0.047429 e ocorre também em $\omega = 4.2$ rad/s para a mesma planta real. Portanto, o compensador projetado foi devidamente validado na frequência.

$$\frac{|e(j\omega)|}{|r(j\omega)|} \leq \delta_r \quad (9.1)$$

$$\frac{|e(j\omega)|}{|r_1(j\omega)|} \leq \delta_F \quad (9.2)$$

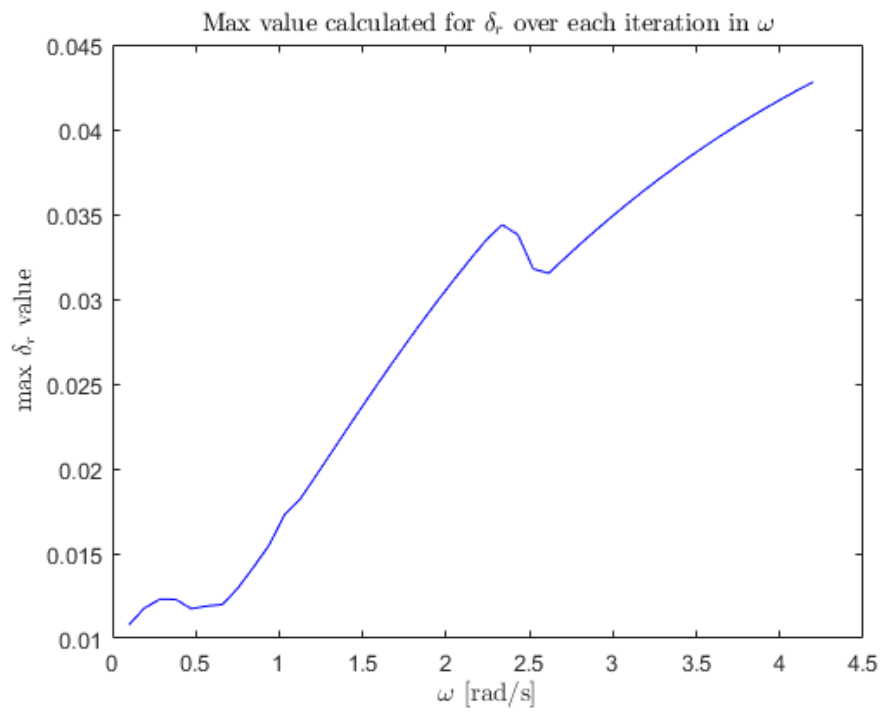


Figura 20: Valor máximo de δ_r para cada frequência ω , simulando para um conjunto de plantas reais.

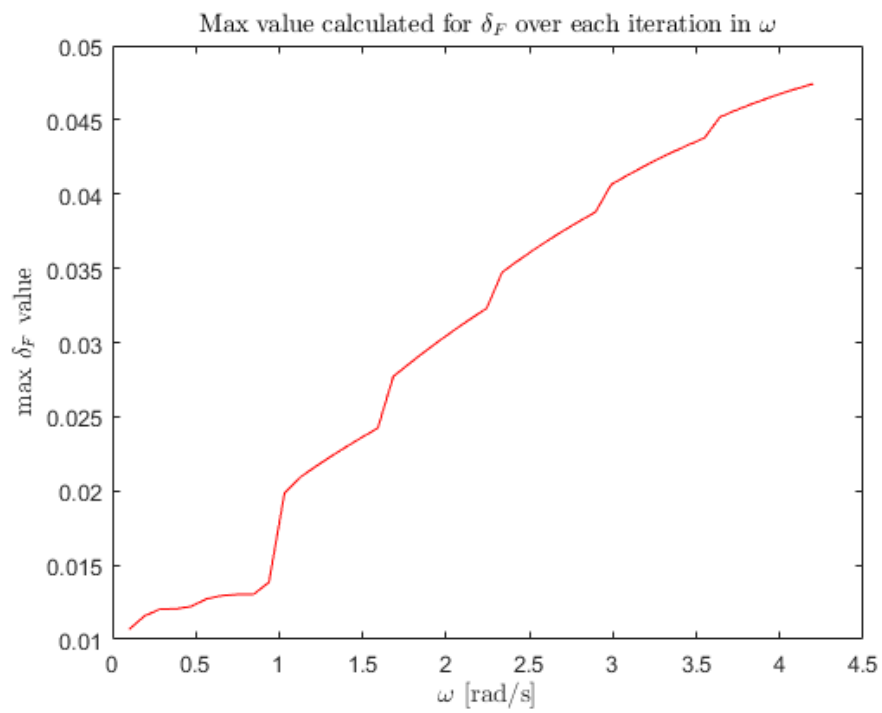


Figura 21: Valor máximo de δ_F para cada frequência ω , simulando para um conjunto de plantas reais.

A fim de avaliar a resposta temporal do sistema de controle, foi primeiro necessário simular a resposta do pré filtro a uma entrada degrau (uma vez que a saída do sistema deve seguir qualitativamente a resposta dinâmica do pré-filtro). Na Figura 22 está representada a resposta ao degrau unitário do pré-filtro projetado. Com este, foi obtido um sobressinal nulo e um tempo de estabilização 2% de 1.1668 segundos, como colocado na Tabela 2.

Tabela 2: Informações da resposta do pré-filtro ao degrau unitário.

<i>Rise Time (t_r)</i>	0.6717
<i>Settling Time ($t_s^{2\%}$)</i>	1.1668
<i>Overshoot (M_p)</i>	0

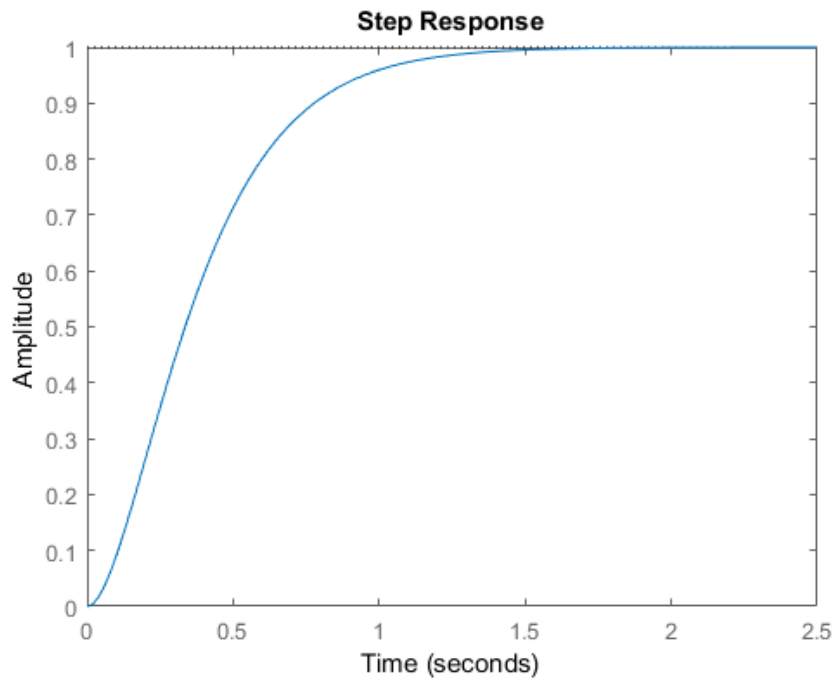


Figura 22: Resposta do pré-filtro à entrada degrau unitário.

Em seguida, foi avaliada a resposta do sistema de controle ao degrau unitário, simulando novamente para diversas expressões de plantas reais. Como a resposta do sistema como um todo sempre apresenta sobressinal nulo (devido ao amortecimento crítico no pré filtro), foi mapeado o tempo de estabilização de cada planta real, de acordo com a variação dos parâmetros da planta K_{QFT} e P_{QFT} . Isso resultou na superfície colocada na Figura 23, que dá um valor máximo de 1.525 segundos, para $K_{QFT} = 12.588$ e $P_{QFT} = 11.168$ (mesma planta real para a qual foram obtidos os maiores valores de tolerância na validação em frequência). Portanto, o tempo de estabilização do sistema é um pouco maior

do que o do pré filtro, mas permanece ainda tranquilamente dentro da faixa permitida nos requisitos.

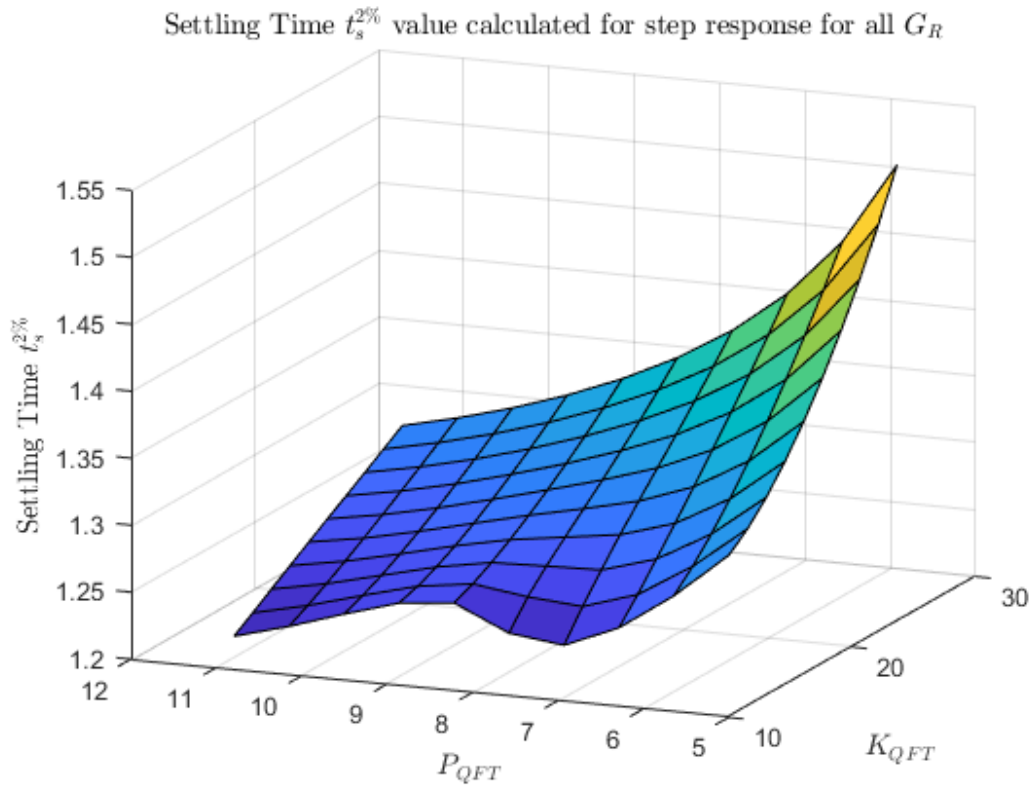


Figura 23: Valores obtidos para o tempo de estabilização 2% do sistema de acordo com as variações paramétricas das plantas reais simuladas.

A partir desta avaliação, selecionou-se esta planta real mais crítica para avaliar a resposta ao degrau com amplitude de 60° (1.0472 rad), a fim de já obter uma validação sobre a condição na qual o sistema irá atuar. O resultado está colocado na Figura 24, que forneceu um tempo de estabilização 2% de 1.525 segundos e sobressinal nulo, como exposto na Tabela 3. Vale ressaltar que a comparação pode ser feita diretamente entre as respostas simuladas, mesmo com amplitudes de degrau diferentes, pois o sistema é linear. Para sistemas lineares, as características da resposta dinâmica são preservadas.

Tabela 3: Informações da resposta do sistema de controle ao degrau de 1.0472 rad.

<i>Rise Time</i> (t_r)	0.683
<i>Settling Time</i> ($t_s^{2\%}$)	1.525
<i>Overshoot</i> (M_p)	0

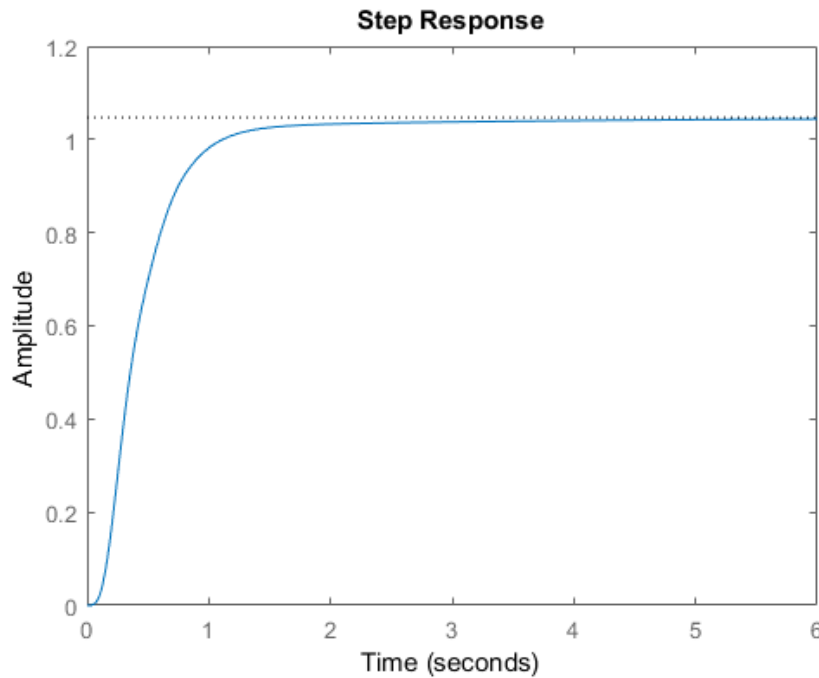


Figura 24: Resposta do sistema de controle à entrada degrau de 1.0472 rad.

Além desta análise, também é necessário avaliar a velocidade angular, a aceleração angular e o esforço de controle do sistema para a condição de operação.

