

UNIVERSIDADE FEDERAL DE MINAS GERAIS
Programa de Pós-Graduação em Engenharia Mecânica

Guilherme de Paula Rúbio

**MODELO COMPUTACIONAL PARA CONTROLE DO MOVIMENTO DE DEDOS DE
PRÓTESES ROBÓTICAS BASEADO EM APRENDIZADO POR REFORÇO**

Belo Horizonte

2025

Guilherme de Paula Rúbio

MODELO COMPUTACIONAL PARA CONTROLE DO MOVIMENTO DE DEDOS DE PRÓTESES ROBÓTICAS BASEADO EM APRENDIZADO POR REFORÇO

Tese apresentada ao Programa de Pós-graduação em Engenharia Mecânica da Universidade Federal de Minas Gerais, como requisito para a obtenção do título de Doutor em Engenharia Mecânica.

Orientador: Claysson Bruno Santos
Vimieiro

Belo Horizonte

2025

R896m	<p>Rúbio, Guilherme de Paula. Modelo computacional para controle do movimento de dedos de próteses robóticas baseado em aprendizado por reforço [recurso eletrônico] / Guilherme de Paula Rúbio. – 2025. 1 recurso online (184 f. : il., color.) : pdf.</p> <p>Orientador: Claysson Bruno Santos Vimieiro.</p> <p>Tese (doutorado) – Universidade Federal de Minas Gerais, Escola de Engenharia.</p> <p>Inclui bibliografia.</p> <p>1. Engenharia mecânica – Teses. 2. Próteses e implantes – Teses. 3. Robótica – Teses. 4. Aprendizado por reforço – Teses. 5. Fisioterapia – Reabilitação – Teses. I. Vimieiro, Claysson Bruno Santos. II. Universidade Federal de Minas Gerais. Escola de Engenharia. III. Título.</p>
	CDU: 621(043)



UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA MECÂNICA

ATA DE DEFESA DE TESE

Ata da sessão pública da defesa de tese de **GUILHERME DE PAULA RÚBIO**, de registro número **2020723209**, aluno do Curso de Doutorado do Programa de Pós-Graduação em Engenharia Mecânica da Escola de Engenharia, Universidade Federal de Minas Gerais, realizada às 14h00 do dia vinte e quatro de outubro de 2025 na modalidade híbrida. A Banca examinadora indicada pelo Colegiado do Programa foi constituída pelos seguintes professores: Dr. Claysson Bruno Santos Vimieiro (orientador - Departamento de Engenharia Mecânica-UFMG), Dr. Alcimar Barbosa Soares (Universidade Federal de Uberlândia-UFU), Dr. Luciano Luporini Menegaldo (Universidade Federal do Rio de Janeiro-UFRJ), Dr. Armando Alves Neto (Departamento de Engenharia Eletrônica-UFMG) e Dr. Antônio Augusto Torres Maia (Departamento de Engenharia Mecânica-UFMG). O presidente da banca examinadora, Dr. Claysson Bruno Santos Vimieiro, abriu a sessão pública de defesa Nº 273/2025 para avaliar a tese intitulada **"MODELO COMPUTACIONAL PARA CONTROLE DO MOVIMENTO DE DEDOS DE PROTESES ROBOTICAS BASEADO EM APRENDIZADO POR REFORÇO"**, requisito final para obtenção do grau de Doutor em Engenharia Mecânica na área de concentração **"Bioengenharia"** e, após dar conhecimento aos presentes do teor das normas regulamentares do trabalho final, passou a palavra ao candidato para apresentação do seu trabalho. Seguiu-se a arguição pelos examinadores com a respectiva defesa do candidato. Após a defesa, os membros da banca examinadora realizaram a avaliação do trabalho sem a presença do candidato para julgamento e expedição do resultado final. Foi atribuída a seguinte indicação: o candidato foi considerado **APROVADO** por unanimidade. O resultado final foi comunicado ao candidato pelo presidente da banca examinadora. Nada mais havendo a tratar, lavrou-se a presente ata que será assinada eletronicamente pelos membros da banca.



Documento assinado eletronicamente por **Claysson Bruno Santos Vimieiro, Professor do Magistério Superior**, em 03/11/2025, às 13:40, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Luciano Luporini Menegaldo, Usuário Externo**, em 03/11/2025, às 14:53, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Armando Alves Neto, Professor do Magistério Superior**, em 03/11/2025, às 14:57, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Alcimar Barbosa Soares, Usuário Externo**, em 03/11/2025, às 16:12, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **Antonio Augusto Torres Maia, Professor do Magistério Superior**, em 04/11/2025, às 09:20, conforme horário oficial de Brasília, com fundamento no art. 5º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no site https://sei.ufmg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **4699493** e o código CRC **7151B777**.

AGRADECIMENTOS

Agradeço aos meus pais **Carlos e Isabel Cristina** e ao meu irmão **Victor** por me possibilitarem a realização desse trabalho acadêmico através do seu apoio e compreensão.

À minha namorada **Suzana** e seus pais, **Marcos Alexandre e Maria de Lourdes**, pela compreensão, palavras de apoio em todos os momentos, sem medir esforços para me auxiliar na realização desse trabalho.

Ao professor **Claysson**, meu orientador, pela confiança em mim depositada, pelos conselhos e pelas orientações dadas para a confecção desse trabalho e durante todos os momentos dessa etapa da minha vida acadêmica.

A toda equipe do projeto prótese, **Arthur, João Paulo, Fernanda, Maria Aparecida, Rina e todos os alunos de iniciação científica**, que foram o corpo fundamental para a realização desse trabalho, ajudando em todas as etapas realizadas.

Aos meus amigos da pós-graduação do **Matheus e Saulo**, pelas conversas, ideias, auxílios e projetos que trabalhamos juntos durante todo esse período.

À toda equipe do **LabBio**, que ajudou a equipe do projeto durante todos esses anos, desde os momentos mais tranquilos aos mais difíceis.

A todo o corpo docente e aos funcionários do **Programa de Pós-Graduação em Engenharia Mecânica (PPGMEC)** da Universidade Federal de Minas Gerais que proporcionaram a realização desse doutorado.

A todos os **amigos e familiares**, por todo o apoio dado a mim durante todo o meu trajeto acadêmico.

À **Financiadora de Estudos e Projetos (Finep)** por todo o aporte financeiro que possibilitou a execução do projeto – Convênios: 01.21.0101.00, Ref.2790/20 e 01.24.0167.00, Ref.2188/22.

O presente trabalho foi realizado com apoio da **Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES)** – Código de Financiamento 001.

Resumo

As próteses robóticas vêm se aproximando das funcionalidades do membro fisiológico, entretanto, elas ainda carecem de sistemas de controle adaptados às necessidades do usuário. O uso da eletromiografia superficial (sEMG) melhora a integração do usuário com o dispositivo sendo utilizada em controladores que o traduzem em movimentos desejados pelo usuário. Apesar da alta acurácia, esses controladores não se adaptam às atividades de vida diária (ATVDs), pois o movimento do dispositivo é feito de forma padronizada, e não apresentam uma transição suave. Para melhorar esse sistema, tenta-se traduzir o sinal em ângulos de juntas, por meio de modelos de regressão, entretanto, dependem de inúmeros parâmetros que variam de usuário para usuário. Visando o desenvolvimento de controladores mioelétricos adaptados ao ambiente e ao usuário para execução das ATVDs, essa tese de doutorado apresentou uma avaliação do uso de técnicas de aprendizado por reforço (RL) para criação de um novo controlador mioelétrico que combina um classificador mioelétrico e um controlador de trajetórias, para execução de trajetórias fisiológicas pelo dispositivo. Inicialmente um teste piloto do controlador de trajetórias foi realizado, utilizando uma rede neural artificial (RNA) treinada RL utilizando o algoritmo *Deep Deterministic Policy Gradient* (DDPG). Nesse teste, o controlador realizou a flexão de um dedo, com dimensões fisiológicas, executando duas trajetórias pré-determinadas, senoidal e linear, controlando o torque de cada articulação. Avaliando a resposta do controlador foi observado que ele definiu estratégias distintas de atuação para cumprir as trajetórias esperadas obtendo um erro médio de $3,98 \pm 2,90$ mm em relação com a trajetória senoidal alvo e $3,22 \pm 1,42$ mm para a trajetória linear alvo. Além disso, com o uso de entradas de seleção de trajetórias proporcionais e de modelos com parâmetros distintos, foi observado que o controlador apresentou respostas distintas, indicando a adaptabilidade do controlador. Avaliando a aplicação da técnica de treinamento para uma prótese real, foi desenvolvido um controlador mioelétrico para a prótese HANDi Hand, modelada em um programa de elementos finitos (FEA). O classificador de movimentos, utilizado como seletor de trajetórias, foi treinado através do uso de técnicas de reconhecimento de padrões, utilizando uma RNA convolucional inspirada na arquitetura *Inception*. O controlador de trajetórias teve como objetivo executar trajetórias, com a prótese, dos movimentos de prensão de força, pinça trípede e bidigital. Para isso, foi utilizado o algoritmo de RL *Twin Delayed Deep Deterministic Policy Gradient* (TD3), e uma rede com células de LSTM convolucional. O classificador de sinais de sEMG foi capaz de apresentar uma acurácia superior a 96% com os dados de um voluntário amputado após a calibração. Em relação ao treinamento do controlador de trajetórias, foi observado que o elevado custo computacional em decorrência do modelo FEA utilizado no ambiente de treinamento, inviabilizou o treinamento para milhares de épocas, indicando que para o uso efetivo da metodologia proposta se faz necessário o estudo de modelos simplificados de atuadores subatuados. Apesar disso, com o controlador de melhor recompensa de treinamento, foi observado que foi utilizada estratégias distintas para execução dos movimentos, tanto com as entradas de treinamento como com os dados de indivíduos amputados.

Palavras-chave: Eletromiografia de Superfície, Redes Neurais Artificiais, Aprendizado por Reforço, Controlador Mioelétrico, Prótese de Membro Superior.

Abstract

Robotic prostheses currently demonstrate an elevated level of functionality, similar to physiological limbs. However, they still lack control systems that are adapted to user needs. The use of surface electromyography (sEMG) has been demonstrated to enhance user integration with the device, with application in controllers that translate the signal into movements desired by the user. Despite their high level of accuracy, these controllers are not adapted to activities of daily living (ADLs), as the device's movement is standardized and does not provide a smooth transition. To enhance this system, researchers are exploring the potential of translating the signal into joint angles through regression models. However, variations in numerous parameters from user to user necessitate careful consideration when implementing these models. Aiming at the development of myoelectric controllers adapted to the environment and the user for the execution of ADLs, this doctoral thesis presented an evaluation of the use of reinforcement learning (RL) techniques to create a new myoelectric controller that combines a myoelectric classifier and a trajectory controller for the execution of physiological trajectories by the device. The pilot test of the trajectory controller was initiated using an artificial neural network (ANN) that had been trained using RL with the Deep Deterministic Policy Gradient (DDPG) algorithm. In this investigation, the controller was employed to flex a finger with physiological dimensions, executing two predetermined trajectories, sinusoidal and linear, controlling the torque of each joint. An evaluation of the controller's response revealed the presence of diverse strategies employed to fulfill the expected trajectories. The mean error observed was $3.98 \text{ mm} \pm 2.90 \text{ mm}$ relative to the target sinusoidal trajectory and $3.22 \text{ mm} \pm 1.42 \text{ mm}$ for the target linear trajectory. In addition, it was observed that the controller presented different responses when different proportional inputs for trajectory selection and models with different parameters were used. This indicates the controller's adaptability. An evaluation of the application of the training technique to a real prosthesis was conducted, resulting in the development of a myoelectric controller for the HANDi Hand prosthesis, which was modelled in a finite element analysis (FEA) program. The motion classifier, employed as a trajectory selector, was trained using pattern recognition techniques, with a convolutional ANN based on the Inception architecture. The trajectory controller was designed to execute trajectories with the prosthesis for force grasping, tripod pinch, and two-finger pinch movements. The RL Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm and a network comprising convolutional LSTM cells were utilized in this context. The sEMG signal classifier demonstrated an accuracy of over 96% with data from an amputee volunteer following calibration. The elevated computational cost of the FEA model employed in the training environment was impractical for extended training over thousands of episodes, suggesting that the effective utilization of the proposed methodology necessitates the investigation of simplified models of underactuated actuators. However, the controller with the highest training reward, it was observed that different strategies were employed to execute the movements, both with the training inputs and with the data from amputees.

Keywords: Surface Electromyography, Artificial Neural Networks, Reinforcement Learning, Myoelectric Controller, Upper Limb Prosthesis.

LISTA DE ILUSTRAÇÕES

Figura 1 – Etapas para o desenvolvimento e utilização de um controlador mioelétrico	23
Figura 2 – Exemplo do posicionamento dos eletrodos para aquisição de sinais de sEMG	25
Figura 3 – Representação da separação das características do sinal de sEMG em relação ao número de movimentos classificados e ao número de usuários	37
Figura 4 – Representação do neurônio biológico (a) e do neurônio artificial (b)	39
Figura 5 – Funções de ativação dos neurônios artificiais: logística (a) , hiperbólica (b) , gaussiana (c) e linear (d)	41
Figura 6 – Exemplo da arquitetura de uma RNA com 2 camadas ocultas.....	43
Figura 7 – Arquitetura típica de RNA com recorrência.....	43
Figura 8 – Diagrama do funcionamento de uma célula LSTM.....	44
Figura 9 – Exemplo da operação de convolução em duas dimensões.....	46
Figura 10 – Diagrama de blocos de uma camada convolucional	46
Figura 11 – Exemplo da evolução do erro de treinamento em função do número de épocas.....	51
Figura 12 – Diagrama do treinamento por aprendizado por reforço	52
Figura 13 – Exemplo de aplicação do voto majoritário com o acúmulo de nove votos	64
Figura 14 – Exemplo de célula de convolução <i>Inception</i> (a) empregada em um modelo de CNN (b) aplicada à classificação de sinais de sEMG	65
Figura 15 – Fluxograma do processo completo de desenvolvimento do trabalho	69
Figura 16 – Localização dos eixos coordenados de referência utilizados na construção do modelo matemático do dedo 2	71
Figura 17 – Posição da ponta do dedo para as trajetórias: senoidal (a) e linear (b)	76
Figura 18 – Arquitetura de RNA utilizada como o Ator no modelo de controle	77
Figura 19 – Arquitetura de RNA utilizada como Crítico no algoritmo DDPG durante o treinamento	78
Figura 20 – Evolução da função de recompensa ao longo dos episódios de treinamento (a) e recompensas obtidas nos episódios completos (b)	82
Figura 21 – Evolução da recompensa do Ator durante os episódios de validação (a) e recompensas obtidas nos episódios completos (b)	83
Figura 22 – Trajetórias realizadas pelo controlador para as entradas senoidais (a) e lineares (b) , comparadas à trajetória desejada	84
Figura 23 – Torques aplicados nas juntas de revolução durante a execução da trajetória senoidal ((a) , (c) e (e)) e da trajetória linear ((b) , (d) e (f))	85
Figura 24 – Rotação relativa das juntas ((a) , (c) e (e)) e velocidade angular relativa das juntas ((b) , (d) e (f)) durante a execução da trajetória senoidal	87

Figura 25 – Rotação relativa das juntas ((a), (c) e (e)) e velocidade angular relativa das juntas ((b), (d) e (f)) durante a execução da trajetória linear	88
Figura 26 – Trajetórias executadas para entradas proporcionais de trajetória: 25% senoidal (a), 50% senoidal (M5050) (b) e 75% senoidal (M7525) (c)	90
Figura 27 – Torque aplicado às juntas durante a execução das entradas proporcionais: junta 1 (a), junta 2 (b) e junta 3 (c)	91
Figura 28 – Rotação relativa das juntas ((a), (c) e (e)) e velocidade angular relativa das juntas ((b), (d) e (f)) com entradas proporcionais apresentadas ao controlador	92
Figura 29 – Variação das posições x (a) e y (b) o longo do tempo para todas as entradas apresentadas ao controlador, em comparação com as posições esperadas utilizadas durante o treinamento	93
Figura 30 – Torque aplicado na junta 1 com o modelo em escala de 90% ((a), (b)), 110% ((c), (d)) e 125% ((e), (f)) nas trajetórias senoidal e linear	96
Figura 31 – Fluxograma do processo de treinamento do controlador mioelétrico ..	100
Figura 32 – Diagrama esquemático do controlador mioelétrico desenvolvido, composto por dois estágios: seletor de trajetórias (treinado por aprendizado supervisionado) e controlador de trajetórias (treinado por aprendizado por reforço)	101
Figura 33 – Prótese HANDi Hand construída no Labbio, utilizada para a implementação e avaliação do controlador mioelétrico proposto	102
Figura 34 – Disposição dos eletrodos utilizados nos bancos de dados DB2, DB3 e DB7 do NINAPRO durante a coleta de sinais de sEMG	104
Figura 35 – Disposição dos sensores flexíveis da CyberGlove II utilizados na aquisição de dados cinemáticos nos bancos de dados DB2 e DB3 (a) e DB7 (b) do NINAPRO	105
Figura 36 – Movimentos realizados pelos participantes durante a aquisição dos bancos de dados DB2, DB3 e DB7 do NINAPRO	106
Figura 37 – Padrões de força realizados pelos participantes durante a aquisição do banco de dados DB2 do NINAPRO	107
Figura 38 – Fluxo das etapas de tratamento dos dados de sEMG utilizados no treinamento e validação do controlador mioelétrico	107
Figura 39 – Tipos de movimentos de pinça e preensão comumente executados durante as ATVDs, conforme protocolo SHAP	108
Figura 40 – Execução da preensão em extensão pela prótese HANDi Hand em bancada experimental	109
Figura 41 – Diagrama esquemático das camadas de convolução em paralelo utilizadas na arquitetura do classificador	110
Figura 42 – Arquitetura da RNA utilizada como classificador de movimentos a partir de sinais de sEMG	112
Figura 43 – Digrama esquemático de interação com o Ansys Mechanical APDL® para construção do ambiente de treinamento do controlador de trajetórias	114

Figura 44 – Mecanismo de atuação do dedo da prótese HANDi Hand durante os movimentos de flexão (a) e extensão (b) do dedo	115
Figura 45 – Modelo do dedo indicador da HANDi Hand construído no <i>software</i> ANSYS Mechanical APDL®	116
Figura 46 – Juntas de revolução (a) e juntas fixas (b) definidas na construção do modelo do mecanismo da prótese	117
Figura 47 – Regiões de contato entre as falanges, a polia e o tendão artificial.....	117
Figura 48 – Condição de contorno utilizada no teste de malha	118
Figura 49 – Coordenadas médias (x e y) e trajetórias esperadas dos episódios: preensão de força ((a) e (b)), pinça trípode ((c) e (d)) e pinça bidigital ((e) e (f)) ...	125
Figura 50 – Arquitetura da RNA utilizada como Ator no controlador de trajetória ...	127
Figura 51 – Arquitetura utilizada nos Críticos para o treinamento do controlador de trajetórias.....	129
Figura 52 – Evolução da acurácia e da função de custo durante o treinamento do classificador nas <i>folds</i> :1 (a) , 2 (b) , 3 (c) , 4 (d) e 5 (e) , bem como as acurácias obtidas para cada fold com os dados de validação (f)	133
Figura 53 – Matrizes de confusão normalizadas em relação às previsões, obtidas com os dados de validação nas <i>folds</i> 1 (a) , 2 (b) , 3 (c) , 4 (d) e 5 (e)	135
Figura 54 – Matriz de confusão normalizada em relação às previsões, obtida com os dados do voluntário 21 do DB7 antes da calibração do classificador.....	136
Figura 55 – Evolução da acurácia e da função de custo durante a calibração do classificador nas <i>folds</i> :1 (a) , 2 (b) , 3 (c) , 4 (d) e 5 (e) , bem como as acurácias obtidas para cada fold com os dados de validação (f)	137
Figura 56 – Matrizes de confusão normalizadas em relação às previsões, obtidas com os dados de validação nas <i>folds</i> 1 (a) , 2 (b) , 3 (c) , 4 (d) e 5 (e) , após a calibração	138
Figura 57 – Evolução da recompensa obtida pelo Ator a cada episódio de treinamento (a) e recompensa nos episódios concluídos (b)	140
Figura 58 – Comparação da variação da coordenada x ao longo do tempo nos movimentos de preensão de força (a) , pinça trípode (b) e pinça bidigital (c) , executada e esperada para as entradas binárias utilizadas no treinamento	142
Figura 59 – Comparação da variação da coordenada x ao longo do tempo nos movimentos de preensão de força (a) , pinça trípode (b) e pinça bidigital (c) , executada e esperada, utilizando o classificador associado ao controlador de trajetória com dados de sEMG de um voluntário do DB3	143
Figura 60 – Comparação da variação da coordenada x ao longo do tempo nos movimentos de preensão de força (a) , pinça trípode (b) e pinça bidigital (c) , executada e esperada, utilizando o classificador associado ao controlador de trajetória a partir dos dados de sEMG do voluntário 21 do DB7.....	145

LISTA DE TABELAS

Tabela 1 – Valores das propriedades do dedo indicador	73
Tabela 2 – Parâmetros utilizados no treinamento do controlador no teste piloto	79
Tabela 3 – Comparação dos erros médios absolutos das trajetórias executadas pelo controlador em relação às trajetórias-alvo	90
Tabela 4 – Principais características dos bancos de dados do NINAPRO, com resumo das informações disponibilizadas.....	104
Tabela 5 – Organização dos neurônios de saída do classificador mioelétrico, indicando a correspondência com os movimentos e a condição de repouso	113
Tabela 6 – Resumo dos parâmetros de treinamento do classificador mioelétrico ..	113
Tabela 7 – Propriedades dos materiais utilizados na construção do modelo.....	116
Tabela 8 – Resumo dos parâmetros utilizados na definição dos contatos do modelo	118
Tabela 9 – Malhas Utilizadas durante o teste de malha.....	119
Tabela 10 – Diferenças entra rotação angular relativa das juntas da prótese em cada uma das malhas testadas.....	119

LISTA DE ABREVIações

ATVD	Atividade de Vida Diária
AI	Inteligência Artificial
AR	Coefficientes de Autorregressão
BSS	Separação de Fonte Cega
CC	Correlação de Concordância
CNN	Redes Neurais Convolucionais
DAF	Análise de Flutuação de Tendência
DAMV	Diferença Absoluta da Média
DASDV	Diferença Absoluta do Desvio Padrão
DDPG	<i>Deep Deterministic Policy Gradient</i>
DRL	Aprendizado por Reforço Profundo
ER	Razão de Energia
FEA	Análise de Elementos Finitos
FFT	Transformada Rápida de Fourier
GDL	Graus de Liberdade
GMM	Modelos de Mistura Gaussiana
ICA	Análise de Componentes Independentes
LDA	Análise de Discriminação Linear
LSTM	Memória de Longo e Curto Prazo
MAV	Máximo Valor Absoluto
MAVS	Valor Absoluto Médio de Inclinação
MDF	Frequência Mediana
ML	Aprendizado de Máquina
MLP	Múltiplas Camadas <i>Perceptron</i>
MNF	Frequência Média
PID	Controlado Proporcional, Integrativo e Derivativo
PPO	<i>Proximal Policy Optimization</i>
PRL	Reconhecimento de Padrões
PS	Densidade de Potência do Espectro
QLDA	Análise de Discriminação Quadrática
ReLU	Unidades Retificadas Lineares

RL	Aprendizado por Reforço
RMS	Raiz Média Quadrática
RNA	Redes Neurais Artificiais
RNN	Rede Neurais Recorrentes
SAC	<i>Soft Actor-Critic</i>
SC	Coeficiente de Esparsidade
sEMG	Eletromiografia de Superfície
SGD	Método Descida Estocástica do Gradiente
SSI	Integral Simples Quadrática
SVM	Máquina de Suporte de Vetores
TD3	<i>Twin Delayed Deep Deterministic Policy Gradient</i>
VAR	Variância
WAMP	Amplitude de Wilson
WL	Comprimento de Onda
WT	Transformada de Wavelet
ZC	Cruzamento de Zero

SUMÁRIO

CAPÍTULO 1: INTRODUÇÃO	16
1.1 Originalidade	19
1.2 Objetivos	21
1.2.1 Objetivo Principal	21
1.2.2 Objetivos Específicos	21
CAPÍTULO 2: REVISÃO BIBLIOGRÁFICA	22
2.1 Sinais Eletromiográficos Superficiais	22
2.2 Sistemas de Predição de Movimentos Utilizando sEMG	24
2.2.1 Aquisição e Condicionamento do Sinal sEMG	24
2.2.2 Extração de Características	30
2.2.3 Classificação dos Sinais e Controle do Dispositivo	38
2.2.3.1 Redes Neurais Artificiais.....	39
2.2.3.2 Aprendizado de Máquina.....	47
2.2.3.3 Aprendizado por Reforço.....	52
2.2.3.4 Classificadores e Controladores Mioelétricos.....	59
CAPÍTULO 3: TESTE PILOTO COM MODELO BIOMECÂNICO	69
3.1 Metodologia	70
3.1.1 Ambiente de Treinamento.....	70
3.1.2 Definição das Trajetórias-alvo	75
3.1.3 Treinamento do Controlador.....	76
3.1.4 Avaliação Computacional do Controlador.....	81
3.2 Resultados e Discussão	82
3.2.1 Respostas Para as Entradas Senoidal e Linear	82
3.2.2 Avaliação com as Entradas de Trajetória Proporcionais.....	89
3.2.3 Avaliação do Controlador de Trajetórias com o Modelo em Escala.....	95
3.2.4 Limitações do Teste Piloto.....	98
CAPÍTULO 4: DESENVOLVIMENTO DO CONTROLADOR MIOELÉTRICO	100
4.1 Metodologia	101
4.1.1 Aquisição e Condicionamento de Dados de sEMG e Cinemáticos	103
4.1.2 Seleção e Tratamento de Dados do NINAPRO.....	107

4.1.3 Arquitetura e Treinamento do Classificador de Movimentos.....	110
4.1.4 Desenvolvimento do Ambiente de Treinamento	114
4.1.4.1 Criação do Modelo de Prótese de Membro Superior	114
4.1.4.2 Função de Recompensa	120
4.1.5 Definição das Trajetórias-alvo	123
4.1.6 Arquitetura e Treinamento do Controlador de Trajetórias	125
4.1.7 Avaliação do Controlador	132
4.2 Resultados e Discussão	133
4.2.1 Classificador de Sinais de sEMG	133
4.2.2 Controlador de Trajetória	139
4.2.3 Limitações	146
CAPÍTULO 5: CONCLUSÃO	148
CAPÍTULO 6: TRABALHOS FUTUROS	150
REFERÊNCIAS BIBLIOGRÁFICAS	151
APÊNDICE A – RESULTADOS COMPLEMENTARES DO TESTE PILOTO	163

CAPÍTULO 1: INTRODUÇÃO

As próteses de membros superiores são dispositivos essenciais para a qualidade de vida de indivíduos amputados, permitindo a realização das atividades da vida diária (ATVDs). As próteses robóticas, com o advento de novos materiais, sistemas microcontrolados e sensores, têm adquirido funcionalidades cada vez mais similares às dos membros biológicos (BERTOS; PAPADOPOULOS, 2019). Apesar dos avanços presentes nestes dispositivos, há ainda uma lacuna que precisa ser aprimorada, que é como eles podem se adaptar ao espaço que estão inseridos, ou executar movimentos mais integrados a cada usuário, proporcionando comandos mais “naturais” e consequentemente melhorando a integração com o usuário durante as ATVDs (LI, K. *et al.*, 2020; ZHOU; ALICI, 2022).

Dentre os tipos de próteses robóticas existentes, as mioelétricas são mais avançadas. Nesse tipo de dispositivo, são utilizados eletrodos de eletromiografia superficial (sEMG) na superfície do coto do usuário que captam os potenciais elétricos gerados pela ativação das unidades motoras. Ou seja, é uma forma eficaz de utilizar os comandos neuronais do usuário de forma não invasiva (BI; FELEKE; GUAN, 2019; LI, K. *et al.*, 2020; MERLETTI; FARINA, 2016). Uma vez medido, todo um sistema de aquisição e classificação é capaz de traduzir o sinal mensurado em movimentos desejados pelos usuários, determinando quais movimentos devem ser executados (LI, K. *et al.*, 2020; SIMAO *et al.*, 2019; ZHOU; ALICI, 2022).

Algumas etapas devem ser realizadas para que se torne possível o uso do sEMG no controle de próteses, sendo elas amplificação, filtragem, extração de características e classificação do movimento desejado (CHEN, M.; ZHOU, 2016; DE LUCA *et al.*, 2006; LI, K. *et al.*, 2020; MERLETTI; FARINA, 2016; NASTARIN; AKTER; AWAL, 2019; NOVAK; RIENER, 2015; RAHMATILLAH; SALAMAT; SOELISTIONO, 2019; SIMAO *et al.*, 2019). Dentre elas, a etapa crucial é a de classificação do movimento, uma vez que ela será responsável por traduzir o sinal em comandos. Apesar de presentes nas próteses mais modernas, o uso dessa técnica não é algo novo, sendo vista em trabalhos como de HUDGINS; PARKER; SCOTT, (1993) e PHILIPSON, (1985) que demonstravam a capacidade do uso desse sinal para determinação de movimentos desejados, e que atualmente podem ter um grau de acurácia superior a 90% (GUO, W. *et al.*, 2017b; VÁSCONEZ *et al.*, 2023).

Com o aumento da capacidade computacional, novos algoritmos para classificar os sinais de sEMG vêm sendo desenvolvidos, sendo divididos em duas categorias que são os métodos de ativação por limiar e os métodos reconhecimento de padrões (PRL) (LI, C. *et al.*, 2018; SCHEME; ENGLEHART, 2011). Os métodos de ativação por limiar baseiam-se na correlação entre a intensidade do sinal sEMG e a execução do movimento, podendo ser associado a atividade de músculos que não estão envolvidos diretamente com o movimento desejado. As técnicas mais utilizadas para esse fim são os sistemas *on/off* associados a máquinas de estados, onde cada músculo ativa um estado da prótese, sendo esses sistemas pouco intuitivos para o uso. (AHMADIZADEH *et al.*, 2017; FAJARDO *et al.*, 2017; KAWAMOTO *et al.*, 2010). O PRL se baseia em técnicas de classificação do sinal de sEMG proporcionando melhor integração do usuário ao sistema, devido ao nível mais alto de informação (SADIKOGLU; KAVALCIOGLU; DAGMAN, 2017). No PRL, a intenção de movimento é determinada por meio das características extraídas do sinal, fazendo-se a correlação entre o sinal lido e o movimento desejado, criando padrões. As técnicas principais para a classificação de padrões no PRL são as redes neurais artificiais (RNA), análise de discriminação linear (LDA), a análise de discriminação quadrática (QLDA), árvore de decisão e máquinas de suporte de vetor (SVM) (NASTARIN; AKTER; AWAL, 2019; RAHMATILLAH; SALAMAT; SOELISTIONO, 2019). Devido ao seu potencial, técnicas de PRL são o maior foco atualmente no desenvolvimento de sistemas de controle mais integrados ao usuário. Entretanto, mesmo com a alta acurácia obtida, superior a 90% (GUO, W. *et al.*, 2017c; VÁSCONEZ *et al.*, 2023), elas ainda têm grandes limitações com o uso para ATVDs, pois limitam suas capacidades de comandos da prótese a um conjunto específico de movimentos, o que não condiz com as ATVDs onde são feitos movimentos não padronizados e dependentes de variadas combinações das articulações (NASTARIN; AKTER; AWAL, 2019; STRAZZULLA *et al.*, 2017).

As próteses comerciais mais avançadas como a Myobock (Ottobock, Alemanha) e Bebionic Hand (Ottobock, Alemanha) geralmente possuem dois tipos de controle, o proporcional e o de ativação de limiar. Segundo JIANG *et al.*, (2025), o primeiro tipo de controle é utilizado em próteses com poucos graus de liberdade, e a amplitude do sinal de sEMG é utilizado para determinar o valor de força de preensão ou o ângulo desejado. Já no segundo, o sistema de controle apenas compara o valor da amplitude do sinal com um valor de referência, em que, a partir do momento em que o limiar é

atingido, a prótese realiza o movimento de pinça desejado. Existe ainda um terceiro grupo de controladores que mistura os dois métodos. Nesse caso, é utilizado um botão ou algum tipo de PRL de sEMG que determinam qual o padrão de pinça desejado. Assim, uma vez que o movimento é selecionado, a prótese utiliza o controle por limiar de ativação ou proporcional para executar a tarefa. Apesar de efetivamente realizar diversos tipos de movimento, eles não são feitos de forma natural, principalmente quando se deseja alterar de um tipo de pinça para outra, o que gera uma falta de intuição do usuário ao usar o dispositivo.

Apesar dos atuais avanços, existe a necessidade de desenvolver controladores mais adaptáveis ao ambiente e aos usuários. Uma forma de se atender a essas necessidades é desenvolver controladores contínuos que determinam posições, velocidades e torques que devem ser executados pelo dispositivo. Alguns autores vêm tentando desenvolver modelos de regressão para associar os ângulos das articulações da mão através das características extraídas do sinal de sEMG, entretanto esses modelos dependem de muitos parâmetros (como rigidez das articulações, massa, centro de massa, entre outros), que variam entre cada usuário (BAKSHI; MANJUNATHA; KUMAR, 2018; BUENO; MONTANO, 2017; GUO, S. *et al.*, 2015; JIANG *et al.*, 2025; LI, K. *et al.*, 2020; LIZHI PAN *et al.*, 2013; REATEGUI *et al.*, 2013). Além disso, não podem ser utilizados diretamente como sinal de controle, uma vez que as próteses não possuem todos os GDLs de uma mão fisiológica. Outros autores propõem modelos de controle híbridos onde combinam os sistemas de PRL com sistemas autônomos de controle. Nesses dispositivos, o classificador determina o movimento de pinça que a prótese deve executar e uma vez que o objeto é “agarrado” controladores de força ou escorregamento atuam e evitam que os objetos sejam soltos pela prótese (DUTRA; SILVEIRA, 2023; GHAZALI *et al.*, 2017).

Para preencher essa lacuna da literatura, de como os movimentos de próteses mioelétricas podem ser adaptados dinamicamente ao espaço e aos usuários, é necessário o desenvolvimento de novas técnicas de controle contínuo. Uma das formas de se obter isso seria com o uso do aprendizado por reforço (RL). No aprendizado por reforço, ao invés de relacionarmos as entradas com as saídas desejadas, nessa técnica o modelo a ser treinado interage com o ambiente de treinamento e então recebe recompensas para as ações por ele escolhidas, determinando assim quais as melhores ações a serem tomadas para cada situação (SUTTON; BARTO, 2018). Com o uso

de aprendizado por reforço profundo (DRL) e o desenvolvimento de algoritmos de treinamento como o *Deep Deterministic Policy Gradients* (DDPG) (LILLICRAP *et al.*, 2015), o *Twin Delayed Deep Deterministic Policy Gradients* (TD3) (FUJIMOTO; VAN HOOFF; MEGER, 2018), o *Soft Actor-Critic* (SAC) (HAARNOJA *et al.*, 2018) e o *Proximal Policy Optimization* (PPO) (SCHULMAN *et al.*, 2017) as técnicas de RL tem demonstrado elevada capacidade no desenvolvimento de controladores robóticos contínuos, que apresentam uma dinâmica complexa e não linear, como o necessário para um controlador de prótese robótica (HAARNOJA *et al.*, 2018; KUMAR *et al.*, 2023; LI, Z. *et al.*, 2020; MOSLEMI; SADEDEL; MOGHADAM, 2024; WANG; HONG; ZHONG, 2022). Entretanto é importante considerar que essa técnica de treinamento demanda elevado custo computacional e tem uma sensibilidade elevada aos parâmetros de treinamento o que traz dificuldades em se obter um resultado satisfatório em alguns casos (VÁSCONEZ *et al.*, 2023). Visto isso, essa tese de doutorado investiga as capacidades das técnicas de RL DDPG e TD3 no treinamento de um modelo de RNA, utilizado em um controlador inteligente para próteses robóticas de membro superior.

1.1 Originalidade

Os métodos atuais para predição de movimento utilizados no controle de próteses de membro superior limitam a funcionalidade do dispositivo a um determinado conjunto de movimentos pré-estabelecidos, o que não é adequado para execução das ATVDs (NASTARIN; AKTER; AWAL, 2019; STRAZZULLA *et al.*, 2017). Assim, é necessário o desenvolvimento de novos controladores adaptativos que associem o sinal de sEMG com movimentos não padronizados, permitindo que o usuário determine as trajetórias que os dedos irão fazer de forma contínua, e ao mesmo tempo se adapte aos objetos que estão sendo manipulados. Dessa forma, se faz necessário que o sistema de controle associe os comandos dos atuadores da prótese com o sinal de sEMG, executando trajetórias que sejam o mais próximo possível da fisiológica, e que consigam ser diferenciadas de acordo com o movimento desejado, não se limitando a realizar sempre as mesmas trajetórias previamente determinadas.

Na literatura é possível observar que existe uma relação entre o sEMG e os estados cinemáticos do membro superior, sendo demonstrada em trabalhos que avaliaram a influência do sinal em movimentos fisiológicos com um grau de liberdade (GDL) (ARANCETA-GARZA; LAKANY; CONWAY, 2013; YANG *et al.*, 2017a). Uma

das formas para se associar os sinais de sEMG com a dinâmica do movimento esperada é através da determinação dos estados cinemáticos do membro utilizando regressores, que demonstram sua capacidade de fazer tal associação, dependem de muitos parâmetros, que se alteram de usuário para usuário, o que dificulta sua utilização para múltiplos usuários (BAKSHI; MANJUNATHA; KUMAR, 2018; BUENO; MONTANO, 2017). Alguns autores tentam desenvolver um controlador discreto através da generalização dos ângulos de junta utilizando RNAs, tentando superar o problema de determinação de parâmetros (DAI; HU, 2020; JIANG *et al.*, 2025; LONG *et al.*, 2023). Por fim, uma última técnica para adaptar os controladores de prótese de membro superior é o controle híbrido, em que as técnicas de PRL são combinadas com controladores autômatos que realizam o controle de deslizamento e força do dispositivo com uso de controladores clássicos, como o proporcional integrativo derivativo (PID) a partir do momento em que a prótese pega algum objeto (DUTRA; SILVEIRA, 2023; GHAZALI *et al.*, 2017; MATOS; CAPSI-MORALES; PIAZZA, 2025).

Pensando no desenvolvimento de um novo controlador capaz de comandar de forma contínua e adaptativa uma prótese, nesse trabalho foi combinado dois modelos em série de RNAs, uma para fazer a classificação do sinal, aproveitando o bom desempenho de técnicas de PRL já utilizadas na literatura, e um controlador de trajetória, treinado por técnicas de RL, capaz de determinar os comandos de atuação do dispositivo. O uso dessa combinação de controlador foi inspirado no funcionamento do sistema neuromuscular. Segundo GALLIVAN *et al.*, (2018) durante a execução de movimentos, o sistema neuromuscular divide suas funções, o centro superior planeja quais tarefas devem ser feitas, enquanto as estruturas do nível intermediário e local avaliam como o membro está e então determinam como movimento deverá ser executado. Visando reproduzir essa divisão, o classificador atuará de forma análoga ao centro superior, definindo os movimentos desejados pelo usuário, enquanto o controlador de trajetórias atuará como as estruturas intermediárias e locais, ou seja, avaliará como a prótese está interagindo com os objetos ou o ambiente, e decidirá como os atuadores devem ser acionados para fazer a trajetória.

Para treinar o modelo que será utilizado no controlador de trajetórias, foram utilizadas as técnicas de aprendizado por reforço DDPG e TD3. Elas se mostraram adequadas para o uso pretendido, uma vez que na literatura são utilizadas principalmente no desenvolvimento de controladores robóticos de múltiplos GDL e em sistema

de planejamento de movimento, sendo esses os objetivos pretendidos com o controlador proposto (ALMAHAMID; GROLINGER, 2022; DONG *et al.*, 2023; LI, Z. *et al.*, 2020; RAZZAGHI *et al.*, 2024).

Assim sendo, essa tese apresenta como **ineditismo** a avaliação de técnicas de RL, computacionalmente, no desenvolvimento de um novo controlador adaptativo híbrido para prótese de membro superior, combinando um sistema de classificação de sinais de sEMG com o controlador contínuo de trajetórias. Essa investigação visa determinar se essa técnica permite o desenvolvimento de controladores inteligentes, capazes de fazer com que o dispositivo execute múltiplas trajetórias de forma adaptativa, auxiliando na integração do usuário com o dispositivo.

1.2 Objetivos

1.2.1 Objetivo Principal

O objetivo principal deste trabalho é avaliar, por meio de análises computacionais, o desempenho e limitações do uso do RL no desenvolvimento de um controlador inteligente para dedos de próteses mioelétricas de membro superior utilizando RNAs treinadas para o reconhecimento de padrões e de planejamento de trajetórias, possibilitando o controle adaptativo do dispositivo.

1.2.2 Objetivos Específicos

- Desenvolver um controlador de torque para articulações de um dedo com dimensões fisiológicas, cumprindo trajetórias previamente especificadas;
- Avaliar o comportamento adaptativo de um controlador de trajetória utilizando o algoritmo DDPG para o treinamento;
- Investigar a combinação de um modelo baseado em elementos finitos e corpos rígidos na criação de ambientes de treinamento para RL.

CAPÍTULO 2: REVISÃO BIBLIOGRÁFICA

Nesta seção são apresentados sobre os sinais eletromiográficos superficiais (sEMG) e como eles podem ser utilizados para o controle de próteses. Além disso, serão apresentados os conceitos de redes neurais artificiais (RNA), de aprendizado de máquinas, de aprendizado por reforço, os algoritmos de aprendizado por reforço *Deep Deterministic Policy Gradients* (DDPG) e *Twin Delayed Deep Deterministic Policy Gradients* (TD3) e suas aplicações no controle robótico.

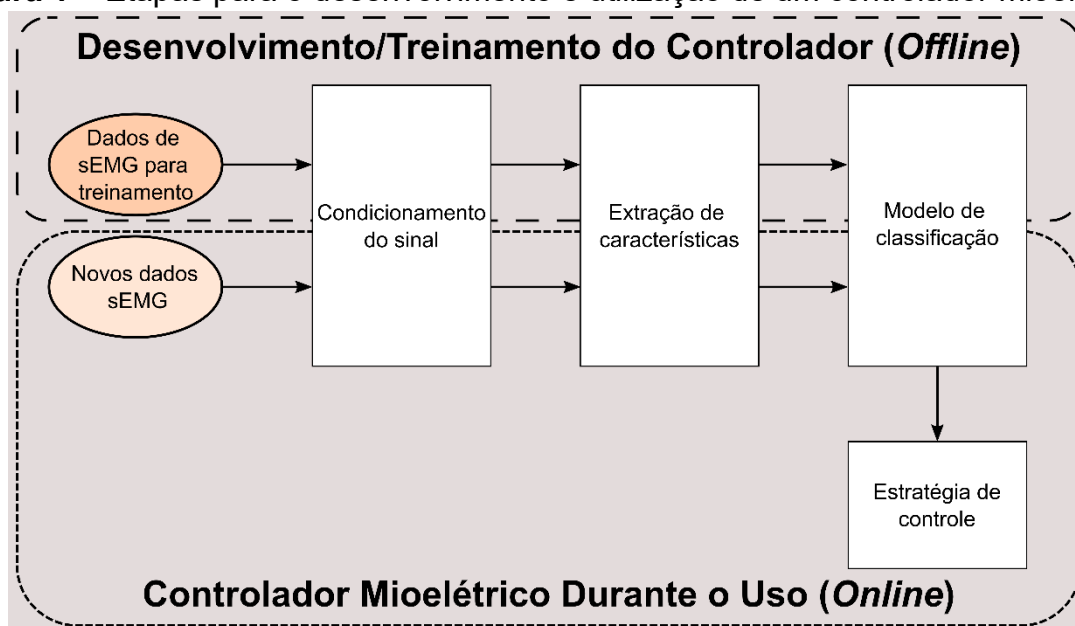
2.1 Sinais Eletromiográficos Superficiais

Segundo MERLETTI; FARINA, (2016) a eletromiografia (EMG) é definida como o registro extracelular da atividade elétrica gerada pelas fibras musculares, onde os eletrodos captam os potenciais elétricos gerados pela ativação das fibras musculares. O termo eletromiografia de superfície (sEMG) é utilizado para representar quando o registro da atividade muscular é feito através do uso de eletrodos posicionados acima da pele, de forma superficial, na região da musculatura que se deseja medir. Uma fibra muscular é eletricamente similar a um axônio de grande diâmetro sem revestimento de mielina, portanto requer alta corrente transmembrana para que um potencial de ação seja transmitido pelo sarcolema. Quando os neurônios motores enviam o potencial de ação para ativação de suas unidades motoras (sinapse neuromuscular), altas correntes são envolvidas no processo de propagação dos potenciais das unidades motoras, que quando somadas geram um campo elétrico extracelular, podendo ser medido pelos eletrodos. O uso desses sinais para controle de prótese vem sendo estudado há muitos anos, como demonstrado por HUDGINS; PARKER; SCOTT (1993) que foi um dos pioneiros no desenvolvimento de controladores mioelétricos, como são conhecidos os sistemas de controle que utilizam sinais de sEMG. Em seu trabalho, foi demonstrado que RNAs do tipo *Multilayer Perceptron* (MLP) podem ser treinadas para mapear padrões não lineares de movimentos de braço e mão, servindo de base para o desenvolvimento desses sistemas.

Para traduzir o sinal de sEMG em comandos para próteses, algumas etapas são necessárias, como pode ser observado na Figura 1. Inicialmente o sinal adquirido com os eletrodos é amplificado, posteriormente um condicionamento do sinal é realizado, a fim de filtrá-lo e separá-lo em janelas. Com o sinal condicionado, suas características são extraídas, e com essas características ele é classificado, ou seja, as

características que foram extraídas são “transformadas” em movimentos desejados por um classificador. Por fim, o sinal traduzido gera um comando executado pela prótese de acordo com a estratégia de controle utilizada (AL-TIMEMY; KHUSHABA; ESCUDERO, 2016; LI, K. *et al.*, 2020; TELLO *et al.*, 2013).

Figura 1 – Etapas para o desenvolvimento e utilização de um controlador mioelétrico



Fonte: Adaptado de LI, K. *et al.*, (2020).

A aquisição do sinal de sEMG é afetada por diversos fatores como posição dos eletrodos, características físicas dos usuários, fadiga muscular e, quando são envolvidos usuários amputados, as características da amputação, ou seja, o sinal varia em função de inúmeras variáveis fisiológicas ou não fisiológicas (ATZORI *et al.*, 2014; VEER, 2015). Devido a essas variações, o desenvolvimento de classificadores generalizados é dificultado. Um dos principais avanços está no desenvolvimento de classificadores para reconhecimento de padrões (PRL), onde os sinais de sEMG são associados a movimentos específicos de pinças, que podem ser executados pela prótese (LI, C. *et al.*, 2018; SCHEME; ENGLEHART, 2011). Nesse tipo de classificador a prótese se limita a realizar movimentos pré-determinados, porém reduz a influência de efeitos adversos da aquisição do sinal, como a posição dos eletrodos, e de características fisiológicas do indivíduo, apresentando efeitos apenas da intensidade do sinal após muitas repetições ou a fatores mecânicos como suor, curtos-circuitos nos eletrodos ou perda de contato com a pele (ATZORI *et al.*, 2014; GUO, W. *et al.*, 2017a).

2.2 Sistemas de Predição de Movimentos Utilizando sEMG

Como apresentado na seção anterior, o estado da arte de controladores mioelétricos consiste no desenvolvimento de controladores associados a classificadores de sinais. Nessa seção será então apresentado como são realizadas cada uma das etapas do sistema apresentado na Figura 1.

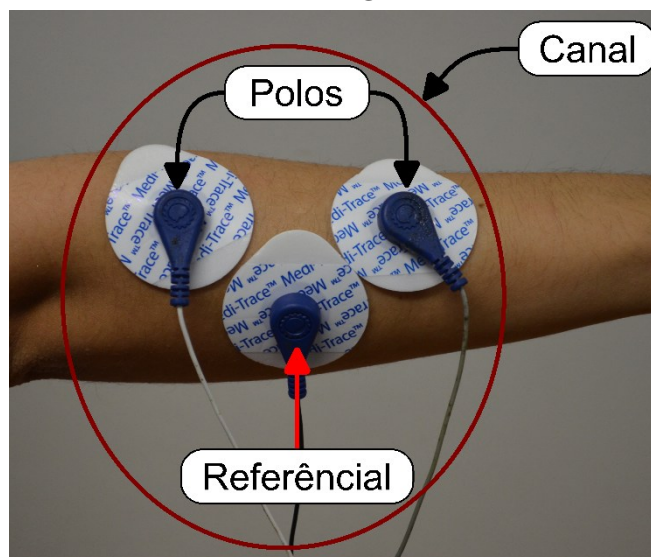
2.2.1 Aquisição e Condicionamento do Sinal sEMG

A primeira etapa do controlador mioelétrico é a aquisição do sinal de sEMG, que deve ser feita com cuidado, pois as propriedades elétricas entre o eletrodo e a pele mudam devido a fatores ambientais, químicos e mecânicos. Os principais fatores que interferem na medição do sinal são: artefatos de movimento causados pela variação ou perda do contato da pele com o eletrodo; movimentos dos cabos que consequentemente levam a variação da corrente parasita de fios soltos carregados; interferência das linhas de energia externas associadas à flutuação da impedância de contato; flutuações da polarização do eletrodo e da distribuição de carga na pele e nos cabos de conexão; e mudanças no nível de ruído devido a mudanças na qualidade do contato. Assim o sistema de aquisição deve ser preparado para lidar com esses distúrbios garantindo a qualidade do sinal medido (MERLETTI; FARINA, 2016).

Boa parte dos distúrbios apresentados no sinal são causados pelo contato incorreto entre o eletrodo e a pele. Visto isso, é necessário selecionar adequadamente o eletrodo a ser utilizado. Os eletrodos de sEMG geralmente são feitos de prata e são posicionados na superfície da pele que está alinhada com o músculo de interesse (Figura 2). Para adquirir o sinal é preciso que o eletrodo possua dois polos posicionados nas extremidades do músculo e um terceiro seja utilizado como referencial, onde cada conjunto de polos é considerado como um canal (MERLETTI; FARINA, 2016).

Dois tipos de eletrodos são utilizados, os eletrodos monopolares (Figura 2) e os eletrodos bipolares. Nos eletrodos monopolares cada eletrodo possui um polo, para adquirir o sinal, utilizam-se três eletrodos, podendo ser posicionados com diferentes distâncias entre si, já o eletrodo referencial pode ser o mesmo para vários canais. Nos eletrodos bipolares os dois polos e o referencial se encontram no mesmo empacotamento, o que facilita o posicionamento, mas limita a distância entre cada um dos polos do eletrodo (SIMAO *et al.*, 2019).

Figura 2 – Exemplo do posicionamento dos eletrodos para aquisição de sinais de sEMG



Fonte: Elaborado pelo autor.

Além do número de polos os eletrodos também podem variar de acordo com a sua interface de contato com a superfície da pele. Segundo LI, K. et al., (2020), MAYOR et al., (2017) e SIMAO et al., (2019) a interface entre a pele e o eletrodo é de suma importância para obter um sinal de qualidade. Atualmente, existem duas opções de eletrodos, os de interface de gel de cloreto de prata ou os secos. No primeiro o eletrodo é envolto com um gel condutivo que ficará em contato com a pele do usuário e garantirá o contato entre a pele e o eletrodo. No segundo caso, o eletrodo é pressionado diretamente contra a pele do indivíduo. Assim eletrodos secos são mais suscetíveis a deslocamentos a se deslocarem do local posicionado. Independentemente do tipo de interface a ser utilizada, a região em que serão posicionados os eletrodos deve ser limpa com álcool e passar pelo processo de tricotomia.

Após escolher o tipo de eletrodo a ser utilizado, outro parâmetro importante é a frequência de aquisição dos dados, necessária para captar as variações do sinal de sEMG. Na literatura não existem valores padrões de frequência de aquisição utilizada. São encontrados trabalhos que utilizaram uma frequência amostral de 8 kHz (REHMAN *et al.*, 2018; RYAIT; ARORA; AGARWAL, 2012), tanto como 2,048 kHz (ARANCETA-GARZA; LAKANY; CONWAY, 2013), outros que trabalham com 2 kHz (GREGORI; CAPUTO; GIJSBERTS, 2018; GUO, W. *et al.*, 2017a; KRASOULIS *et al.*, 2017; LIANG PENG *et al.*, 2013; NAIK; AL-TIMEMY; NGUYEN, 2016), assim como 1,5 kHz (SEGIL; WEIR, 2015); 1,024 kHz (ALY; YOUSSEF; FATHY, 2018; BETTHAUSER *et al.*, 2016; TANG; YU; CANG, 2016). Entretanto a grande maioria dos trabalhos utilizam

a frequência de 1 kHz (AL-TIMEMY; KHUSHABA; ESCUDERO, 2016; AZAB; AR-VANCH; MIHAYLOVA, 2017; GUO, W. *et al.*, 2017b; HUANG *et al.*, 2018; MAYOR *et al.*, 2017; REATEGUI *et al.*, 2013; TELLO *et al.*, 2013; VEER, 2015). As frequências menos utilizadas são de 0,2 kHz, (LI, C. *et al.*, 2018); e 0,1 kHz (YANG; GU; *et al.*, 2017). Apesar da variação nos valores de frequência amostral, é importante ressaltar que 95% da potência do sinal de sEMG se concentra nas frequências de 5 até 400 ou 500 Hz, assim aplicando o teorema de amostragem de Nyquist–Shannon, é necessária uma frequência amostral mínima entre 800 e 1000 Hz (LI, W.; SHI; YU, 2021).

O próximo fator influente na montagem do sistema de aquisição de sinal é a posição dos canais e observam-se alguns padrões. Na maioria dos trabalhos, os eletrodos são colocados de forma equidistante na superfície próxima a posição do músculo de interesse, pois o posicionamento dos eletrodos não precisa estar exatamente sobre a musculatura de interesse, visto que uma pequena variação não acarreta mudanças significativas no desempenho dos classificadores (BAKSHI; MANJUNATHA; KUMAR, 2018; MAYOR *et al.*, 2017; YANG; GU; *et al.*, 2017).

Uma vez captado pelo eletrodo, o sinal de sEMG então passa por um amplificador de sinal e então é filtrado e janelado (LI, K. *et al.*, 2020; RYAIT; ARORA; AGARWAL, 2012; SIMAO *et al.*, 2019). Quando se faz o processo de filtragem do sinal, deve-se ter cuidado ao escolher o filtro que retirará os ruídos, uma vez que o sinal de sEMG apresenta uma razão sinal-ruído muito baixa. Logo, a utilização de determinados filtros pode resultar na perda de dados significativos do sinal (WARIS *et al.*, 2014).

Para escolha adequada dos filtros utilizados deve-se conhecer as fontes de ruído do sinal. A primeira fonte de ruído é proveniente dos próprios componentes eletrônicos utilizados na captação do sEMG, podendo estar presentes com frequência de 0 até elevados kHz. Para eliminar esse tipo de ruído deve-se utilizar eletrodos com componentes que geram poucos ruídos no sistema (LI, K. *et al.*, 2020). A segunda fonte de ruído dos sinais de sEMG são os distúrbios eletromagnéticos causados pela rede elétrica. Para isso, é utilizado um filtro do tipo *notch* que retira as frequências de 50 e 60 Hz, que são as frequências utilizadas para transmissão de energia elétrica (LI, K. *et al.*, 2020; SIMAO *et al.*, 2019). A última fonte de ruído vem dos distúrbios gerados pelos artefatos de movimento, pelos movimentos dos cabos, pelas flutuações da impedância de contato, pelas flutuações da polarização do eletrodo e distribuição de carga na pele e nos cabos de conexão, além de mudanças na qualidade do contato

(MERLETTI; FARINA, 2016). Boa parte desses ruídos são presentes a baixas frequências, de 0 a 20 Hz, e por isso os sinais de sEMG geralmente passam por filtros passa-faixa de frequência de 20 a 500 Hz, assim eliminando os ruídos de movimento e assegurando a preservação da potência do sinal (LI, K. et al., 2020; LI, W.; SHI; YU, 2021; NAZMI et al., 2016; SIMAO et al., 2019).

Apesar de a retirada das duas primeiras fontes de ruído serem relativamente simples, a última fonte de ruído não há consenso na literatura de como deve ser retirada. Alguns autores retiram apenas os ruídos provenientes da rede elétrica aplicando o filtro do tipo *notch* de 50 ou 60 Hz (ARANCETA-GARZA; LAKANY; CONWAY, 2013; HERLE, 2018; MAYOR et al., 2017), ou do filtro de Hampel (GREGORI; CAPUTO; GIJSBERTS, 2018; KRASOULIS et al., 2017). Quando são aplicados filtros para retirar a terceira fonte de ruído muitas combinações e tipos de filtros são utilizados. Encontram-se combinações como a aplicação do filtro *notch* com filtros de passa alta de 5 Hz e passa baixa de 309 Hz (TELLO et al., 2013); filtros de passa faixa de 10 a 500 Hz, 10 a 400 Hz, 20 a 450 Hz, 20 a 500 Hz, 30 a 450 Hz e 100 a 500 Hz (ALY; YOUSSEF; FATHY, 2018; AMSUSS et al., 2014; BETTHAUSER et al., 2016; DING et al., 2014; FAJARDO; LEMUS; ROHMER, 2015; GUO, W. et al., 2017b; REHMAN et al., 2018; STURMA et al., 2015; TANG; YU; CANG, 2016). O filtro *Butterworth* também é utilizado como filtro passa alta, passa baixa ou passa faixa. Ele pode ser utilizado como passa alta de 20 Hz, frequência em que se encontra os ruídos de terceiro tipo, e com 40 Hz (LIU et al., 2016; NAIK; AL-TIMEMY; NGUYEN, 2016; PAN; HARMODY; HUANG, 2018; YOUNG, A. J. et al., 2013), também como passa baixa de fase zero, de primeira ordem com corte de 1 Hz e de segunda ordem com corte de 5 Hz (ATZORI et al., 2014; NAZEMI; MALEKI, 2014; PAN; HARMODY; HUANG, 2018). Como filtro de passa faixa, são utilizados filtros *Butterworth* de quarta ordem com banda de frequência de 20 a 400 Hz (LIZHI PAN et al., 2013). Alguns autores chegam a desenvolver novas técnicas para tentar retirada dos ruídos, como REATEGUI et al. (2013), que utilizou a técnica de variação total de retirada de ruído o tratando como um problema de otimização baseado na função de custo, apresentando resultados satisfatórios na mitigação do ruído quando comparados com os filtros passa alta, passa baixa e passa faixa utilizados.

Além de filtrar o sinal, alguns autores o retificam e o suavizam, melhorando a qualidade. A suavização é feita utilizando a raiz média quadrática (RMS) do sinal, ou

fazendo sua normalização com base no maior valor absoluto medido (MAV), técnica também utilizada para generalizar os classificadores para PRL. Em alguns casos a redução da frequência amostral também é utilizada para retirar as frequências mais elevadas de ruído e reduzir a quantidade de dados, melhorando o treinamento do classificador (LIU *et al.*, 2016; PAN; HARMODY; HUANG, 2018; SEGIL; WEIR, 2015). Também podem ser utilizadas técnicas mais avançadas para redução do ruído e da quantidade necessária de eletrodos para se obter elevada acurácia do classificador utilizando a análise de componentes independentes (ICA) (LI, K. *et al.*, 2020). O ICA é uma técnica de separação de fonte cega (BSS) tendo como objetivo decompor o sinal de sEMG medido, de forma instantânea, em componentes independentes. Essa técnica, associada ao algoritmo de decomposição rápida de ICA (também conhecida de lcasto), pode ser utilizada para reduzir o ruído e a quantidade de eletrodos necessários (NAIK; AL-TIMEMY; NGUYEN, 2016).

A última etapa do condicionamento e pré-processamento do sinal de sEMG é o janelamento. O janelamento consiste em subdividir o sinal no tempo em pequenos intervalos, em que esses intervalos serão analisados pelo classificador. Nessa etapa deve-se tomar cuidado com o tamanho da janela escolhida, pois uma janela muito longa acarreta o aumento da acurácia de classificação, porém gera atraso entre o comando do usuário e a execução da tarefa pela prótese, já uma janela muito curta pode gerar uma baixa acurácia de classificação, mas não gera atrasos na execução da tarefa (BI; FELEKE; GUAN, 2019).

Como demonstrado por MAYOR *et al.* (2017), quando se utiliza sinais de sEMG para o controle de dispositivos protéticos o tempo entre o usuário enviar o comando e a prótese executá-lo não deve ser maior que 300 ms. Em seu trabalho, foi observado que existe uma relação direta entre o tamanho da janela utilizada e a acurácia do classificador, em que quanto maior fosse o tamanho da janela, melhor o classificador funcionava, apresentando um limite a 300 ms, onde nenhum aumento significativo na acurácia do classificador foi observado. Foi também avaliada a sobreposição entre janelas, observando que quanto maior fosse o tempo de sobreposição, melhor era o desempenho do classificador, entretanto isso afetava diretamente o atraso. No trabalho de ALY; YOUSSEF; FATHY (2018) é também demonstrado que o melhor desempenho do controlador ocorre com uma janela de 250 ms. Assim, ao se escolher a janela de tempo utilizada, deve-se considerar que seu tamanho seja o maior possível,

garantindo elevada acurácia. Entretanto, faz-se necessário o uso da técnica de sobreposição, para reduzir o atraso, evitando que o tempo de sobreposição excessivo eleve o custo computacional.

O tamanho padrão da janela para uma boa performance do controlador mioelétrico para PRL ainda não foi determinado pela literatura em que o tempo da janela e da sobreposição dependem dos equipamentos utilizados e das estratégias de controle (LIU *et al.*, 2016). Pode-se observar na literatura largura de janelas de 70 ms (TANG; YU; CANG, 2016), de 100 ms (ARANCETA-GARZA; LAKANY; CONWAY, 2013; HUANG *et al.*, 2018; PAN; HARMODY; HUANG, 2018; REATEGUI *et al.*, 2013; SEGIL; WEIR, 2015), de 128 ms (AMSUSS *et al.*, 2014), de 150 ms (AL-TIMEMY *et al.*, 2016; AL-TIMEMY; KHUSHABA; ESCUDERO, 2016), de 200 ms (ATZORI *et al.*, 2014; BETTHAUSER *et al.*, 2016; GREGORI; CAPUTO; GIJSBERTS, 2018; GUO, W. *et al.*, 2017a; LIANG PENG *et al.*, 2013; POWAR; CHEMMANGAT, 2017), de 250 ms (LIU *et al.*, 2016; YOUNG, A. J. *et al.*, 2013; YOUNG, AARON J. *et al.*, 2012), de 256 ms (KRASOULIS *et al.*, 2017; NAIK; AL-TIMEMY; NGUYEN, 2016), de 300 ms (GUO, W. *et al.*, 2017b), de 3 s (VEER, 2015) e de 5 s (TELLO *et al.*, 2013).

Já para sobreposição, a grande maioria dos trabalhos utiliza 50 ms de incremento entre uma janela e outra (AMSUSS *et al.*, 2014), assim os valores de sobreposição variam de 20 ms a 225 ms (AL-TIMEMY *et al.*, 2016; AL-TIMEMY; KHUSHABA; ESCUDERO, 2016; AMSUSS *et al.*, 2014; ARANCETA-GARZA; LAKANY; CONWAY, 2013; BETTHAUSER *et al.*, 2016; GREGORI; CAPUTO; GIJSBERTS, 2018; GUO, W. *et al.*, 2017b, a; HUANG *et al.*, 2018; KRASOULIS *et al.*, 2017; NAIK; AL-TIMEMY; NGUYEN, 2016; REATEGUI *et al.*, 2013; TANG; YU; CANG, 2016; TELLO *et al.*, 2013; YOUNG, A. J. *et al.*, 2013; YOUNG, AARON J. *et al.*, 2012), ou não apresentam nenhuma sobreposição (POWAR; CHEMMANGAT, 2017; SEGIL; WEIR, 2015). De maneira geral, observa-se que quanto maior a sobreposição, melhor foi o desempenho do classificador. Essas relações são demonstradas no trabalho de KULWA *et al.*, (2024) em que avaliando diferentes tamanhos de janelas e sobreposições observou que para redes neurais convolucionais (CNN) uma sobreposição de 75% do tamanho da janela de dados utilizada resulta em maior acurácia ao classificador assim como uma janela de 200 ms para situações em que poucos canais são utilizados.

2.2.2 Extração de Características

Após o condicionamento de sinal ser realizado, inicia-se a etapa de extração de características. Três métodos são utilizados para a extração de características: as análises de propriedades do sinal de sEMG no domínio do tempo; no domínio da frequência e no domínio do tempo e da frequência. Cada um desses métodos contém um conjunto de características que são extraídas, podendo ser combinadas ou não, sendo aplicadas a cada janela (BI; FELEKE; GUAN, 2019; LI, K. *et al.*, 2020).

As características no domínio do tempo são calculadas diretamente do sinal de sEMG e seus valores variam em função do tempo (BI; FELEKE; GUAN, 2019). Como o sinal medido não precisa passar por nenhuma transformação antes de ter suas propriedades extraídas, grande parte dos trabalhos que envolvem o uso de dados de sEMG para controle de próteses as utilizam, devido à baixa complexidade computacional necessária para sua implementação (BI; FELEKE; GUAN, 2019; LI, K. *et al.*, 2020; YOUNG, AARON J. *et al.*, 2012). As principais características no domínio do tempo utilizadas são a raiz média quadrática (RMS), o EMG integrado (iEMG), o cruzamento de zero (ZC), o comprimento de onda (WL), a variância (VAR) e o valor médio absoluto (MAV) (BI; FELEKE; GUAN, 2019).

A raiz média quadrática (RMS), indica a intensidade da ativação muscular em cada janela, espera-se que essa propriedade reflita o nível de ativação das unidades motoras (ALY; YOUSSEF; FATHY, 2018; LI, C. *et al.*, 2018; LI, K. *et al.*, 2020; NAZEMI; MALEKI, 2014; POWAR; CHEMMANGAT, 2017; SEGIL; WEIR, 2015; TANG; YU; CANG, 2016; TELLO *et al.*, 2013). O RMS pode ser calculado de forma discreta conforme a Equação (1), onde N é o número total de amostras de uma janela de tempo e x_i são os valores de tensão elétrica de cada amostra.

$$RMS = \sqrt{\frac{1}{N} \cdot \sum_{i=1}^N x_i^2} \quad (1)$$

Assim como o RMS, o valor integrado do EMG (iEMG) também indica a intensidade da ativação muscular, entretanto o RMS modela o sinal como um processo Gaussiano aleatório, enquanto o iEMG é apenas a soma absoluta do sinal, o que explica sua maior utilização em aplicações clínicas em aplicações clínicas (NAZEMI; MALEKI, 2014; POWAR; CHEMMANGAT, 2017; TELLO *et al.*, 2013; VEER, 2015).

Outra característica extraída é o cruzamento de zeros (ZC), que é relativamente simples de ser calculada e por isso é amplamente utilizada. O ZC indica a quantidade de vezes que a amplitude do sinal de sEMG cruza o eixo zero. Fisiologicamente, quando uma contração muscular é sustentada, um maior número de unidades motoras é ativado, por consequência, mais cruzamentos no eixo zero o sinal passa a apresentar. Assim o ZC fornece uma estimativa do número de unidades motoras acionadas de forma simples, e facilmente integrada às demais propriedades extraídas (GUO, S. *et al.*, 2015; GUO, W. *et al.*, 2017a; WARIS *et al.*, 2014). Pode-se calcular o ZC utilizando a Equação (2) onde x é o valor de tensão elétrica de cada amostra, sgn é a função que extrai o sinal do valor analisado, e ε representa o limiar do cálculo, definido empiricamente fim de reduzir a influência de ruídos.

$$ZC = \sum_{i=1}^{N-1} I\{sgn(x_i \cdot x_{i+1}) \cap |x_i - x_{i+1}| \geq \varepsilon\}, \quad I\{x\} = \begin{cases} 1; & se\ x < 0 \\ 0; & se\ x \geq 0 \end{cases} \quad (2)$$

Outra característica muito utilizada do sinal de sEMG é o comprimento de onda (WL). O WL quantifica a complexidade do sinal de sEMG, sendo definido como a soma cumulativa da variação da forma de onda ao longo da janela de tempo da janela analisada (ALY; YOUSSEF; FATHY, 2018; GUO, W. *et al.*, 2017a; LI, C. *et al.*, 2018; NAZEMI; MALEKI, 2014; POWAR; CHEMMANGAT, 2017; TELLO *et al.*, 2013). O WL é calculado pela Equação (3) onde x representa a tensão elétrica de cada amostra do sinal medido e N corresponde ao número de amostras da janela.

$$WL = \frac{1}{N} \sum_{i=1}^{N-1} |x_{i+1} - x_i| \quad (3)$$

A variância (VAR) indica a potência do sinal de sEMG, uma vez que a média do sinal é próxima a zero, refletindo a variabilidade natural do recrutamento das unidades motoras. A VAR corresponde à média quadrática dos desvios das amostras em relação à média da janela; valores baixos indicam que as amostras estão próximas da média (NAZEMI; MALEKI, 2014; POWAR; CHEMMANGAT, 2017; TELLO *et al.*, 2013). Além da variância, pode-se utilizar o desvio padrão da amplitude do sinal, definido como a raiz quadrada da VAR (LI, C. *et al.*, 2018). A variância é calculada conforme a Equação (4), em que N representa o número de amostras da janela, x_i o valor da tensão elétrica de cada amostra e \bar{x} é a média amostral da janela.

$$VAR = \frac{1}{N-1} \cdot \sum_{i=1}^N (x_i - \bar{x})^2 \quad (4)$$

O valor médio absoluto (MAV) é uma característica amplamente utilizada PRL (BI; FELEKE; GUAN, 2019). O MAV é definido como a média dos valores absolutos das amostras em uma janela (GUO, W. *et al.*, 2017a; LI, C. *et al.*, 2018; NAZEMI; MALEKI, 2014; POWAR; CHEMMANGAT, 2017; WARIS *et al.*, 2014). Comparada com as demais características extraídas do domínio do tempo do sEMG, o MAV apresenta os melhores resultados em controladores baseados em separação de classes de movimentos (WARIS *et al.*, 2014). O MAV é definido pela Equação (5), onde N é o número de amostras da janela analisada e x_i o valor de tensão de cada amostra.

$$MAV = \frac{1}{N} \cdot \sum_{i=1}^N |x_i| \quad (5)$$

A mudança do sinal da inclinação (SSC) é outra característica amplamente utilizada no domínio do tempo. Diferentemente do ZC, a SSC indica quantas vezes a inclinação da curva do sinal muda, de decrescente (negativa) para crescente (positiva), ou vice-versa, sem que o sinal necessariamente cruze o eixo (GUO, W. *et al.*, 2017a). A SSC é definida pela Equação (6), em que sgn representa a função que extrai o sinal do valor analisado, ε é o limiar para executar o cálculo, N é o número de amostras da janela, x_i e x_{i-1} representam a tensão elétrica medida em cada amostra e $I\{x\}$ é definido pela Equação (7).

$$SSC = \sum_{i=1}^{N-1} I\{[sgn(x_i - x_{i-1}) \cdot sgn(x_i - x_{i+1})] \cap |x_i - x_{i-1}| \geq \varepsilon \cap |x_i - x_{i-1}| \geq \varepsilon\} \quad (6)$$

$$I\{x\} = \begin{cases} 1; & \text{se } x > 0 \\ 0; & \text{se } x \leq 0 \end{cases} \quad (7)$$

Outras propriedades temporais são também avaliadas na literatura como a energia total (ES), a função da inclinação do sinal, a razão de energia (ER), a correlação de concordância (CC), a amplitude de Wilson (WAMP), que avalia a coincidência entre duas variáveis do sinal, o coeficiente de esparsidade (SC), o valor absoluto médio de inclinação (MAVS), a integral simples quadrática (SSI), a análise de flutuação de tendência (DAF), a diferença absoluta do desvio padrão (DASDV), a diferença

absoluta da média (DAMV) e a entropia da amostra (SampEn), embora sejam menos frequentes na literatura (ALY; YOUSSEF; FATHY, 2018; BAKSHI; MANJUNATHA; KUMAR, 2018; DING *et al.*, 2014; GUO, S. *et al.*, 2015; HERLE, 2018; HUDGINS; PARKER; SCOTT, 1993; NAZEMI; MALEKI, 2014; VEER, 2015; WARIS *et al.*, 2014).