A velocidade angular e a aceleração angular foram obtidas derivando a resposta do sistema ao degrau, que representa a evolução da posição angular no tempo. Tais resultados estão colocados nas Figuras 25 e 26, respectivamente. O pico de velocidade angular está próximo de 2.4 rad/s, estando bem abaixo do limite máximo estipulado nos requisitos (de 4.189 rad/s). Também na aceleração, atinge-se um pico de valor inferior ao máximo estipulado nos requisitos (atinge-se um valor próximo a 16 rad/s², sendo o limite máximo de 20 rad/s²). Vale ressaltar que a operação de derivação amplifica ruídos. Portanto, projetou-se um filtro digital de fase zero para o sinal da aceleração (com corte em 5 Hz), pois esta apresentava muito ruído dependendo da simulação. Nesta configuração final do controlador, praticamente não houve ruído no sinal de aceleração analisado. Isto pode ser percebido pelas curvas vermelha e azul coincidindo na Figura 26.

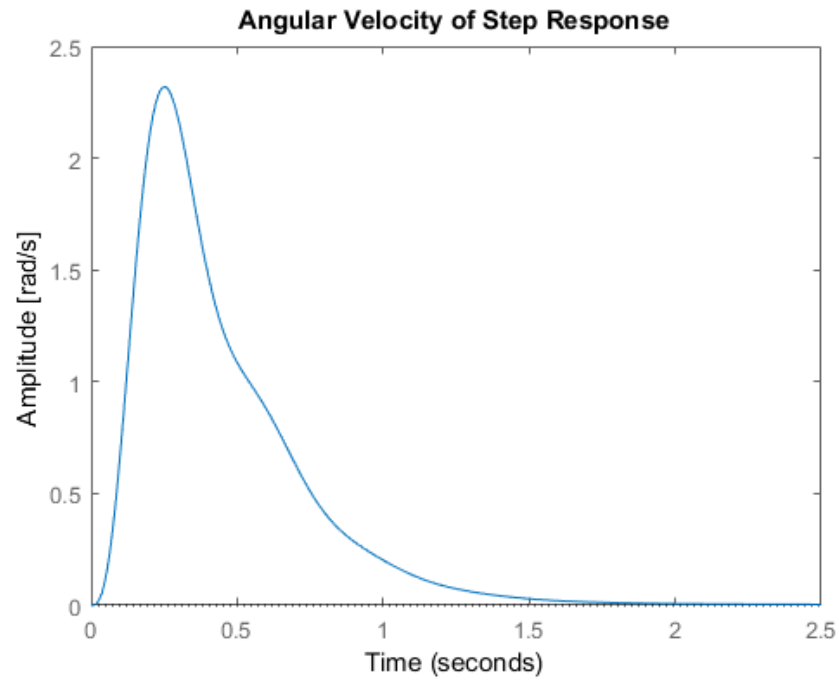


Figura 25: Velocidade angular do sistema a partir da derivação da resposta à entrada degrau de 1.0472 rad.

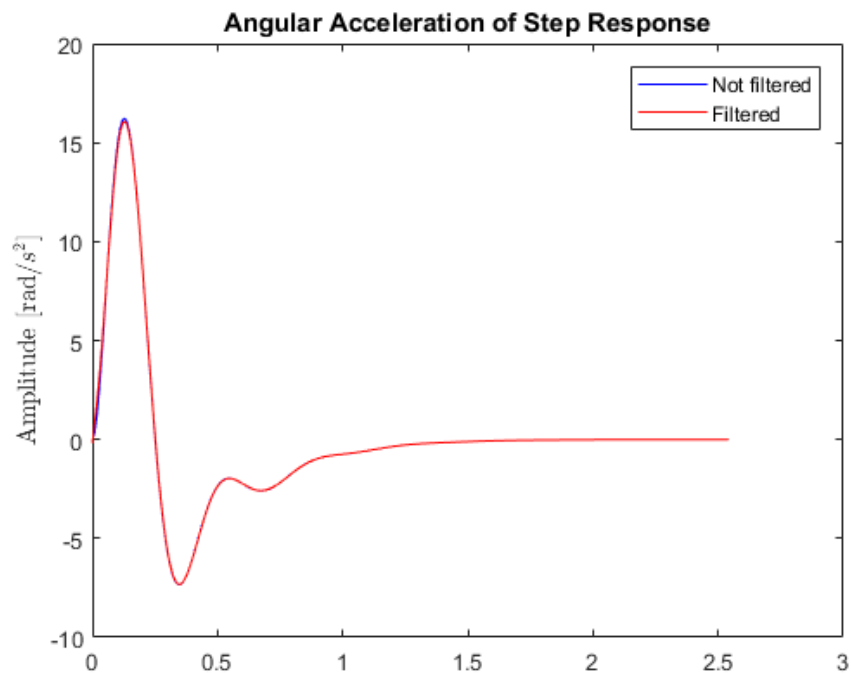


Figura 26: Aceleração angular do sistema a partir de duas derivações da resposta à entrada degrau de 1.0472 rad. A curva vermelha representa o este sinal filtrado por um filtro digital de fase zero com frequência de corte em 5 Hz.

Por fim, o esforço de controle $u(s)$ obtido para a resposta do sistema ao degrau de 1.0472 rad está representado na Figura 27. Percebe-se que há um pico de tensão de aproximadamente 2.6 V , sendo este um valor viável de implementar na prática.

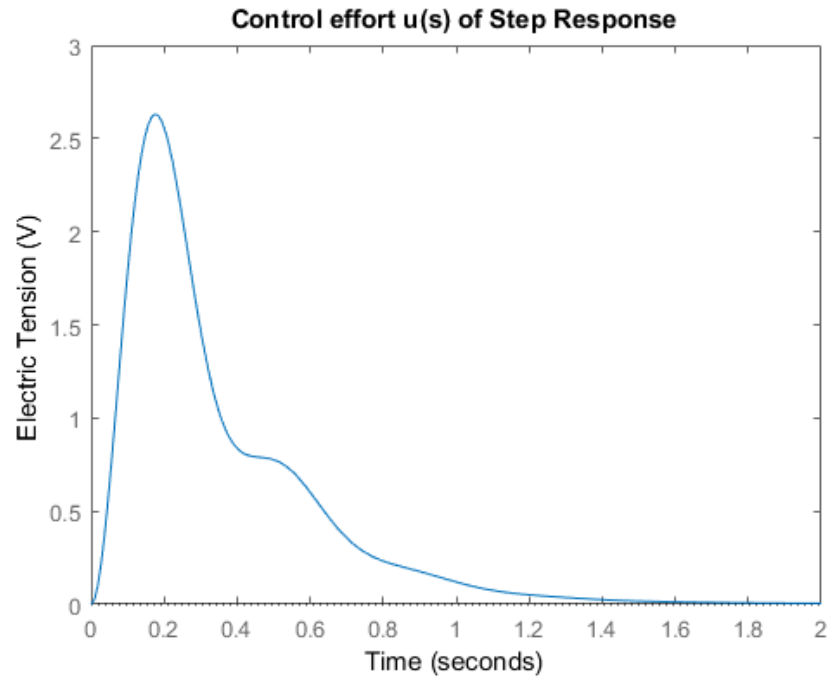


Figura 27: Esforço de controle do sistema para a resposta à entrada degrau de 1.0472 rad.

Portanto, analisando a resposta temporal do sistema em termos de posição angular, velocidade angular, aceleração angular e esforço de controle, o compensador foi também devidamente validado no tempo.

10 INTEGRAÇÃO DAS PARTES - SIMULAÇÕES

Devido à pandemia de COVID-19, não foi possível realizar a implementação física do presente trabalho e seus consequentes testes na bancada do laboratório. Portanto, foram desenvolvidos códigos para simular os sistemas funcionando de forma integrada e validar seu correto funcionamento. Para maior clareza, serão inicialmente descritas as partes que compõe a simulação de cada subsistema (pré-processamento e classificação; e controle) nas seções 10.1 e 10.2 respectivamente. Em seguida, a simulação do sistema devidamente integrado está exposta na seção 10.3. Vale adiantar que todas estas simulações foram realizadas utilizando o Python. Além disso, todos os códigos estão disponíveis no *GitHub* do projeto [32].

10.1 Simulação Integrada - Pré-Processamento

Para a simular o pré-processamento, foi criado um script Python *run.py* que acessa uma pasta contendo arquivos do: filtro, algoritmo de redução dimensional (ICA, CCA e PCA), os modelos treinados (um para cada algoritmo não-supervisionado) e os dados para a simulação. É selecionado uma linha do conjunto de dados para simular um sinal novo chegando, com frequência escolhida de 250Hz, passando pelo filtro, pelo algoritmo de pré-processamento e pelo modelo. Foi implementado um gráfico adaptativo que mostra os últimos 4s de simulação, permitindo que o usuário veja em tempo real como o sistema está se comportando, como indica a Figura 28. O usuário também pode escolher qual algoritmo utilizar na simulação colocando o nome da técnica na chamada do script pelo terminal (por exemplo, 'python run.py cca' para rodar a simulação utilizando o CCA).

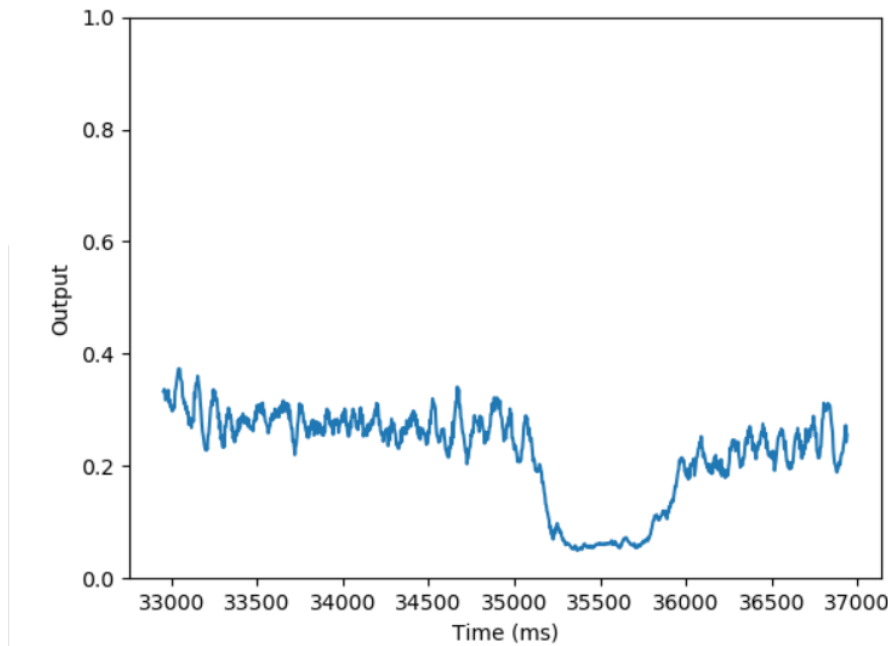


Figura 28: Print da saída do modelo (probabilidade da detecção do movimento), em tempo real, gerado pelo Script.

Como dito anteriormente, para a escolha do , a metodologia escolhida foi atribuir um custo relacionado à taxa de falso positivo e um à taxa de falso negativo. O *threshold* que minimiza o custo total é o utilizado no modelo.

10.2 Simulação Integrada - Controle

O objetivo aqui é simular a parte de controle de forma a se aproximar ao que seria sua implementação física. Para a implementação na bancada de teste seria necessário codificar as funções de transferência do pré-filtro e do controlador em um microcontrolador, a fim de gerar a tensão elétrica que serviria de entrada para a planta física. O *encoder* do motor seria o responsável pela medição da saída da planta física e subsequente realimentação deste sinal na malha de controle no microcontrolador. Como não é possível utilizar a planta física (devido à pandemia de COVID-19), a função de transferência que descreve a mesma (de acordo com a modelagem analítica desenvolvida no presente trabalho) foi também colocada no código. Dessa forma, foi desenvolvido um código de simulação que permite a observação das variáveis de controle, de acordo com o sinal vindo da parte de pré-processamento (mais especificamente o sinal de classificação).

Com o intuito de facilitar a utilização e organizar o código de simulação do controle, foi desenvolvida uma estrutura baseada em 2 classes. A primeira, denominada *LLC*, implementa o controle de baixo nível, armazenando todas as variáveis de interesse na instância do objeto desta classe. A segunda classe é *HLC*, na qual é implementado o controle de alto nível do projeto, responsável por receber o sinal da classificação do pré-processamento e aplicar a uma lógica de estados que comanda o controle de baixo nível (ou seja, comanda a movimentação do exoesqueleto).

Nas subseções seguintes é explicado com mais detalhe a implementação das classes *LLC* e *HLC*, seus principais métodos e funcionamento. O código que implementa essa arquitetura está devidamente comentado e disponível no GitHub [32].

10.2.1 Classe para Controle de Baixo Nível - *LLC*

A fim de implementar no código as funções de transferência do sistema é necessário voltar ao domínio do tempo (através da Transformada Inversa de Laplace) e então resolver numericamente as EDOs obtidas. A abordagem mais simples para a resolução numérica das EDOs é utilizar o método das diferenças finitas [35]. Isto é feito pois com as diferenças finitas não é necessário utilizar um algoritmo iterativo para o cálculo de cada ponto, o que torna o algoritmo mais rápido e mais adequado para a execução em tempo real.

A simulação deve seguir o fluxo de sinal representado no diagrama de blocos do sistema (Figura 3). Para tanto, foram isoladas nas equações as saídas de cada função de transferência. Vale lembrar que as funções de transferência do pré-filtro, compensador e planta estão dadas nas eqs. (8.2), (8.7) e (8.3) respectivamente. Outro fator importante é que o projeto foi realizado para diversas plantas, uma vez que foi desenvolvido um controle robusto, de acordo com as variações de parâmetros estipuladas e os critérios de robustez. Para garantir que as simulações vão atender todas as plantas, selecionou-se os parâmetros que forneceram os piores resultados nas simulações do controle, como exposto na Seção 9.2. Logo, assume-se que $K_{QFT} = 12.588$ e $P_{QFT} = 11.168$.

Para a solução das EDOs usando o MDF, aproximou-se as primeiras derivadas pela primeira diferença regressiva de 3 pontos, que possui erro na ordem do passo ao quadrado h^2 . Já para as segundas derivadas, utilizou-se a segunda diferença regressiva, que possui erro na ordem do passo h . Ambas as aproximações são dadas pelas eqs. (10.1) e (10.2) de forma genérica.

$$f'_i = \frac{3f_i - 4f_{i-1} + f_{i-2}}{2h} + O(h^2) \quad (10.1)$$

$$f''_i = \frac{f_i - 2f_{i-1} + f_{i-2}}{h^2} + O(h) \quad (10.2)$$

No caso das EDOs do sistema, o passo é dado pelo período de amostragem T_s , que se relaciona com a aquisição de dados por meio dos sensores do sistema (no caso deste trabalho seria o *encoder*). Os resultados obtidos para a solução aproximada das EDOs do sistema estão colocadas nas eqs. (10.3), (10.4), (10.5) e (10.6). Note que, nestas equações, utilizou-se o subscrito *ant* para denotar o ponto anterior (correspondente ao índice $i - 1$); e o subscrito *ant2* para o ponto anterior ao anterior (correspondente ao índice $i - 2$).

$$y = \frac{12.588 T_s^2 u + y_{ant}(2 + 22.336 T_s) - y_{ant2}(1 + 5.584 T_s)}{1 + 16.752 T_s} \quad (10.3)$$

$$u = \frac{e(73.602 + 2 T_s) - 98.136 e_{ant} + 24.534 e_{ant2} + 4 u_{ant} - u_{ant2}}{3 + 2 T_s} \quad (10.4)$$

$$e = r - y \quad (10.5)$$

$$r = \frac{25 T_s^2 r_1 + r_{ant}(2 + 20 T_s) - r_{ant2}(1 + 5 T_s)}{1 + 15 T_s + 25 T_s^2} \quad (10.6)$$

O sistema foi simulado até atingir a referência desejada (60°) com tolerância de 1° . Além disso utilizou-se $T_s = 0.01$, correspondendo a um período de amostragem de 10 ms . Para efeito de comparação, simulou-se o sistema também utilizando a biblioteca *control* do *Python*. Esta fornece resultados iguais aos simulados com o Matlab para validação do projeto de controle. A evolução da referência de controle (r), do esforço de controle (u) e da saída da planta (y) estão expostas, respectivamente, nas Figuras 29, 30 e 31.

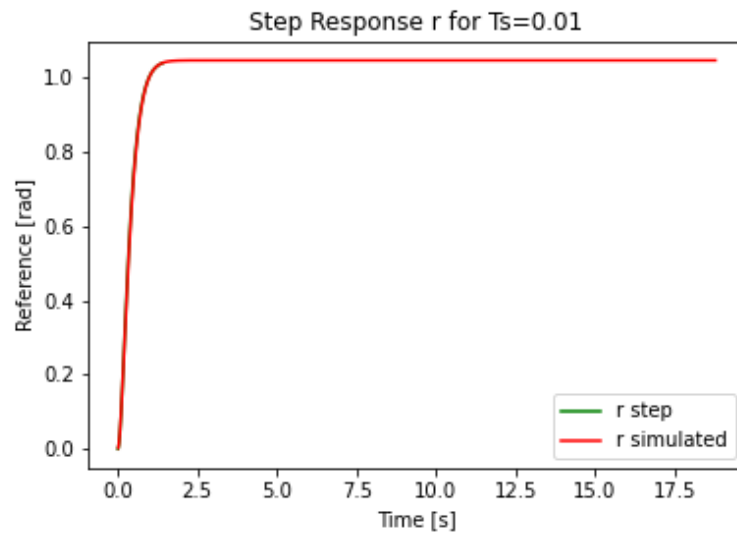


Figura 29: Simulação da evolução da referência de controle comparada ao valor esperado da evolução desta variável. Foi utilizada tolerância de 1° e referência de entrada de 60° .

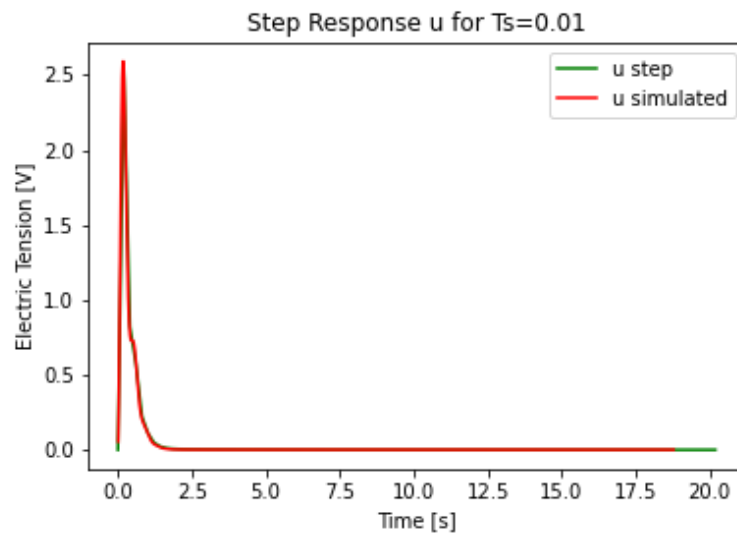


Figura 30: Simulação da evolução do esforço de controle comparada ao valor esperado da evolução desta variável. Foi utilizada tolerância de 1° e referência de entrada de 60° .

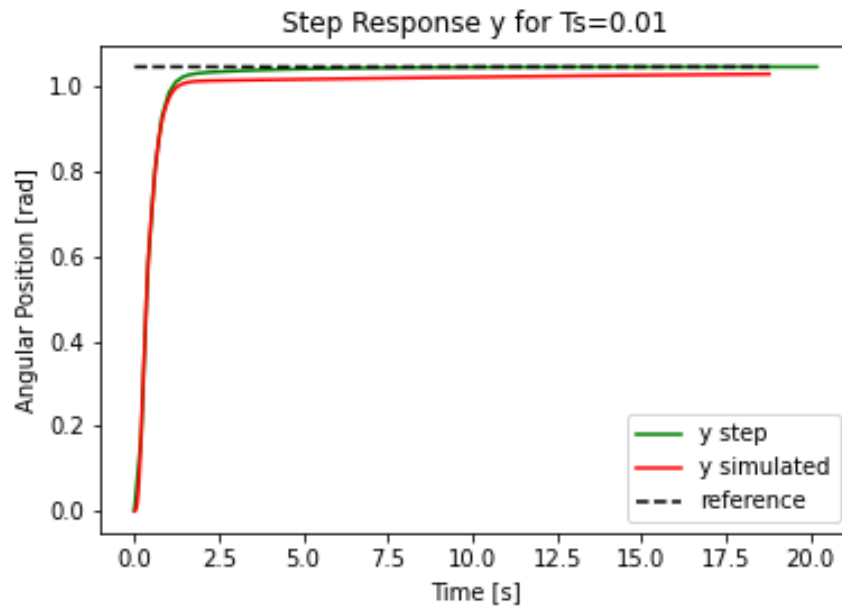


Figura 31: Simulação da evolução da saída da planta comparada ao valor esperado da evolução desta variável. Foi utilizada tolerância de 1° e referência de entrada de 60° .

Independentemente da forma de aproximação adotada para resolução aproximada das EDOs pelo MDF, surge um problema decorrente de um fenômeno conhecido como dissipação numérica (ou difusão numérica) [36,37]. Como estão sendo resolvidas de forma aproximada 3 EDOs em cima de um mesmo sinal (em sequência), o erro vai se acumulando. Dessa forma o método "perde energia" para a resolução numérica adequada. Isso explica o "erro estacionário" que aparece no resultado da simulação da saída (curva vermelha da Figura 31), comparado à simulação da resposta ao degrau esperada para o sistema, feita utilizando a biblioteca de controle do *Python* (curva verde da mesma figura).

Como o intuito da simulação é suprir a impossibilidade de realizar a implementação física do projeto devido à pandemia, os sinais que realmente devem seguir o valor esperado (sem considerável erro numérico) são a referência do controle e o esforço de controle. Isto porque estes seriam implementados no microcontrolador como foram implementados nas simulações. Partindo desta premissa, contornou-se o problema trazido pela dissipação numérica, que aparece de forma relevante apenas para y , aumentando um pouco a tolerância para encerrar o controle. Isto foi feito ao aumentar de 1° para 2° . Tal valor foi obtido através de um ajuste fino, a fim de obter um tempo de controle semelhante ao tempo de estabilização obtido na simulação de validação do controle (Seção 9.2). O resultado desse ajuste está colocado na Figura 32, no qual se observa um tempo de controle de aproximadamente 1.8 segundos (ligeiramente maior do que o tempo de estabilização

obtido na validação, de 1.525 segundos).

O principal método da classe *LLC*, que simula a malha de controle, é o *Do_LLC()*. Para mais detalhes nos métodos, veja o código com comentários no GitHub [32].

A análise do uso de diferentes métodos de aproximação por MDF, discutida em mais detalhe, está exposta no Apêndice E.

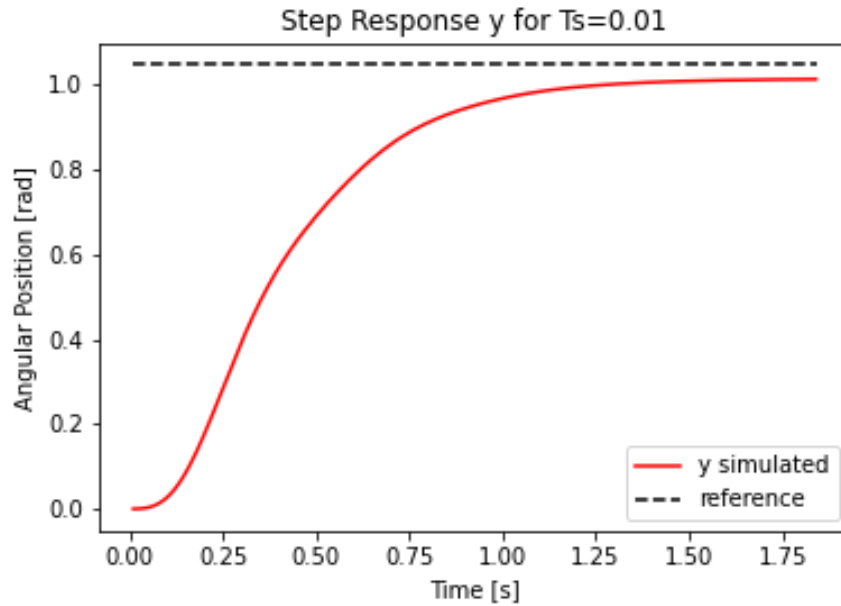


Figura 32: Simulação da evolução da saída da planta após ajuste da tolerância. Foi utilizada tolerância de 2° e referência de entrada de 60° .

10.2.2 Classe para Controle de Alto Nível - *HLC*

O propósito do controle de alto nível é comandar a atuação do sistema, baseado em uma lógica pré-definida. O sistema pode estar em apenas duas configurações: repouso ou atuado. Dessa forma é possível fazer a analogia dessas duas configurações com a flexão e extensão do cotovelo, uma vez que o exoesqueleto é acoplado ao antebraço. Portanto, o controle de alto nível foi projetado como uma máquina de estados, que possui apenas dois estados: flexão e extensão. Ao transicionar entre esses dois estados, o sistema simula a atuação sobre o exoesqueleto por meio do *LLC*.

Vale ressaltar que, por motivos de segurança, o sistema não pode transicionar de estado antes de completar a atuação anterior. Portanto, se o exoesqueleto está executando flexão e recebe o comando de extensão, este último comando não é executado e é perdido. O mesmo se aplica para o processo inverso. Além disso, não é possível realizar mais de uma

flexão seguida ou mais de uma extensão seguida. Isso levaria o sistema a configurações angulares indesejadas, que machucariam o usuário.

O diagrama simplificado dessa máquina de estados foi exposto anteriormente por meio da Figura 4, ilustrando a lógica de funcionamento do controle de alto nível.

O principal método da classe *HLC*, que executa a máquina de estados, é o *Do_HLC()*. Vale ressaltar que a classe possui como um dos atributos um objeto da classe *LLC*, a fim de simular o controle de baixo nível (como explicado na subseção 10.2.1). Dessa forma, a atuação do exoesqueleto ocorre utilizando a malha de controle projetada, tanto ao flexionar como ao estender.

Para mais detalhes nos métodos, veja o código com comentários no *GitHub* do projeto [32].

10.3 Simulação Integrada - Sistema Completo

O objetivo da simulação do sistema integrado e completo é mimetizar o que ocorreria na implementação física. Partindo desta visão, a implementação física teria dois processos ocorrendo de forma simultânea: o processamento dos sinais de EEG e o controle do exoesqueleto. Sendo assim, não seria possível simular este comportamento com chamadas de funções comuns, de forma sequencial. A solução encontrada foi simular o sistema de forma *multithread*. Para tanto, foi utilizada a biblioteca *threading* do *Python*. A partir desta, é possível definir funções para cada uma rodar em uma *thread* separada, ou seja, definem-se funções para rodar em paralelo.