As características no domínio do tempo apresentam variadas combinações para aplicação em classificadores como relatado na literatura. Resultados satisfatórios de acurácia são apresentados combinando o MAV com ZC, SSC e WL (GUO, W. *et al.*, 2017a; HUANG *et al.*, 2018; LIANG PENG *et al.*, 2013; YOUNG, A. J. *et al.*, 2013); ou com ZC, VAR e WAMP (WARIS *et al.*, 2014); ou com MAVS, ZC, SSC e WL (HUDGINS; PARKER; SCOTT, 1993). O RMS também é combinado de diversas formas, encontrando-se combinações do RMS com STD (AZAB; ARVANCH; MIHAYLOVA, 2017); com ZC, SSC, WL (AMSUSS *et al.*, 2013, 2014); com SD, VAR, SSI e iEMG (VEER, 2015); ou uma combinação mais abrangente com o iEMG, o MAV, WL, ES, ER, CC (NAZEMI; MALEKI, 2014). A VAR também é combinada com WL, iEMG, ZC, SSC e WAMP (FAJARDO; LEMUS; ROHMER, 2015).

Diferentemente das características extraídas no domínio do tempo, as características no domínio da frequência utilizam a densidade da potência do espectro (PS) do sinal sendo computadas por métodos paramétricos e utilizando a transformada rápida de Fourier (FFT) (BI; FELEKE; GUAN, 2019; LI, K. *et al.*, 2020). Assim, extrair características no domínio da frequência requer um custo computacional elevado, o que limita a sua aplicação prática (BI; FELEKE; GUAN, 2019). As características do domínio da frequência mais utilizadas são os coeficientes de autorregressão (AR), o espectro de potência (PS), a frequência média (MNF), a frequência mediana (MDF) e a razão de frequências (FR) (BI; FELEKE; GUAN, 2019; LI, K. *et al.*, 2020).

A MDF é definida como o valor de frequência que divide o PS em duas metades (RYAIT; ARORA; AGARWAL, 2012). Em alguns trabalhos se utiliza a frequência mediana modificada (MMDF), que indica a frequência onde o espectro de potência dividida em duas regiões de soma de amplitude igual (ALY; YOUSSEF; FATHY, 2018). Já a MNF, corresponde à média das frequências obtidas no PS, entretanto alguns trabalhos optam por utilizar a frequência média modificada (MMNF), que é a soma do produto das amplitudes do PS por sua frequência dividida pela soma total da intensidade na janela, resultando em uma média ponderada (ALY; YOUSSEF; FATHY, 2018). Os AR tratam o sinal como um modelo em que cada amostra é uma combinação linear

das amostras prévias mais um erro de ruído branco (ALY; YOUSSEF; FATHY, 2018). A FR relaciona os componentes de baixa frequência do sinal com os componentes de alta frequência (ALY; YOUSSEF; FATHY, 2018).

Dentre as combinações das características do domínio do tempo com as características do domínio da frequência, podem ser observadas combinações do RMS, com a função de inclinação, ES, e MNF (RYAIT; ARORA; AGARWAL, 2012). Também é observada a combinação dos coeficientes da transformada de Fourier (FT) com o MAV, WL, VAR, SSC e ZC (BETTHAUSER *et al.*, 2016). Os AR de quarta ordem podem ser combinados com a RMS, o MAV, a VAR, o WL e o ZC (NAIK; AL-TIMEMY; NGUYEN, 2016) ou com o MAV, o WL e a VAR em escala logarítmica (KRASOULIS *et al.*, 2017) e alguns autores os combinam com o MAV, VAR, VAR em escala logarítmica, RMS, DASDV, DAMV e iEMG (BAKSHI; MANJUNATHA; KUMAR, 2018). Os AR de sexta ordem também são utilizados, porém combinados com o MAV, o ZC, a SSC e o WL (GUO, W. *et al.*, 2017a). Observa-se que as características do domínio da frequência de AR de sexta ordem, FR, MMNF e MMDF são combinadas com as características temporais de RMS, WL e WAMP (ALY; YOUSSEF; FATHY, 2018).

O último método de extração de características utilizado na literatura são as características extraídas no regime do tempo e da frequência, essas características são utilizadas para identificar mudanças fisiológicas, como por exemplo, a fadiga muscular. Quando são analisados tais tipos de mudanças, avalia-se a energia do sinal tanto no domínio do tempo como no domínio da frequência. Devido à complexidade envolvida nas transformações para o regime da frequência, tais propriedades demandam elevado custo computacional para serem calculadas. As principais características do domínio do tempo e frequência utilizam as transformadas curtas de Fourier (STFT) e as transformadas de *wavelet* (WT) (BI; FELEKE; GUAN, 2019; LI, K. *et al.*, 2020).

As características temporais, combinadas com características do domínio do tempo e da frequência observadas são o RMS com MAV, MAVS, ZC, SSC, WL, histograma, transformada marginal de *wavelet* (wDWT) e a normalização da combinação de todas as características utilizadas (ATZORI *et al.*, 2012). Existem também trabalhos que se utilizam das transformadas de *wavelet* para reconstruírem o sinal de sEMG, removendo ruídos e posteriormente do sinal reconstruído extraem características temporais como RMS, ZC e DFA (GUO, S. *et al.*, 2015).

Quando analisamos como cada característica extraída influencia no desempenho do classificador, observamos que os estudos apresentam conclusões distintas, o que mostra a dependência das escolhas da característica extraídas com a aplicação do controlador. Comparando várias características que podem ser extraídas DING et al., (2014) observou que utilizando como classificador uma Máquina de Vetores de Suporte (SVM), para classificar 8 movimentos, e comparando as características de densidade média espectral (MPSD), AR, coeficientes de *wavelet* (WC), MAV, WL, ZC, SSC e SC, observou-se que a maior acurácia, aproximadamente 93%, foi obtida quando utilizado o SC. Essa análise leva em conta apenas a região transiente do sinal de sEMG, desconsiderando as características estacionárias, o que levou ao melhor desempenho do classificador. Em contraponto, no classificador desenvolvido por VEER, (2015) somente com a combinação das 5 características utilizadas (RMS, SD, VAR, SSI e iEMG) é que se obteve desempenho satisfatório na predição de força desejada utilizando o sinal de sEMG.

Em outro trabalho desenvolvido por LIU et al., (2016) foram avaliadas seis características extraídas do sinal de sEMG para o PRL, sendo elas o MAV, o MAV1, a VAR, a RMS, o WL, os quatro primeiros AR e a transformada empacotada de wavelet (WPT). Foi então observada que o MAV e o WPT obtiveram as melhores acurácias de classificação (75,85% e 75,12% respectivamente), entretanto o WPT consome um maior tempo de treinamento do classificador, já as combinações lineares do MAV com o RMS apresentam uma maior robustez do classificador.

O trabalho de ALY; YOUSSEF; FATHY, (2018) demonstra que combinar AR de sexta ordem com RMS, WL e WAMP apresentam os resultados mais promissores para a aplicação do PRL, pois apresentaram elevada acurácia com baixa complexidade computacional. Quando adicionadas às características do sinal no regime da frequência os melhores resultados foram obtidos utilizando AR de sexta ordem ao invés de quarta ordem, assim como utilizando características do domínio do tempo, entretanto ao inserir as propriedades do domínio da frequência a complexidade computacional é aumentada, o que pode prejudicar o desempenho de aplicações em tempo real.

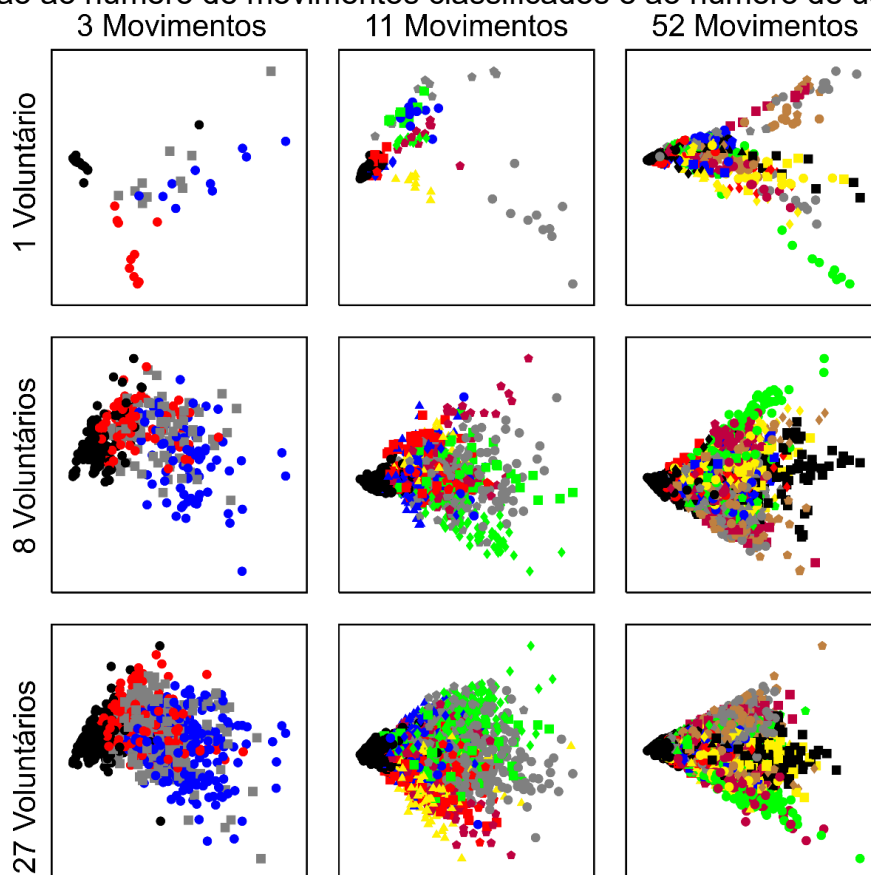
No trabalho de MAYOR et al., (2017) ao estudar características dos domínios do tempo, da frequência e tempo-frequência, observou que quanto maior o número de características extraídas maior era a acurácia do classificador obtida, entretanto, a partir de seis características o aumento de desempenho do classificador não era

significativo. No trabalho de HERLE, (2018) as características de AR, SampEn, MAV, MAV1, RMS, WL, ZC e SSC foram analisadas com o classificador LDA e observou-se que o melhor desempenho ocorria quando era combinado os AR de sexta ordem com SampEn, SSC, ZC, MAV, MAV1 e WL.

Assim é possível concluir que todas as características podem ser combinadas das mais diversas formas e a escolha geralmente se baseia em um processo de tentativa e erro (BI; FELEKE; GUAN, 2019; LI, K. *et al.*, 2020). Independentemente da combinação utilizada, quando deseja-se fazer um controlador mioelétrico utilizando o PRL faz-se necessário que as características combinadas sejam capazes de agrupar cada movimento a ser classificado em uma região única no espaço amostral das características. Como demonstrado na Figura 3, utilizando o método da análise dos componentes principais (PCA) observa-se que, quanto maior o número de movimentos classificados, ou a quantidade de pessoas que se avalia o sinal, pior é a separação dos movimentos, necessitando assim que sejam extraídas muitas características, aumentando a complexidade do classificador e conseqüentemente o custo computacional (ATZORI *et al.*, 2012; MAYOR *et al.*, 2017).

Devido à necessidade de aumentar a quantidade de características extraídas, alguns autores propõem um pré-processamento das propriedades do sinal antes de enviá-los ao classificador. Um dos métodos mais utilizados é o do PCA, pois ele é capaz de reduzir o número de dados de entrada do classificador a um conjunto de autovalores de maior influência no espaço das características, resultando em maior acurácia do classificador, uma vez que características que pouco contribuem para a separação dos movimentos são retiradas (ATZORI *et al.*, 2012; LI, C. *et al.*, 2018; LIANG PENG *et al.*, 2013). Essa técnica é necessária para aplicações em tempo real que necessitam de muitas características para prever os movimentos desejados, como demonstrado por LI *et al.* (2018), que reduziram o número de características necessárias para o classificador de 32 para apenas 8, utilizando os 8 primeiros componentes principais do PCA, reduzindo assim a complexidade computacional do controlador por eles desenvolvido sem afetar o desempenho.

Figura 3 – Representação da separação das características do sinal de sEMG em relação ao número de movimentos classificados e ao número de usuários



Fonte: Adaptado de ATZORI et al., (2012).

Por fim, autores propuseram novos métodos de extração de características para obter um melhor desempenho, combinando de formas inovadoras com as características tradicionalmente extraídas. Nos trabalhos de AL-TIMEMY et al., (2016) e AL-TIMEMY; KHUSHABA; ESCUDERO, (2016) foi proposto um novo método de extração de característica, por ele nomeado Descritores Espectrais de Potência no Domínio do Tempo (TD-PSD), que consiste em um conjunto de características temporais que descreve o espectro de potência do sinal diretamente do domínio do tempo, sem a necessidade de FFT. Observou-se que o novo método de extração de características quando aplicado ao classificador apresentou um baixo erro de predição, porém uma grande variabilidade, sendo possível construir um controlador robusto com até 6 movimentos classificados. Entretanto em situações de ATVds, é observado um erro superior a 60% na classificação se não são utilizados sinais de variadas forças aplicadas no treinamento. Quando utilizados sinais de variadas intensidades de força, o erro caiu para menos de 10%, evidenciando que o sinal de sEMG varia com a força, uma vez que a técnica de TD-PSD depende da direção do espectro de potência do sinal,

que está diretamente relacionado a força muscular da contração. Além desse trabalho, HUANG et al., (2018) utilizou quatro tradicionais características temporais (MAV, WL, ZC, SSC) para criar um modelo de mistura gaussiana (GMM), a fim de retirar as sinergias musculares que são do sinal de sEMG, para melhorar o desempenho de classificadores de PRL e de modelos de regressão.

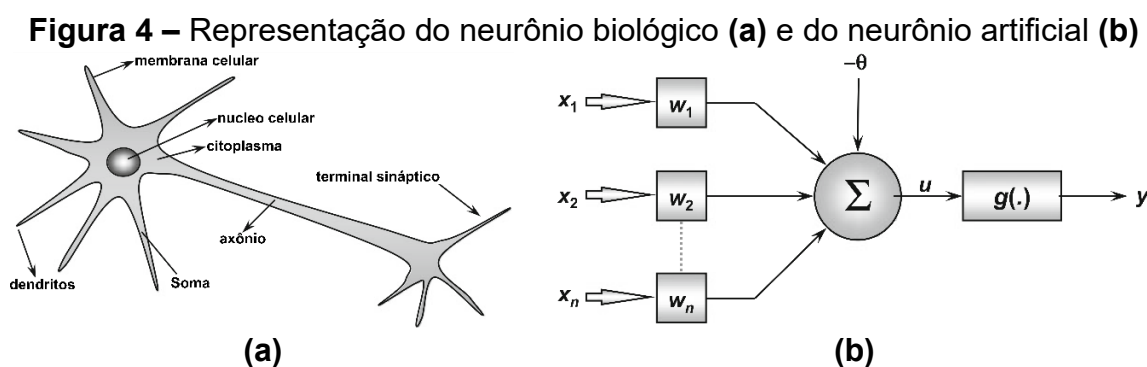
Além da influência nos sistemas de classificação, o tipo de característica extraída também impacta significativamente no desempenho de regressores. No trabalho de JIANG et al., (2025) foi evidenciado que para previsões contínuas de ângulos das articulações dos dedos, faz-se necessário o uso de características temporais e de frequência. Em seu teste de importância de variável, observou que as características temporais foram mais influentes na resposta do modelo do que as características de frequência, representando uma influência de 40%, 52,6% e 29,2% nos resultados de desempenho no coeficiente de correlação, no coeficiente de determinação e no erro médio quadrático. Entretanto, ao avaliar a influência no desempenho das características por canal de sEMG, observa-se que em alguns canais ambos os tipos de características influenciavam no resultado, demonstrando que tanto características temporais quanto de frequência são combinadas pelo modelo para executar as previsões contínuas de ângulos.

2.2.3 Classificação dos Sinais e Controle do Dispositivo

Após obter as características do sinal de sEMG de cada janela, a próxima etapa é transformá-las em movimentos desejados, para isso utiliza-se um classificador. Para traduzir o sinal de sEMG observam-se três abordagens principais, a classificação de movimentos discretos, estimativa de movimento de junta contínua e estimativa da rigidez de junta. Para o controle de próteses são utilizadas as classificações de movimentos discretos ou a estimativa de movimento de junta contínua (LI, K. *et al.*, 2020). Controladores mioelétricos utilizam técnicas de PRL para classificar os movimentos, associando as características extraídas do sinal de sEMG com um conjunto específico de posturas, ou movimentos desejados. Os principais métodos de classificação de sinais utilizados são: análise discriminante linear (LDA), modelos de mistura Gaussiana (GMM), máquinas de vetor de suporte (SVM) e redes neurais artificiais (RNA) (BI; FELEKE; GUAN, 2019; LI, K. *et al.*, 2020).

2.2.3.1 Redes Neurais Artificiais

As RNAs são algoritmos computacionais que se inspiram no funcionamento do neurônio biológico para relacionar os dados de entrada com a saída desejada. O neurônio biológico (Figura 4a) consiste em um conjunto de dendritos que recebem os estímulos bioelétricos dos neurônios anteriores que são somados pelo núcleo celular. Quando a soma desses estímulos supera o limiar de ativação, o neurônio é ativado e passa o sinal para o próximo, através do axônio. O neurônio artificial (Figura 4b) funciona de forma semelhante, as entradas fornecidas (x_1, x_2, \dots, x_n) são somadas de forma ponderada por pesos (w_1, w_2, \dots, w_n), quando a soma ponderada das entradas supera o limiar de ativação (θ) é acionado o neurônio artificial conforme a sua função de ativação ($g(\cdot)$) (BRAGA; CARVALHO; LUDERMIR, 2000; DA SILVA *et al.*, 2017).



Fonte: Adaptado de DA SILVA *et al.* (2017).

Matematicamente, o neurônio artificial é representado pela Equação (8), onde a saída (y) de um neurônio corresponde à função de ativação $g(u)$. A função de ativação depende da soma ponderada das entradas x_i pelo seu peso w_i subtraído o limiar de ativação θ do neurônio, sendo o potencial de ativação (u) (BRAGA; CARVALHO; LUDERMIR, 2000; DA SILVA *et al.*, 2017; GOODFELLOW; BENGIO; COURVILLE, 2017; NIELSEN, 2019).

$$y = g(u), \quad u = \sum_{i=1}^n w_i \cdot x_i - \theta \quad (8)$$

Existem várias funções de ativação para um neurônio artificial, e estas são divididas em duas classes, as funções de ativação parcialmente diferenciáveis e as funções de ativação totalmente diferenciáveis. As funções de ativação parcialmente diferenciáveis são funções que em algum ponto sua derivada não existe, essas funções

incluem o degrau unitário simples, o degrau bipolar e a rampa simétrica. Devido à ausência de derivada em determinados pontos, as funções de ativação parcialmente diferenciáveis não são frequentemente utilizadas nos algoritmos de treinamento modernos, sendo preferível o uso de funções de ativação totalmente diferenciáveis (BRAGA; CARVALHO; LUDERMIR, 2000; DA SILVA *et al.*, 2017).

A principal função de ativação totalmente diferenciável é a função logística, definida pela Equação (9). Como demonstrado na Figura 5a, a função logística possui um valor mínimo de 0 e máximo de 1 e sua inclinação depende do valor da constante β , definindo assim a abruptidade da resposta do neurônio (DA SILVA *et al.*, 2017). A variável u é o potencial de ativação do neurônio. Uma forma particular da função logística ocorre quando o valor de β é igual a 1, conhecida como função sigmoideal, sendo essa a sua forma mais utilizada para desenvolvimento de RNAs modernas. Os neurônios que utilizam a função de ativação sigmoideal são conhecidos como neurônios sigmoideais (GOODFELLOW; BENGIO; COURVILLE, 2017; NIELSEN, 2019).

$$g(u) = \frac{1}{1 + e^{-\beta \cdot u}} \quad (9)$$

Outra função utilizada é a função hiperbólica, definida pela Equação (10). Os valores máximos e mínimos da função são -1 e 1, respectivamente (Figura 5b). A variável β é o parâmetro que controla a inclinação da curva, e u é o potencial de ativação do neurônio (DA SILVA *et al.*, 2017).

$$g(u) = \frac{1 - e^{-\beta \cdot u}}{1 + e^{-\beta \cdot u}} \quad (10)$$

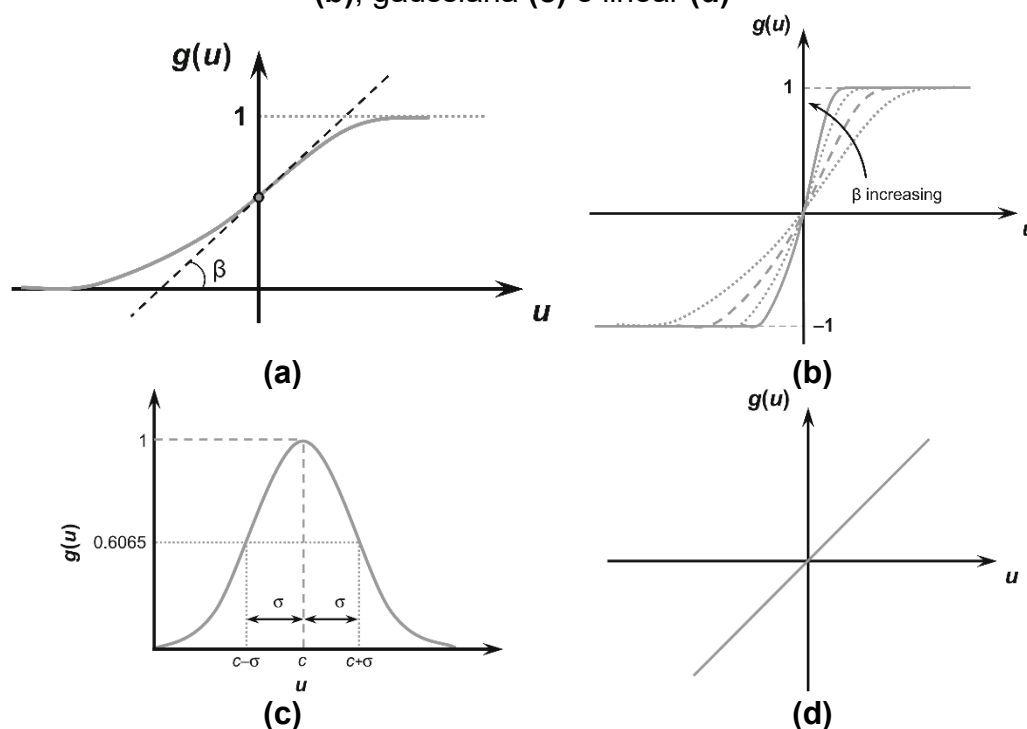
Outra função totalmente diferenciável é a função de ativação Gaussiana (Figura 5c), definida pela Equação (11). Essa função é baseada na distribuição de probabilidade Gaussiana, ou seja, o potencial de ativação é colocado a uma distância relativa à média, assumindo valor máximo 1. A variável c é o centro da função de ativação e a variável σ é o desvio padrão da distribuição Gaussiana, que indica a dispersão da função de ativação. A variável u é o potencial de ativação do neurônio (DA SILVA *et al.*, 2017).

$$g(u) = e^{-\frac{(u-c)^2}{2 \cdot \sigma^2}} \quad (11)$$

A função de ativação utilizada em modelos de ajuste de curva é a função linear (Figura 5d). Essa função de ativação corresponde ao próprio valor do potencial de ativação do neurônio (u), conforme definido na Equação (12) (DA SILVA *et al.*, 2017).

$$g(u) = u \quad (12)$$

Figura 5 – Funções de ativação dos neurônios artificiais: logística (a), hiperbólica (b), gaussiana (c) e linear (d)



Fonte: Adaptado de DA SILVA *et al.*, (2017).

Em aplicações de aprendizado profundo é comum utilizarmos um outro tipo de função de ativação, a retificada linear, onde os neurônios que a utilizam são chamados de Unidades Retificadas Lineares (ReLU). Essa função é similar à função de ativação linear, com a diferença de que seu valor é retificado, isto é, para potenciais de ativação (u) negativos o valor de saída é sempre zero, como representado na Equação (13). A grande vantagem dessa função é que ela possui gradientes elevados e consistentes, garantindo que a segunda derivada seja sempre zero e que a primeira derivada seja sempre 1 para a região ativada da função. Isso assegura melhor desempenho dos gradientes durante o aprendizado quando comparado com funções de ativação que introduzem efeitos de segunda ordem nos gradientes (GOODFELLOW; BENGIO; COURVILLE, 2017).

$$g(u) = \max(0, u) \quad (13)$$

Uma das grandes desvantagens de neurônios ReLU é que eles não conseguem ser otimizados quando os potenciais de ação são menores ou iguais a zero. Para superar essa desvantagem, foram propostas na literatura generalizações dos neurônios ReLU. Uma das generalizações mais utilizadas é substituí-los por neurônios que usam uma função de ativação do tipo *leaky ReLU* (Equação (14)). Nessa função, para potenciais de ativação menores ou iguais a zero a resposta é uma reta com inclinação reduzida, com uma pequena inclinação definida pelo parâmetro α , inferior a 1. Uma alternativa é utilizar funções paramétricas de ReLU (PReLU), em que a inclinação da reta (α), para potenciais negativos, é ajustada durante o treinamento (GOODFELLOW; BENGIO; COURVILLE, 2017).

$$g(u) = \max(0, u) + \alpha \min(0, u) \quad (14)$$

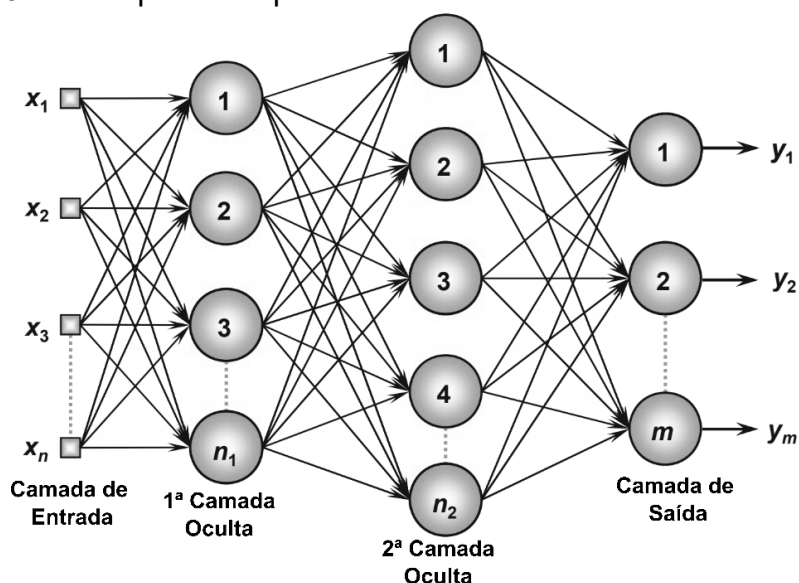
Para problemas de reconhecimento de padrões (PRL) os neurônios da camada de saída geralmente apresentam uma função de ativação do tipo *softmax*. Nessa função de ativação, os potenciais de ativação de todos os neurônios da camada são levados em consideração e assim a soma das funções de ativação de cada um deles é sempre igual a 1, sendo sempre positiva. Nessa função o objetivo é que a camada se comporte como uma função de probabilidade, em que o neurônio mais ativado terá o maior valor enquanto os demais terão seu valor mínimo, no caso do sEMG, corresponde ao movimento desejado (DA SILVA *et al.*, 2017; NIELSEN, 2019). A função *softmax* é definida pela Equação (15), onde l representa a camada que será aplicada à função *softmax*, e j representa o neurônio cuja ativação será calculada, u_j representa o potencial de ativação do neurônio cuja função de ativação é calculada, u_i é o potencial de ativação de todos os neurônios da camada e n o número de neurônios da camada.

$$g(u_j^l) = \frac{e^{u_j^l}}{\sum_{i=1}^n e^{u_i^l}} \quad (15)$$

O neurônio artificial pode ser combinado assim como no neurônio biológico, onde a saída de um neurônio é utilizada como entrada do próximo neurônio, conforme exemplificado na Figura 6 (BRAGA; CARVALHO; LUDERMIR, 2000; DA SILVA *et al.*, 2017).

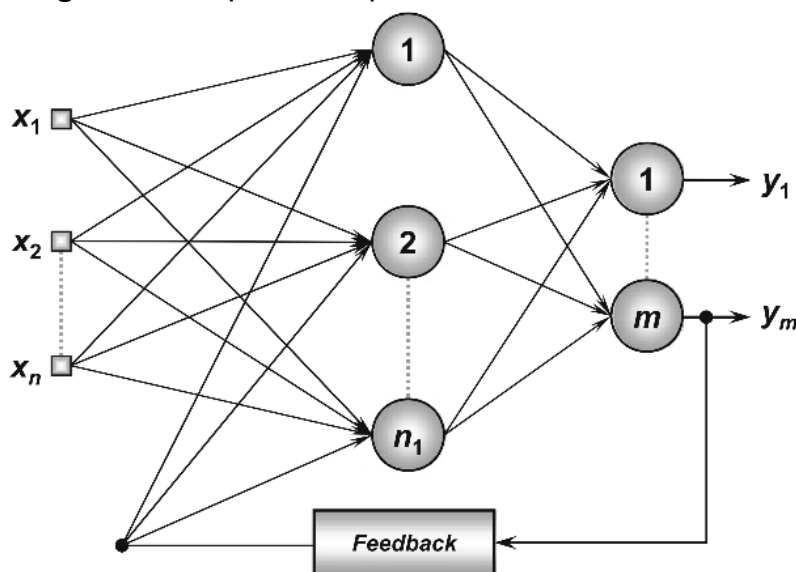
A Figura 6 representa um tipo de combinação dos neurônios artificiais, conhecida como arquitetura de múltiplas camadas de resposta direta, também conhecido como rede de Múltiplas Camadas de Perceptron (*Multi Layer Perceptron – MLP*). Trata-se de apenas um tipo de arquitetura, entretanto vários tipos de arquiteturas podem ser utilizados. Entre as estruturas mais comuns observa-se as redes neurais de recorrência (RNN) como apresentadas na Figura 7 as respostas também são utilizadas como entrada da RNA, empregada na predição de eventos futuros, como séries temporais (DA SILVA *et al.*, 2017).

Figura 6 – Exemplo da arquitetura de uma RNA com 2 camadas ocultas



Fonte: Adaptado de DA SILVA *et al.*, (2017).

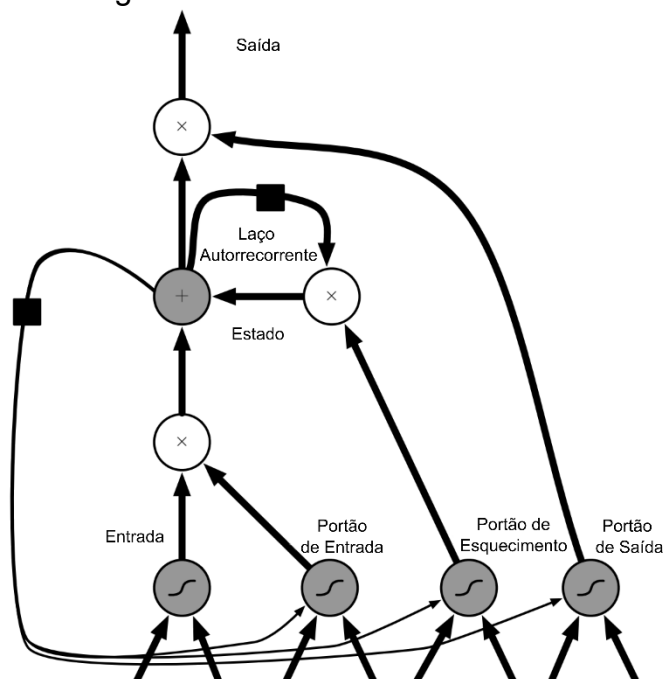
Figura 7 – Arquitetura típica de RNA com recorrência



Fonte: Adaptado de DA SILVA *et al.*, (2017).

Em arquiteturas de RNNs é comum utilizarmos arquiteturas mais complexas desenvolvidas especialmente para fazer a retroalimentação dos dados de saída à camada de entrada. Um dos tipos de arquitetura muito utilizadas são as células de Memória de Longo e Curto Prazo (*Long Short-Term Memory* – LSTM). Conforme ilustrado na Figura 8, a primeira grande contribuição dessas células foi o emprego dos laços autorrecorrentes, que permitem que os gradientes sejam propagados por longos períodos. O laço de autorrecorrência exige neurônios ocultos com parâmetros ajustados durante o treinamento, assegurando o ajuste adequado da escala de integração temporal, de acordo com a sequência de entrada apresentada (GOODFELLOW; BENGIO; COURVILLE, 2017).

Figura 8 – Diagrama do funcionamento de uma célula LSTM



Fonte: Adaptado de GOODFELLOW; BENGIO; COURVILLE, (2017).

Outra característica importante é a presença de portões de entrada, saída e “esquecimento” para o controle do fluxo de informação dentro da célula. O portão de “esquecimento” para o controle do fluxo de informação dentro da célula. O portão de “esquecimento” é controlado por um neurônio ($f^{(t)}$), geralmente sigmoide, definindo a importância da informação de realimentação ($s^{(t-1)}$) no laço autorrecorrente para determinação do estado atual da célula ($s^{(t)}$). O neurônio responsável por controlar o portão das entradas externas ($g^{(t)}$) ajusta o peso que as entradas atuais possuem no estado atual da célula LSTM. Por fim, o portão de saída é o neurônio ($q^{(t)}$) que ajusta o peso do estado atual na saída da célula LSTM ($h^{(t)}$) (GOODFELLOW; BENGIO;

COURVILLE, 2017). A Equação (18) representa as equações de resposta da célula LSTM, onde σ representa a função de ativação dos neurônios dos portões de controle, $\mathbf{x}^{(t)}$ o vetor de entrada da célula e b , \mathbf{U} e \mathbf{W} representam o viés e as matrizes dos pesos das entradas dos laços recorrentes dos neurônios dos portões de controle, respectivamente.