Dessa forma, foram definidas 3 *threads* para a simulação e monitoramento da simulação do sistema integrado: uma para o pré-processamento, uma para o controle, e uma para gerar os gráficos dos resultados em "tempo real".

A função de pré-processamento importa os sinais de EEG e aplica uma das abordagens de pré-processamento nos sinais. Nesta, também é executada a classificação, que indica se deve ocorrer flexão ou não.

A função de controle recebe continuamente a classificação dos sinais tratados de EEG, e manda isso para o controle de alto nível. Este simula a atuação (ou não) sobre o exoesqueleto. Vale ressaltar que o modelo de classificação só é capaz de detectar sinais de flexão, não de extensão. Isso porque os dados utilizados no projeto também só apresentam marcação para flexão. Para contornar este problema, foi imposto que o sistema de controle

recebe um sinal de extensão 2 segundos após a flexão. Tal valor foi escolhido por estar próximo à média de valores observados nos dados dos experimentos.

Por fim, foi definida uma função para plotar informações para o acompanhamento da simulação em "tempo real". A simulação não é exatamente em tempo real, pois existe um pequeno tratamento temporal da informação para poder *plotá-lo* corretamente. Entretanto, os *plots* ocorrem de forma sincronizada e com intervalos de tempo adequados, de forma a dar a impressão que as informações estão sendo fornecidas em tempo real de processamento. O gráfico gerado apresenta informações do ângulo executado pelo exoesqueleto em comparação com o ângulo interno do cotovelo, medido experimentalmente e fornecido no conjunto de dados utilizado neste projeto.

PARTE IV

RESULTADOS E DISCUSSÃO

11 RESULTADOS

Os resultados finais, obtidos a partir do desenvolvimento do presente trabalho, serão expostos nas seções seguintes. Note que foi feita a separação entre a parte de pré-processamento e classificação (seção 11.1), a parte de controle (seção 11.2) e a parte dos sistemas funcionando de forma integrada (11.3), para maior clareza.

11.1 Resultados do Pré-Processamento

Os valores obtidos de AUC-ROC, sensibilidade e especificidade dos 3 algoritmos de pré-processamento e seus respectivos modelos de classificação estão descritos, respectivamente, nas Tabelas 4, 5 e 6. Há, nas tabelas referentes ao PCA e ao CCA, a dimensão final dos dados após a aplicação do algoritmo.

A média e desvio-padrão (indicado dentro dos parênteses) da métrica AUC-ROC dos modelos referentes ao PCA, ao ICA e ao CCA são, respectivamente, 0.803 (0.024), 0.927 (0.053) e 0.627 (0.029). A dimensão final dos dados após a aplicação do PCA ficou entre 12 a 25 dimensões, uma redução de 59.0% a 80.3%. Já a dimensão final dos dados após a aplicação do CCA variou entre 3 ou 4, representando uma redução de 93.4% ou 95.1%.

Tabela 4: Número de componentes principais encontrado, AUC-ROC, Sensibilidade e Especificidade para o algoritmo PCA.

ID	Nº componentes	AUC	Sensibilidade	Especificidade
Sujeito 1	14	0.77	0.652	0.744
Sujeito 2	14	0.79	0.632	0.789
Sujeito 3	15	0.81	0.645	0.804
Sujeito 4	22	0.81	0.790	0.671
Sujeito 5	20	0.85	0.819	0.734
Sujeito 6	21	0.80	0.733	0.744
Sujeito 7	18	0.79	0.698	0.735
Sujeito 8	22	0.83	0.767	0.738
Sujeito 9	21	0.77	0.747	0.652
Sujeito 10	12	0.78	0.661	0.771
Sujeito 11	13	0.78	0.882	0.515
Sujeito 12	14	0.80	0.710	0.740
Sujeito 13	25	0.83	0.725	0.790
Sujeito 14	16	0.83	0.701	0.763
Sujeito 15	19	0.81	0.799	0.693

Tabela 5: AUC-ROC, Sensibilidade e Especificidade para o algoritmo ICA.

ID	AUC	Sensibilidade	Especificidade
Sujeito 1	0.81	0.730	0.748
Sujeito 2	0.93	0.759	0.968
Sujeito 3	0.86	0.648	0.897
Sujeito 4	0.96	0.908	0.909
Sujeito 5	0.96	0.911	0.900
Sujeito 6	0.97	0.876	0.937
Sujeito 7	0.85	0.678	0.901
Sujeito 8	0.96	0.879	0.908
Sujeito 9	0.93	0.813	0.932
Sujeito 10	0.99	0.951	0.963
Sujeito 11	0.89	0.810	0.822
Sujeito 12	0.92	0.790	0.973
Sujeito 13	0.98	0.937	0.905
Sujeito 14	0.97	0.920	0.903
Sujeito 15	0.92	0.802	0.879

Tabela 6: Dimensão final encontrada, AUC-ROC, Sensibilidade e Especificidade para o algoritmo CCA. Os casos marcados com asterisco apresentaram descontinuidade no espaço dos dados.

ID	Nº componentes	AUC	Sensibilidade	Especificidade
Sujeito 1	3	0.62	0.414	0.763
Sujeito 2	4	0.67	0.549	0.698
Sujeito 3	4	0.62	0.484	0.710
Sujeito 4*	3	0.59	0.756	0.387
Sujeito 5	4	0.64	0.457	0.760
Sujeito 6	3	0.63	0.388	0.788
Sujeito 7	4	0.61	0.302	0.832
Sujeito 8	4	0.62	0.388	0.782
Sujeito 9	4	0.60	0.693	0.441
Sujeito 10	4	0.57	0.842	0.264
Sujeito 11	4	0.63	0.687	0.484
Sujeito 12*	3	0.67	0.589	0.681
Sujeito 13*	4	0.62	0.745	0.430
Sujeito 14	4	0.67	0.848	0.422
Sujeito 15	4	0.65	0.533	0.707

11.2 Resultados do Controle

A classe *HLC*, que implementa o controle de alto nível, comanda a atuação do exoesqueleto (simulada pela classe *LLC*) para atingir dois estados de posições pré definidas. No estado flexionado, o ângulo interno do cotovelo é de 60° , enquanto no estado estendido, este é de 120° . Para propósitos de controle, o ângulo é monitorado de forma local. Portanto, a flexão ocorre indo de 0 a 60° ; e a extensão de 0 a -60° . A evolução temporal destes movimentos pode ser observada nas Figuras 33 e 34, obtidas a partir das simulações com o *HLC*.

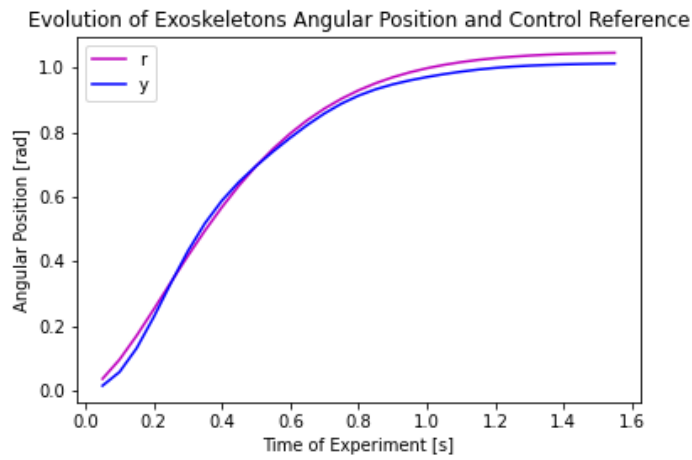


Figura 33: Evolução temporal da posição angular do exoesqueleto em movimento de flexão, comparado à referência de controle recebida.

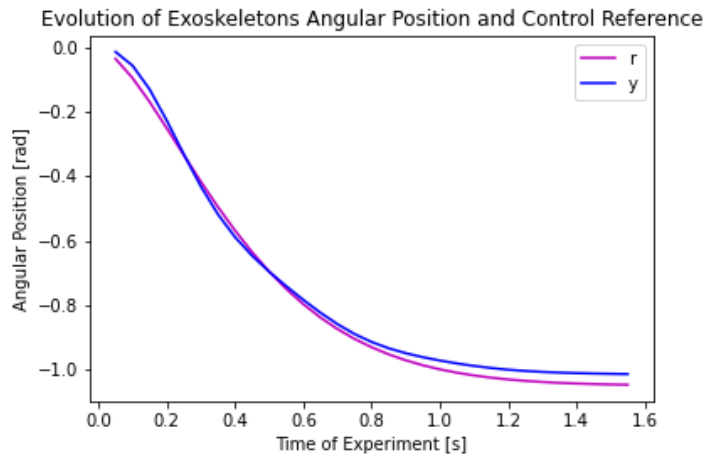


Figura 34: Evolução temporal da posição angular do exoesqueleto em movimento de extensão, comparado à referência de controle recebida.

O esforço de controle também é monitorado ao longo da atuação. Isto pode ser observado a partir das Figuras 35 e 36, para flexão e extensão respectivamente. Para finalidade de monitorar a evolução do controle, também é armazenado o erro ao longo da atuação. Este pode ser observado nas Figuras 37 e 38.

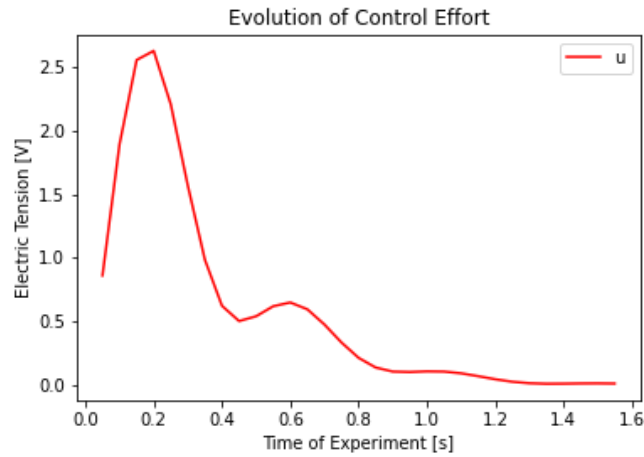


Figura 35: Evolução temporal do esforço de controle para o exoesqueleto em movimento de flexão.

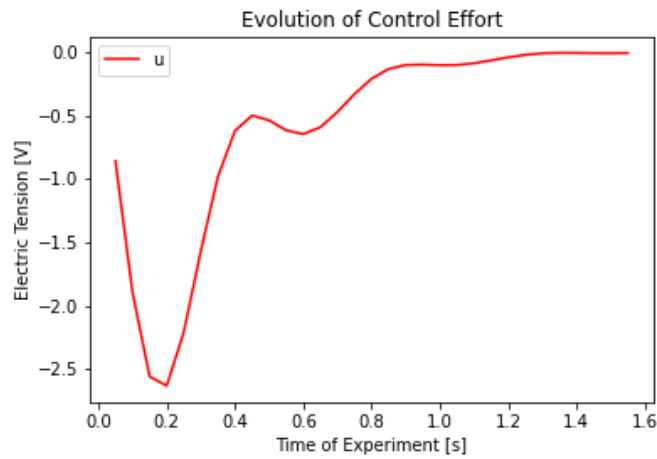


Figura 36: Evolução temporal do esforço de controle para o exoesqueleto em movimento de extensão.

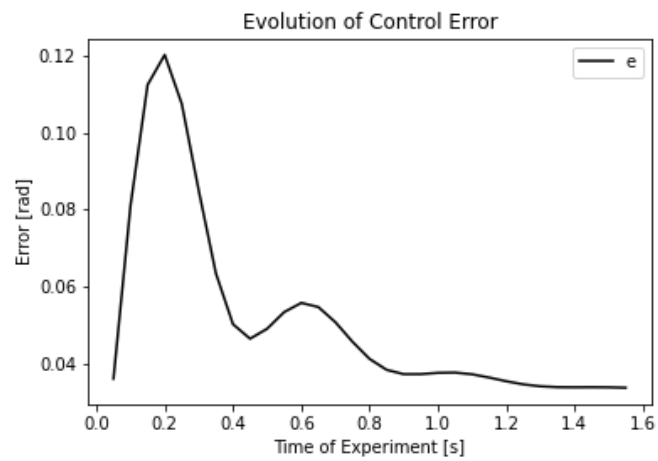


Figura 37: Evolução temporal do erro de controle para o exoesqueleto em movimento de flexão.

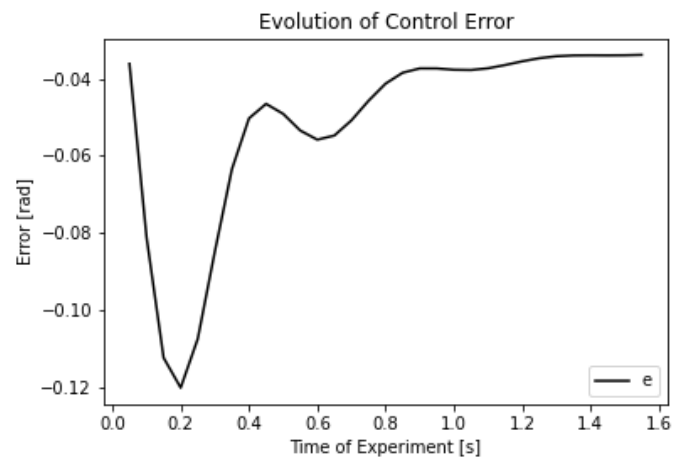


Figura 38: Evolução temporal do erro de controle para o exoesqueleto em movimento de extensão.