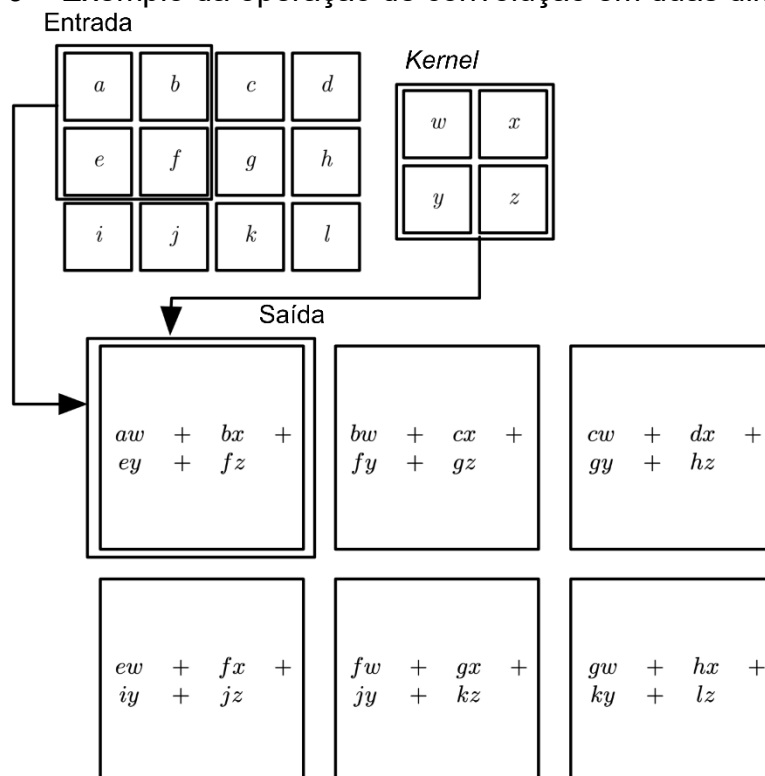
$$\begin{cases} h^{(t)} = \tanh(s^{(t)}) q^{(t)} \\ s^{(t)} = f^{(t)} s^{(t-1)} + g^{(t)} \sigma(b + \mathbf{U}\mathbf{x}^{(t)} + \mathbf{W}\mathbf{h}^{(t-1)}) \\ f^{(t)} = \sigma(b^f + \mathbf{U}^f \mathbf{x}^{(t)} + \mathbf{W}^f \mathbf{h}^{(t-1)}) \\ g^{(t)} = \sigma(b^g + \mathbf{U}^g \mathbf{x}^{(t)} + \mathbf{W}^g \mathbf{h}^{(t-1)}) \\ q^{(t)} = \sigma(b^o + \mathbf{U}^o \mathbf{x}^{(t)} + \mathbf{W}^o \mathbf{h}^{(t-1)}) \end{cases} \quad (16)$$

Uma outra arquitetura de rede muito utilizada é a das Redes Neurais Convolucionais (*Convolutional Neural Network – CNN*). As CNN são uma rede de neurônios que ao invés de fazer operações matriciais, determinam o valor de saída por operações de convolução em pelo menos uma de suas camadas. As operações de convolução são uma separação das entradas em pequenos grupos (*kernels*) que fazem a soma ponderada desse pequeno conjunto de dados, como pode-se observar na operação de convolução de um dado de entrada em duas dimensões, apresentado na Figura 9. A principal vantagem desse sistema é que ele envolve três conceitos fundamentais para o aprendizado de máquina (*Machine Learning – ML*), as interações esparsas, o compartilhamento de parâmetros e as representações equivariantes, que alinhadas reduzem a necessidade de capacidade computacional durante o treinamento. Devido a essas características, as CNNs são muito utilizadas quando a posição da entrada tem significância para as respostas, como em séries temporais, classificação de imagens entre outros problemas (GOODFELLOW; BENGIO; COURVILLE, 2017).

Uma típica camada de CNN consiste em três estágios cruciais (Figura 10), as camadas de convolução paralelas, que através das operações de convolução produzem um conjunto de potenciais de ação. Depois da etapa de convolução faz-se necessário que os potenciais de ação passem pela camada da função de ativação (chamada de estágio de detecção). Por fim, a camada de função de *Pooling*, que modifica a saída para as próximas camadas. O *Pooling* mais utilizado é o *max pooling* que consiste na extração do valor máximo entre um determinado conjunto de saídas das operações de convolução. A combinação dessas camadas contribui para que o

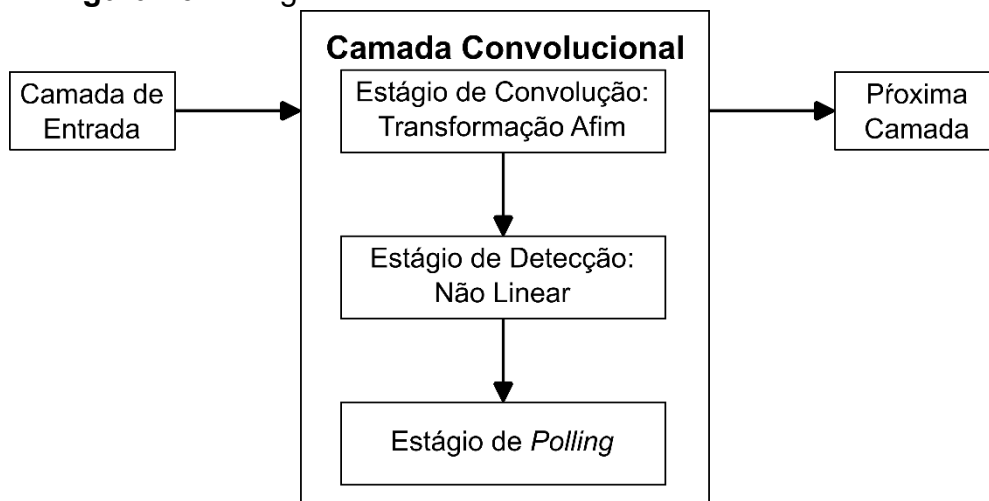
modelo represente invariância à translação da entrada, ou seja, além de conseguir extrair características específicas através das operações de convolução, a combinação dessas operações com as funções de *pooling* auxiliam na identificação da translação da entrada, garantindo que o modelo resulte em uma mesma resposta para características que apenas mudaram de posição com o tempo (GOODFELLOW; BENGIO; COURVILLE, 2017).

Figura 9 – Exemplo da operação de convolução em duas dimensões



Fonte: Adaptado de GOODFELLOW; BENGIO; COURVILLE, (2017).

Figura 10 – Diagrama de blocos de uma camada convolucional



Fonte: Adaptado de GOODFELLOW; BENGIO; COURVILLE, (2017).

A determinação do número de neurônios de cada camada da RNA, assim como sua arquitetura, determina o desempenho do modelo. De maneira geral algumas regras devem ser previamente seguidas. Inicialmente o número de neurônios da camada de entrada é igual ao número de dados que serão fornecidos a rede, no caso do sEMG o número de características extraídas. O número de neurônios da camada de saída é determinado pelo número de respostas que a RNA deve dar para um determinado sistema, por exemplo, os movimentos desejados de pinça. A grande dificuldade na elaboração de modelos de RNA é a determinação da quantidade de neurônios das camadas ocultas e o tipo de arquitetura para representar adequadamente os dados. Além do número de neurônios, também deve ser determinado o valor do peso e limiar de ativação de cada neurônio. Para tal, empregam-se algoritmos de aprendizado de máquina (ML) (BI; FELEKE; GUAN, 2019; BRAGA; CARVALHO; LUDERMIR, 2000; DA SILVA *et al.*, 2017; GOODFELLOW; BENGIO; COURVILLE, 2017; LI, K. *et al.*, 2020; NIELSEN, 2019).

2.2.3.2 Aprendizado de Máquina

O procedimento de ML consiste em algoritmos utilizados para ajustar os valores dos parâmetros de cada neurônio utilizado na RNA. Para isso, primeiramente precisa-se de um banco de dados que possui as respostas que um determinado sistema apresenta quando está sujeito a uma determinada entrada. A construção desse banco de dados é o passo fundamental dos algoritmos de ML, pois através dele serão ajustados os pesos e os limiares de ativação de cada neurônio. Assim, antes de iniciar o treinamento, o banco de dados é dividido em duas partes, os dados para treinamento, que representam de 60 a 90% dos dados selecionados de forma aleatória, e os dados para validação, que representam outros 10 a 40% dos dados restantes, sendo esse processo conhecido por validação cruzada (*cross-validation*). Após a separação, os dados para treinamento são divididos em pequenos lotes, ou *batches*, que serão utilizadas para cada iteração de treinamento do algoritmo. Quanto todas as *batches* forem utilizadas para o treinamento, completa uma época (BRAGA; CARVALHO; LUDERMIR, 2000; DA SILVA *et al.*, 2017; GOODFELLOW; BENGIO; COURVILLE, 2017; NIELSEN, 2019). Algumas técnicas visam melhorar a capacidade generalista da rede limitando o acesso aos bancos de dados a cada treinamento. Uma dessas técnicas é o *k-fold* em que os dados de treinamento são divididos em grupos, e então a cada

rodada de treinamento um dos grupos é deixado “de fora” do treinamento, sendo utilizado para validação. Ao final das rodadas de treinamento, escolhe-se o modelo que apresentou o melhor desempenho entre cada grupo (GOODFELLOW; BENGIO; COURVILLE, 2017; KRASOULIS *et al.*, 2017; SIMAO *et al.*, 2019; TANG; YU; CANG, 2016).

Para considerar que a rede está “treinada”, executam-se várias épocas até que a rede apresente respostas próximas às esperadas. Após o treinamento finalizado ela é testada com os dados para validação comparando as respostas dadas com as respostas esperadas determinando assim, o nível de acurácia que a rede possui com dados externos aos de treinamento, verificando a sua capacidade de generalização. Esse procedimento é conhecido como aprendizado supervisionado (*offline*). Existem outras formas de treinar a rede, utilizando o aprendizado sem supervisão ou o aprendizado por reforço, tanto *offline* como *online* (BRAGA; CARVALHO; LUDERMIR, 2000; DA SILVA *et al.*, 2017; GOODFELLOW; BENGIO; COURVILLE, 2017; NIELSEN, 2019).

Com a ideia geral de como funciona o treinamento, faz-se necessário entender a matemática envolvida nele. O algoritmo mais utilizado para ML é o algoritmo de *backpropagation*, que se baseia na resposta dada pela RNA com uma determinada entrada para ajustar os pesos reduzindo o erro do modelo. Assim o ML é um problema de otimização em que tenta-se reduzir o erro entre as respostas dadas e as respostas esperadas, utilizando para isso o método descida estocástica do gradiente (*Stochastic Gradient Descent* – SGD) (GOODFELLOW; BENGIO; COURVILLE, 2017; NIELSEN, 2019). A Equação (17) apresenta a aplicação do SGD aplicado para encontrar os valores do peso (w_{ji}^l) e o limiar de ativação (θ_j^l) do neurônio (i) de uma determinada camada (l) relativo a uma das entradas (j), onde C_{x_k} representa a função de custo, que mensura o erro da saída, do neurônio para cada uma das entradas da *batch*, onde m é o número de amostras que a *batch* possui, η representa a taxa de aprendizagem, que indica a velocidade de atualização dos pesos e limiares e t é a *batch* que está calculando.

$$\begin{cases} w_{ji}^l(t+1) = w_{ji}^l(t) - \frac{\eta}{m} \cdot \sum_{k=1}^m \frac{\partial C_{X_k}}{\partial w_{ji}^l} \\ \theta_i^l(t+1) = \theta_i^l(t) - \frac{\eta}{m} \cdot \sum_{k=1}^m \frac{\partial C_{X_k}}{\partial \theta_i^l} \end{cases} \quad (17)$$

O aprendizado depende do tipo de função de custo e de suas derivadas parciais. A função de custo mais utilizada é o erro médio quadrático definido na Equação (18), onde $Y_i(X_k)$ representa o valor da resposta esperada, y_i é a resposta dada pelo neurônio e n representa o número de neurônios da camada (BRAGA; CARVALHO; LUDERMIR, 2000; DA SILVA *et al.*, 2017; GOODFELLOW; BENGIO; COURVILLE, 2017; NIELSEN, 2019).

$$C_{X_k} = \frac{1}{2 \cdot n} \cdot \sum_{i=1}^n (Y_i(X_k) - y_i)^2 \quad (18)$$

A função de custo *cross-entropy* é amplamente utilizada e está definida na Equação (19) onde $Y_i(X_k)$ é o valor da resposta desejada, y_i é a resposta dada pelo neurônio e n é o número de neurônios da camada. A principal característica dessa função é provocar mudanças bruscas nos parâmetros quando o erro é elevado e mudanças suaves quando o erro é reduzido (GOODFELLOW; BENGIO; COURVILLE, 2017; NIELSEN, 2019).

$$C_{X_k} = \sum_{i=1}^n [Y_i(X_k) \cdot \ln y_i + (1 - Y_i(X_k)) \cdot \ln(1 - y_i)] \quad (19)$$

Após definida a função de custo utilizada no algoritmo de ML, por meio do algoritmo *backpropagation*, inicia-se com a determinação aleatórias dos parâmetros da RNA e apresentando uma *batch* de dados à rede para se obter a resposta. Para cada uma das respostas obtidas aplica-se a função de custo escolhida e suas derivadas parciais (BRAGA; CARVALHO; LUDERMIR, 2000; DA SILVA *et al.*, 2017; GOODFELLOW; BENGIO; COURVILLE, 2017; NIELSEN, 2019). Primeiramente atualiza-se os parâmetros da camada de saída através da Equação (20), onde $w_{ij}^{(L)}$ representa o peso do neurônio i com a entrada j da última camada, η é a taxa de aprendizagem, m é o número de amostra da *batch*, $\delta_{ik}^{(L)}$ é o erro local para cada amostra (k) da *batch*,

$g(u_{jk}^{(L-1)})$ é a função de ativação do neurônio da camada anterior e $u_{ik}^{(L)}$ é o potencial de ativação para cada entrada apresentada do neurônio ajustado.

$$\begin{cases} w_{ij}^{(L)}(t+1) = w_{ij}^{(L)}(t) - \frac{\eta}{m} \cdot \sum_{k=1}^m \delta_{ik}^{(L)} \cdot g(u_{jk}^{(L-1)}) \\ \theta_i^{(L)}(t+1) = \theta_i^{(L)}(t) - \frac{\eta}{m} \cdot \sum_{k=1}^m \delta_{ik}^{(L)} \\ \delta_{ik}^{(L)} = \frac{\partial C_{x_k}}{\partial g(u_{ik}^{(L)})} \cdot \frac{\partial g(u_{ik}^{(L)})}{\partial u_{ik}^{(L)}} \end{cases} \quad (20)$$

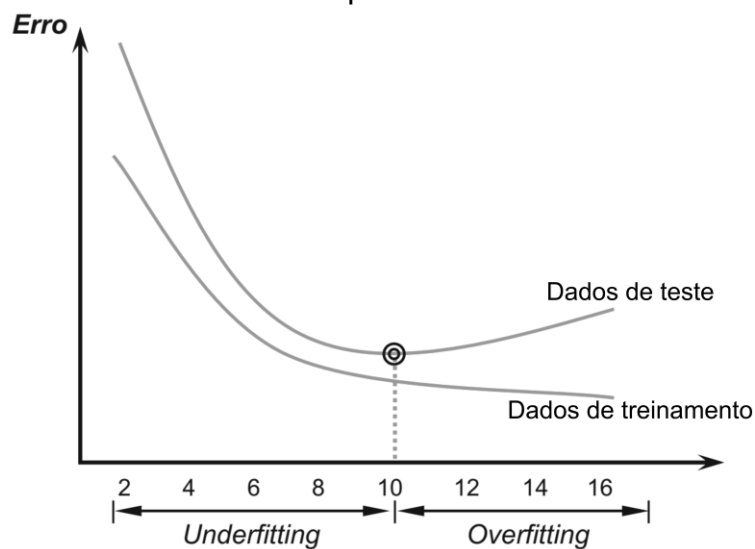
Após a atualização dos parâmetros da última camada, procede-se à atualização das camadas anteriores utilizando a Equação (21) (BRAGA; CARVALHO; LUDERMIR, 2000; DA SILVA *et al.*, 2017; GOODFELLOW; BENGIO; COURVILLE, 2017; NIELSEN, 2019). Na Equação (21) $n^{(l+1)}$ representa o número de neurônios da camada posterior à que está sendo atualizada, $\delta_{nk}^{(l+1)}$ representa o erro local calculado para cada neurônio da camada posterior para cada entrada da *batch* e $w_{nj}^{(l+1)}(t+1)$ representa os valores dos pesos atualizados da camada posterior, que recebem como entrada a saída do neurônio em atualização.

$$\begin{cases} w_{ij}^l(t+1) = w_{ij}^l(t) - \frac{\eta}{m} \cdot \sum_{k=1}^m \delta_{ik}^l \cdot g(u_{jk}^{(l-1)}) \\ \theta_i^l(t+1) = \theta_i^l(t) - \frac{\eta}{m} \cdot \sum_{k=1}^m \delta_{ik}^l \\ \delta_{ik}^l = \left(\sum_{n=1}^{n^{(l+1)}} \delta_{nk}^{(l+1)} \cdot w_{nj}^{(l+1)}(t+1) \right) \cdot \frac{\partial g(u_{ik}^l)}{\partial u_{ik}^l} \end{cases} \quad (21)$$

Um dos principais desafios durante o treinamento é garantir que a RNA generalize os dados, de forma que o modelo não apresente respostas enviesadas ao conjunto de dados apresentados. Como o treinamento depende do ajuste da RNA por cada época, a quantidade de épocas utilizadas implica diretamente nos resultados da rede. Se poucas épocas forem utilizadas, a rede apresentará um erro grande (*underfitting*), entretanto caso muitas épocas forem utilizadas a tendência é que a rede obtenha o valor de função de custo próximo a zero, o que garante que ela tenha um bom desempenho quando as entradas forem iguais às entradas utilizadas no treinamento, porém tenha um erro grande para entradas diferentes (*overfitting*). Como demonstrado

na Figura 11, quando o número de épocas é aumentado o erro com os dados de treinamento diminui, enquanto o erro com os dados de teste aumenta. Assim existe um ponto ótimo em que os dados de teste terão o menor erro possível, levando a uma rede com maior capacidade de generalização. Conhecer esse ponto não é possível antes de iniciarmos o treinamento, para tal, emprega-se um algoritmo de monitoramento chamado *early stopping*. Nesse algoritmo o erro com os dados de teste é avaliado a cada época e quando estes não reduzem mais o treinamento é encerrado restaurando os parâmetros para o momento de melhor desempenho do modelo (BRAGA; CARVALHO; LUDERMIR, 2000; DA SILVA *et al.*, 2017; GOODFELLOW; BENGIO; COURVILLE, 2017).

Figura 11 – Exemplo da evolução do erro de treinamento em função do número de épocas



Fonte: Adaptado de DA SILVA *et al.*, (2017).

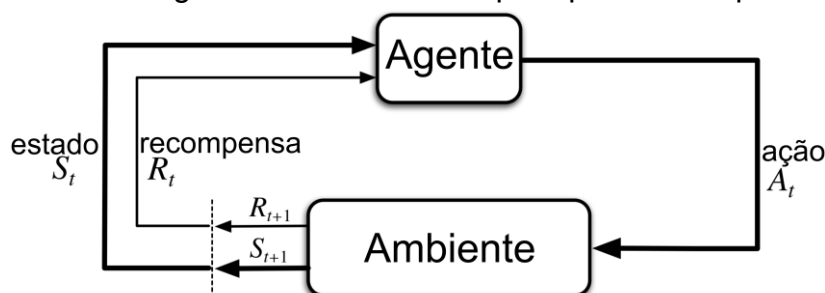
Outra forma de evitar o *overfitting* é a adoção de estratégias de regularização durante o treinamento. Uma das técnicas muito utilizadas é a de atenuação de pesos, que tem como objetivo regularizar os parâmetros da rede, encontrando os menores possíveis por meio da adição de uma penalização na função de custo. Outra técnica muito empregada em treinamento profundo é o uso de camadas de *Dropout* inseridas entre as camadas ocultas da RNA. Esse tipo de camada tem como finalidade excluir algumas das saídas da camada anterior, de acordo com a probabilidade de exclusão de cada saída apenas durante as etapas de treinamento da rede. Assim o *Dropout* visa criar modelos alternativos da RNA, simulando o treinamento simultâneo de múltiplas redes (GOODFELLOW; BENGIO; COURVILLE, 2017).

2.2.3.3 Aprendizado por Reforço

No aprendizado supervisionado, conforme descrito na Sessão 2.2.3.2, o algoritmo de treinamento compara as respostas dadas pela RNA com as respostas esperadas, calculando o erro, dessa forma ensinando a RNA a reduzir os erros em relação a um conjunto de dados, entretanto pode-se treinar RNAs utilizando outra técnica, o aprendizado por reforço (*Reinforcement Learning* – RL). No RL, ao invés da RNA ser ajustada pelo erro, ela é ajustada por meio de uma função de recompensa, ou seja, ela é treinada para obter a maior recompensa possível ao interagir com um determinado sistema (SUTTON; BARTO, 2018).

A Figura 12 apresenta o diagrama de treinamento utilizando o RL. A RNA atua como Agente, o elemento que decidirá quais ações tomar, assim, ela determina de acordo com a sua política qual ação será realizada no ambiente (A_t). Ao ser efetuada a ação, o Ambiente então retorna com uma recompensa (R_{t+1}), determinada de acordo com a resposta do ambiente, e o estado (S_{t+1}) em que se encontra. Essas informações então são recebidas novamente pelo Agente que tomara novas ações a fim de aumentar a recompensa recebida. Em resumo, o treinamento por RL faz com que o Agente procure maximizar a recompensa recebida através da interação com o ambiente (SUTTON; BARTO, 2018).

Figura 12 – Diagrama do treinamento por aprendizado por reforço



Fonte: Adaptado de SUTTON; BARTO, (2018).

Para que o processo de RL aconteça, outros elementos são necessários além dos apresentados na Figura 12, que são uma política, um sinal de recompensa, uma função de valor e um modelo do ambiente. A política é definida como as regras utilizadas para que o Agente execute uma ação, ou seja, a forma como o Agente vai mapear, dado um estado do Ambiente, a ação tomada. O sinal de recompensa é o valor de recompensa imediata que o Agente receberá ao executar uma ação no Ambiente. Como o objetivo do Agente é obter a recompensa máxima, o sinal de recompensa é

responsável por alterar os parâmetros da política (pesos e os limiares de ativação dos neurônios, caso ele seja uma RNA) durante o treinamento (SUTTON; BARTO, 2018).

Apesar da política e do sinal de recompensa em um primeiro momento parecem ser o suficiente para o RL, eles ainda não complementam todo o necessário para o treinamento. O sinal de recompensa dá apenas a recompensa imediata de uma ação que o Agente toma, entretanto, determinadas ações que imediatamente podem ter um alto valor de recompensa, a longo prazo podem levar o Agente a ações indesejadas. Então, para solucionar esse problema, em vez de treinar o Agente utilizando o sinal de recompensa é utilizada uma função valor, que consiste em uma função que indica a probabilidade de uma determinada ação levar o Agente ao estado final de maior recompensa. Assim, ações que imediatamente teriam uma recompensa baixa podem ter uma função valor muito alta, desde que essa ação tenha uma maior probabilidade de levar o Agente ao estado desejado. O grande desafio do RL é determinar a função valor para cada sistema (SUTTON; BARTO, 2018).

O último elemento, apesar de opcional, é de extrema importância para mapear as recompensas obtidas devido as ações tomadas, o modelo do ambiente. Esse modelo corresponde à representação física de um ambiente, que será manipulado pelo Agente, assim, dado uma ação, é capaz de prever o estado futuro, definindo a função valor e atualizando as políticas durante o treinamento (SUTTON; BARTO, 2018).

Uma das classes de algoritmos utilizados para o RL é o Ator-Crítico, onde uma RNA é utilizada para tomada de ações, o Agente/Ator, e outra RNA é utilizada para definir a função valor durante o treinamento, o Crítico. No RL, o treinamento ocorre por passos de tempo, então define-se qual é o valor de recompensa para cada passo de acordo com as ações tomadas pelo Ator. A cada passo de tempo ajusta-se os parâmetros da RNA da função valor (w) e os parâmetros da RNA do Ator (θ) para que as ações tomadas façam com que o ambiente se comporte como esperado. Os parâmetros da função valor são ajustados utilizando a Equação (22), onde α^w é a taxa de aprendizado da função valor, R_{t+1} é a recompensa obtida no próximo passo com os parâmetros atuais, γ a taxa de desconto para previsões futuras, $\hat{v}(S_{t+1}, w_t)$ é a função valor de um estado futuro com os parâmetros atuais, $\hat{v}(S_t, w_t)$ é o valor da função valor do estado atual e λ^w é a taxa de decaimento de rastreamento da elegibilidade dos parâmetros da função valor (SUTTON; BARTO, 2018).

$$\begin{cases} w_{t+1} = w_t + \alpha^w \cdot \delta_t \cdot z^w \\ \delta_t = R_{t+1} + \gamma \cdot \hat{v}(S_{t+1}, w_t) - \hat{v}(S_t, w_t) \\ z^w = \gamma \cdot \lambda^w \cdot z_{t-1}^w + \nabla \hat{v}(S_t, w_t) \end{cases} \quad (22)$$

Após o ajuste dos parâmetros da RNA da função de valor, no mesmo passo de tempo, ajustam-se os parâmetros da RNA do Ator (θ). A Equação (23) representa as expressões utilizadas para ajustar os parâmetros da RNA do Ator para cada passo temporal, onde α^θ é a taxa de aprendizagem, R_{t+1} é a recompensa obtida no próximo passo com os parâmetros atuais, γ a taxa de desconto para previsões futuras, $\hat{v}(S_{t+1}, w_t)$ é a função valor de um estado futuro com os parâmetros atuais, $\hat{v}(S_t, w_t)$ é o valor da função valor do estado atual, λ^θ é a taxa de decaimento de rastreamento da elegibilidade dos parâmetros e $\pi(A_t|S_t, \theta_t)$ é a probabilidade de uma determinada ação ser tomada pela rede dado os estados e os parâmetros atuais, isto é, a probabilidade de resposta da rede para o estado atual (SUTTON; BARTO, 2018).

$$\begin{cases} \theta_{t+1} = \theta_t + \alpha^\theta \cdot \delta_t \cdot z^\theta \\ \delta_t = R_{t+1} + \gamma \cdot \hat{v}(S_{t+1}, w_t) - \hat{v}(S_t, w_t) \\ z^\theta = \gamma \cdot \lambda^\theta \cdot z_{t-1}^\theta + \gamma^t \cdot \nabla \ln(\pi(A_t|S_t, \theta_t)) \end{cases} \quad (23)$$

O Ator-Crítico tradicional apresenta dificuldades em desempenhar bem com problemas de aprendizado profundo, levando a um grande tempo de treinamento para obter a convergência. Além disso, esse algoritmo está restrito a ações discretas, como de classificação, necessitando de modificações para o seu emprego no aprendizado de aproximadores de resposta contínua. Assim, na literatura são encontrados diversos algoritmos baseados no algoritmo do Ator-Crítico. Entre esses algoritmos está o *Deep Deterministic Policy Gradients* (DDPG), primeiramente apresentado por LILLICRAP et al., (2015), que tem sido amplamente utilizado no treinamento de modelos aplicados em múltiplas áreas como o controle contínuo de trajetória em manipuladores robóticos com múltiplos GDL, controle de carros e veículos aéreos autônomos, entre outros (AL-MAHAMID; GROLINGER, 2022; DONG et al., 2023; LI, Z. et al., 2020; RAZZAGHI et al., 2024; SUMIEA et al., 2024).

Na literatura, encontram-se diversos trabalhos que avaliam o DDPG em controladores adaptativos. SUN et al., (2020) desenvolveu um sistema de controle de cruzeiro adaptativo para assistência de direção que foi capaz de melhorar a segurança e eficiência de desempenho de um veículo pesado de três eixos em situações de tráfego. CHEN, P.; HAN, (2023) empregou o DDPG para construir um controlador para

melhorar a eficiência de extração de potência de geradores eólicos, a fim de estabilizar a energia gerada reduzindo as cargas. No controle robótico temos, por exemplo, o trabalho de CHEN, Z. et al., (2024) que desenvolveu um controlador de trajetória para braços robóticos hidráulicos, melhorando seu controle de movimento. Assim, observa-se que o algoritmo DDPG vem sendo utilizado nas mais diversas tarefas com muita eficiência, entretanto, não foram encontrados trabalhos, até o presente momento, que o utilizam para o desenvolvimento de controles de trajetórias em próteses robóticas utilizando sinais de sEMG.

O DDPG (Algoritmo 1) foi inicialmente proposto por LILLICRAP et al., (2015) como uma proposta de Ator-Crítico para solução de problemas que apresentam espaço de ação contínuo. O sucesso desse algoritmo decorre da utilização de uma abordagem livre de modelo e independente de política, combinado as abordagens utilizadas no Ator-Crítico com o método *Deep Q Network* (DQN), o que possibilita seu uso em RNAs utilizadas como regressores (MNIH et al., 2013). Além disso, ele utiliza as técnicas de memória de repetição (*replay buffer*) e de uso de redes-alvos (*target networks*) para auxiliar na convergência do modelo (NUGROHO et al., 2023). Em razão dessa combinação, essa técnica permite que o modelo aprenda políticas de espaço contínuo de ação, com múltiplos estados e com dinâmicas não lineares complexas (MEYES; SCHEIDERER; MEISEN, 2018).

Além do DDPG, outros algoritmos também vêm sendo utilizados na literatura como o *Proximal Policy Optimization* (PPO) (SCHULMAN et al., 2017) e o *Soft Actor-Critic* (SAC) (HAARNOJA et al., 2018). O PPO é uma variação dos métodos de gradiente que tem como objetivo otimizar a política do modelo diretamente sem o uso da função valor, reduzindo assim a variação dos gradientes e melhorando a convergência do modelo conforme o número de amostras aumenta. Quando comparado com o DDPG foi observado que quanto maior a complexidade da tarefa, apesar de ter um melhor desempenho de treinamento, o PPO não representava uma melhora significativa em relação ao desempenho do treinamento utilizando o DDPG, como foi relatado por KUMAR et al., (2023).

Algoritmo 1: DDPG

```

1  Inicialize os parâmetros  $\theta^Q$  do Crítico  $Q(s, a|\theta^Q)$  e  $\theta^\mu$  do Ator  $\mu(s|\theta^\mu)$ 
2  Inicialize as redes alvo  $Q'$  e  $\mu'$  com os parâmetros  $\theta^{Q'} \leftarrow \theta^Q, \theta^{\mu'} \leftarrow \theta^\mu$ 
3  Inicialize a memória de repetição  $R$ 
4  para episódio de 1 até  $M$  faça:
5      Receba os estados iniciais  $s_1$  do ambiente
6      para  $t$  de 1 até  $T$  faça:
7          Selecione uma ação de acordo com a política atual e o ruído usando:
              
$$a_t \leftarrow \mu(s_t|\theta^\mu) + \epsilon, \quad \epsilon \approx \mathcal{N}(0, \sigma)$$

8          Execute a ação  $a_t$  e observe a recompensa  $r_t$  e o próximo estado  $s_{t+1}$ 
9          Armazene  $(s_t, a_t, r_t, s_{t+1})$  em  $R$ 
10         Pegue uma miniamostra aleatória com  $N$  amostras de transição  $(s_n, a_n, r_n, s_{n+1})$  de  $R$ 
11         Determine  $\delta_n = r_n + \gamma Q'(s_{n+1}, \mu'(s_{n+1}|\theta^{\mu'})|\theta^{Q'}) - Q(s_n, a_n|\theta^Q)$ 
              Atualize os parâmetros do Crítico com um passo de gradiente descendente usando:
12             
$$\nabla_{\theta^Q} \frac{1}{N} \sum_n (Q(s_n, a_n|\theta^Q) - \delta_n)^2$$

              Atualize a política do Ator com um passo de gradiente ascendente usando:
13             
$$\nabla_{\theta^\mu} \frac{1}{N} \sum_n Q(s_n, \mu(s_n|\theta^\mu)|\theta^Q)$$

              Atualize as redes alvo:
14             
$$\theta^{Q'} \leftarrow \tau \theta^Q + (1 - \tau) \theta^{Q'}$$

             
$$\theta^{\mu'} \leftarrow \tau \theta^\mu + (1 - \tau) \theta^{\mu'}$$

15         fim-para
16 fim-para

```

O SAC é um algoritmo estocástico que se baseia no emprego do arcabouço de entropia máxima que tende a melhorar a estabilidade e a exploração do modelo durante o treinamento (HAARNOJA *et al.*, 2018; WANG; HONG; ZHONG, 2022). MOSLEMI; SADEDEL; MOGHADAM, (2024) demonstraram em seu trabalho comparativo que conforme a complexidade da função de recompensa aumentava o DDPG conseguia superar o desempenho do SAC. Observa-se que para dinâmicas e funções de recompensa complexas o DDPG consegue apresentar um melhor desempenho em relação aos demais algoritmos.

Apesar do DDPG ter um bom desempenho, ele sofre problemas de convergência em alguns ambientes, principalmente quando o Crítico tende a superestimar a função valor nos estágios iniciais do treinamento, levando à exploração de regiões que

causam divergência. Para solucionar esse problema, foi proposta uma atualização do DDPG conhecida como *Twin Delayed Deep Deterministic Policy Gradients* (TD3) inicialmente proposta por FUJIMOTO; VAN HOOFF; MEGER, (2018). O TD3 introduz três modificações no DDPG, a técnica de aprendizado dupla, adição de ruído nas ações determinadas pela rede alvo e a atualização atrasada da política e das redes alvo (FUJIMOTO; VAN HOOFF; MEGER, 2018; MORALES, 2020).

A estratégia de aprendizado duplo consiste em utilizar duas funções valor (ou seja, duas RNAs como Crítico, Q_1 e Q_2) para encontrar o valor estimado alvo da função valor (δ), através do valor mínimo entre as duas RNAs alvo, Q'_1 e Q'_2 . A Equação (40) representa a estimativa alvo da função valor, onde observa-se que dada uma recompensa de um estado atual (r_t), o valor alvo é determinado pela soma descontada pela taxa de desconto γ . A soma descontada é proveniente do valor mínimo das funções alvo dos dois Críticos dado em resposta ao próximo estado e a ação suavizada (a') dada pelo Ator alvo, utilizando os parâmetros atuais das RNAs alvo. Por fim, d representa se o passo do episódio é ou não um passo terminal, assumindo valor 1 caso seja terminal e 0 para estados não terminais (MORALES, 2020).

$$\delta(r_n, s_{n+1}, d_n) = r_n + \gamma(1 - d_n) \min_{i=1,2} \left(Q'_i \left(s_{n+1}, a'(s_{n+1}) \middle| \theta^{Q'_i} \right) \right) \quad (24)$$

A ação suavizada constitui a segunda modificação introduzida pelo TD3 ao DDPG. A ideia de suavizar a ação é aplicar um ruído Gaussiano tanto nas ações tomadas pelo agente durante a manipulação do ambiente, quanto durante a estimativa dos valores esperados da função valor utilizando o Ator alvo (μ'). O uso dessa técnica garante ao TD3 uma melhor extrapolação, pois o ruído adicionado atua como um regularizador, uma vez que induz respostas mais generalizadas da rede utilizada como Crítico para ações similares. Dessa forma, essa técnica previne que a RNA da política convirja para ações inadequadas nos estágios iniciais do treinamento devido a inacurácia inicial do Crítico, pois com o ruído os valores das ações serão mais abrangentes durante o treinamento (MORALES, 2020). A Equação (25) representa a ação suavizada alvo, observa-se que o valor utilizado é limitado pela faixa de ações válidas permitidas ao ambiente ($[a_{min}, a_{max}]$). Outra característica importante é que o ruído Gaussiano (\mathcal{N}) possui desvio padrão (σ) entre uma faixa de 10 a 20 % do valor máximo permitido às ações, entretanto a faixa de valores de ruído permitida também é limitada

ao intervalo $[\epsilon_{min}, \epsilon_{max}]$, que geralmente tem um valor igual a metade do intervalo das ações (FUJIMOTO; VAN HOOFF; MEGER, 2018; MORALES, 2020).

$$a'(s) = clip(\mu'(s|\theta^{\mu'}) + clip(\epsilon, \epsilon_{min}, \epsilon_{max}), a_{min}, a_{max}), \quad \epsilon \sim \mathcal{N}(0, \sigma) \quad (25)$$

A última modificação introduzida pelo TD3 refere-se ao atraso na atualização da política e das redes-alvo, enquanto os Críticos apresentam maior frequência de atualização. O uso dessa atualização tenta garantir que a política só seja atualizada quando o Crítico esteja mais estável, o que é particularmente relevante nos estágios iniciais do treinamento, quando o valor do Crítico varia muito e não apresenta valores estáveis para guiar a política (MORALES, 2020). Geralmente os Críticos são atualizados a cada iteração de treinamento, enquanto a política e a rede-alvo são atualizadas a cada duas iterações de treinamento (FUJIMOTO; VAN HOOFF; MEGER, 2018).

O Algoritmo 2 representa o algoritmo padrão do TD3 apresentado por FUJIMOTO; VAN HOOFF; MEGER, (2018) e MORALES, (2020). Observa-se que o processo de treinamento é bem parecido com o apresentado no Algoritmo 1, modificando a forma como é tomada a ação da rede-alvo do Ator, do uso dos dois Críticos para estimar o valor desejado da função valor e do desacoplamento da atualização do Ator com a atualização do Crítico. Outro aspecto relevante é que ambas as redes do Crítico são atualizadas independentemente, assim elas não compartilham nenhum parâmetro. Por fim, apenas a rede do primeiro Crítico é utilizada para atualizar a rede do Ator, como ocorre no DDPG.

Algoritmo 2: TD3

```

1 Inicialize os Críticos  $Q_1(s, a|\theta^{Q_1})$  e  $Q_2(s, a|\theta^{Q_2})$  e  $\theta^\mu$  do Ator  $\mu(s|\theta^\mu)$ 
2 Inicialize as redes alvo  $Q'_1, Q'_2$  e  $\mu'$  com os parâmetros  $\theta^{Q'_1} \leftarrow \theta^{Q_1}, \theta^{Q'_2} \leftarrow \theta^{Q_2}$  e  $\theta^{\mu'} \leftarrow \theta^\mu$ 
3 Inicialize a memória de repetição  $R$ 
4 para episódio de 1 até  $M$  faça:
5   Receba os estados iniciais  $s_1$  do ambiente
6   para  $t$  de 1 até  $T$  faça:
7     Selecione uma ação de acordo com a política atual e o ruído:
8        $a_t \leftarrow \text{clip}(\mu(s_t|\theta^\mu) + \epsilon, a_{\min}, a_{\max}), \quad \epsilon \approx \mathcal{N}(0, \sigma)$ 
9     Execute a ação  $a_t$  e observe a recompensa  $r_t$  e o próximo estado  $s_{t+1}$  e  $d$ 
10    Armazene  $(s_t, a_t, r_t, s_{t+1}, d)$  em  $R$ 
11    se é hora de atualizar então
12      Pegue uma miniamostra aleatória com  $N$  amostras  $(s_i, a_i, r_i, s_{i+1}, d)$  de  $R$ 
13      Determine a ação alvo suavizada
14         $a'(s) = \text{clip}(\mu'(s|\theta^{\mu'}) + \text{clip}(\epsilon, \epsilon_{\min}, \epsilon_{\max}), a_{\min}, a_{\max}), \quad \epsilon \sim \mathcal{N}(0, \sigma)$ 
15      Determine o valor alvo da função valor
16         $\delta(r_n, s_{n+1}, d_n) = r_n + \gamma(1 - d_n) \min_{i=1,2} (Q'_i(s_{n+1}, a'(s_{n+1})|\theta^{Q'_i}))$ 
17      Atualize separadamente os Críticos com um passo de gradiente descendente usando:
18         $\nabla_{\theta^{Q_i}} \frac{1}{N} \sum_n (Q_i(s_n, a_n|\theta^{Q_i}) - \delta(r_n, s_n, d_n))^2, \quad i = 1,2$ 
19      se passo de atualização da política então
20        Atualize a política com um passo de método de gradiente ascendente usando:
21         $\nabla_{\theta^\mu} \frac{1}{N} \sum_n Q_1(s_n, \mu(s_n|\theta^\mu)|\theta^{Q_1})$ 
22        Atualize as redes alvo individualmente:
23           $\theta^{Q'_i} \leftarrow \tau\theta^{Q_i} + (1 - \tau)\theta^{Q'_i}, \quad i = 1,2$ 
24           $\theta^{\mu'} \leftarrow \tau\theta^\mu + (1 - \tau)\theta^{\mu'}$ 
25    fim-se
26  fim-se
27 fim-para
28 fim-para

```

2.2.3.4 Classificadores e Controladores Mioelétricos

Dadas as arquiteturas utilizadas e os métodos de ML, os classificadores apresentam grande capacidade de determinar o movimento a ser executado pela prótese a partir das características do sinal de sEMG. Dentre as técnicas mais utilizadas na literatura são observados: SVM; SVM utilizando o método dos mínimos quadrados

(LS-SVM) (ATZORI *et al.*, 2012, 2014; DING *et al.*, 2014; GUO, S. *et al.*, 2015; GUO, W. *et al.*, 2017b; LIU *et al.*, 2016; MAYOR *et al.*, 2017; NAZEMI; MALEKI, 2014; WARIS *et al.*, 2014; YANG; GU; *et al.*, 2017; YANG; YANG; *et al.*, 2017) e o SVM quadrático (QSVM) (ALY; YOUSSEF; FATHY, 2018); também o LDA, (AL-TIMEMY; KHUSHABA; ESCUDERO, 2016; AMSUSS *et al.*, 2013, 2014; DING *et al.*, 2014; GREGORI; CAPUTO; GIJSBERTS, 2018; GUO, W. *et al.*, 2017b, a; JARRASSE *et al.*, 2017; KRASOULIS *et al.*, 2017; LIANG PENG *et al.*, 2013; LIU *et al.*, 2016; MAYOR *et al.*, 2017; NAIK; AL-TIMEMY; NGUYEN, 2016; NAZEMI; MALEKI, 2014; POWAR; CHEMMANGAT, 2017) e a análise de discriminante quadrático (QDA) (ALY; YOUSSEF; FATHY, 2018); as RNAs utilizando MLP (DING *et al.*, 2014; FAJARDO; LEMUS; ROHMER, 2015; GUO, S. *et al.*, 2015; NAZEMI; MALEKI, 2014; VEER, 2015); o algoritmo de *k-nearest neighbors* (k-NN) (ALY; YOUSSEF; FATHY, 2018; LIU *et al.*, 2016; MAYOR *et al.*, 2017; POWAR; CHEMMANGAT, 2017); o *random forests* (RF) (LIU *et al.*, 2016); a regressão de crista incremental (IRR) (STRAZZULLA *et al.*, 2017); e o classificador Naive Bayes (POWAR; CHEMMANGAT, 2017).

Quanto ao desempenho dos classificadores para predição de movimento ATZORI *et al.*, (2012) observou que com aumento de movimentos a serem classificados ocorre um aumento significativo na taxa percentual de erro de classificação, sendo que para poucos movimentos classificados o erro foi de 7,5%, enquanto com as mesmas configurações de rede e treinamento o erro para muitos movimentos subia para 20%. A redução da acurácia observada por ATZORI *et al.*, (2012) foi também observada no trabalho de JARRASSE *et al.*, (2017). Outra observação do seu trabalho é que o desvio padrão do erro era relativamente alto, o que indica que ocorre uma grande discrepância entre os sinais de sEMG de diferentes indivíduos. WARIS *et al.*, (2014) em seu trabalho obteve uma acurácia média de 89,6 % utilizando o SVM para classificar 5 movimentos. Enquanto ATZORI *et al.*, (2014) obteve uma acurácia de 82,77 % quando foi utilizado o SVM para classificar 52 movimentos. VEER, (2015) utilizando MLP para classificar 4 movimentos utilizando 2 canais apresentou o menor erro médio quadrático de 0,003 após 19 épocas com 3 neurônios na única camada oculta utilizada. Utilizando 10 canais de sEMG para classificar 5 movimentos STRAZZULLA *et al.*, (2017) obteve uma taxa de execução da tarefa de 95,6 % com atividades manuais de mão única e 83,6 % para atividades bimanuais utilizando o IRR.

No trabalho de NAZEMI; MALEKI, (2014), foram comparados os classificadores LDA, LS-SVM e MLP, utilizando diferentes tipos de características do sinal de sEMG. Foi observado que independentemente das características utilizadas a MLP apresentava a maior acurácia de classificação, de 96,43 % contra 84,23% do LDA e 85,19 % do LS-SVM. Embora apresentasse bom desempenho de classificação, a MLP era o classificador que apresentava o maior custo computacional, por isso os classificadores mioelétricos utilizando LDA eram os mais utilizados na época, pois eram computacionalmente mais leves, o que compensava a menor acurácia. Já DING et al., (2014) em seu trabalho observou que utilizando apenas a característica de coeficientes de esparsidade do sEMG para classificar 8 movimentos não houve diferença estatisticamente significativa no desempenho entre os desempenhos do LDA, do SVM e da MLP. Outra comparação foi feita por ALY; YOUSSEF; FATHY, (2018) comparando o k-NN, o QDA e o QSVM e obteve a maior acurácia utilizando o QDA para classificar 5 movimentos (99,5 %) apresentando baixa variabilidade de acurácia entre os voluntários que testaram o classificador. MAYOR et al., (2017) comparando o SVM, k-NN e o LDA para classificar 13 movimentos obteve a maior acurácia com o SVM (98,8%), seguindo do k-NN (94,6%) e por último o LDA (71,9%), demonstrando que o aumento da quantidade de movimentos a serem classificados o LDA deixa de ser um bom classificador, requerendo classificadores mais robustos.

Um dos grandes problemas na aplicação de classificadores para PRL está na generalização para aplicações em tempo real. GUO, S. et al., (2015) demonstrou que as RNAs tendem a apresentar *overfitting* com os dados de treinamento, o que prejudica seu desempenho quando comparadas com outros métodos, como o SVM, e resulta em uma queda brusca da acurácia de movimento, mesmo que durante os testes de validação no treinamento a acurácia seja elevada, em torno de 97%. Isso se dá ao fato de o sinal de sEMG ser não estacionário e variável com o tempo e por indivíduo.

Para melhorar a acurácia dos controladores mioelétricos, outras técnicas podem ser utilizadas combinando sinais de sEMG com outros sensores. No trabalho de LIU et al., (2016) foi demonstrado que quando associava o sinal de sEMG com o de acelerômetros a acurácia do classificador SVM chegou a 88,72 %, cerca de 13% maior do que a acurácia do mesmo sistema sem o uso dos dados de acelerometria. (GUO, W. et al., 2017a) viu melhora no classificador LDA e no classificador SVM quando combinou sinais de sEMG com de espectrografia de infravermelho por proximidade

(NIRS), obtendo 86,7% e 86,6% de acurácia respectivamente na classificação de 5 movimentos com 12 canais de sEMG, valores superiores ao mínimo de acurácia necessário para evitar frustração do usuário, que é de 85%. KRASOULIS et al., (2017) observou que ao associar o sEMG com sensores inerciais foi obtido o melhor desempenho do classificador do que com o acelerômetro, apresentando 82,7% de acurácia na classificação de 12 movimentos. GUO, W. et al., (2017a) demonstrou que a melhor combinação se dá com sensores de mecanomiografia, conseguindo uma acurácia de 95,1% com indivíduos não amputados.

Quanto aos parâmetros que podem influenciar o desempenho do classificador, LIANG PENG et al., (2013) observaram que seu classificador utilizando LDA, que classificava 7 movimentos com 4 canais de sEMG, tinha seu desempenho prejudicado conforme a posição de inclinação do punho mudava, assim como WARIS et al., (2014). Já AMSUSS et al., (2013) observou que o fator de maior influência é o tempo de uso, onde foi demonstrado que, após cinco dias de uso do classificador pelos usuários, obteve-se o maior valor de acurácia, reforçando assim a necessidade de treinamento para aplicação do sistema em próteses.

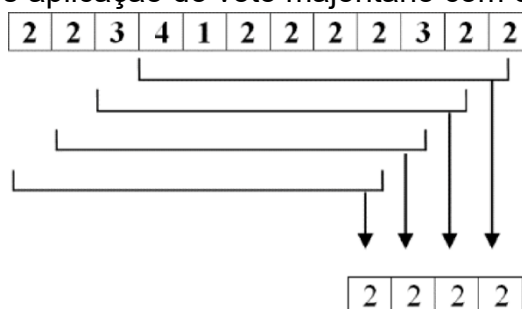
YANG; GU; et al., (2017) em seu trabalho observou que a força e a forma de aquisição de dados influenciam no desempenho do classificador. Avaliou-se o classificador SVM treinado com dados em quatro condições do sinal adquirido, em uma condição estática em contração constante, o sinal adquirido em movimentos dinâmicos executando ATVDs, o sinal adquirido e movimentos dinâmicos de ATVD com variação de postura e o sinal adquirido e movimentos dinâmicos de ATVD com variação de postura e força. Observou-se que a maior acurácia era obtida quando os dados utilizados para o treinamento pertenciam ao mesmo indivíduo que utilizava o sistema. Entretanto quando outro indivíduo utilizava o sistema treinado com dados de outro indivíduo a acurácia era inferior a 70%, sendo o pior caso quando o treinamento era feito com dados estáticos. Observando todas essas influências foi proposto que, para desenvolver um sistema de sEMG capaz de ser generalista, é necessário que sejam utilizados dados das mais diversas formas das ATVDs e da maior diversidade de indivíduos possíveis. As mesmas conclusões foram observadas por GREGORI; CAPUTO; GIJSBERTS, (2018) que utilizando o LDA treinado com dados estáticos observou uma queda de acurácia do classificador para o uso em ambiente controlado de aproximadamente 80% para menos de 40% durante as ATVDs.

MOHAMMADREZA; FONSECA; JIANG, (2025) avaliaram quatro protocolos de treinamento para classificadores de movimento de sEMG, analisando a robustez e a generalidade do sistema em aplicações de tempo real. Foi avaliada a eficiência do classificador em situações em que os dados foram coletados sem a prótese, com a prótese em situações estáticas, durante movimentos dinâmicos utilizando a prótese de maneira guiada e livre. Em seu trabalho foi demonstrado que o desempenho do controlador cai drasticamente em aplicações em condições reais devido ao peso dos objetos e os movimentos de braço, o que evidencia que, para um melhor desempenho, é necessário que o banco de dados contenha dados coletados durante o uso da prótese, melhorando a robustez e a generalidade do controlador, como apontado pelos demais autores.

Dada a grande variabilidade do sinal, e sua influência por diversos fatores externos, alguns autores focam em criar tipos de classificadores para melhorar o desempenho em tempo real. SEGIL; WEIR, (2015) desenvolveram o algoritmo de soma de vetores (VSA), que utiliza o RMS do sEMG para determinar um vetor de ativação entre os canais e com isso, mapear os músculos que estão sendo ativados para determinar o movimento a ser efetuado. Foi observado em seus testes que o controle de velocidade teve o melhor desempenho, entretanto, o algoritmo dependia da manutenção prolongada da contração pelo usuário, o que dificulta o controle de posição. BETTHAUSER et al., (2016) utilizou a técnica de classificação por representação de esparsidade (SRC), muito utilizada em reconhecimento facial, dada a sua capacidade de separar uma quantidade densa de características. Apesar de limitado à classificação de um único movimento, o SRC demonstrou um desempenho de classificação maior que o LDA quando apresentado a dados que estavam fora do banco de dados de treinamento.

Outra forma de tentar melhorar a acurácia de classificação dos sinais é utilizando métodos de pós processamento da classificação. O método conhecido por voto majoritário (Figura 13) é o mais utilizado, em que ao invés de o controlador receber diretamente o movimento desejado classificado, é feito um acúmulo de classificações, em que a classe mais recorrente era escolhida e enviada como o movimento desejado para o controlador, melhorando assim a suavidade do movimento da prótese, pois não ocorrem mudanças bruscas de movimento (AL-TIMEMY; KHUSHABA; ESCUDERO, 2016; MAYOR et al., 2017).

Figura 13 – Exemplo de aplicação do voto majoritário com o acúmulo de nove votos

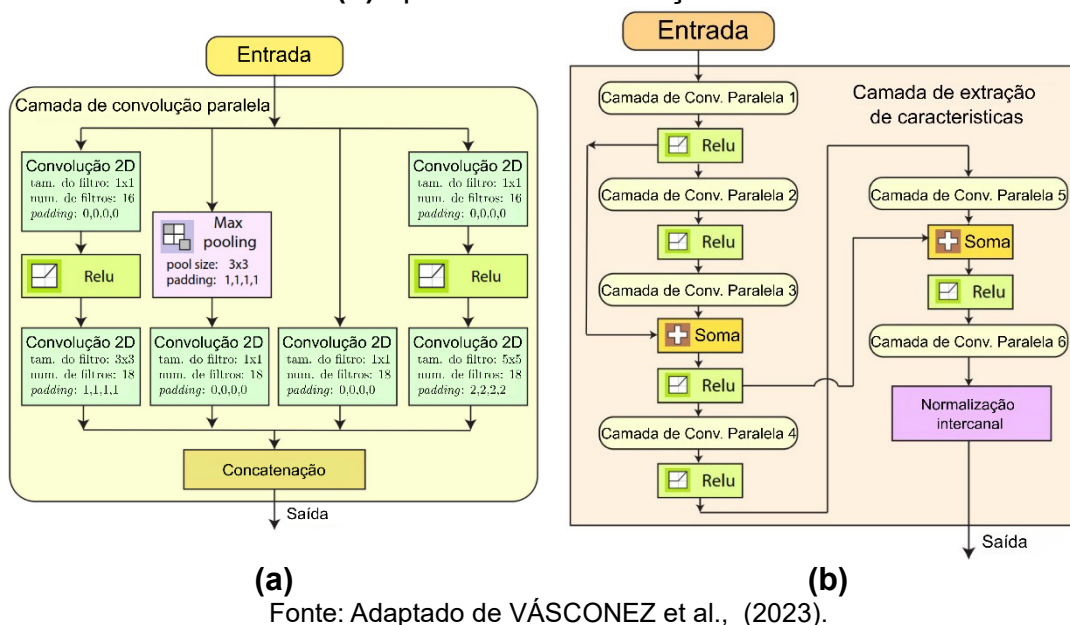


Fonte: AL-TIMEMY; KHUSHABA; ESCUDERO, (2016).

A desvantagem do voto majoritário é o atraso inserido no sistema, uma vez que cada voto corresponde a uma janela de dados apresentada ao controlador. Assim, deve-se escolher o número mínimo de votos necessário para não acrescentar um atraso significativo ao movimento e ao mesmo tempo ter uma melhora significativa no desempenho da classificação (AL-TIMEMY *et al.*, 2016; AL-TIMEMY; KHUSHABA; ESCUDERO, 2016; LIU *et al.*, 2016).

Com o aumento do recurso computacional, outras arquiteturas vêm sendo exploradas como as CNNs e as RNN. VÁSCONEZ *et al.*, (2023) fez um trabalho comparativo utilizando uma arquitetura de CNN baseada em módulos *Inception* desenvolvida pela GoogLeNet, conforme ilustrado na Figura 14, sendo essa arquitetura muito utilizada na classificação de imagens, devido a sua característica de extração de características utilizando operações de convolução em paralelo. Neste trabalho a arquitetura foi utilizada na extração de características do domínio do tempo e da frequência para comparar o desempenho de treinamento utilizando o algoritmo DQN de RL e o treinamento supervisionado. Observou-se que o aprendizado supervisionado apresentou uma acurácia de 90,49%, enquanto no RL a acurácia máxima observada foi de 76,27%. Além de um desempenho superior de classificação, nessas condições o treinamento supervisionado leva um menor tempo computacional, em que o modelo foi treinado em duas horas, enquanto o modelo em RL levava quarenta e cinco horas para ser treinado.

Figura 14 – Exemplo de célula de convolução *Inception* (a) empregada em um modelo de CNN (b) aplicada à classificação de sinais de sEMG



Fonte: Adaptado de VÁSCONEZ et al., (2023).

Após a classificação do sinal, a última etapa do controlador mioelétrico consiste no envio do sinal ao controlador da prótese, responsável por executar o movimento desejado. O controlador mais utilizado é a máquina de estados finitos. Uma vez que é determinado o movimento desejado, a máquina de estados finitos indica para a prótese o movimento (estado) escolhido, permitindo assim que o usuário desative ou ative o movimento conforme a contração muscular classificada, buscando tornar o movimento o mais natural (FAJARDO; LEMUS; ROHMER, 2015; KRASOULIS *et al.*, 2017; SEGIL; WEIR, 2015). Controladores do tipo *fuzzy* também são utilizados, posteriormente à classificação do movimento (LI, C. *et al.*, 2018), assim como controles proporcionais, porém requerem a utilização da característica de RMS do sEMG para determinar a intensidade da atividade muscular (STURMA *et al.*, 2015).

Apesar de os sistemas de PRL apresentarem valores elevados de acurácia, superior a 90%, a lógica de funcionamento do sistema dificulta seu uso nas ATVDs. O PRL associa o sinal de sEMG com uma pose desejada, porém durante as ATVDs, inúmeras combinações de angulação, velocidade e torque das articulações são utilizadas sem apresentar um padrão definido (NASTARIN; AKTER; AWAL, 2019; STRAZZULLA *et al.*, 2017). Além disso, a mudança de movimento nas próteses que utilizam máquinas de estados é muito abrupta, e os movimentos não apresentam transições suaves, dificultando a utilização do dispositivo (JIANG *et al.*, 2025). Assim são necessários classificadores que, em vez de associarem o sEMG a poses, o associem com

o ângulo, a velocidade ou o torque das juntas da prótese. Para isso, pode-se utilizar os regressores para estimar o movimento de junta contínua para a execução do movimento (LI, K. *et al.*, 2020).

Uma das principais formas de traduzir o sinal de sEMG em ângulos articulares, velocidades ou torques é por meio de biomodelos musculares ou musculoesqueléticos. Nesse tipo de abordagem o sinal de sEMG é utilizado como dado de entrada para o modelo de contração muscular, como o modelo de Hill, e então é estimado o ângulo que o membro irá fazer. Apesar de mais próximo ao funcionamento biológico, ao ser utilizado em ativação de próteses em tempo real esse modelo apresenta baixa acurácia, pois depende de inúmeros parâmetros para ser calibrado, como rigidez das articulações, que variam entre indivíduos e ao longo do tempo. Dado isso, a aplicação de modelos musculoesqueléticos é mais utilizada na avaliação muscular clínica e não para controle de próteses (BI; FELEKE; GUAN, 2019; LI, K. *et al.*, 2020).

Outra forma de traduzir o sinal de sEMG em valores contínuos das juntas é utilizando métodos de regressão, tanto lineares, polinomiais e RNAs. As regressões lineares e polinomiais apresentam maior estabilidade nas posições previstas, porém são menos generalistas, o que dificulta o uso do sistema para vários usuários, uma vez que se faz necessário um modelo de regressão individual para cada indivíduo. Já as RNAs, apesar de menos estáveis, apresentam a maior generalização (BAKSHI; MANJUNATHA; KUMAR, 2018; BUENO; MONTANO, 2017; LI, K. *et al.*, 2020; PAN; HARMODY; HUANG, 2018; SIMAO *et al.*, 2019).

A regressão realizada por RNAs, como demonstra o trabalho de TANG; YU; CANG, (2016), apresenta elevada acurácia quando se combina o sinal de sEMG medido em várias condições de carregamento e força, pois o desempenho do regressor depende diretamente da força utilizada e da velocidade da contração. Assim, um sistema eficaz é alcançado com a combinação de múltiplos dados nas mais diversas situações, dependendo ainda da integração de outros sensores além do sEMG.

Algumas arquiteturas mais atuais como CNN e RNN também vêm sendo utilizadas como regressores de ângulo das articulações da mão. JIANG *et al.*, (2025) propôs uma arquitetura de CNN, chamada *TF2AngleNet*, capaz de prever ângulos contínuos dos dedos da mão com coeficiente de correlação de 94,7 %, R^2 de 89,2% e erro médio quadrático de 89,2 %. Na arquitetura proposta foram utilizadas duas redes CNN, uma voltada para transcrição de características de frequência e outra para

características temporais. Para predição de ângulos as redes foram concatenadas e o decodificador das características foi feito por uma terceira CNN. Neste trabalho foi demonstrado que o desenvolvimento de sistemas contínuos de predição de movimentos se mostra essencial para o desenvolvimento de prótese mais adaptativas ao usuário, entretanto, embora eficaz, os autores realizaram testes apenas em um modelo virtual que emulava uma mão real e não uma prótese, o que não demonstrou a capacidade do uso desse sistema em condições reais de aplicação.

Do ponto de vista dos controladores adaptativos, o principal foco recai sobre os controladores híbridos de força de preensão ou escorregamento. Nesses sistemas o sinal de EMG é utilizado para definir qual movimento o usuário deseja fazer até o momento em que a prótese encosta no objeto, em que um sistema autônomo assume o controle do dispositivo para garantir que o objeto não caia da mão. Esse tipo de controlador é demonstrado no trabalho de MATOS; CAPSI-MORALES; PIAZZA, (2025) onde se combinou um sistema de sEMG de limiar de ativação, com um controlador proporcional integrativo. Nesse trabalho o sistema autônomo de controle de escorregamento foi implementado por meio de um sensor de força magnético capaz de medir a força normal e de escorregamento feito pelo usuário. Com esses valores o sinal de controle é determinado para garantir que o objeto não escorregasse da prótese. Foi demonstrado em seu trabalho que o sistema foi eficaz em realizar a transição entre sistemas de controle e conseguir compensar a força para prevenir o escorregamento. Abordagens semelhantes são observadas no trabalho de DUTRA; SILVEIRA, (2023) e GHAZALI et al., (2017), que garantiram que a prótese mantenha firme o objeto utilizando controladores proporcionais, derivativo-integrativos; do tipo Fuzzy e modelo de predição com filtros de Kalman. Apesar de conseguirem efetuar as tarefas propostas, esse tipo de controlador híbrido foca apenas em adaptar a preensão e não o movimento da prótese, o que é essencial para facilitar o uso do dispositivo e tornar o controle mais suave.

Conforme observado, na literatura os controladores mioelétricos se baseiam na predição de movimentos desejados pelo usuário para controle da prótese. O uso de RNAs vem se mostrando uma boa alternativa para o desenvolvimento dos classificadores, uma vez que apresentam maior acurácia quando é observada a classificação de uma grande variedade de movimentos, ou quando a quantidade de características é elevada (DING *et al.*, 2014; GUO, S. *et al.*, 2015; NAZEMI; MALEKI, 2014;

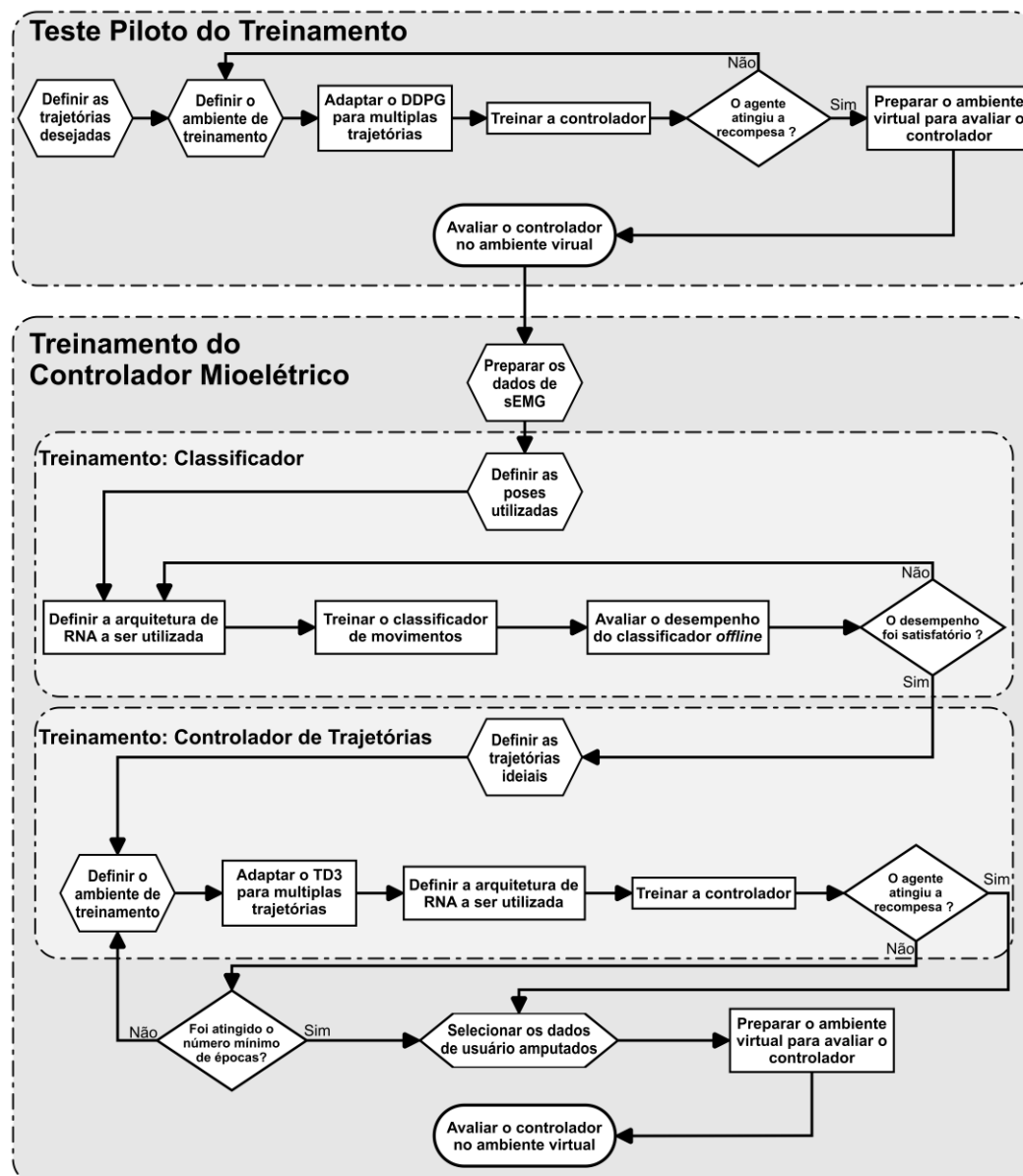
VÁSCONEZ *et al.*, 2023; VEER, 2015). Apesar da alta acurácia, superior a 90%, o uso de estratégias de controle que envolvem principalmente as máquinas de estados tornam a transição de movimentos abruptas e pouco intuitivas para o usuário, o que demonstra a lacuna a ser preenchida no desenvolvimento desse tipo de controlador (JIANG *et al.*, 2025).

Para melhorar a integração do usuário ao dispositivo, sugere-se o uso de regressores de posição das articulações dos dedos, que apresentam uma elevada correlação, quando implementados por meio de CNNs (JIANG *et al.*, 2025). Entretanto esses trabalhos ainda se limitam a prever os movimentos considerando os GDL de uma mão fisiológica e não nos GDLs presentes no dispositivo a serem controlados (BAKSHI; MANJUNATHA; KUMAR, 2018; BI; FELEKE; GUAN, 2019; BUENO; MONTANO, 2017; JIANG *et al.*, 2025; LI, K. *et al.*, 2020; PAN; HARMODY; HUANG, 2018; SIMAO *et al.*, 2019). Outra abordagem recorrente são os controladores autômatos, que isolam a etapa de classificação de movimento da manipulação do objeto pelo dispositivo, garantindo a sua apreensão (DUTRA; SILVEIRA, 2023; GHAZALI *et al.*, 2017; MATOS; CAPSI-MORALES; PIAZZA, 2025). O uso de RL na robótica vem demonstrando uma elevada capacidade em se desenvolver controladores robóticos, principalmente envolvendo o controle de trajetórias (CHEN, Z. *et al.*, 2024). Essas soluções robóticas podem ser valiosas para o desenvolvimento de controladores mais integrados ao usuário, sendo capazes de reproduzir trajetórias desejadas associadas ao movimento desejado, determinado pelo sinal de sEMG, e proporcionando movimentos mais previsíveis ao usuário. Apesar de promissor, é importante notar, como alertado por VÁSCONEZ *et al.*, (2023), que a técnica do RL tem alto custo computacional e nenhuma garantia de que o objetivo pretendido seja alcançado. Assim é valiosa a investigação das capacidades do RL em seus algoritmos clássicos, DDPG e TD3, para o desenvolvimento de controladores mioelétricos para próteses robóticas, demonstrando as capacidades e limitações dessa técnica.

CAPÍTULO 3: TESTE PILOTO COM MODELO BIOMECÂNICO

Para o desenvolvimento deste trabalho, seguiu-se o fluxograma apresentado na Figura 15, com o objetivo de avaliar a capacidade das técnicas DDPG e TD3 no desenvolvimento de um controlador mioelétrico adaptativo de trajetória.

Figura 15 – Fluxograma do processo completo de desenvolvimento do trabalho



Fonte: Elaborado pelo autor.

Como apresentado na Figura 15, o trabalho foi dividido em duas etapas principais: o teste piloto de treinamento e o treinamento do controlador mioelétrico. Cada uma das etapas foi explicada separadamente. Neste capítulo foram apresentados a metodologia, os resultados e as discussões referentes ao teste piloto de treinamento,

enquanto no Capítulo 4 foi apresentado o desenvolvimento do treinamento do controlador mioelétrico para uma prótese de membro superior.

O teste piloto avaliou as capacidades do DDPG no treinamento de um controlador para a execução de diferentes trajetórias. Foi então desenvolvido um modelo biomecânico simplificado do dedo indicador, que possibilitava atuar no torque de cada uma das articulações. O controlador deveria flexionar o dedo completamente em um segundo executando duas trajetórias, uma com os ângulos das articulações variando de forma senoidal e outra linear. Os torques aplicados nas articulações, determinados continuamente, deveriam ser adaptados para executar ambas as trajetórias utilizando entradas de trajetória distintas e com modificações nas condições ambientais. Para execução do treinamento, construiu-se um ambiente de treinamento e ajustou-se o algoritmo DDPG para o uso com múltiplas trajetórias. Por fim, avaliou-se o controlador alterando as entradas e o ambiente de treinamento, verificando se o controlador se adequou às novas condições. Todas as etapas foram feitas utilizando um *Desktop* com processador AMD Ryzen 9 7950X, placa de vídeo NVIDIA GeForce RTX 4090 e 128 GB de memória RAM. As etapas apresentadas do teste piloto já foram previamente publicadas pelo autor no trabalho intitulado “*A New Proposal for Intelligent Continuous Controller of Robotic Finger Prostheses Using Deep Deterministic Policy Gradient Algorithm Through Simulated Assessments*” (RÚBIO; COSTA; VIMIEIRO, 2025).