11.3 Resultados da Integração dos Sistemas

O funcionamento integrado das partes do sistema foi simulado em Python, como explicado na seção 10.3. A partir deste código, o resultado é plotado em uma janela de simulação. Esta é devidamente ajustada para dar a impressão de que os resultados estão sendo plotados em tempo real. Para conseguir avaliar o resultado obtido pelo sistema como um todo, é plotada a evolução do ângulo do exoesqueleto juntamente com os dados de ângulo interno do cotovelo, para um dos sujeitos do estudo. Um dos *frames* deste resultado dinâmico está apresentado na Figura 39, gerada a partir dos dados do sujeito 15 e utilizando ICA como pré-processamento. Nesta figura, é possível observar o movimento de flexão e extensão, tanto do exoesqueleto como do braço do sujeito no experimento. O tempo de pré-processamento dos sinais utilizando PCA e ICA ficaram na ordem de $2ms$, enquanto que o CCA apresentou um tempo de $4.8ms$. Logo, os 3 algoritmos conseguiram cumprir o requisito de tempo definido.

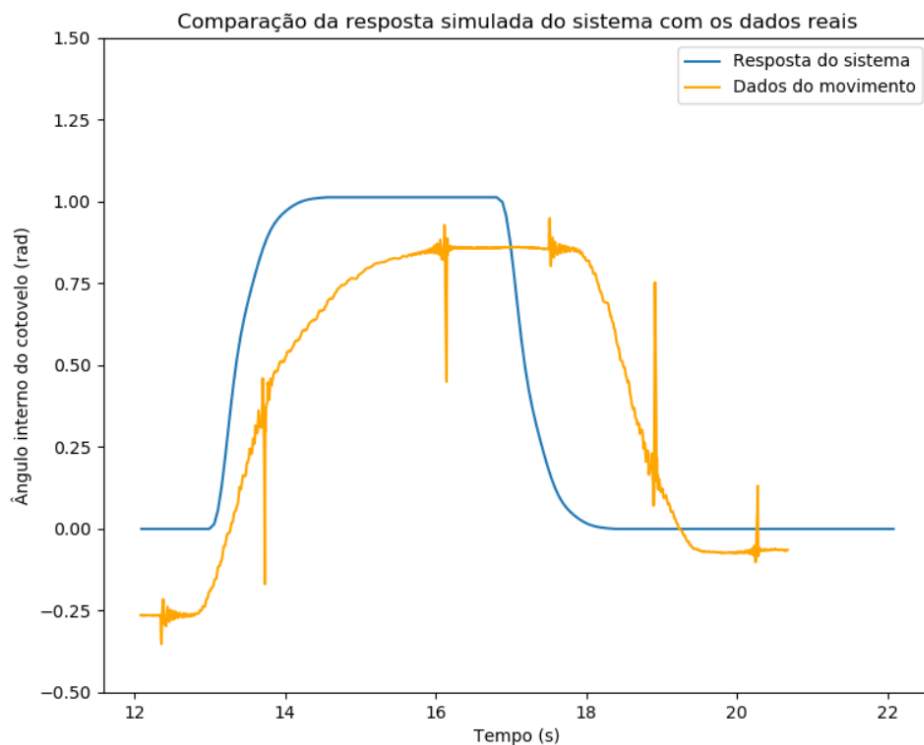


Figura 39: Comparação entre os resultados obtidos simulando o sistema como um todo e os dados experimentais de evolução do ângulo interno do cotovelo. Simulação e dados do sujeito 15, utilizando ICA como pré-processamento.

12 DISCUSSÃO

As discussões a respeito dos resultados obtidos com o presente trabalho serão expostas nas seções seguintes. Note que foi feita a separação entre a parte de pré-processamento e classificação (seção 12.1), a parte de controle (seção 12.2) e a parte dos sistemas funcionando de forma integrada (12.3), para maior clareza.

12.1 Discussão para o Pré-Processamento

A partir da análise das tabelas da Seção 11.1, é possível observar uma grande diferença nos valores de AUC-ROC entre os 3 pré-processamentos, além de um comportamento parecido entre os pacientes.

Os valores de AUC-ROC foram superiores com o pré-processamento do ICA em todos os experimentos. Esse resultado pode ser explicado dado que o objetivo do ICA é separar sinais que estão misturados, como dito anteriormente, e nesse problema há mistura dos sinais cerebrais no volume entre a superfície do couro cabeludo (onde se captura os sinais elétricos com os eletrodos) e o cérebro (onde são originados os sinais elétricos).

Os menores valores de AUC-ROC, por sua vez, foram encontrados no algoritmo CCA. Esse resultado pode ser explicado pela alta complexidade da redução dimensional, visto que sua magnitude foi superior a 90%. Essa extrema redução talvez necessite de um algoritmo de classificação com complexidade superior a de um SVM, como uma Rede Neural. Três sujeitos apresentaram conjunto de dados descontinuo que, embora não penalizem o algoritmo, torna o processo de estimar os parâmetros do CCA mais difícil.

O PCA, por sua vez, obteve um resultado intermediário, dado que ele realiza uma transformação linear de baixa complexidade.

Dado que somente um dos pacientes era canhoto, não há como verificar se a metodologia desenvolvida nesse trabalho seria invariante a essa característica. Essa questão fica aberta para pesquisas futuras.

12.2 Discussão para o Controle

Analisando todas as figuras apresentadas na Seção 11.2, de resultados de controle, percebe-se que os sinais de extensão são espelhados em relação ao eixo temporal, quando comparados com os mesmos sinais de flexão. Este é um resultado esperado, uma vez que o sistema é SLIT.

Tal fato reforça a validade da abordagem incremental adotada para o controle de baixo nível. Apesar do *encoder* presente na bancada ser absoluto, não seria possível mandar referências absolutas para o controle de baixo nível, da forma com a qual foi projetado e implementado. Isso porque a posição de repouso do sistema é considerada como o zero global. Dessa forma, ao solicitar a extensão do braço em coordenadas absolutas, seria enviada uma referência nula para o pré-filtro. Por ser nulo, este sinal não sofreria influência do pré-filtro, ocasionando a anulação do efeito dinâmico do mesmo. Portanto, a resposta de extensão do exoesqueleto seria influenciada apenas pelo controlador, gerando um comportamento dinâmico indesejado.

Com isso em mente, para a operação na bancada de testes, seria necessário acrescentar uma pequena lógica no controle de baixo nível, para converter as coordenadas absolutas lidas pelo *encoder* em coordenadas locais/incrementais. Outra possibilidade seria operar o sistema fora da condição de ângulo absoluto zero. Com isso poderiam ser utilizadas coordenadas globais no controle. Entretanto, esta abordagem não é muito segura, podendo acarretar em graves problemas.

Outro ponto interessante a ser ressaltado é que o efeito da dissipação numérica ainda aparece na simulação da saída do modelo; mesmo ajustada com uma tolerância maior. Nas Figuras 33 e 34 fica evidente o distanciamento da saída da planta (curva azul) em relação à referência de controle (curva magenta) entre 0.7 e 1.6 segundos. Este fenômeno já foi mencionado na subseção 10.2.1 e discutido com mais detalhe no Apêndice E, mas vale ressaltar a sua existência no resultado da simulação do controle. Porém, este não seria um problema para a implementação física, uma vez que a dissipação numérica só aparece de forma significativa para o sinal de saída, mas não para o esforço de controle (como também foi apresentado na subseção 10.2.1).

Por fim, observando as curvas de evolução do erro de controle (Figuras 37 e 38), é perceptível a diminuição do mesmo ao longo da ação de controle. Isso demonstra que o sistema de controle está sendo capaz de atingir a referência desejada. O erro só não se torna nulo na simulação devido à questão da dissipação numérica na mesma.

12.3 Discussão para a Integração dos Sistemas

Avaliando a Figura 39, que mostra o resultado da simulação do movimento comparado com os dados do movimento real, é percebido um comportamento qualitativamente satisfatório. As duas curvas, além de iniciarem em um tempo próximo, apresentam um formato semelhante, o que mostra que o controle projetado tem sucesso em reproduzir o movimento real desejado para o braço humano.

Olhando com maior detalhe o início da subida das duas curvas (próximo dos 13 segundos de simulação), percebe-se que o movimento real inicia ligeiramente antes do que o movimento simulado (aproximadamente 0.1 segundos antes). Tal resultado é esperado, uma vez que o sistema de pré-processamento e classificação identifica os sinais cerebrais de iminência de movimento, com um intervalo entre 30ms a 130ms antes da ação. Apenas neste momento o controle é acionado. Talvez um sistema que identificasse a intenção de movimento eliminasse essa pequena latência. Entretanto, esta é tão pequena que não prejudica a presente abordagem de ser implementada com sucesso na prática.

Uma grande diferença entre os sinais está no movimento de extensão (descida das curvas). A curva simulada (azul) desce bem antes da curva real (laranja). Entretanto, tal comportamento ocorre pois o classificador não identifica o movimento de retorno (extensão), pois estes dados não estavam disponíveis no *dataset* utilizado para o projeto. Essa consideração já foi feita na seção 10.3, sendo seu efeito perceptível no resultado final do projeto.

Por fim, é evidente que os dados experimentais não estabilizam exatamente nos ângulos esperados (0 ou 60°). Tal comportamento se justifica pelo fato de o experimento ser realizado com a posição livre do braço, ou seja, não tem nenhum tipo de delimitação física para indicar a posição inicial e final para o indivíduo. O sujeito, portanto, movimenta o braço atingindo as referências de forma aproximada. O mesmo não ocorre com o exoesqueleto, que tem bem definidos os ângulos de cada estado (flexão e extensão). Esta acurácia atingida com o uso do exoesqueleto pode ser uma vantagem a ser explorada em diversas aplicações, dentre elas a realização de exercícios de fisioterapia.

PARTE V

CONCLUSÕES E CONSIDERAÇÕES FINAIS

13 CONCLUSÕES

A escolha do algoritmo de pré-processamento é crucial quando trabalhando com sinais de EEG. Nesse trabalho, foi possível observar como as características distintas das três técnicas de *Feature Transform* resultaram em modelos com AUC-ROC bem diferentes. A abordagem não-linear utilizando o CCA demonstrou um alto poder de redução dimensional, entretanto seu respectivo modelo não conseguiu detectar os padrões necessários para a classificação correta da intenção de movimento. Por outro lado, os algoritmos lineares conseguiram realizar tal tarefa com excelência, com o ICA alcançando um valor de AUC-ROC médio acima de 0.90.

O controlador robusto de posição angular é capaz de realizar com sucesso o movimento desejado (flexão e extensão). Além disso, os resultados obtidos se assemelham qualitativamente aos valores observados experimentalmente para os mesmos movimentos. Vale ressaltar que o retorno do exoesqueleto para a posição inicial, em movimento de extensão, não é recebido do classificador. Para isso ser possível, será necessário coletar dados de EEG que identifiquem o movimento de extensão. Com isso, os modelos poderão ser retreinados a fim de identificar este tipo de movimento.

Em termos de simulação, o fenômeno de dissipação numérica prejudica um pouco o resultado final, o que pôde ser contornado aumentando um pouco a tolerância para a parada do controle (a 2 graus da referência). Já para a implementação prática, este fenômeno não será relevante e não prejudicará o funcionamento real do sistema.

Tirando as considerações feitas ao longo do trabalho, e que serão reforçadas no capítulo 14, a implementação física vai exigir, basicamente, pequenas adaptações nos códigos. Isso porque os códigos deste projeto foram estruturados de maneira a facilitar a posterior implementação física do mesmo.

14 CONSIDERAÇÕES FINAIS

Uma continuação óbvia deste trabalho é a sua implementação física na bancada de testes do laboratório. Pela forma com a qual foram construídos os códigos de simulação do projeto, não deve ser difícil sua adaptação para o projeto físico. Um ponto importante para esta adaptação é a necessidade de converter o ângulo lido do *encoder* absoluto para uma medida local/incremental (como explicado na seção 11.2). Tal alteração deve ser feita dentro da parte de aquisição do sinal no controle de baixo nível. A implementação física dos algoritmos de pré-processamento e classificação, entretanto, será uma tarefa de baixa complexidade, pois só é necessário mover os códigos para um sistema embarcado.

Outro aspecto para evoluir o trabalho é passar a identificar mais ângulos de movimento, não só 0 e 60° como está no momento. Isso exigiria um esforço maior na parte de pré-processamento e classificação. Dentro desta ideia, fazer o sistema identificar movimentos de retorno (extensão) do braço também é um próximo passo. Para ambos os caminhos aqui apontados, será necessária a aquisição de novos dados de EEG, identificando tais tipos de movimento.

Para melhorar o desempenho do pré-processamento do CCA, seria necessário utilizar um algoritmo de classificação de maior complexidade, como dito anteriormente na Seção 12.1, como Redes Neurais, Redes Convolucionais ou Redes Recorrentes. Além disso, combinar com outros tipos de pré-processamento, como *Feature Extract* e *Feature Selection*, pode melhorar também o desempenho sem precisar alterar o classificador.