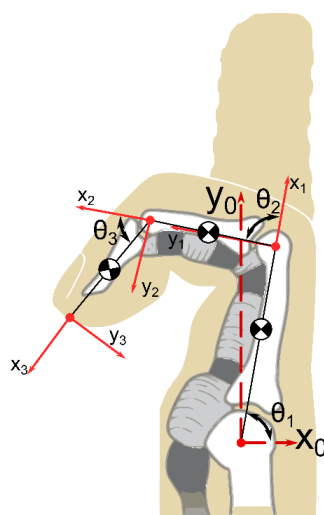
3.1 Metodologia

3.1.1 Ambiente de Treinamento

A primeira etapa do treinamento do controlador de trajetórias do dedo biomecânico consistiu na construção de um ambiente de treinamento, responsável por simular as ações tomadas pelo Agente e determinar os estados futuros e a recompensa da ação. A primeira etapa de elaboração do ambiente de treinamento foi o desenvolvimento do modelo dinâmico do dedo 2. Posteriormente, foi necessário desenvolver a função de recompensa, responsável por quantificar o desempenho do Agente durante suas interações com o ambiente através da comparação entre a posição desejada da ponta do dedo com a posição que está sendo executada, a partir dos comandos de torque definidos pela RNA. Nessa seção serão apresentados com detalhes cada etapa de elaboração do ambiente de treinamento.

Para a construção do modelo matemático, que representará o comportamento dinâmico do dedo, utilizou-se o método proposto por SPONG; HUTCHINSON; VIDYASAGAR, (2020). Cada falange foi considerada como um elo rígido e inextensível e as articulações definidas como juntas de revolução simples, sem amortecimento e sem rigidez. Para determinação dos eixos coordenados, foi considerada a convenção de Denavit-Hartenberg, conforme ilustrado na Figura 16. A fim de simplificar o problema, o dedo foi considerado com movimentos planares, assim os eixos z_0, z_1, z_2 e z_3 foram normalmente localizados em relação ao plano. O eixo coordenado de referência $o_0x_0y_0z_0$, teve sua origem o_0 coincidente com o centro da articulação metacarpofalangeana (junta 1) e o eixo x_0 foi localizado horizontalmente. O eixo coordenado $o_1x_1y_1z_1$ teve sua origem localizada no centro da articulação interfalangeana proximal (junta 2) e o eixo coordenado $o_2x_2y_2z_2$ teve sua origem localizada no centro da articulação interfalangeana distal (junta 3). O último eixo coordenado $o_3x_3y_3z_3$ teve sua origem localizada no centro da ponta do dedo, e foi utilizado para determinar a posição da ponta do dedo no espaço.

Figura 16 – Localização dos eixos coordenados de referência utilizados na construção do modelo matemático do dedo 2



Fonte: Elaborado pelo autor.

As equações de movimento do modelo foram obtidas pelo método de Euler-Lagrange. Assim, as equações diferenciais gerais do movimento foram definidas pela Equação (26), em que θ_i corresponde ao ângulo de cada uma das falanges em relação ao eixo x_{i-1} ; d_{kj} representa os elementos da matriz de inércia $D(\theta)$; c_{ijk} refere-

se aos símbolos de Christoffel; g_k é o termo associado a energia potencial gravitacional total P ; e τ_k é o torque externo aplicado em cada junta.

$$\sum_{j=1}^3 d_{kj}(\theta)\ddot{\theta}_j + \sum_{i=1}^3 \sum_{j=1}^3 c_{ijk}(\theta)\dot{\theta}_i\dot{\theta}_j + g_k(\theta) = \tau_k, \quad k = 1,2,3 \quad (26)$$

A matriz inercial $D(\theta)$ foi definida pela Equação (27), em que m_i representa a massa de cada falange; I_i representa o momento de inércia de cada falange com relação ao eixo z_i ; e J_{vi} é a a matriz Jacobiana geométrica de velocidade linear com relação ao centro de massa de cada falange.

$$D(\theta) = \sum_{i=1}^3 m_i J_{v_i}(\theta)^T J_{v_i}(\theta) + \begin{bmatrix} \sum_{i=1}^3 I_i & \sum_{i=2}^3 I_i & I_3 \\ \sum_{i=2}^3 I_i & \sum_{i=2}^3 I_i & I_3 \\ I_3 & I_3 & I_3 \end{bmatrix} \quad (27)$$

As matrizes Jacobiana geométrica da velocidade linear foram definidas pelas Equações (28), (29) e (30) em que l_i representa o comprimento de cada falange, l_{c_i} a distância entre o centro da articulação anterior e o centro de massa da falange i , as variáveis s_1 , s_{12} e s_{123} são representações das funções $\sin(\theta_1)$, $\sin(\theta_1 + \theta_2)$ e $\sin(\theta_1 + \theta_2 + \theta_3)$ e as variáveis c_1 , c_{12} , c_{123} representam as funções $\cos(\theta_1)$, $\cos(\theta_1 + \theta_2)$ e $\cos(\theta_1 + \theta_2 + \theta_3)$.

$$J_{v_1}(\theta) = \begin{bmatrix} -l_{c_1}s_1 & 0 & 0 \\ l_{c_1}c_1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (28)$$

$$J_{v_2}(\theta) = \begin{bmatrix} -l_1s_1 - l_{c_2}s_{12} & -l_{c_2}s_{12} & 0 \\ l_1c_1 + l_{c_2}c_{12} & l_{c_2}c_{12} & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (29)$$

$$J_{v_3}(\theta) = \begin{bmatrix} -l_1s_1 - l_2s_{12} - l_{c_3}s_{123} & -l_2s_{12} - l_{c_3}s_{123} & -l_{c_3}s_{123} \\ l_1c_1 + l_2c_{12} + l_{c_3}c_{123} & l_2c_{12} + l_{c_3}c_{123} & l_{c_3}c_{123} \\ 0 & 0 & 0 \end{bmatrix} \quad (30)$$

Os símbolos de Christoffel foram expressos pela Equação (31).

$$c_{ijk} = \frac{1}{2} \left(\frac{\partial d_{kj}}{\partial \theta_i} + \frac{\partial d_{ki}}{\partial \theta_j} + \frac{\partial d_{ij}}{\partial \theta_k} \right) \quad (31)$$

Os elementos associados à energia potencial total P foram definidos conforme demonstrado na Equação (32).

$$g_k = \frac{\partial P}{\partial \theta_k} \quad (32)$$

Para resolver o sistema de equações diferenciais e então determinar os estados cinemáticos do modelo (posição e velocidade de cada junta), foi utilizado o método de Runge-Kutta de 4/5 ordem, utilizando um passo de tempo de 10^{-5} s. Adotou-se como posição inicial do modelo os ângulos $\theta_1 = \pi/2$ rad, $\theta_2 = 0$ rad e $\theta_3 = 0$ rad, coincidente com a posição de dedo completamente estendido. Para limitar o espaço de solução foi considerado que cada junta poderia rotacionar no máximo $\pi/2$ rad com relação à posição inicial.

Como parâmetros dimensionais do modelo, foi utilizada uma simplificação na geometria do dedo. Cada falange foi considerada como um tronco de cone elíptico, apresentando uma densidade uniforme de 1160 kgm^{-3} conforme indicado por THOMAS; ZENI; WINTER, (2023). Para determinar os comprimentos e larguras do tronco de cone e as propriedades necessárias para a solução das equações, utilizou-se os dados definidos por VERGARA; AGOST; GRACIA-IBÁÑEZ, (2018), que determinaram as medidas antropométricas das mãos da população adulta masculina da Espanha. Com essas características foram então definidas as propriedades necessárias para a solução do modelo, conforme apresentado na Tabela 1.

Tabela 1 – Valores das propriedades do dedo indicador

Propriedade	Falange Proximal ($i = 1$)	Falange Medial ($i = 2$)	Falange Distal ($i = 3$)
l_i (m)	45,30E-3	25,60E-3	24,50E-3
l_{ci} (m)	28,70E-3	17,80E-3	13,40E-3
m_i (kg)	17,10E-3	6,40E-3	3,60E-3
I_i ($\text{kg} \cdot \text{m}^2$)	3,33E-6	4,36E-7	1,44E-7

Fonte: Elaborado pelo autor.

Além do modelo, foi necessário desenvolver a função de recompensa que avaliou o desempenho do Agente durante o treinamento. Devido a restrição da faixa de operação das articulações de $\pi/2$ rad, foi necessário desenvolver uma função de

penalidade, para situações em que o torque determinado pelo controlador levava o dedo para ângulos de articulação fora da faixa permitida, e uma função de erro de trajetória, utilizada quando o torque definido pelo controlador mantinha o ângulo da junta dentro da faixa de operação, de modo que a função mapeava a diferença entre a posição que a ponta do dedo se encontrava em relação à posição desejada.

A função de penalidade só era utilizada quando o ângulo da junta 1 fosse menor que $\pi/2$ rad ou maior que π rad, ou caso as juntas 2 e 3 apresentassem um ângulo menor que 0 rad ou maior que $\pi/2$ rad. A Equação (33) apresenta a função de penalidade, em que E_i representa o valor que foi excedido da posição limite permitida como demonstrado nas Equações (34) e (35). As penalidades variavam de -10 a -20 de forma a avaliar a extrapolação dos limites até 0,5 rad. Para extrapolações maiores que 0,5 rad o valor de penalidade era aproximadamente -20. Além da penalidade, caso os limites fossem extrapolados, o episódio de treinamento era interrompido indicando uma falha para o Agente durante o treinamento.

$$R_p = 10 \exp\left(\frac{\ln 0,001}{0,5} \sqrt{(E_1)^2 + (E_2)^2 + (E_3)^2}\right) - 20 \quad (33)$$

$$E_1 = \begin{cases} \frac{\pi}{2} - \theta_1, & \theta_1 < \frac{\pi}{2} \\ \theta_1 - \pi, & \theta_1 > \pi \\ 0, & \text{em outros casos} \end{cases} \quad (34)$$

$$E_i = \begin{cases} \theta_i, & \theta_i < 0 \\ \theta_i - \frac{\pi}{2}, & \theta_i > \frac{\pi}{2} \\ 0, & \text{em outros casos} \end{cases}, \quad i = 2,3 \quad (35)$$

Quando a ação determinada pelo Agente não extrapolava os limites de posição da junta, a função de erro de trajetória era fornecida ao algoritmo de treinamento. A Equação (36) representa a função de recompensa, em que E_x e E_y são os erros entre as coordenadas x e y esperadas e executadas pela ação do agente respectivamente. Como pode ser observado, a função de erro de trajetória sempre dava valores maiores que a função de penalidade, garantindo que o modelo possa distinguir ações que levam as juntas para fora dos limites aceitos do espaço de ações das ações que mantem o dedo dentro do espaço de trabalho. Assim, a função de erro de trajetória fornece valores entre 0 e -5, mapeando erros absolutos de posição entre 0 e 0,2 m, mantendo um erro constante para erros maiores que 0,2 m.

$$R = \begin{cases} 5 \exp\left(\frac{\ln 0,001}{0,2} \sqrt{(E_1)^2 + (E_2)^2}\right) - 5, & \frac{\pi}{2} < \theta_1 < \pi \text{ e } 0 < \theta_i < \frac{\pi}{2} \\ R_p, & \text{em outros casos} \end{cases} \quad (36)$$

Todo o ambiente apresentado foi elaborado utilizando a biblioteca Gymnasium, desenvolvida em Python, pela facilidade de integração com os algoritmos de treinamento de RL (BROCKMAN *et al.*, 2016).

3.1.2 Definição das Trajetórias-alvo

Para o teste piloto, duas trajetórias distintas para a execução da flexão do dedo foram escolhidas, na primeira o dedo realizou a flexão completa de forma que a posição angular de cada uma das articulações variasse senoidalmente, já na segunda trajetória, os ângulos das articulações deveriam variar linearmente. A escolha dessas trajetórias foi realizada com o objetivo de avaliar o comportamento do controlador em duas situações dinâmicas distintas. Na primeira, a velocidade do movimento decaía até zero ao final do movimento (trajetória senoidal), já na segunda, a velocidade do movimento era mantida constante durante todo o deslocamento (trajetória linear). Em ambas as trajetórias as articulações partiam da posição de repouso executavam a flexão completa em 1 s. Simulando a janela de sinal de sEMG, o valor de referência de posição era atualizado a cada 40 ms, em vez de todo o passo de treinamento.

Para a trajetória senoidal, a variação da rotação e velocidade angular de cada junta foi determinada pela Equação (37), em que t representa o tempo.

$$\begin{cases} \theta_1 = \frac{\pi}{2} \sin\left(\frac{\pi}{2} t\right) + \frac{\pi}{2} \\ \theta_2 = \theta_3 = \frac{\pi}{2} \sin\left(\frac{\pi}{2} t\right) \\ \dot{\theta}_1 = \dot{\theta}_2 = \dot{\theta}_3 = \left(\frac{\pi}{2}\right)^2 \cos\left(\frac{\pi}{2} t\right) \end{cases} \quad (37)$$

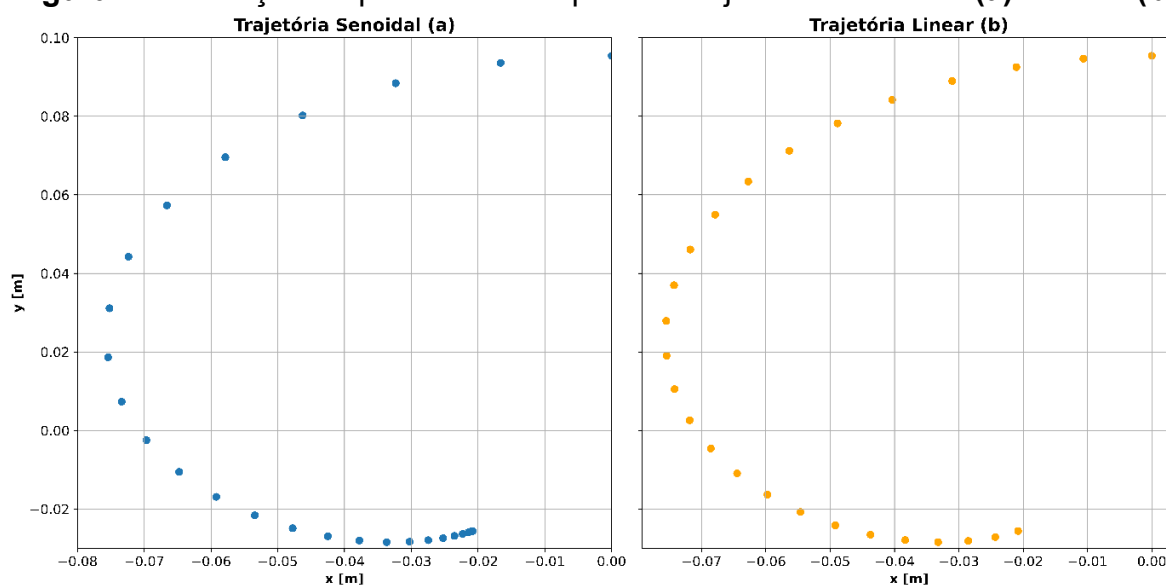
Para a trajetória linear, a variação da rotação angular e da velocidade de cada junta foram determinadas pela Equação (38), em que t representa o tempo.

$$\begin{cases} \theta_1 = \frac{\pi}{2}(t + 1) \\ \theta_2 = \theta_3 = \frac{\pi}{2} t \\ \dot{\theta}_1 = \dot{\theta}_2 = \dot{\theta}_3 = \frac{\pi}{2} \end{cases} \quad (38)$$

Com as variações da rotação e da velocidade angular de cada junta definidas, foi necessário determinar a matriz de transformação homogênea do modelo para determinar a posição da ponta do dedo para cada passo de tempo conforme a Equação (39). A Figura 17 apresenta a posição esperada da ponta do dedo para a trajetória senoidal (Figura 17a) e linear (Figura 17b).

$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = \begin{Bmatrix} l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3) \\ 0 \end{Bmatrix} \quad (39)$$

Figura 17 – Posição da ponta do dedo para as trajetórias: senoidal (a) e linear (b)



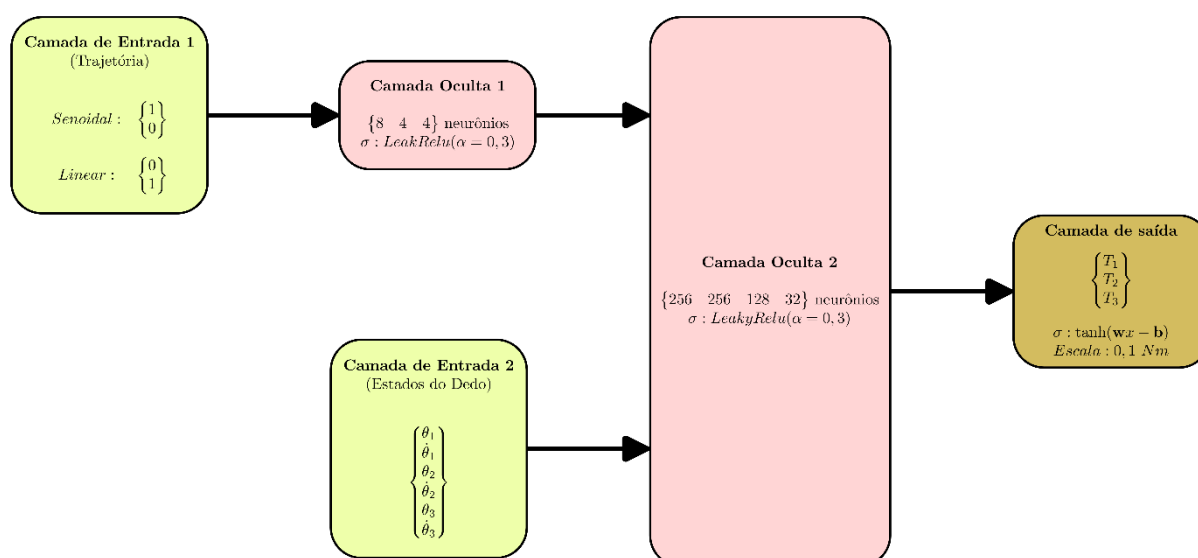
Fonte: Elaborado pelo autor.

3.1.3 Treinamento do Controlador

O controlador proposto neste trabalho almejava combinar duas entradas distintas, os estados cinemáticos da prótese e a seleção de trajetória desejada. Para o treinamento, a seleção de trajetória foi binária, ou seja, para a trajetória senoidal a entrada utilizada era $\{1 \ 0\}^T$, enquanto para a trajetória linear $\{0 \ 1\}^T$. As entradas binárias foram escolhidas para simular as saídas dos classificadores de sinais, que utilizam função de ativação do tipo softmax na camada de saída. Como o objetivo do teste piloto era emular uma situação, utilizou-se a saída de um classificador de movimentos que utiliza sinais de sEMG como entrada de seleção de trajetórias. As entradas de treinamento deveriam ser similares às saídas do classificador de movimentos, garantindo que a soma delas fosse um e que dois neurônios, relativos a cada movimento predito, fossem utilizados.

Para obter a arquitetura simplificada foi utilizada uma MLP, em que as entradas dos estados cinemáticos foram separadas das entradas de seleção de trajetória, como apresentado na Figura 18. As entradas de seleção de trajetórias passaram por 3 camadas ocultas com 8, 4 e 4 neurônios (Camada Oculta 1). A saída da Camada Oculta 1 foi concatenada com a entrada referente aos estados da prótese, e foram conectados com camadas ocultas com 256, 256, 128 e 32 neurônios (Camada Oculta 2). Todos os neurônios das camadas ocultas utilizaram a função de ativação do tipo *leaky* ReLU (Equação (14)) com parâmetro α de 0,3, valor padrão da implementação da biblioteca Keras (CHOLLET, 2015). Para camada de saída foram utilizados três neurônios com função de ativação do tipo tangente hiperbólica escalonados em 0,1 Nm, utilizados com valor de torque aplicado em cada uma das juntas. A função do tipo tangente hiperbólica foi utilizada para garantir que o valor de torque variasse de $-0,1$ Nm a $0,1$ Nm com uma distribuição não linear, garantindo que os valores limites raramente eram atingidos. Os parâmetros de inicialização foram aleatoriamente definidos limitando os parâmetros dos neurônios de saída entre $-0,05$ e $0,05$, garantindo a não extrapolação das respostas iniciais da rede.

Figura 18 – Arquitetura de RNA utilizada como o Ator no modelo de controle

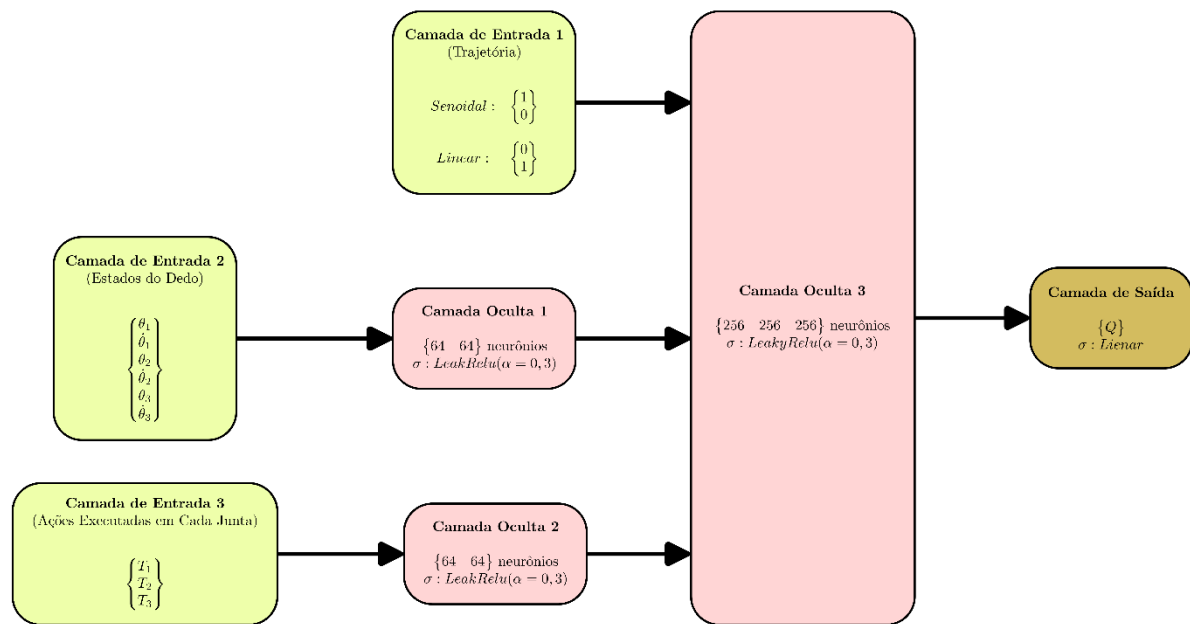


Fonte: Elaborado pelo autor.

Para a execução do DDPG também foi necessário desenvolver a RNA que atuará como Crítico durante o treinamento. Essa rede precisou de três entradas para funcionar corretamente. A primeira entrada foi referente à seleção da trajetória desejada, enquanto a segunda recebia os estados cinemáticos do modelo e a última

entrada relativa às ações que o Ator tomou durante o treinamento. Para garantir uma arquitetura simplificada, foi utilizada novamente uma MLP, em que a segunda entrada foi conectada a duas camadas ocultas com 64 neurônios, assim como a terceira entrada. As saídas das camadas ocultas foram concatenadas com a primeira, e então foram conectadas a outras três camadas ocultas com 256 neurônios. A saída da RNA do Crítico consistiu em apenas um neurônio com função de ativação linear, responsável por determinar a função valor para o treinamento, uma vez que a distribuição dos valores da função valor deve ser uniforme. Para todos os neurônios das camadas ocultas, foi utilizada a função de ativação do tipo *leaky* ReLU (Equação (14)), com parâmetro α de 0,3. A Figura 19 representa a arquitetura utilizada no Crítico.

Figura 19 – Arquitetura de RNA utilizada como Crítico no algoritmo DDPG durante o treinamento



Fonte: Elaborado pelo Autor.

Para o treinamento, algumas modificações foram feitas no algoritmo padrão do DDPG. Durante o treinamento, os episódios foram apresentados sequencialmente, assim uma época de treinamento foi considerada como a sequência de um episódio de trajetória senoidal seguido de um episódio de trajetória linear. Outra alteração foi na forma como a posição desejada era tratada, como explicado anteriormente, para simular um sistema com sEMG as posições desejadas eram alteradas a cada 40 ms, enquanto o passo temporal de execução da tarefa foi de 1 ms. Assim, o Ator teria 40 tentativas para levar a ponta do dedo de uma posição para outra, e totalizando 1000 passos totais para cada episódio.

O Algoritmo 3 foi empregado para o treinamento do controlador, onde a apresentação sequencial de cada uma das trajetórias é observada na linha 6. A cada 50 episódios de treinamento, ou caso em algum dos episódios a recompensa superasse o valor de -650, o processo de verificação do agente era iniciado, como demonstrado na linha 30. O processo de verificação consistia em executar o controlador no ambiente, porém sem ruído e sem atualizar seus parâmetros, a fim de verificar a resposta real do Ator. O treinamento só foi finalizado quando o Ator conseguiu completar ambas as trajetórias e obteve recompensas superiores a -650 pontos. O valor alvo de recompensa utilizado foi de -650 pontos, pois representava um erro absoluto médio de cerca de 3 mm entre a trajetória executada e a trajetória desejada. Outro ponto importante implementado no algoritmo é a separação das amostras utilizadas tanto na memória de repetição como nas miniamostras (linha 19). Essa distinção acontece para limitar as amostras de falha do Ator como sendo apenas 20 % das amostras armazenadas (2×10^4) e da miniamostra (6) utilizada no treinamento. Como demonstrado no trabalho de LI, Z. et al., (2020), quando o ambiente de treinamento apresenta penalidades, a eficiência de treinamento é reduzida caso apenas amostras de falha sejam utilizadas, sendo assim aconselhável que a cada atualização a miniamostra possua tanto amostras de falha como de não falha, sendo a última em maior quantidade.

Para garantir a etapa de extrapolação e exploração do treinamento, foi utilizado como ruído o processo de Ornstein-Uhlenbeck, com desvio padrão igual à 10^{-5} , para gerar o ruído das ações tomadas pelo Ator, conforme indicado na documentação das bibliotecas TensorFlow e Keras para Python utilizadas no desenvolvimento do trabalho (CHOLLET, 2015; MARTÍN ABADI *et al.*, 2015). Um resumo dos parâmetros de treinamento utilizados está apresentado na Tabela 2.

Tabela 2 – Parâmetros utilizados no treinamento do controlador no teste piloto

Parâmetro	Valor
Taxa da aprendizagem do ator	10^{-4}
Taxa de aprendizagem do crítico	10^{-3}
Quantidade de amostras armazenadas	10^5 (2×10^4 falhas e 8×10^4 não falhas)
Tamanho da miniamostra	32 (6 falhas e 28 não falhas)
Taxa de suavização para atualização das redes alvo (τ)	10^{-5}
Taxa de desconto (γ)	0,99
Recompensa alvo mínima	-650
Otimizador	Adam

Fonte: Elaborado pelo autor.

Algoritmo 3: DDPG adaptado para o treino do controlador do dedo 2.

```

1  Inicialize o Crítico  $Q(i, s, a|\theta^Q)$  e o Ator  $\mu(i, s|\theta^\mu)$  com  $-0,05 \leq \theta^\mu \leq 0,05$ 
2  Inicialize as redes alvo  $Q'$  e  $\mu'$  com os parâmetros  $\theta^{Q'} \leftarrow \theta^Q, \theta^{\mu'} \leftarrow \theta^\mu$ 
3  Inicialize a memória de repetição  $B$ 
4  Inicialize o contador de episódios ( $E \leftarrow 0$ )
5  enquanto (recompensa do Ator na trajetória senoidal ( $R_s$ ) e linear ( $R_l$ )  $\leq -650$ ) e (pas-
    sos de executados na trajetória senoidal ( $T_s$ ) e linear ( $T_l$ )  $\leq 1000$ ) faça:
6  para episódio em [senoidal, linear] faça:
7  | Inicie a variável de recompensa do episódio:  $R \leftarrow 0$ 
8  | Receba os estados iniciais  $s_0$  do ambiente
9  | se episódio = senoidal então:
10 | | Determine a entrada de trajetória sendo:  $i \leftarrow \{1 \ 0\}^T$ 
11 | senão:
12 | | Determine a entrada de trajetória sendo:  $i \leftarrow \{0 \ 1\}^T$ 
13 | fim-se
14 | para t de 1 até T=1000 faça:
15 | | Seleccione uma ação de acordo com a política atual e o ruído usando:
16 | |  $a_t \leftarrow clip(\mu(i_t, s_t|\theta^\mu) + \epsilon, a_{min}, a_{max}), \quad \epsilon \approx \mathcal{N}(0, \sigma)$ 
17 | | Escolha a posição alvo para os próximos 40 passos de acordo com a trajetória
18 | | de treinamento
19 | | Execute a ação  $a_t$  e observe a recompensa  $r_t$ , o próximo estado  $s_{t+1}$  e se o es-
20 | | tado extrapolou os limites de rotação da junta ( $f_t$ )
21 | | Armazene  $(i_t, s_t, a_t, r_t, i_{t+1}, s_{t+1}, d_t, f_t)$  em  $B$ 
22 | | Pegue uma miniamostra aleatória com 6 passos de falha e 26 passos sem fa-
23 | | lha de transição  $(i_n, s_n, a_n, r_n, i_{n+1}, s_{n+1}, f_t)$  de  $B$ 
24 | | Determine  $\delta_{n+1} = r_n + \gamma Q'(i_{n+1}, s_{n+1}, \mu'(i_{n+1}, s_{n+1}|\theta^{\mu'})|\theta^{Q'})$ 
25 | | Atualize os parâmetros do crítico com um passo de gradiente descendente
26 | | usando:
27 | | 
$$\nabla_{\theta^Q} \frac{1}{N} \sum_n (Q(i_n, s_n, a_n|\theta^Q) - \delta_{n+1})^2$$

28 | | Atualize a política do ator com um passo de gradiente ascendente usando:
29 | | 
$$\nabla_{\theta^\mu} \frac{1}{N} \sum_n Q(i_n, s_n, \mu(i_n, s_n|\theta^\mu)|\theta^Q)$$

30 | | Atualize as redes alvo:  $\theta^{Q'} \leftarrow \tau\theta^Q + (1 - \tau)\theta^{Q'}$  e  $\theta^{\mu'} \leftarrow \tau\theta^\mu + (1 - \tau)\theta^{\mu'}$ 
31 | | se  $f_t$  é terminal:
32 | | | Interromper
33 | | fim-se
34 | | Atualize a recompensa do agente:  $R \leftarrow R + r_t$ 
35 | fim-para
36 Atualize:  $E \leftarrow E + 1$  e  $S \leftarrow t$ 

```

Algoritmo 3(continuação): DDPG adaptado para o treino do controlador do dedo 2.

```

30   |   se  $E$  é múltiplo de 50 ou ( $R \geq -650$  e  $S = 1000$ ):
31   |   | Avalie o agente no ambiente sem ruído e determine  $R_s, S_s, R_l$  e  $S_l$ 
32   |   fim-se
33   fim-para
34 fim-enquanto

```

3.1.4 Avaliação Computacional do Controlador

Para avaliar o controlador virtualmente, o modelo desenvolvido novamente foi utilizado, porém agora sem o uso da função de recompensa. Três avaliações foram executadas, a primeira foi uma avaliação de validação do treinamento, em que foram verificadas as respostas do controlador para as entradas de trajetórias utilizadas durante o treinamento. Na segunda avaliação, foram utilizadas entradas proporcionais de trajetória, em que foram escolhidas como proporção 25% da trajetória senoidal ($i = \{0,25 \ 0,75\}^T$), 50% senoidal ($i = \{0,5 \ 0,5\}^T$) e 75% senoidal ($i = \{0,75 \ 0,25\}^T$). Para a última avaliação realizada foram feitas alterações nos parâmetros do ambiente de treinamento avaliando se o controlador se adaptava a essas mudanças. Para isso o modelo foi escalonado em três volumes distintos, sendo avaliadas as respostas para o modelo com 90%, 110% e 125% do volume original. As escalas de 90% e de 110 % foram determinadas através das variações dimensionais fisiológicas conforme VERGARA; AGOST; GRACIA-IBÁÑEZ, (2018), já o valor de 125% foi adotado para avaliar um modelo que extrapolava os parâmetros simulando um cenário com maior massa e momento de inércia.

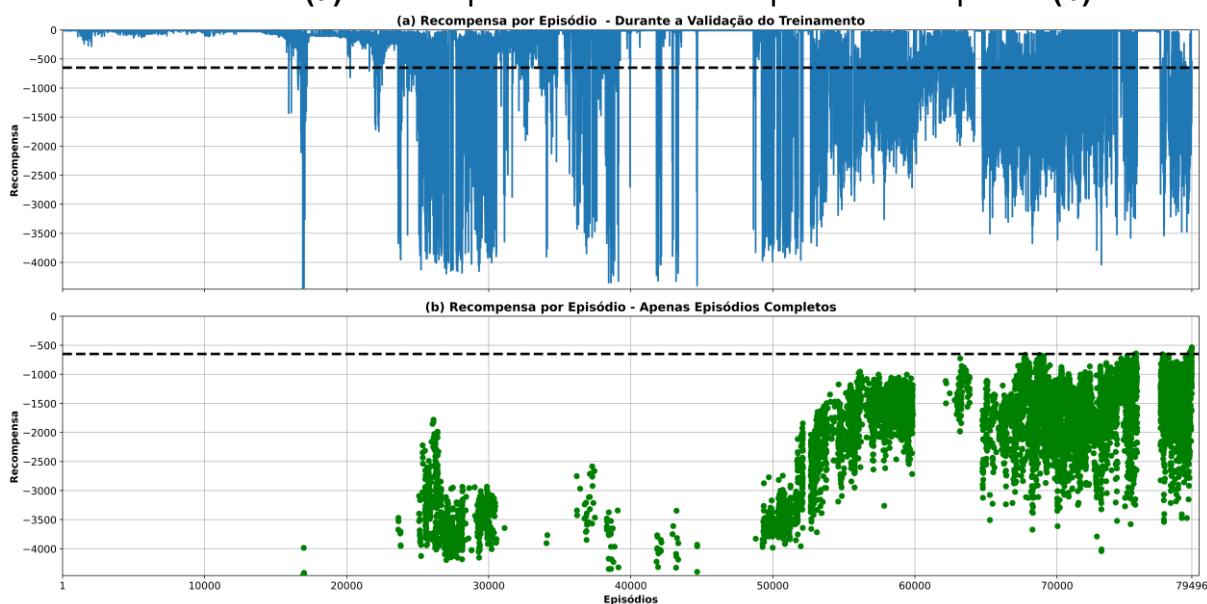
3.2 Resultados e Discussão

Os resultados do teste piloto foram divididos em três seções. Primeiramente foram apresentadas as análises sobre a validação do treinamento utilizando as entradas binárias de seleção das trajetórias senoidal e linear. Posteriormente foram apresentadas as avaliações referentes as entradas proporcionais de seleção de trajetórias M2575, M5050 e M7525. Por fim, apresentam-se analisado comportamento do controlador utilizando o modelo em escalas volumétricas de 90%, de 110% e de 125%.

3.2.1 Respostas Para as Entradas Senoidal e Linear

Inicialmente foi observada a evolução do treinamento do Ator com a avaliação da recompensa em cada episódio (Figura 20). Observa-se na Figura 20a, que inicialmente o Ator apresentou uma grande dificuldade em completar a tarefa, sendo utilizados aproximadamente 16000 episódios para completá-la pela primeira vez. Após a primeira execução completa da tarefa, o treinamento apresentou sequências de episódios em que a tarefa foi completada sequencialmente, como apresentado no gráfico de dispersão da recompensa dos episódios completos (Figura 20b). Ao final do treinamento, após 79496 episódios (ou 39748 épocas), o Ator atingiu a recompensa-alvo estipulada de -650 (linha tracejada representada no gráfico) pontos, marcando assim o término do treinamento.