REFERÊNCIAS

- [1] *Cartilha do Censo 2010 - Pessoas com Deficiência*, 2010.
- [2] O. Zander and C. Kothe, “Towards passive brain–computer interfaces: applying brain–computer interface technology to human–machine systems in general,” *Journal of Neural Engineering*, vol. 8, 2011.
- [3] R. Jonathan and W. Elizabeth, *Brain-Computer Interfaces: Principles and Practice*. Oxford, 2011.
- [4] F. Lotte, L. Bougrain, A. Cichocki, M. Clerc, M. Congedo, A. Rakotomamonjy, and F. Yger, “A review of classification algorithms for eeg based brain–computer interfaces: a 10 year update,” *Journal of Neural Engineering*, vol. 15, 2018.
- [5] P. García-Laencina, G. Rodríguez-Bermudez, and J. Roca-Dorda, “Exploring dimensionality reduction of eeg features in motor imagery task classification,” *Expert Systems with Applications*, vol. 41, pp. 5285–5295, 2014.
- [6] K. Pearson, “Liii. on lines and planes of closest fit to systems of points in space,” *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 6, pp. 559–572, 1901.
- [7] C. Jutten and J. Herault, “Blind separation of sources, part i: An adaptive algorithm based on neuromimetic architecture,” *Signal Processing*, vol. 24, pp. 1–10.
- [8] Y. Wu, T. H. Huang, C. Y. Lin, S. J. Tsai, and P. Wang, “Classification of eeg motor imagery using support vector machine and convolutional neural network,” *2018 International Automatic Control Conference (CACs)*, 2018.
- [9] H. S. Lo and S. Q. Xie, “Exoskeleton robots for upper-limb rehabilitation: State of the art and future prospects,” *Medical engineering & physics*, vol. 34, no. 3, pp. 261–268, 2012.
- [10] J. L. Pons, “Rehabilitation exoskeletal robotics,” *IEEE Engineering in Medicine and Biology Magazine*, vol. 29, no. 3, pp. 57–63, 2010.
- [11] C. Yang, J. Zhang, Y. Chen, Y. Dong, and Y. Zhang, “A review of exoskeleton-type systems and their key technologies,” *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, vol. 222, no. 8, pp. 1599–1612, 2008.
- [12] K. Anam and A. A. Al-Jumaily, “Active exoskeleton control systems: State of the art,” *Procedia Engineering*, vol. 41, pp. 988–994, 2012.
- [13] J. Rosen, M. B. Fuchs, and M. Arcan, “Performances of hill-type and neural network muscle models—toward a myosignal-based exoskeleton,” *Computers and Biomedical Research*, vol. 32, no. 5, pp. 415–439, 1999.

- [14] M. T. Mason, “Compliance and force control for computer controlled manipulators,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 11, no. 6, pp. 418–432, 1981.
- [15] G. P. Furtado, “Formulation of impedance control strategy as an optimal control problem,” Master’s thesis, Universidade de São Paulo, São Paulo, 2018.
- [16] M. A. Gomes, G. L. M. Silveira, and A. A. Siqueira, “Gait pattern adaptation for an active lower-limb orthosis based on neural networks,” *Advanced Robotics*, vol. 25, no. 15, pp. 1903–1925, 2011.
- [17] J. A. Lee and M. Verleyson, *Nonlinear dimensionality reduction*. Springer, 2007.
- [18] A. Forner-Cordero, V. Quadrado, S. Tsagbey, and B. Smits-Engelsman, “Improved learning a coincident timing task with a predictable resisting force,” *Motor Control*, vol. 22, pp. 117–133, 2018.
- [19] R. S. Souza, “Design and prototyping of a development platform for exoskeleton research,” Master’s thesis, Universidade de São Paulo, São Paulo, 2018.
- [20] D. Nguyen, L. E. Brown, J. W. Coburn, D. A. Judelson, A. D. Eurich, A. V. Khamoui, and B. P. Uribe, “Effect of delayed-onset muscle soreness on elbow flexion strength and rate of velocity development,” *The Journal of Strength & Conditioning Research*, vol. 23, no. 4, pp. 1282–1286, 2009.
- [21] R. C. Luo, B.-H. Shih, and T.-W. Lin, “Real time human motion imitation of anthropomorphic dual arm robot based on cartesian impedance control,” pp. 25–30, 2013.
- [22] P. Ofner, A. Schwarz, J. Pereira, and G. Müller-Putz, “Upper limb movements can be decoded from the time-domain of low-frequency eeg,” *PloS one*, vol. 12(8): e0182578, 2017.
- [23] T. Kohonen, “Self-organized formation of topologically correct feature maps,” *Synergetics of the Brain*, vol. 23, pp. 264–273, 1982.
- [24] P. Demartines and J. Herault, “Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets,” *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 148–154, 1997.
- [25] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, pp. 273–297, 1995.
- [26] G. van Rossum, “Python tutorial,” Tech. Rep. CS-R9526, Centrum voor Wiskunde en Informatica (CWI), Amsterdam, May 1995.
- [27] J. Cho, A. Paiva, S.-P. Kim, J. Sanchez, and J. Príncipe, “Self-organizing maps with dynamic learning for signal reconstruction,” *Neural Networks*, pp. 274–284, 2007.
- [28] S. Haykin, *Redes Neurais: Princípios e prática*. Bookman, 2001.
- [29] O. Oguntoyinbo *et al.*, “Pid control of brushless dc motor and robot trajectory planning simulation with matlab®/simulink®,” 2009.

- [30] J. J. da Cruz, “Notas de aula de introdução ao projeto de sistemas de controle robustos,” tech. rep., Escola Politécnica da Universidade de São Paulo, São Paulo, BR, 2020.
- [31] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [32] F. A. de Moraes Machado and A. E. M. Colognesi, “Repositório git: Uso de pré-processamento não-linear de sinais de eeg para controle de exoesqueleto de membro superior.” disponível em https://github.com/FelipeAugustoMachado/TCC_Mortal. Publicado em 2020.
- [33] J. Herault, C. Jausions-Picaud, and A. Guérin-Dugué, “Curvilinear component analysis for high-dimensional data representation: I. theoretical aspects and practical use in the presence of noise,” pp. 625–634, 06 1999.
- [34] P. Demartines, “Organization measures and representations of the kohonen maps,” 1992.
- [35] P. A. F. Enabe, “Teoria da elasticidade linear: Torção uniforme.” available at https://www.researchgate.net/publication/344483432_Teoria_da_Elasticidade_Linear_Torcao_Uniforme. Acessado em: 04-10-2020.
- [36] L. N. Trefethen, “Finite difference and spectral methods for ordinary and partial differential equations.” unpublished text, 1996, available at <http://people.maths.ox.ac.uk/trefethen/pdetext.html>. Acessado em: 04-10-2020.
- [37] D. W. Zingg, “Lecture 7 - numerical dissipation.” NASA lecture, available at https://www.nas.nasa.gov/assets/pdf/ams/2018/introtocfd/Intro2CFD_Lecture7_Zingg.pdf. Acessado em: 04-10-2020.

APÊNDICE A – REVISÃO BIBLIOGRÁFICA

A fim de obter uma visão geral do uso de exoesqueletos para reabilitação, pesquisou-se na Plataforma Scopus (no início do mês de março) pelas palavras chave 'exoskeleton AND rehabilitation AND "state of the art"', o que trouxe 87 resultados. Filtrando estes para apenas aqueles da área 'Engineering' e documento tipo 'Review', obteve-se 12 resultados. Destes, selecionou-se 1 trabalho, que trouxe mais 1. Os resultados indicam que o uso de exoesqueletos robóticos para recuperação de movimento é algo bastante estudado nas últimas duas décadas e que apresenta resultados extremamente satisfatórios [9, 10].

Para entender melhor o sinal de eletroencefalografia (EEG) e sistemas *brain-computer interface* (BCI), a busca inicial na plataforma Scopus foi 'EEG AND BCI'. O resultado foi de 6926 resultados, e desses foram selecionados apenas as Revisões, totalizando 107 trabalhos. Filtrando pelo conteúdo, foram selecionadas 2 revisões. A seguir, foi feita uma outra busca para ver como é abordado esse tema usando Machine Learning para classificação: 'EEG AND BCI AND "Classification"'. Houve 3500 resultados, sendo 153 Revisões. Destes, foi selecionado apenas um trabalho. A terceira pesquisa foi para conhecer quais técnicas são utilizadas para Redução Dimensional. A pesquisa foi 'EEG AND BCI AND "Dimension Reduction"', obtendo 24 documentos, onde 3 foram selecionados. A última busca nesse mesmo tema foi 'EEG AND BCI AND "Dimensionality Reduction"', com 62 resultados, foram selecionados 4 trabalhos. Foi encontrado que o sinal coletado do EEG possui alguns problemas a serem tratados, sendo o maior deles a qualidade do sinal. Por possuir um volume entre o cérebro e o sensor, constituído de líquido cefalorraquidiano, crânio e o couro cabeludo [3], os sinais criados por diferentes regiões do cérebro acabam se misturando. Essa característica torna o processamento do sinal uma tarefa extremamente difícil e com resultados insatisfatórios para sistemas de *Brain-Computer Interface* (BCI) [3].

Uma forma de contornar esse problema é a utilização de Inteligência Artificial (IA). Existem diversas abordagens para pré-processamento [4], sendo que elas são divididas em

3: Feature Selection (FS), Feature Extraction (FE) e Feature Transform (FT). Embora FE seja a abordagem mais utilizada nesse tipo de sinal, FT é capaz de alcançar melhores resultados [5], sendo os algoritmos de FT mais utilizados o *Principal Components Analysis* (PCA) [6], o Local Fisher Discriminant Analysis (LFDA) e o *Independent Component Analysis* (ICA). Nenhum dos algoritmos a seguir foram encontrados na pesquisa: Curvilinear Component Analysis, Non-Linear Mapping, Geodesic Non-Linear Mapping e Curvilinear Distance Analysis.

Após o pré-processamento, é utilizado algum modelo de classificação ou de regressão para converter o sinal de EEG processado em uma variável discreta/contínua que alimentará o controlador. Dos diversos modelos diferentes que tentam executar essa tarefa [4], aqueles que apresentam melhores resultados são os não lineares [8], embora que esses modelos não possuam interpretabilidade como os modelos lineares apresentam.

Após definir uma rotina de pré-processamento e escolher o modelo a ser utilizado, é necessário verificar se essa escolha atende aos requisitos. Para isso, é feito um teste do modelo de forma *offline*, i.e., utilizando um conjunto de dados adquiridos anteriormente. Se as métricas do modelo cumprem os requisitos, o modelo é submetido ao teste *online*, i.e., processando os dados e tomando decisões em tempo real, enviando sua saída para o controlador do exoesqueleto.

Para a revisão bibliográfica da parte de controle pesquisou-se, inicialmente, por 'control AND exoskeleton AND "state of the art"', o que deu 71 resultados dos quais escolheu-se 1, que trouxe mais 3 referências. A partir da leitura de alguns trabalhos do próprio Laboratório de Biomecatrônica do Departamento de Engenharia Mecatrônica da Poli, obteve-se 1 tese sobre controle de impedância que se mostrou relevante para a revisão bibliográfica do presente trabalho. Essa busca revelou que atualmente existem diversas configurações mecânicas de exoesqueletos, sendo adaptadas para as mais variadas aplicações [9–12]. Revisões na literatura abordando este tema são facilmente encontradas. Entretanto, revisões sobre os sistemas de controle aplicados a esses exoesqueletos não são tão comuns [12].

Os sistemas de controle comumente utilizados baseiam-se em quatro grandes aspectos: modelo; hierarquia; parâmetros físicos e uso. A construção do modelo é dada pela composição de duas partes: o modelo dinâmico do exoesqueleto e o modelo dos músculos do operador [9]. A obtenção deste primeiro modelo dinâmico pode ser feita de forma analítica, por modelagem matemática; por meio de técnicas de identificação de sistemas; ou mesmo utilizando Inteligência Artificial, sendo esta uma técnica mais moderna. Por

outro lado, o intuito de modelar também o músculo é prever o movimento realizado pelo mesmo dentro da dinâmica do sistema como um todo [13].

O controle costuma ser estruturado em 3 níveis de hierarquia, sendo que estes são projetados de acordo com o uso desejado e os parâmetros físicos que serão controlados [9]. Os parâmetros mais comuns nestas aplicações são a posição, a velocidade e a força/torque. Este último é extremamente relevante para adequar a operação do exoesqueleto sem haver nenhum dano sobre o operador. Uma técnica interessante que realiza o controle de força/torque juntamente com o controle de posição é o controle de impedância [14]. A partir desta, é possível controlar a interação entre o usuário e o exoesqueleto de forma direta. Tal técnica é bastante avançada e possibilita a obtenção de resultados muito satisfatórios para aplicações reais [15].

O enfoque do presente trabalho está na implementação de um controle de posição orientado por sinais de referência de EEG. Sendo assim, técnicas de controle tão avançadas não se mostram necessárias em uma primeira abordagem. O projeto de um controlador robusto para controle de posição e, eventualmente também da velocidade, já é suficiente para demonstrar a aplicação desejada. Um exemplo de uso deste tipo de controlador aplicado ao controle de exoesqueleto foi demonstrado por [16].

APÊNDICE B – SIMULAÇÕES DO ALGORITMO SOM

Neste apêndice serão expostas simulações realizadas com o algoritmo SOM e seu variante SOM-DL, comparando os algoritmos. O objetivo é decidir qual algoritmo utilizar para que se obtenha uma boa representação do espaço de dados. Para isso, serão feitas análises do MSE (*Mean Squared Error*) em função do número de neurônios. Todas as simulações possuirão o número de iterações definidos em [23]: 1000 iterações para organização do mapa e 500 iterações por neurônio, para alcançar a especialização.

B.1 Simulações de MSE

Nessas simulações, foram analisadas como se variava o MSE conforme aumenta o mapa. Foram analisados 5 mapas de tamanhos diferentes: 15x15, 20x20, 25x25, 30x30 e 35x35, e o MSE foi calculado utilizando 5000 pontos. Os resultados estão indicados na Figura 40.

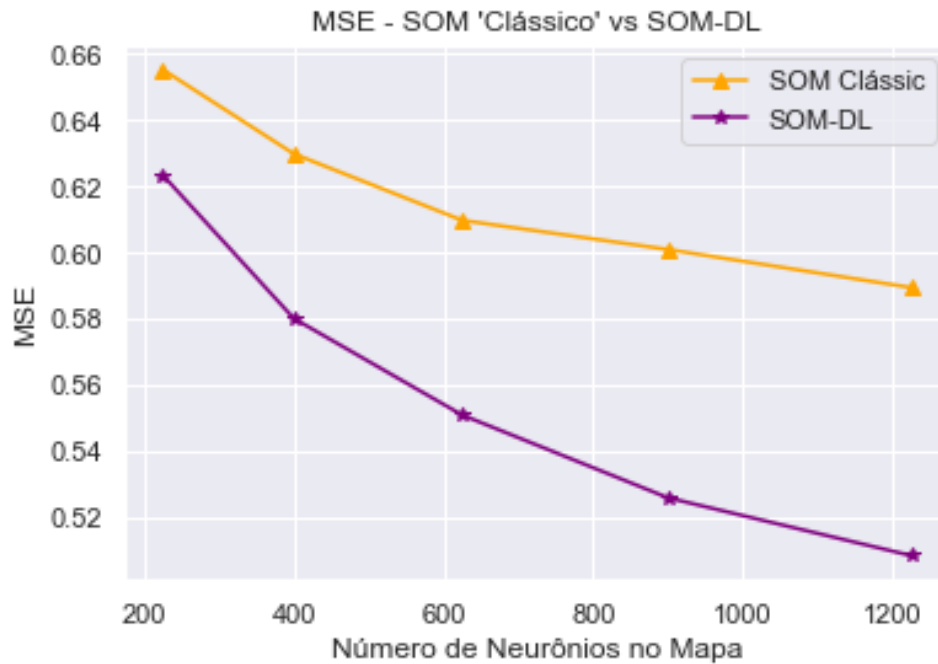


Figura 40: Comparação do MSE entre o SOM clássico e seu variante, SOM-DL.

Observa-se que o variante SOM-DL é capaz de alcançar um MSE menor, comparado ao SOM clássico. Pode-se ver esse comportamento nas Figuras 41 e 42. Esse resultado faz com que a utilização do variante ser mais adequado à problemática abordada.

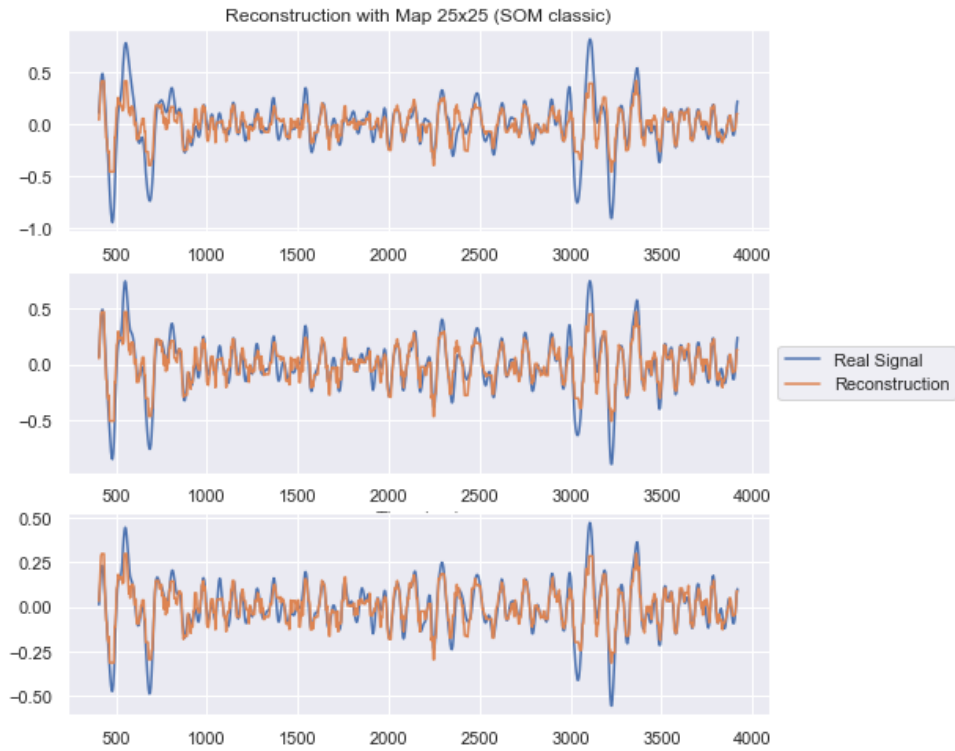


Figura 41: Sinal Original e Sinal reconstruído com SOM clássico.

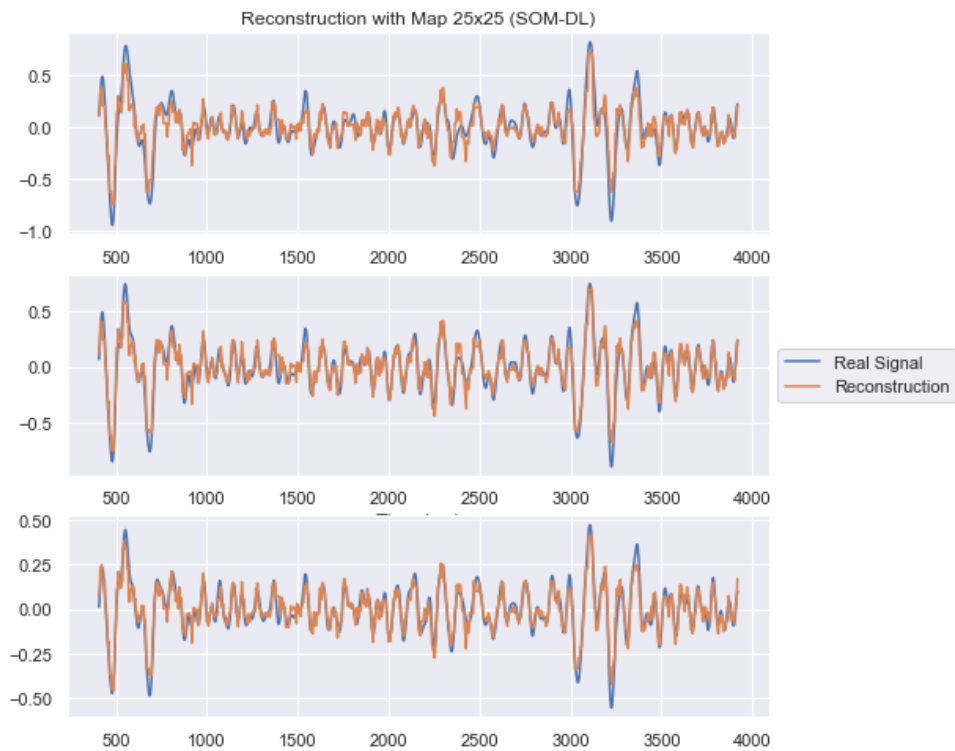


Figura 42: Sinal Original e Sinal reconstruído com SOM-DL.

APÊNDICE C – SIMULAÇÕES DOS MODELOS 'REAIS'

Neste apêndice serão expostas as simulações realizadas para verificar o efeito das dinâmicas não modeladas sobre o modelo nominal original. O objetivo é observar se o efeito das dinâmicas não modeladas consideradas aparecem em frequências que coincidam com as frequências de funcionamento do sistema na prática. Para tanto, serão consideradas apenas as incertezas não estruturadas nas simulações (ou seja, não serão levadas em conta incertezas paramétricas).

C.1 Simulação para o Braço

Com base na modelagem realizada na sub-subseção 7.3.2.1, foram feitas simulações para avaliar a resposta em frequência do sistema, considerando o modelo do exoesqueleto adicionando a dinâmica não modelada de deformação do mesmo.

Para efeito de simulação, considerou-se $V(s) = 1$ e com isso, a função de transferência para o modelo do braço deformável (antes dada por eq. (7.30)) será dada pela eq. (C.1).

$$G_{R,braco,dinamicas}(s) = \theta_n(s) \quad (C.1)$$

O valor da inércia do exoesqueleto em relação ao eixo de rotação do motor, foi obtido através do modelo CAD do exoesqueleto (juntamente com a aplicação do Teorema dos Eixos Paralelos). O desenho mecânico deste está colocado no Anexo A. Como a simulação foi realizada com um elevado número de discretização, aproximou-se o coeficiente de elasticidade Δk das molas ao valor do módulo de elasticidade do material do exoesqueleto (alumínio 6061). Este valor também foi obtido através do CAD.

Note que a constante elétrica k_e é obtida através da eq. (7.20). Além disso, assumiu-se o motor atuando com eficiência máxima.

O código que realiza a simulação do sistema e obtém sua resposta em frequência está disponível no GitHub [32]. Vale ressaltar que o erro multiplicativo l_m é dado pela eq. (C.2).

$$|\Delta m(j\omega)| = \left| \frac{G_R(j\omega) - G(j\omega)}{G(j\omega)} \right| \leq l_m(\omega) \quad (C.2)$$

Na Figura 43 está colocado o resultado da simulação. Note que está sendo ilustrado o erro multiplicativo em dB , para a evolução da frequência angular ω em escala logarítmica.

Analisando a Figura 43 e as variáveis internas do algoritmo [32], percebe-se que o máximo erro multiplicativo atingido representa um erro relativo de 0.11%. Este pico ocorre próximo à frequência angular de 1000 rad/s. Como a faixa de operação imposta pelos requisitos do projeto vai até a frequência angular de aproximadamente 4.2 rad/s, fica evidente que a dinâmica de deformação não terá efeito algum para a operação real da planta. Além disso, vale observar que mesmo na frequência onde ocorre o pico do erro, este é extremamente pequeno. Isso é fisicamente justificável, uma vez que o exoesqueleto tem comprimento de 15cm, o que pode ser considerado com um comprimento pequeno.

Chega-se à conclusão de que a incerteza não estruturada proveniente da deformação do exoesqueleto é desprezível. Portanto, não será considerada no projeto.

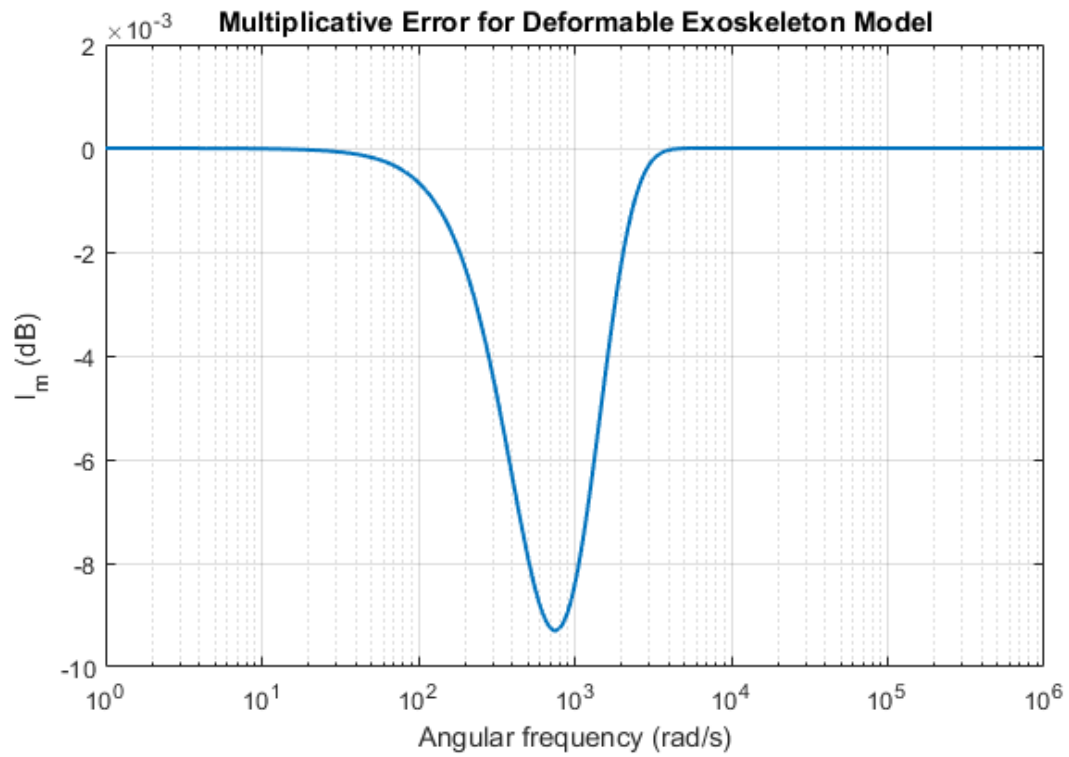


Figura 43: Erro multiplicativo para o modelo dinâmico do exoesqueleto que considera deformações, para uma discretização de $n = 1000$ elementos.

C.2 Simulação para o Motor

A análise feita para a inclusão de dinâmica não modelada para o motor segue o mesmo raciocínio desenvolvido no Apêndice C.2. O algoritmo também está disponível no GitHub [32] e calcula o erro multiplicativo considerando a adição do efeito da indutância no motor BLDC. Note que o erro multiplicativo é dado pela eq. (C.2).

O resultado da simulação está colocado na Figura 44. A partir desta, percebe-se que para uma frequência de 10 rad/s, o erro multiplicativo já é considerável. Entretanto, é necessário interpretar o significado deste resultado antes de concluir se a dinâmica da indutância é desprezível para o projeto ou não. Esta frequência deve ser comparada à frequência de entrada sobre a qual o motor estará sujeito. Ou seja, isso poderia ser enxergado como a frequência da corrente que vai entrar no motor. A partir desta visão, percebe-se que o motor, por ser um motor de corrente contínua, não irá sofrer com variação significativa na entrada. Ou seja, a frequência sobre a qual o motor estará sujeito será extremamente pequena (bem menor do que 1 rad/s). A maior variação de entrada no motor se dará durante a atuação de controle, mas esta também terá uma frequência extremamente baixa, uma vez que o movimento desejado deve ser suave e sem trancos.

Dessa forma, pode-se concluir que o motor estará sujeito a frequências muito menores do que 1 rad/s, e nesta faixa o erro multiplicativo é desprezível (vide Figura 44). Portanto, a incerteza não estruturada proveniente do efeito dinâmico da indutância do motor BLDC será desprezada no projeto.