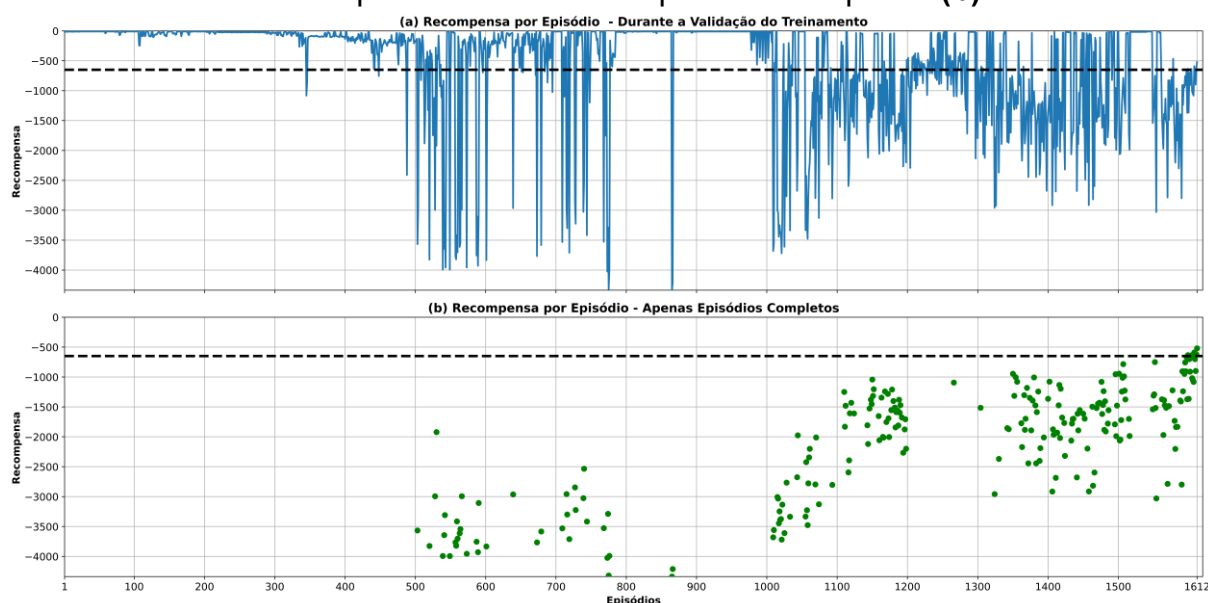
Figura 20 – Evolução da função de recompensa ao longo dos episódios de treinamento **(a)** e recompensas obtidas nos episódios completos **(b)**



Fonte: Elaborado pelo autor.

Para garantir que o Ator de fato estava evoluindo durante o treinamento a cada 50 épocas, ou caso a recompensa do episódio completo fosse superior à recompensa-alvo, uma validação das ações do Ator era realizada, em que ele executava a tarefa sem a presença do ruído para ambas as entradas. A Figura 21 apresenta a evolução da função de recompensa durante os episódios de validação do Ator. Como demonstrado na Figura 21a, diversos episódios de validação não foram concluídos, apresentando maior consistência na finalização da tarefa nos últimos episódios de validação, onde o Ator atingiu a capacidade de completar a tarefa com uma recompensa superior a recompensa-alvo durante a validação (-650 – linha tracejada representada no gráfico), como ilustrado na Figura 21b. Com os dados de validação de treinamento, verificou-se que o controlador foi devidamente treinado e adquiriu a capacidade de solucionar a tarefa.

Figura 21 – Evolução da recompensa do Ator durante os episódios de validação **(a)** e recompensas obtidas nos episódios completos **(b)**

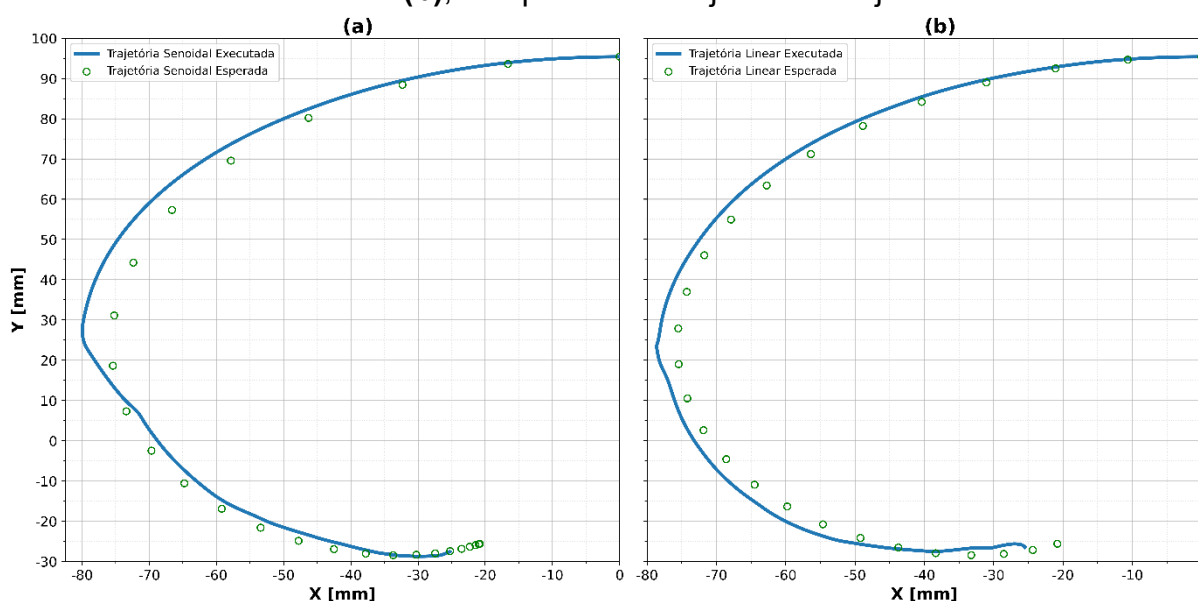


Fonte: Elaborado pelo autor.

Avaliando as respostas do controlador, é observado na Figura 22 que ele foi capaz de realizar ambas as trajetórias aproximadamente, tendo melhor desempenho na execução da trajetória linear. Durante a trajetória senoidal (Figura 22a) a distância média entre a posição desejada do dedo e a executada foi de $3,99 \pm 2,90$ mm, enquanto que para a trajetória linear (Figura 22b) a distância média foi de $3,22 \pm 1,42$ mm, indicando quantitativamente o melhor desempenho. Verificou-se que o erro de trajetória aumentava conforme a distância com relação à origem no eixo x aumentava,

sendo o maior erro observado no ponto -80,25 mm para a trajetória senoidal e -78,24 mm na trajetória linear, pontos onde a rede realiza correções mais abruptas a trajetória. Dividindo a trajetória em duas etapas, antes da correção abrupta e após a correção abrupta, observa-se que, antes, o erro médio era de $5,96 \pm 3,01$ mm para a trajetória senoidal e $3,40 \pm 1,66$ mm para a linear. Após a correção, o controlador ajusta a trajetória e a distância média é de $3,14 \pm 2,40$ mm para a trajetória senoidal e $3,09 \pm 1,20$ mm para a linear.

Figura 22 – Trajetórias realizadas pelo controlador para as entradas senoidais (a) e lineares (b), comparadas à trajetória desejada



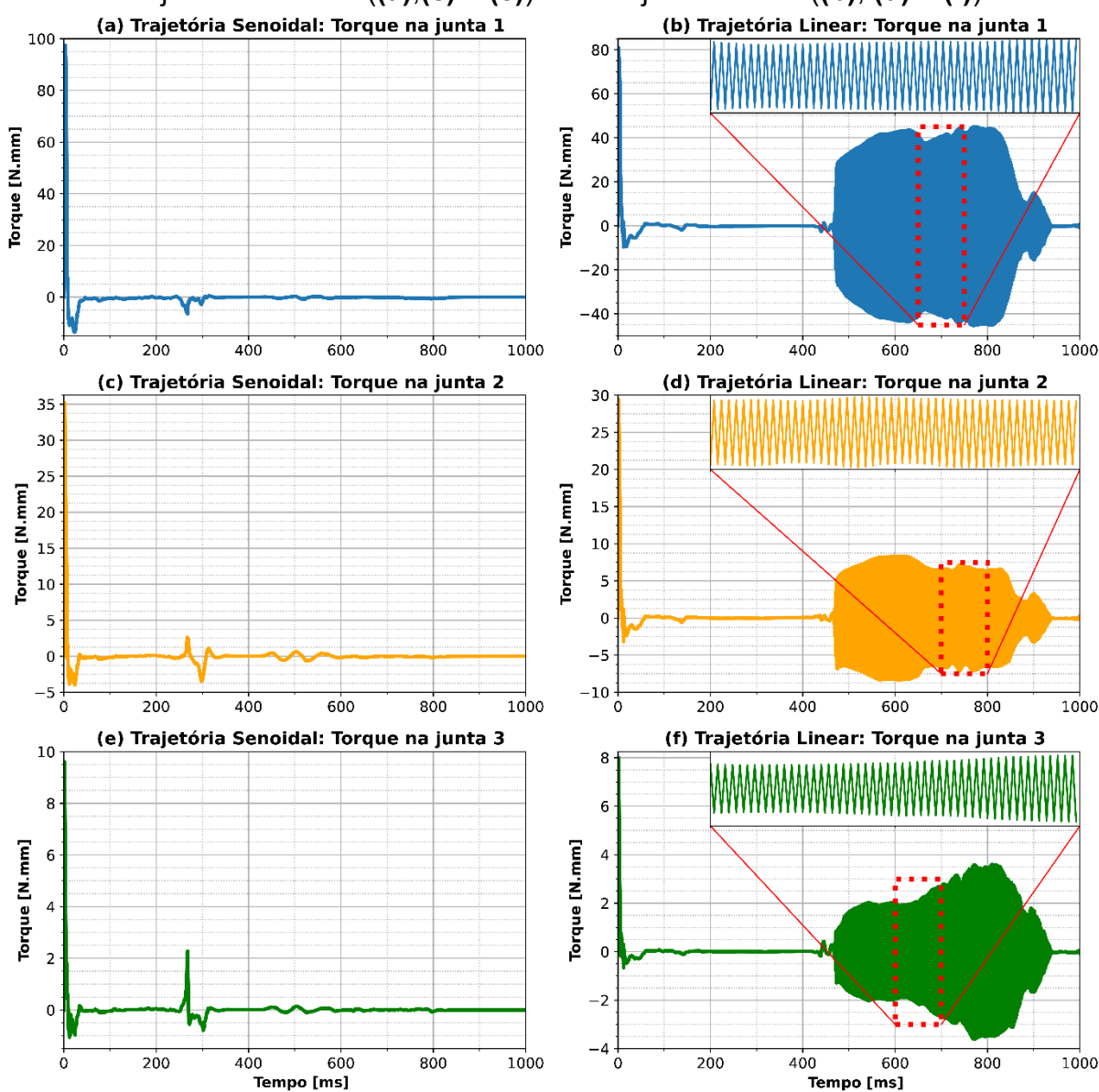
Fonte: Elaborado pelo autor.

Outro ponto notado nas trajetórias desenvolvidas é o ponto final onde a rede encerrou o movimento, em ambas as trajetórias, foi antes do ponto final esperado (-20.8 mm, -25.6 mm), encerrando o movimento no ponto (-25.3 mm, -27.6 mm) durante a execução da trajetória senoidal, representando uma distância de 4,88 mm, e no ponto (-25.5 mm, -26.4 mm) na execução da trajetória linear, representando um erro absoluto de 4,77 mm. Quanto aos erros máximos absolutos, foi observado um erro de 16,24 mm e de 10,56 mm para a execução das trajetórias senoidal e linear respectivamente, enquanto os erros mínimos absolutos foram 0,081 mm e 0,225 mm.

Na Figura 23 são apresentados os torques aplicados nas juntas, definidos pelo controlador, durante a execução das trajetórias no ambiente virtual sem ruído. Observou-se que o maior torque ocorreu na primeira junta com um valor de 97,5 Nmm e 80,9 Nmm na execução das trajetórias senoidal e linear. Após o pico de torque

observou-se que apenas valores baixos de torque foram utilizados, a fim de estabilizar o movimento. A maior diferença observada nas atuações ocorreu quando o ponto máximo da coordenada x foi atingido, 250 ms para a trajetória senoidal e 425 ms para a trajetória linear, a partir desse ponto as atuações mudaram abruptamente. Na trajetória senoidal ocorreram pequenas oscilações (Figura 23a, Figura 23c e Figura 23e) seguidas por momentos de torque zero. Já para a trajetória linear, oscilações constantes até o final do movimento foram realizadas (Figura 23b, Figura 23d e Figura 23f).

Figura 23 – Torques aplicados nas juntas de revolução durante a execução da trajetória senoidal ((a),(c) e (e)) e da trajetória linear ((b), (d) e (f))



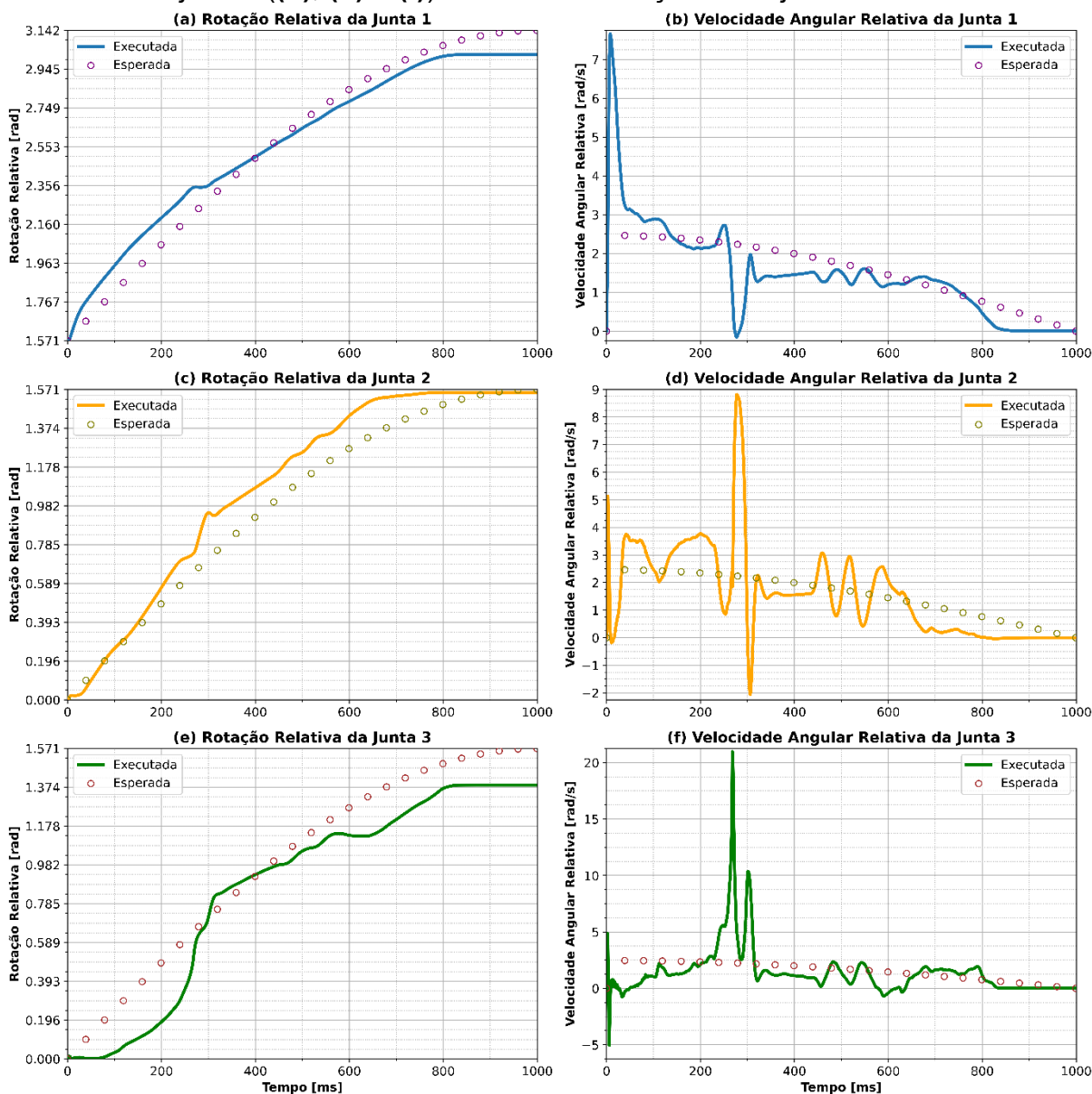
Fonte: Elaborado pelo autor.

A mudança de estratégia de controle observada na Figura 23 pode estar relacionada ao padrão de velocidade esperado para cada uma das trajetórias de treinamento. Na trajetória senoidal (Figura 23a, Figura 23c e Figura 23e) esperava-se que a velocidade se comporte como uma função cosseno, ou seja, no primeiro passo de tempo o dedo deveria sair do repouso para a primeira posição com uma velocidade de 2,47 rad/s para então reduzir a zero no final do movimento. Essa característica esperada na velocidade foi identificada pela rede, pois o torque inicial foi elevado para acelerar o movimento e então conforme posição final desejada se aproximava o valor de torque reduziu-se à aproximadamente zero. Já para a trajetória linear (Figura 23b, Figura 23d, Figura 23f) era esperado que a velocidade das juntas fosse constante em 1,57 rad/s, o que explica o padrão oscilatório apresentado, uma vez que para manter o movimento com uma velocidade constante o sistema necessitava fazer corrigir qualquer mudança na trajetória, assim compensando as respostas dinâmicas do dedo a cada ação. Essa análise evidenciou que a RNA utilizada foi capaz de diferenciar quais seriam os padrões de movimento para cada trajetória selecionada para melhor cumprir o padrão esperado.

Analisando os valores das variáveis de estado durante a execução da trajetória senoidal (Figura 24) observa-se que a rotação relativa no decorrer do tempo foi próxima das curvas esperadas. A Figura 24a e a Figura 24e mostram que a rotação de ambas as juntas foram constantes após 800 ms em uma posição angular inferior à esperada, entretanto a rotação relativa da junta 2 (Figura 24c) atinge a posição esperada antes do término do movimento (800 ms). Esses estados mostram que um dos motivos de a trajetória não atingir a posição final alvo foi o erro entre as posições executadas nas juntas 1 e 3. A velocidade angular relativa na junta 1 (Figura 24b) obteve o maior valor, tendo um pico nos passos iniciais do movimento, com magnitude de 7,66 rad/s, devido ao alto torque na junta para superar as forças inerciais e para que o movimento se iniciasse. Como a junta 1 foi responsável por mover todo o dedo, consequentemente ela necessitou de um impulso. Após o pico, a velocidade reduz e se mantém próximo ao esperado, tornando-se zero a 850 ms. Observando as juntas 2 e 3 (Figura 24d e Figura 24f) é notável que o pico de velocidade ocorre a 275 ms, ou seja, após a correção de trajetória observada na Figura 23a, o que indica que esse pico é o responsável pela correção abrupta observada. Assim como na junta 1 as

juntas 2 e 3 apresentam um padrão próximo do esperado apresentando uma velocidade igual a zero por volta de 850 ms.

Figura 24 – Rotação relativa das juntas ((a), (c) e (e)) e velocidade angular relativa das juntas ((b), (d) e (f)) durante a execução da trajetória senoidal

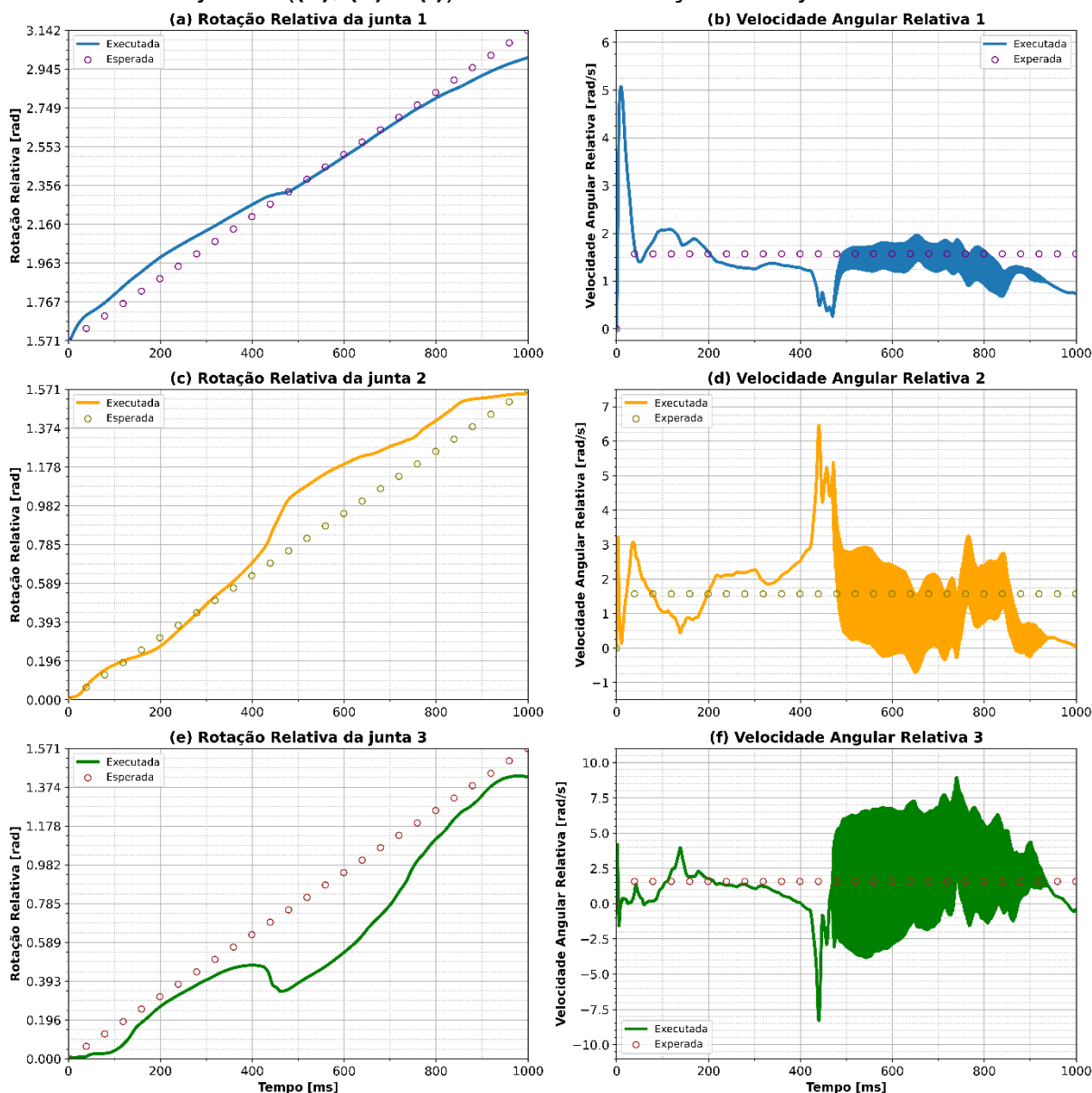


Fonte: Elaborado pelo autor.

Na Figura 25, apresentam-se as variáveis de estado durante a execução da trajetória linear, onde verificou-se que a rotação mais próxima da esperada ocorreu na junta 1 (Figura 25a), entretanto a posição final foi inferior à desejada. A rotação relativa da junta 2 (Figura 25c) também foi próxima à rotação-alvo no início do movimento e ficou estagnada em uma posição próxima à ideal. A rotação relativa da junta 3 (Figura 25e) apresentou alteração significativa em sua linearidade a 425 ms, tempo em que a

correção da trajetória é observada na Figura 22b. Esse comportamento indica que a correção da trajetória foi motivada pelo movimento da junta 3.

Figura 25 – Rotação relativa das juntas ((a), (c) e (e)) e velocidade angular relativa das juntas ((b), (d) e (f)) durante a execução da trajetória linear



Fonte: Elaborado pelo autor.

Avaliando as velocidades angulares (Figura 25b, Figura 25d e Figura 25f), foi observado um comportamento inicial da junta 1 similar ao do movimento senoidal (Figura 24b), entretanto com um pico de velocidade menor (5.07 rad/s). Diferentemente do observado na trajetória senoidal, a maior velocidade ocorreu na junta 3 com um valor de 8.94 rad/s. O comportamento observado nos torques aplicados nas juntas (Figura 23) também foi observado nas velocidades das juntas, apresentando um

padrão oscilatório em torno da velocidade constante esperada. Observa-se que esse padrão começa quando o controlador corrige a trajetória executada a 425 ms.

Com as variáveis de estado apresentadas pode-se notar que o controlador apresentou uma maior dificuldade em atingir o ponto final da trajetória senoidal, quando comparada com a trajetória linear (Figura 24, Figura 25). Uma das explicações para essa observação se deve ao fato de todos os pontos da trajetória linear serem igualmente espaçados, assim, entre os pontos finais o erro sempre aumentava, o que favorecia a rede na tentativa de atingi-los. Já para a trajetória senoidal, os pontos finais encontram-se muito próximos, dificultando para a RNA aprender os pontos finais desejados, uma vez que a distância ao ponto alvo pouco aumentava.

3.2.2 Avaliação com as Entradas de Trajetória Proporcionais

Outra avaliação feita, no controlador treinado, foi a sua resposta às entradas proporcionais nos neurônios responsáveis pela seleção de trajetória. Três proporções foram utilizadas nos neurônios das entradas senoidais e lineares respectivamente: 0,25 por 0,75 (M2575); 0,5 por 0,5 (M5050); e 0,75 por 0,25 (M7525). A Figura 26 apresenta a trajetória da ponta do dedo de cada uma das entradas utilizadas. Como esperado, a proporção M2575 (Figura 26a) apresentou maior similaridade com a entrada linear binária (Figura 22b), apresentando uma correção mais suave. Para as entradas M5050 e M7525 (Figura 26b e Figura 26c) as trajetórias foram mais próximas da entrada senoidal binária (Figura 22a). A correção apresentada com as entradas binárias também foi observada nas entradas proporcionais, a partir do ponto máximo da coordenada x .

Para quantificar as diferenças e as similaridades entre as trajetórias, foi calculado o erro médio absoluto entre a trajetória executada com as entradas proporcionais com as trajetórias alvo, linear e senoidal, para cada passo de tempo de execução da tarefa, apresentado na Tabela 3. A definição da trajetória esperada para as entradas proporcionais, foi obtida a partir da proporcionalidade entre posição angular das juntas da trajetória linear e senoidal, semelhantes às utilizadas na entrada da avaliação. Corroborando com o observado na Figura 26, o erro da entrada M2575 foi maior para a trajetória senoidal do que para a linear, quantificando a similaridade observada. Para a entrada M5050 foi observado que os erros comparados com trajetórias-alvo linear e senoidal foram os maiores, com uma pequena tendência de proximidade com a

trajetória senoidal, como observado na Figura 26. Com a entrada M7525 a distância foi mais próxima da trajetória senoidal alvo do que linear, mostrando a similaridade.

Figura 26 – Trajetórias executadas para entradas proporcionais de trajetória: 25% senoidal (a), 50% senoidal (M5050) (b) e 75% senoidal (M7525) (c)

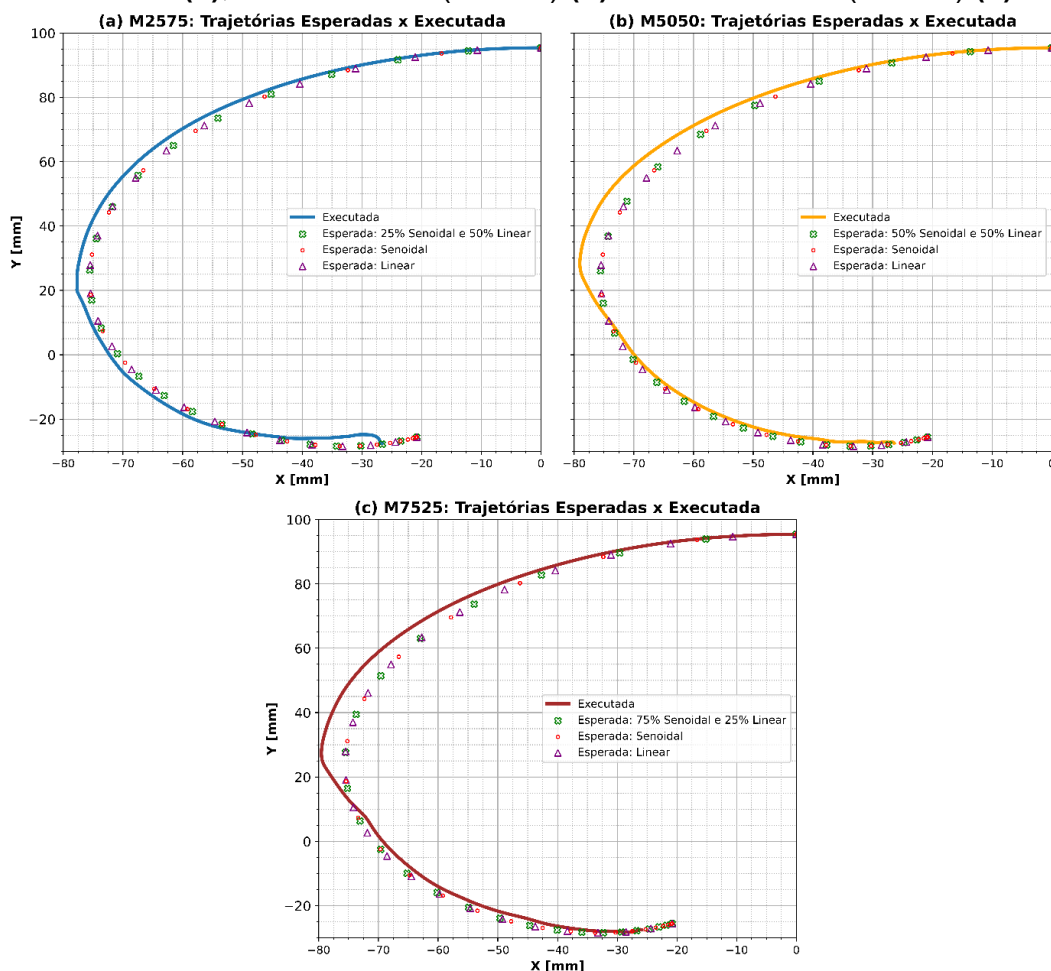


Tabela 3 – Comparação dos erros médios absolutos das trajetórias executadas pelo controlador em relação às trajetórias-alvo

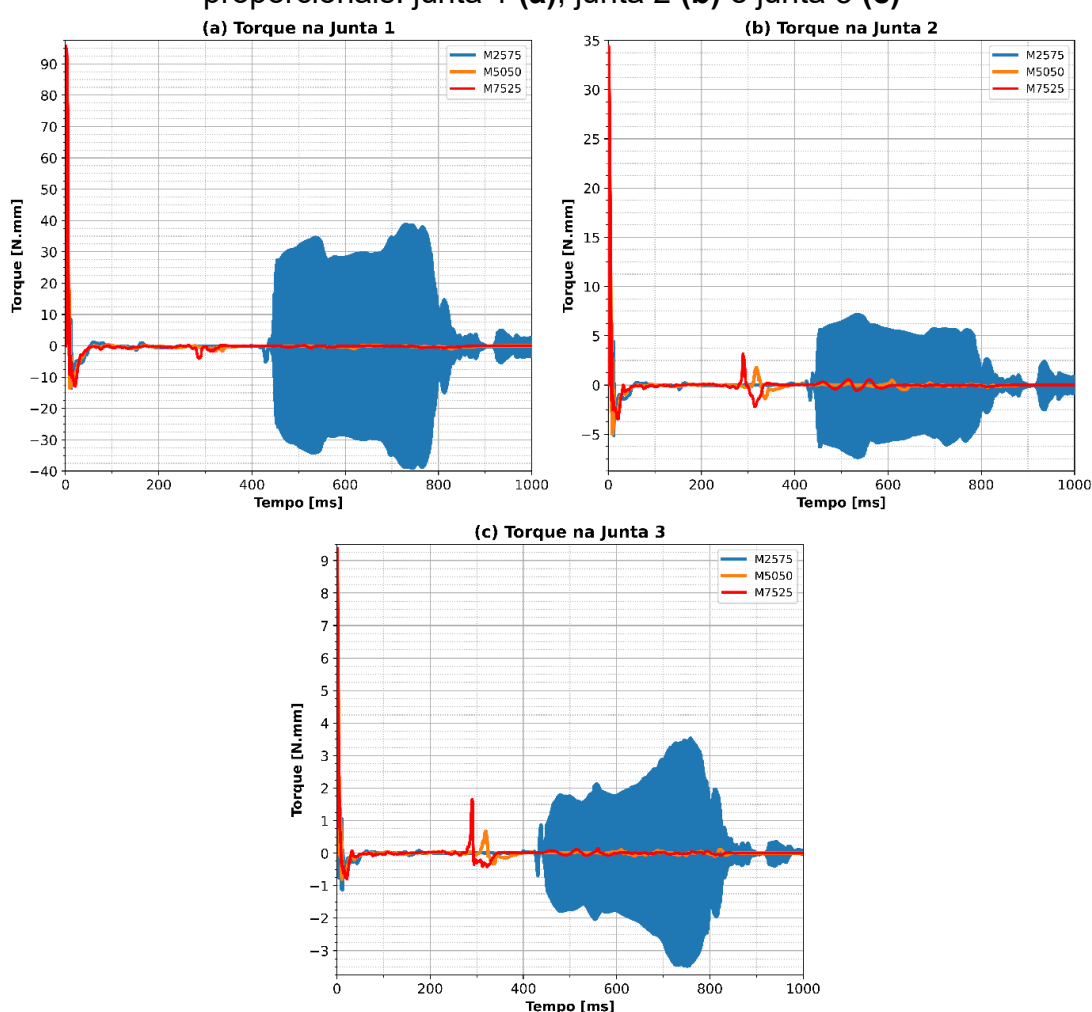
Entrada	Trajetória				
	M2575	M5050	M7525	Linear	Senoidal
M2575	$4,952 \pm 2,467$	—	—	$4,428 \pm 2,146$	$22,53 \pm 10,91$
M5050	—	$5,976 \pm 2,973$	—	$17,92 \pm 9,607$	$8,178 \pm 4,096$
M7525	—	—	$5,001 \pm 3,100$	$22,40 \pm 12,43$	$4,330 \pm 2,429$

Fonte: Elaborado pelo autor.

Observando os torques em cada junta (Figura 27), pode-se notar a similaridade dos comportamentos da entrada M2575 com a entrada linear (Figura 23b, Figura 23d e Figura 23e), enquanto o comportamento das entradas M5050 e M7525 apresentaram comportamento próximo ao da entrada senoidal da entrada senoidal (Figura 23a,

Figura 23c e Figura 23e). Verifica-se nessas duas entradas que a correção aconteceu coincidentemente com o pico de torque observado entre 250 e 350 ms, próximo ao ponto de correção da entrada senoidal, porém com valores menores. Já para a entrada M2575 observa-se que o comportamento oscilatório observado a partir de 400 ms, coincidente com a correção de trajetória observada. Esse comportamento demonstra que, de fato, o padrão oscilatório está relacionado com valores elevados de entrada no neurônio responsável pela seleção da trajetória linear.