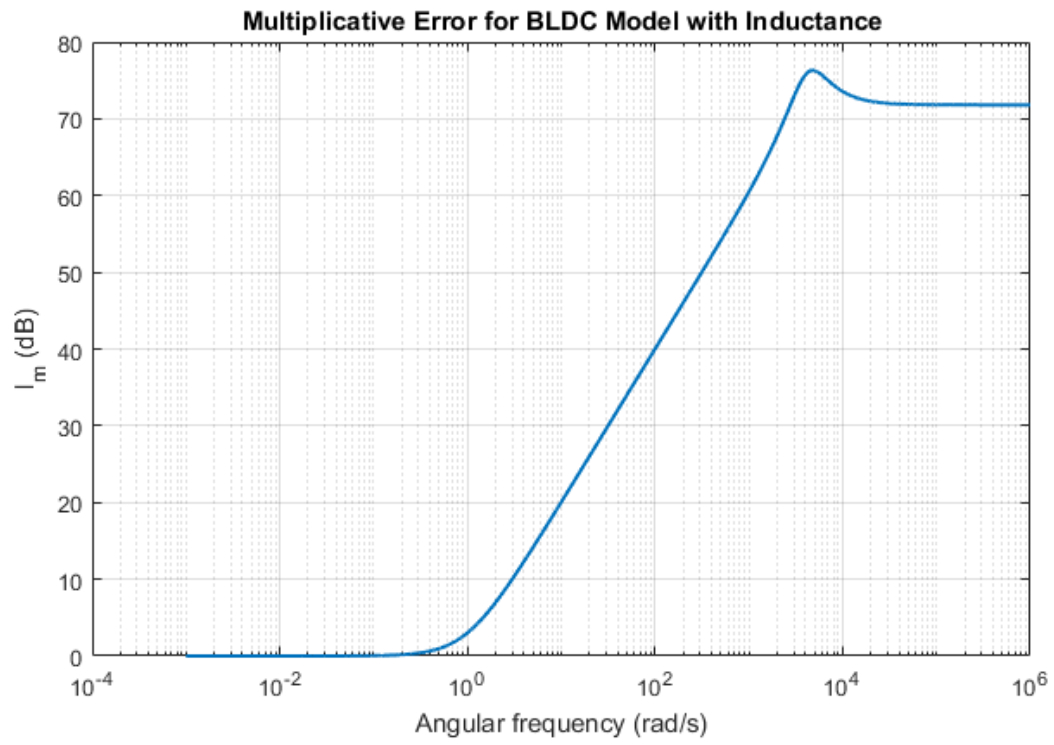


Figura 44: Erro multiplicativo para o modelo dinâmico do motor BLDC que considera o efeito da indutância.

APÊNDICE D – VALIDAÇÕES DO SISTEMA DE CONTROLE

O algoritmo está disponível no GitHub [32] e implementa a validação do sistema de controle tanto em frequência como no tempo.

Essencialmente, o código é dividido em 3 partes. A primeira parte para a definição de parâmetros, funções de transferência da malha de controle e variáveis para analisar os resultados das simulações. Na segunda parte realiza-se uma varredura em frequência para avaliar a resposta do sistema quando excitado em várias frequências (dentro da faixa de operação do sistema). Por fim, na terceira parte, plota-se as análises finais para a validação do sistema de controle.

Existem alguns pontos do código que valem ser ressaltados (apesar de todo o código estar devidamente comentado e explicado). Ao definir a variável *input* como a entrada senoidal do sistema, soma-se um *offset* de 3 ao sinal. Isso é feito pois avalia-se razões de sinais da planta para calcular a tolerância. Os sinais da planta serão também senoides, de mesma frequência, mas de amplitudes diferentes. Portanto, se fosse mantido o sinal de entrada com *offset* nulo, ocorreriam eventuais divisões por zero. Essa ideia segue a mesma lógica da técnica de *Laplace smoothing*.

Para as validações, armazena-se valores máximos de interesse em vetores, a fim de visualizar a evolução desses valores para diferentes frequências ou parâmetros da planta ao final do código.

No fim do código, utiliza-se a função de transferência da planta que proporcionou os resultados mais críticos (valores maiores de *delta* e maior tempo de estabilização). Com isso, tem-se a certeza de que se está validando o sistema de controle para o pior cenário real possível (dentro das considerações de robustez do sistema).

APÊNDICE E – LLC - DISSIPACÃO NUMÉRICA

Como explicado na Seção 10.2.1, utilizou-se o método das diferenças finitas (MDF) para resolver as equações diferenciais do sistema. Pela estrutura do sistema de controle, a saída de uma EDO é a entrada da outra, e dessa forma, as 3 EDOs do sistemas são aplicadas sobre um mesmo sinal. Isso faz com que seja acumulado erro em cima do sinal, devido à natureza aproximada da resolução das equações diferenciais por MDF. Tal situação gera o fenômeno da dissipação numérica.

A fim de visualizar esse efeito, foram realizadas 3 aproximações diferentes para resolução das EDOs. A primeira abordagem foi utilizar apenas a primeira diferença regressiva para aproximar a derivada, o que traz um erro na ordem do passo h . Para a segunda derivada, aplicava-se a primeira diferença regressiva novamente sobre a variável já derivado da mesma forma. Vale ressaltar que esta não é uma boa abordagem, pois o erro para a segunda derivada se acumula, sendo maior do que a ordem do passo h . A fórmula da primeira diferença regressiva está dada de forma genérica na eq. (E.1). Esta abordagem foi denominada abordagem I.

$$f'_i = \frac{f_i - f_{i-1}}{h} + O(h) \quad (\text{E.1})$$

Para amenizar esse efeito de erro acumulado na segunda derivada, o sistema foi reequacionado utilizando a segunda derivada regressiva para aproximar a derivada segunda. Esta aproximação segue a eq. (10.2) e possui erro da ordem do passo h . Tal abordagem foi denominada abordagem A.

Por fim, para obter um resultado ainda mais preciso, foi substituída a primeira derivada regressiva pela primeira derivada regressiva de 3 pontos. Isso reduz o erro para ordem do passo ao quadrado h^2 , seguindo a eq. (10.1). Esta abordagem foi denominada abordagem B.

Para efeito de comparação, o sistema foi simulado para as três abordagens e comparado com a simulação usando a biblioteca *control* do *Python*. O resultado está exposto na Figura 45. Vale ressaltar que, para as abordagens I e A foram utilizados passos de 0.01. Já para a abordagem B, foi utilizado passo de 0.1. Isso foi feito pra ressaltar o efeito da dissipação numérica.

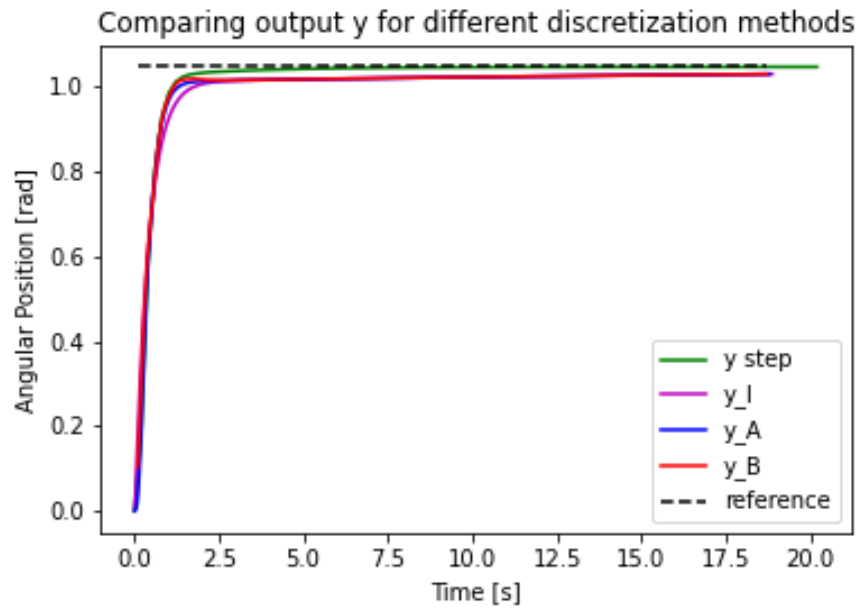


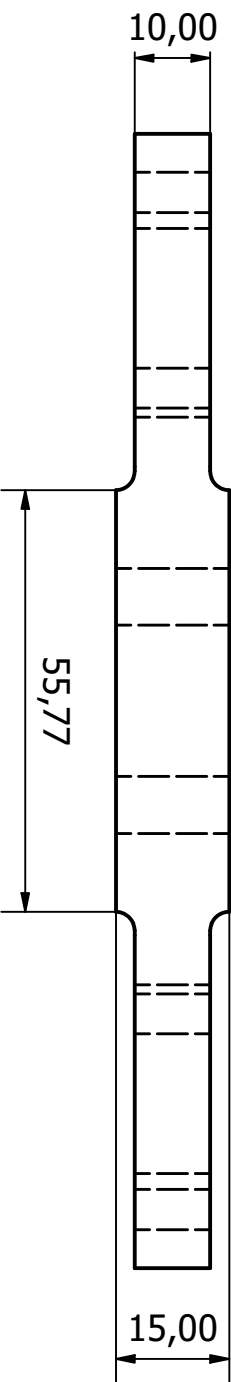
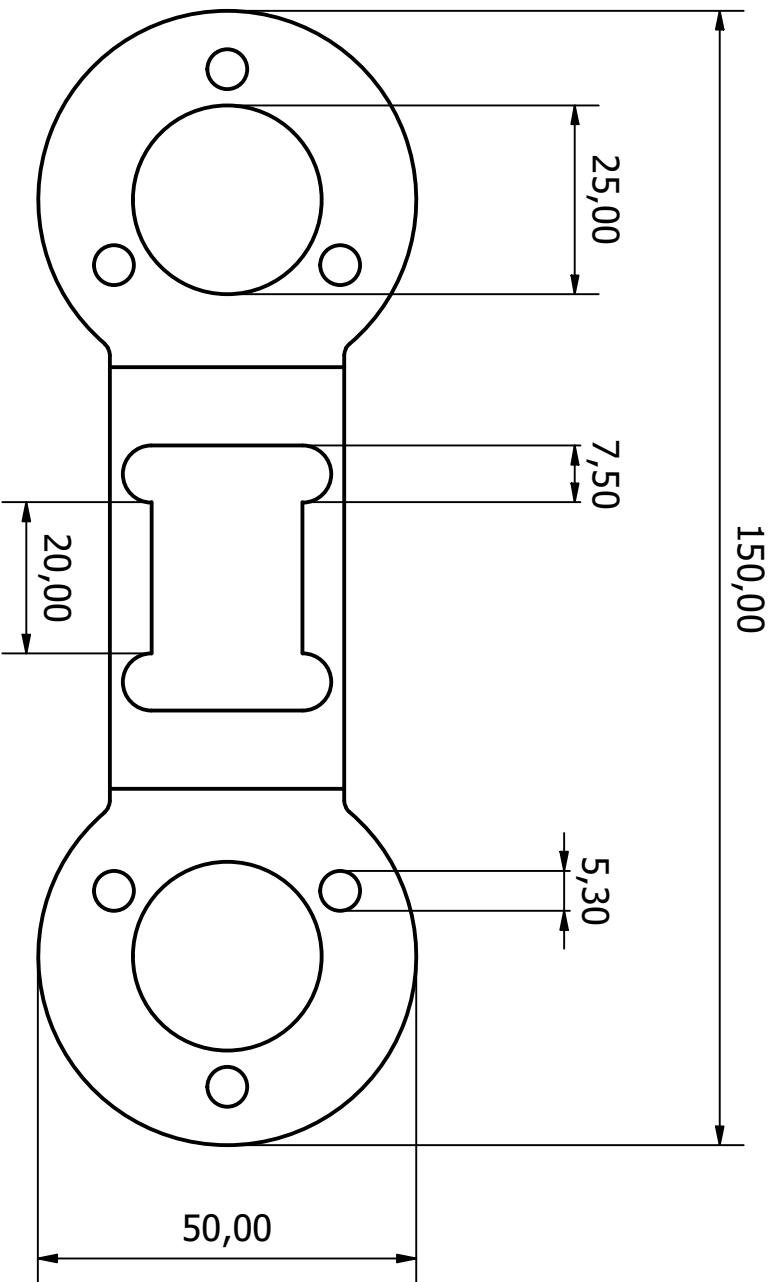
Figura 45: Simulação da evolução da saída da planta para diferentes aproximações numéricas, comparadas ao valor esperado da evolução desta variável. Foi utilizada tolerância de 1° e referência de entrada de 60° .

Nota-se que a curva vermelha, referente à simulação B, consegue acompanhar bem a curva verde (simulação ideal) em sua parte inicial, mas depois cai no mesmo platô que as outras duas abordagens caem (curvas I e A, respectivamente magenta e azul). Com uma discretização mais grosseira há menos pontos para serem calculados, portanto é acumulado menos erro ao longo do processo. Por isso há essa "sobrevida" da curva vermelha neste pequeno pico, que depois sofre forte efeito da dissipação numérica. Dessa forma fica evidente que o problema na simulação é de natureza numérica, e não do controle ou do modelo em si.

Vale ainda observar que as curvas para A e B realmente se aproximam bem mais à curva esperada (verde), como já era de se esperar.

ANEXO A – DESENHO MECÂNICO DO EXOESQUELETO

A seguir, segue o desenho mecânico do elo do exoesqueleto, com as suas cotas em milímetros.



Designed by	Checked by	Approved by	Date	Date
João Paulo				10/05/2020

Elo_bancada	Edition	Sheet
		1 / 1