Figura 27 – Torque aplicado às juntas durante a execução das entradas proporcionais: junta 1 (a), junta 2 (b) e junta 3 (c)

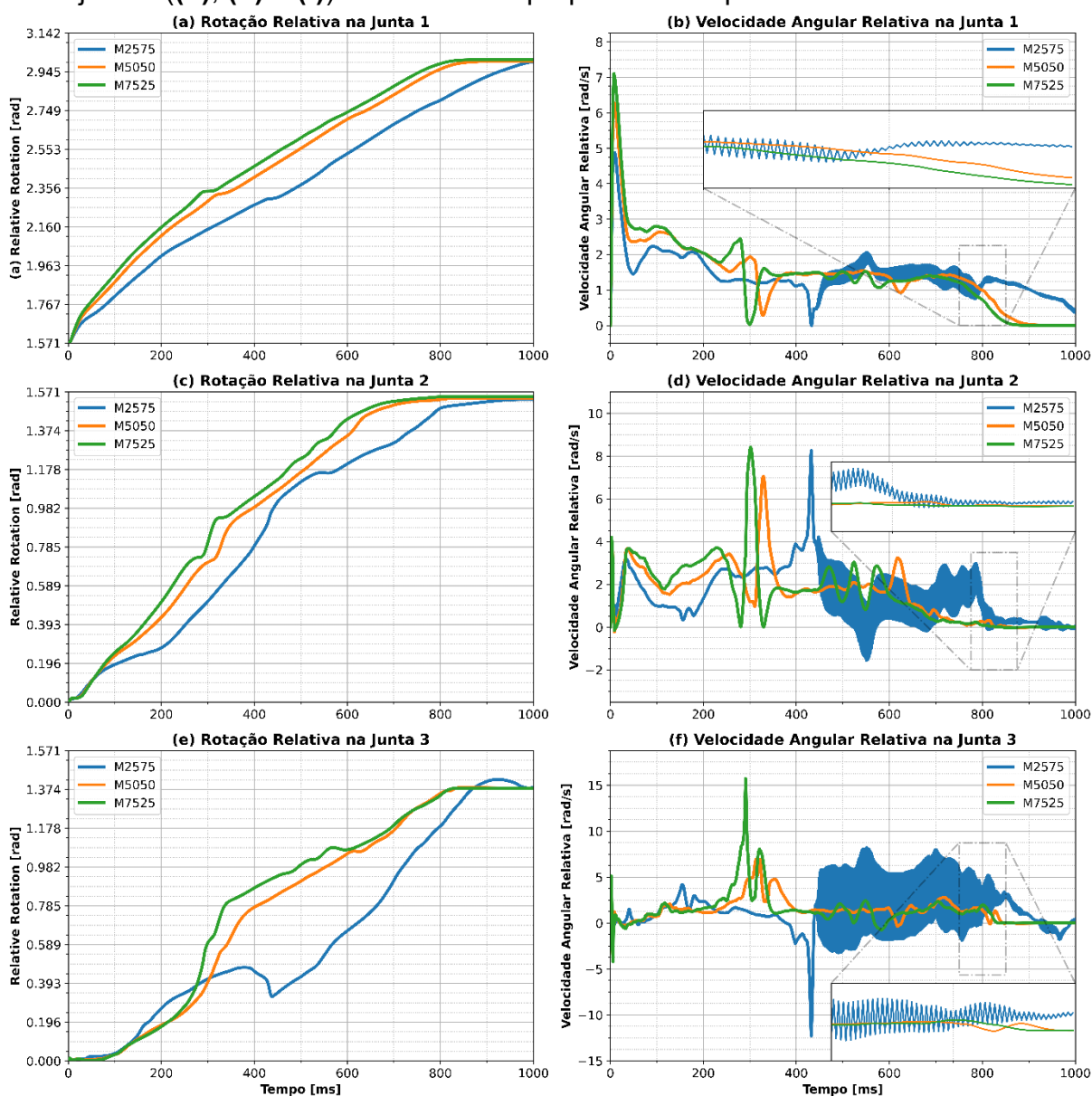


Fonte: Elaborado pelo autor.

Avaliando a evolução das variáveis de estado durante a execução da trajetória (Figura 28), também foram observadas similaridades no comportamento das variáveis de estado das entradas linear e senoidal binárias (Figura 24 e Figura 25). Para a entrada M2575 foi notado o padrão oscilatório na velocidade angular relativa (Figura 28b, Figura 28d e Figura 28f) a partir de 400 ms, como esperado devido à proximidade com

a entrada linear, entretanto, ao invés de essa oscilação ocorrer em torno de uma velocidade constante, ela apresenta um decaimento até atingir um valor próximo a zero, isso deve-se ao pouco estímulo dado ao neurônio da entrada senoidal. As entradas M5050 e M7525 apresentaram comportamento de velocidade angular relativa semelhante ao da trajetória senoidal binária. Também observa-se pela rotação angular relativa (Figura 28a, Figura 28c e Figura 28e) que as curvas apresentaram um aspecto próximo ao esperado, e que todas elas finalizaram o movimento no mesmo ângulo de 3,00 rad para a junta 1, 1,54 rad para a junta 2 e 1,38 rad para a junta 3.

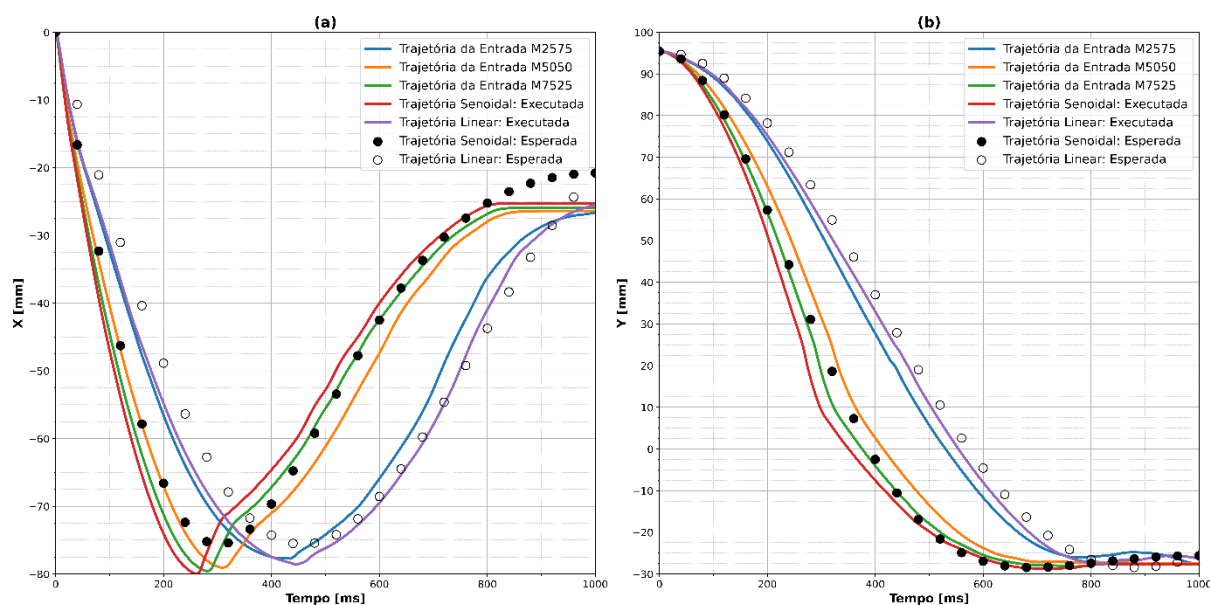
Figura 28 – Rotação relativa das juntas ((a), (c) e (e)) e velocidade angular relativa das juntas ((b), (d) e (f)) com entradas proporcionais apresentadas ao controlador



Fonte: Elaborado pelo autor.

Por fim foi avaliada a variação das coordenadas x e y no tempo (Figura 29) para todas as entradas em comparação com as coordenadas esperadas das trajetórias linear e senoidal. Com a avaliação das posições, é notável a capacidade da RNA em diferenciar as trajetórias aproximando a entrada puramente linear e M2575 das posições esperadas da trajetória linear, enquanto apresentou uma proximidade das entradas puramente senoidal, M5050 e M7525 com a trajetória senoidal esperada. Analisando a evolução das coordenadas na componente x (Figura 29a), observou-se com maior clareza a coincidência da correção aplicada nos pontos onde a coordenada absoluta x atingiu os valores máximos (aproximadamente 250 ms para a senoidal e 425 ms para a linear). Para todas as entradas o modelo finalizou com uma posição em x entre 32,5 mm e 35,0 mm. Avaliando a componente y (Figura 29b) observa-se que a proximidade da trajetória foi maior, conseguindo atingir o valor final próximo ao esperado, e não apresentando uma correção abrupta, evidenciando a influência do aumento do valor da coordenada x na correção efetuada.

Figura 29 – Variação das posições x (a) e y (b) o longo do tempo para todas as entradas apresentadas ao controlador, em comparação com as posições esperadas utilizadas durante o treinamento



Fonte: Elaborado pelo autor.

Com os resultados obtidos das entradas proporcionais, alguns pontos observados nos testes com as entradas binárias foram comprovados. O primeiro ponto a ser confirmado foi que de fato a mudança nas estratégias de controle ocorrem quando a coordenada x da ponta do dedo atinge o maior valor, posição coincidente com o início

do comportamento oscilatório das curvas de torque da entrada puramente linear (Figura 23) e a combinação M2575 (Figura 27), reforçando que o padrão oscilatório é um padrão de atuação adotado pelo controlador. Comparando as curvas de torque com a curva de posição no eixo x (Figura 29a), verificou-se que o ponto de correção também era coincidente com a mudança abrupta na tendência da posição esperada para a trajetória (componente x da velocidade). Essa observação pode indicar uma sensibilidade do modelo há mudanças abruptas nas velocidades do movimento do dedo. Reforçando essa hipótese, avaliando a posição da ponta do dedo no tempo (Figura 29), observou-se o aumento do erro com o aumento da coordenada em x . Essa influência ocorreu devido ao aumento do torque na ponta do dedo nessa posição, levando assim a uma maior instabilidade do controlador e conseqüentemente a uma maior dificuldade em se manter a posição. Outra observação foi que a rede aplicou correções de trajetória efetuada com a mudança na inclinação da coordenada x esperada, de negativa para positiva (Figura 29a), posição essa onde também foi observada a mudança das estratégias de controle. Essa correção pode estar relacionada à incapacidade do controlador em atingir a componente x da posição final desejada, uma vez que a redução na inclinação da curva pode representar dificuldade em determinar o torque esperado, fazendo que ele seja zero.

Quanto ao comportamento do controlador, verificou-se que nos primeiros passos tem-se um pico de torque e conseqüentemente de velocidade (Figura 23, Figura 24, Figura 25, Figura 27 e Figura 28), responsável por tirar o dedo do repouso. Após o pico inicial, o torque é invertido, porém com um valor inferior, uma vez que o dedo já se encontra em movimento. Analisando a Figura 27 e a Figura 30, constatou-se que o torque foi praticamente zero em quase todo tempo até atingir o valor máximo da coordenada x , onde ocorram as mudanças de estratégia de controle. Isso indica que provavelmente a estratégia de controle aprendida foi de aproveitar a força inercial dos corpos para fazer o controle de movimento para a condição em que a coordenada x aumentaria. Quando o ponto absoluto máximo de x foi atingido (Figura 29a), verificou-se que a mudança de estratégia foi alterada novamente, pois a combinação de baixo momento de inércia do dedo com a necessidade de velocidade constante da junta fez com que controlador atuasse de forma oscilatória para manter a velocidade constante quando o neurônio da trajetória linear era estimulado, e freando o final do movimento quando o neurônio da trajetória senoidal era estimulado.

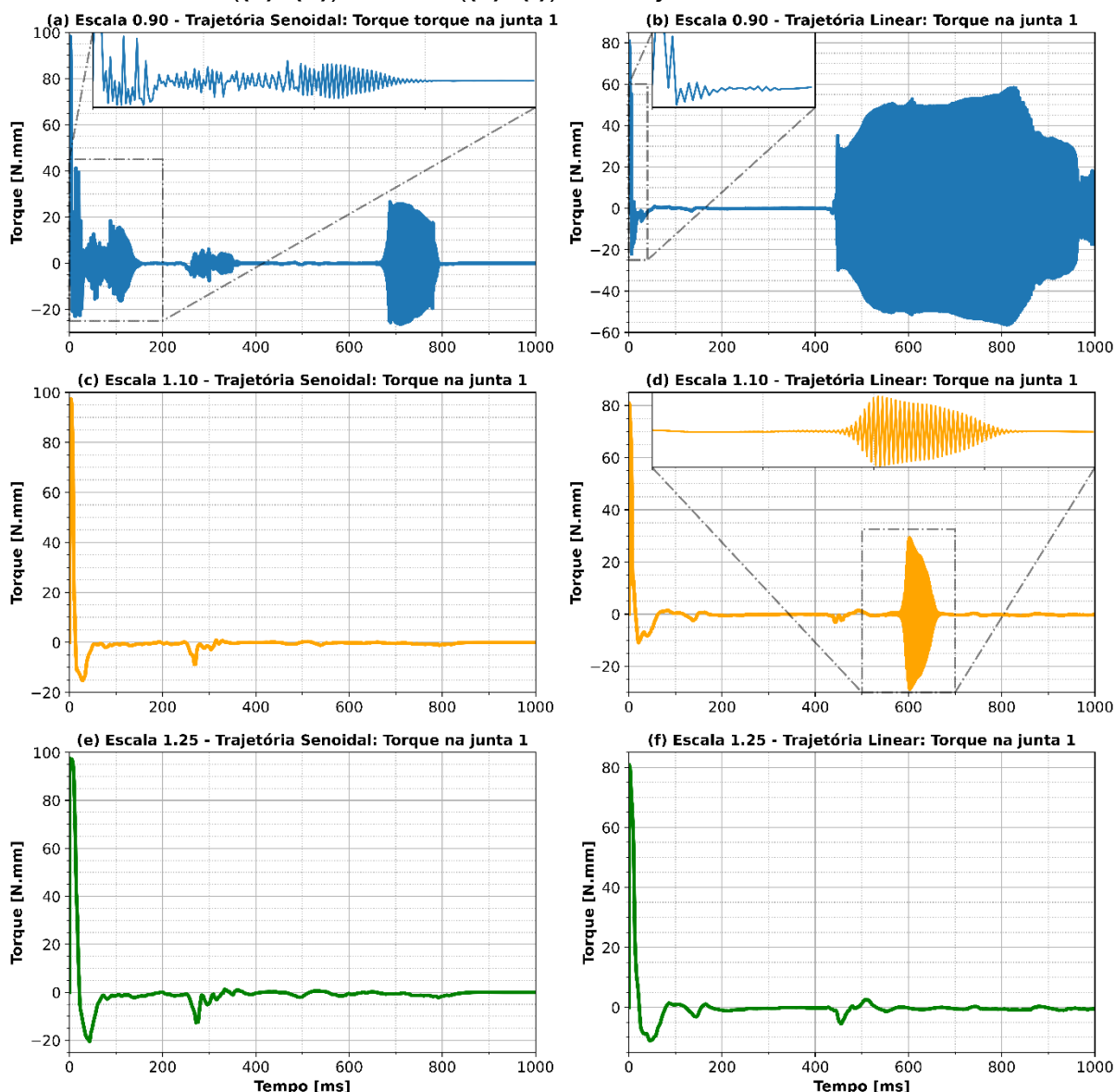
Por fim, avaliando as combinações proporcionais das trajetórias (Figura 29), observa-se que a RNA estabeleceu uma relação não proporcional entre as entradas de trajetória. Como esperado com a entrada M2575, a estratégia de controle e a trajetória executada foram próximas à trajetória linear esperada, enquanto para a entrada M7525, a trajetória executada foi mais próxima da senoidal. Entretanto, para a entrada da M5050, foi adotada uma estratégia mais aproximada da utilizada com a entrada senoidal binária e não uma estratégia intermediária entre as duas trajetórias, indicando assim a não proporcionalidade da combinação, com uma tendência para a trajetória senoidal com valores intermediários de entrada no neurônio linear.

3.2.3 Avaliação do Controlador de Trajetórias com o Modelo em Escala

Avaliando a capacidade do controlador em generalizar as respostas observadas para situações fora do ambiente de treinamento, foram refeitas as avaliações com as entradas binárias e proporcionais, porém com o modelo em escalas volumétricas de 90 %, de 110 % e de 125 %. Os erros médios observados para as entradas linear e senoidal binárias foram respectivamente $2,76 \pm 1,21$ mm e $3,50 \pm 2,61$ mm para a escala de 90 %; $3,42 \pm 1,75$ mm e $4,07 \pm 2,96$ mm para a escala de 110 %; e $3,92 \pm 2,03$ mm e $4,64 \pm 3,40$ mm para escala de 125%.

Na Figura 30 são apresentados os torques aplicados na junta 1 durante a execução das entradas linear e senoidal binárias com as três escalas. Observa-se que o comportamento das respostas foi diferente do apresentado no modelo original, o que indica que o sistema se adaptou às mudanças ambientais. Para o dedo mais leve, com 90 % do volume original (Figura 30a e Figura 30b), foi obtida uma resposta oscilatória para ambas trajetórias, diferentemente do observado originalmente (Figura 23 e Figura 27), devido à menor inércia do modelo, que o deixava mais sensível às manipulações do controlador. Para o modelo mais pesado, conseqüentemente com maior inércia (Figura 30e e Figura 30f), a resposta do controlador se modificou, e nenhum padrão oscilatório foi observado para nenhuma das entradas. Essas observações indicam que o controlador treinado utilizando o DDPG foi capaz de adaptar a sua estratégia dada as mudanças de entrada e de ambiente, uma vez que a tarefa foi cumprida com um erro médio absoluto próximo do observado nas condições originais do ambiente. As demais avaliações com as entradas binárias e proporcionais para os modelos em escala são apresentadas no Apêndice A.

Figura 30 – Torque aplicado na junta 1 com o modelo em escala de 90% ((a), (b)), 110% ((c), (d)) e 125% ((e), (f)) nas trajetórias senoidal e linear



Os resultados obtidos com o modelo original e em escala, indicam que o controlador apresentou capacidade de capturar corretamente o comportamento dinâmico e elaborar melhores estratégias para cumprir as trajetórias-alvo. Como indicado com as respostas do modelo original (Figura 23 e Figura 27), a hipótese de que o sistema aprendeu a utilizar a inércia como estratégia de controle é corroborada pelos resultados apresentados nos testes de generalidade (Figura 30). Como observado para o dedo com menor volume (90% do volume original) a estratégia de controle foi oscilatória para ambas as trajetórias, indicando que devido à massa reduzida e consequentemente inércia, o modelo se comportou de forma mais instável, o que necessitava de

intervenções mais constantes do controlador. Já para o dedo de maior volume (125% do volume original), para cumprir a trajetória esperada a estratégia oscilatória não foi utilizada em nenhum momento, devido à maior dificuldade em se movimentar o dedo, uma vez que ele apresentava maior volume, e conseqüentemente maior inercia. Essas avaliações nos permitiram concluir que a rede treinada se adequou às mudanças inerciais do ambiente e elaborou novas estratégias para cumprir as trajetórias esperadas.

Avaliando os resultados obtidos em um contexto de aplicação da metodologia do teste piloto para o desenvolvimento de controladores de prótese robóticas de membro superior, foi observado que o controlador desenvolvido apresentou as duas características principais desejadas sendo elas a capacidade de diferenciar as entradas de trajetória e a adaptação às mudanças do ambiente. Observa-se, inicialmente, que o sistema foi capaz de, dadas as entradas proporcionais, como indicadas nos resultados da Sessão 3.2.2, adaptar as trajetórias executadas, o que indicava que era o possível utilizar essa técnica no desenvolvimento de controladores que não executassem apenas movimentos padronizados, como os observados na literatura (SADIKOGLU; KAVALCIOGLU; DAGMAN, 2017; VÁSCONEZ *et al.*, 2023; YING *et al.*, 2024). Os resultados presentes demonstram que é possível utilizar com entrada do controlador as respostas de um classificador de movimentos de sEMG treinado por supervisão, como apresentado no trabalho de VÁSCONEZ *et al.*, (2023) e muitos outros da literatura (Seção 2.2.3.4) para determinar as entradas de seleção de trajetória, combinado com os estados atuais da prótese, relacionando de forma indireta as características extraídas do sEMG em comandos de atuação que devem ser enviados ao dispositivo, permitindo ao usuário um controle individualizado da execução dos movimentos.

Avaliando os sinais de controle obtidos na avaliação da RNA (Figura 23, Figura 27 e Figura 30), ela apresentou a capacidade de diferenciar atuações de controle dada a entrada de trajetória e as mudanças no ambiente, o que favoreceu o uso da rede como um controlador adaptativo para próteses para múltiplas trajetórias, dando um enfoque maior em como o dispositivo executa o movimento e não apenas na força de contato ou a velocidade de execução da tarefa, como são feitos nos controladores autônomos observados na literatura (DUTRA; SILVEIRA, 2023; GHAZALI *et al.*, 2017), podendo proporcionar ao usuário mais autonomia e previsibilidade ao controlar o dispositivo. Assim, os resultados do teste piloto indicaram que a proposta de controlador

mioelétrico de trajetórias avaliado pode ser utilizado na criação de controladores mio-elétricos não padronizados, com potencial de gerar um dispositivo mais adaptado às ATVDs e as necessidades do usuário.

3.2.4 Limitações do Teste Piloto

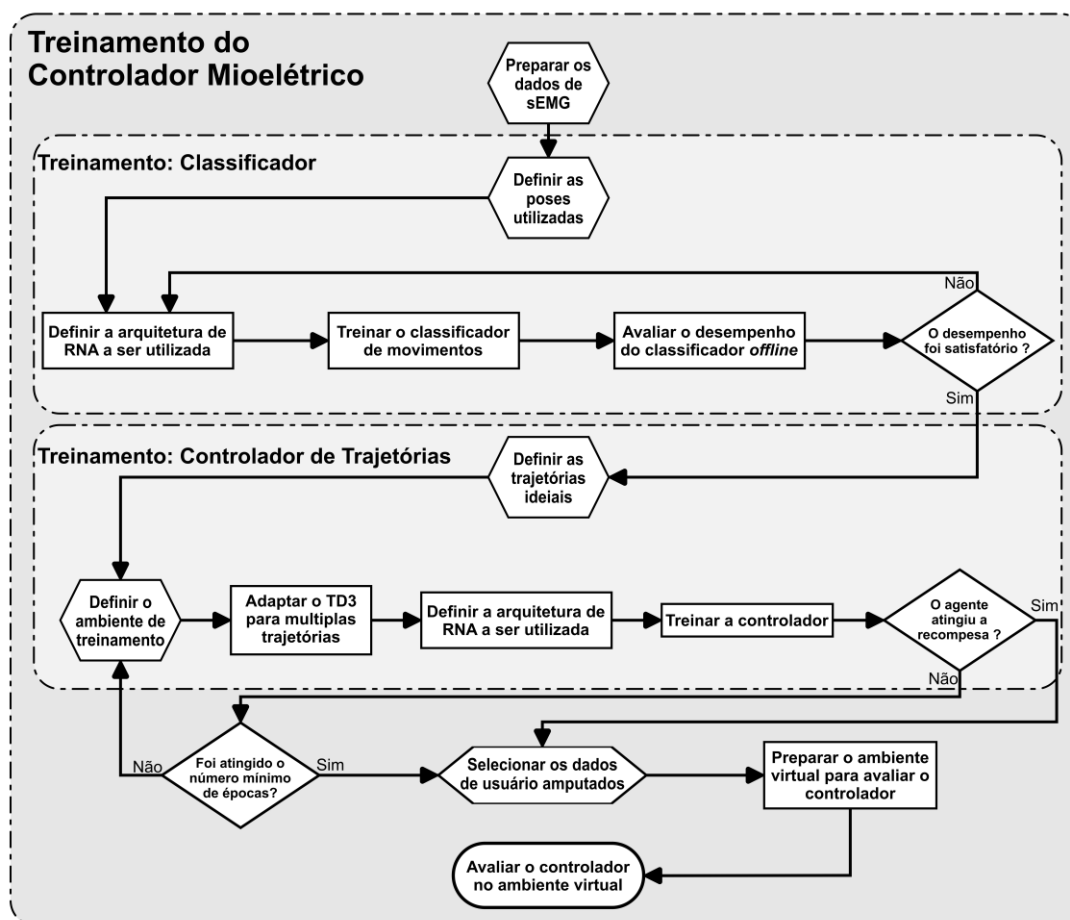
Para execução do teste piloto, algumas simplificações no modelo dinâmico do ambiente de treinamento precisaram ser feitas limitando as respostas observadas. A primeira delas foi considerar as falanges do dedo como corpos rígidos, ou seja, qualquer deformação foi desconsiderada. Apesar dessa consideração, ela apresenta-se razoável, e não tem grandes impactos na dinâmica do dedo, uma vez que apenas pequenas deformações podem ocorrer no movimento. Outra limitação do modelo foi o uso da geometria de cada falange como um tronco de cone elíptico, com distribuição de massa uniforme. Essa consideração pode ter impacto nos resultados uma vez que elas podem alterar a posição real do centro de massa do dedo. Além disso, foi optado por desconsiderar os efeitos de amortecimento e de rigidez das articulações. Essa simplificação foi adotada dada a dificuldade em se obter esses dados devido à grande variabilidade desses parâmetros para cada indivíduo. Como a determinação desses parâmetros estava fora do escopo do teste piloto, focado na aplicação do algoritmo DDPG para o desenvolvimento de um controlador inteligente de múltiplas trajetórias, elas não foram avaliadas. A consideração desses elementos pode afetar a estabilidade do modelo dinâmico e consequentemente a resposta da RNA, uma vez que a consideração desses termos insere torques resistivos e componentes de dissipação de energia para cada junta. Apesar dessas limitações, como foi observado no teste de adaptabilidade do controlador (Figura 30), o sistema demonstrou capacidade de perceber as mudanças ambientais o que é um bom indicativo de que o modelo poderia se adaptar a condições reais onde os efeitos de rigidez e amortecimento das juntas estão presentes. Outra limitação existente em relação ao controlador está no fato de não terem sido realizadas avaliações experimentais. Assim, como pode ser observado, refinamentos podem ser feitos para melhor avaliar a capacidade do treinamento por reforço em desenvolver modelos de controle adaptativo de múltiplas trajetórias em dispositivos reais. Entretanto, as avaliações apresentadas, apesar das limitações, dão o aporte teórico sobre a capacidade da metodologia, não impactando negativamente nas análises realizadas.

Outra limitação importante foi o uso de apenas o DDPG na avaliação da capacidade do aprendizado por reforço no treinamento de controladores adaptativos. Apesar de muito utilizado no desenvolvimento de controladores robóticos, é conhecido pela literatura que esse algoritmo é sensível aos hiperparâmetros de tamanho de memória de armazenamento, à taxa de aprendizado e aos parâmetros iniciais (KUMAR *et al.*, 2023). Assim, outros algoritmos como o TD3, PPO e SAC também poderiam ser utilizados, uma vez que eles podem apresentar uma convergência mais rápida do modelo ou conseguem lidar melhor com as sensibilidades do que o DDPG para o treinamento aqui proposto (HAARNOJA *et al.*, 2018; KUMAR *et al.*, 2023; WANG; HONG; ZHONG, 2022). Apesar dessa limitação, o uso dos outros algoritmos teria um maior impacto em levar o Ator a atingir a recompensa mínima desejada mais rapidamente. Assim, se faz necessário futuramente, avaliar os demais algoritmos nesse mesmo ambiente de treinamento criado para entender a real aplicabilidade de algoritmos de aprendizado por reforço no desenvolvimento de controladores de trajetória adaptativo para próteses de membro superior. Apesar dessa limitação, o controlador desenvolvido no teste piloto teve a capacidade de definir diferentes estratégias de controle, adaptando suas respostas a seleção de trajetória e as variações ambientais.

CAPÍTULO 4: DESENVOLVIMENTO DO CONTROLADOR MIOELÉTRICO

Ao finalizar o teste piloto, a segunda etapa do trabalho foi avaliar a metodologia empregada no desenvolvimento de um controlador mioelétrico para uma prótese robótica real. A Figura 31 apresenta o fluxograma apenas das etapas de desenvolvimento do controlador mioelétrico, que serão apresentadas nesse capítulo.

Figura 31 – Fluxograma do processo de treinamento do controlador mioelétrico



Fonte: Elaborado pelo autor.

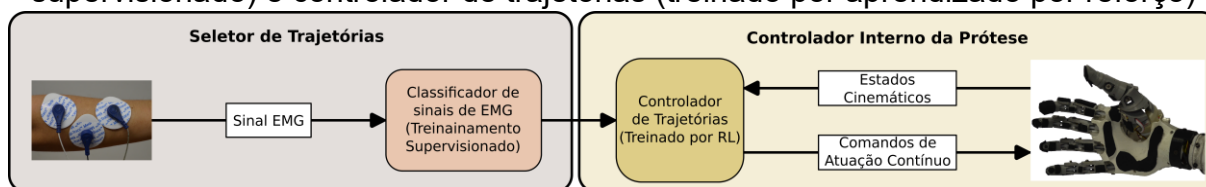
No desenvolvimento do controlador mioelétrico, pretendeu-se avaliar as capacidades e as limitações da técnica de RL no desenvolvimento de um controlador mioelétrico adaptativo, utilizando um modelo de prótese real e dados de sEMG de indivíduos hígidos e amputados. Na Figura 31, pode-se observar que para o treinamento do controlador da prótese, foi necessário desenvolver o seu modelo dinâmico e definir trajetórias baseadas nas trajetórias fisiológicas de indivíduos hígidos, extraídas de um banco de dados público. Para acelerar o treinamento, foi utilizado o algoritmo TD3, uma evolução do DDPG, e os parâmetros foram ajustados para atingir o melhor

desempenho possível no treinamento. Para a avaliação das respostas executadas pelo controlador, utilizando dados de sEMG, foi necessário o desenvolvimento de um classificador de movimentos. Utilizando um banco de dados público, os sinais de sEMG foram tratados e tiveram suas características extraídas. Com o classificador e o controlador treinados, eles foram avaliados utilizando os dados de indivíduos amputados em um ambiente virtual. Novamente, todas as etapas de tratamento de dados, de definição de modelo, de treinamento e de validação virtuais foram feitas utilizando-se um Desktop com um processador AMD Ryzen 9 7950X, uma placa de vídeo NVIDIA GeForce RTX 4090 e 128 GB de memória RAM.

4.1 Metodologia

A fim de reproduzir o controlador treinado por RL no teste piloto no dispositivo real, fez-se necessário o desenvolvimento de duas RNAs. Como apresentado na Figura 32, o controlador da prótese se divide em dois estágios, um seletor de trajetórias e um controlador de trajetórias. Para o desenvolvimento do seletor de trajetórias foi utilizada a técnica já consagrada para treinamento de classificadores de movimentos de sEMG, o aprendizado supervisionado. O seletor era responsável por traduzir as características extraídas do sinal em uma função de probabilidade de movimento utilizando neurônios do tipo *softmax* na camada de saída. A saída do seletor alimentou o controlador de trajetórias que combinou a saída do seletor com os estados atuais da prótese para determinar os sinais de atuação necessários para cumprir a trajetória desejada pelo usuário.

Figura 32 – Diagrama esquemático do controlador mioelétrico desenvolvido, composto por dois estágios: seletor de trajetórias (treinado por aprendizado supervisionado) e controlador de trajetórias (treinado por aprendizado por reforço)

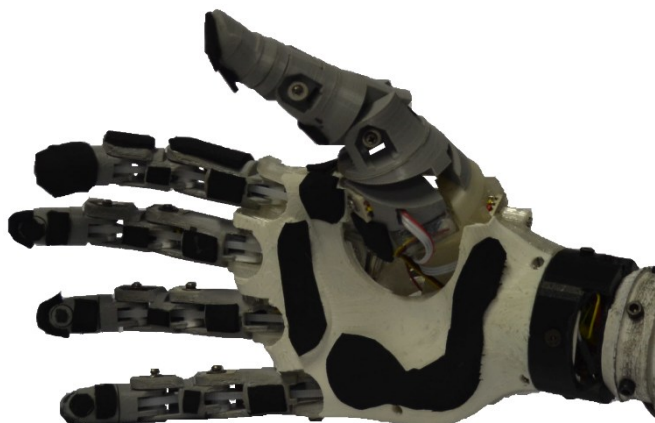


Fonte: Elaborado pelo autor.

A prótese escolhida para a implementação do controlador foi a HANDi Hand (Figura 33), pois se tratava de uma prótese de plataforma aberta, o que facilitaria o processo de avaliação das características funcionais do dispositivo, e conseqüentemente o desenvolvimento do modelo do dispositivo. A HANDi Hand foi desenvolvida

pelo laboratório BLINC do *Alberta Machine Intelligence Institute (Amii)* da Universidade de Alberta (Canadá) por BRENNIS; DAWSON; PILARSKI, (2017). Como sistema de atuação ela utiliza de tendões artificiais, o que garante uma adaptabilidade dos dedos aos objetos quando segurados. Possui 6 motores, o que permite que ela tenha 6 graus de liberdade (GDL) (flexão e extensão de cada dedo, e adução e abdução do dedo um). Além disso, pode ser fabricada utilizando técnicas de impressão 3D. A HANDi Hand possui potenciômetros entre cada falange, permitindo a medição de posição angular, e sensores de força do tipo FSR na ponta de cada um dos dedos. Segundo o autor, as características da prótese permitem que a força máxima de prensão seja de 4,2 N. Dadas as características e de sua instrumentação, apresenta-se como uma solução ideal para avaliar o controlador proposto nesse trabalho, e foi construída no Laboratório de Bioengenharia da UFMG (Labbio).

Figura 33 – Prótese HANDi Hand construída no Labbio, utilizada para a implementação e avaliação do controlador mioelétrico proposto



Fonte: Elaborado pelo autor.

Com o modelo de prótese definido, nessa seção serão apresentadas todas as etapas de desenvolvimento das RNAs para desenvolver o controlador para HANDi. Inicialmente serão demonstrados quais dados de sEMG e cinemáticos foram selecionados e os tratamentos feitos. Posteriormente serão descritos os parâmetros de treinamento do classificador. Por fim, será apresentado o desenvolvimento do controlador de trajetórias, apresentando o ambiente de treinamento e suas definições, as trajetórias-alvo escolhidas e o algoritmo de treinamento escolhido.

4.1.1 Aquisição e Condicionamento de Dados de sEMG e Cinemáticos

Para o treinamento e validação do controlador mioelétrico, foi necessário o uso de dados de eletromiografia superficial. Devido à necessidade de um grande volume de dados, para que os classificadores de sEMG apresentem uma boa acurácia, optou-se por utilizar um banco de dados público para as etapas de treinamento e validação, o NINAPRO.

O NINAPRO, desenvolvido inicialmente por ATZORI et al., (2012) foi idealizado para servir de suporte para pesquisadores de controladores mioelétricos para próteses de membro superior, visando impulsionar o desenvolvimento desses sistemas aumentando a quantidade de dados que possam ser utilizados com diversos tipos de usuário. Hoje o NINAPRO conta com 10 bancos de dados que possuem dados de sEMG combinado com sensores inerciais, de força, de rastreamento ocular, entre outros. A Tabela 4 apresenta um resumo dos bancos de dados disponibilizados.

Dentre os bancos de dados disponíveis o DB2, o DB3, o DB7 e o DB9 foram escolhidos, sendo os dois primeiros utilizados o treinamento do classificador, e o terceiro para a validação do treinamento com o voluntário 21, o único que possuía amputação adquirida do DB7, e o último para extração da variação angular de cada articulação da mão durante a execução dos movimentos. Os bancos de dados DB2, DB3 e DB7 foram desenvolvidos nos trabalhos de ATZORI et al., (2014) e KRASOULIS et al., (2017) utilizando 12 eletrodos de sEMG Delsys® Trigno™ IM Wireless System, um eletrodo seco de sEMG de canal único com três polos acoplado com sensores inerciais de 9 GDL (acelerômetro, giroscópio e magnetômetro), apresentando uma frequência amostral de 2 kHz para os dados sEMG e 128 Hz para os dados inerciais. Dos 12 eletrodos, oito deles foram igualmente espaçados em volta do antebraço à 3 cm do cotovelo enquanto os quatro restantes foram posicionados nos músculos do extensor dos dedos, do flexor dos dedos, do bíceps branquial e do tríceps branquial. A superfície onde foram colocados os eletrodos foi limpa com álcool isopropílico 70% e uma bandagem elástica sem látex foi utilizada envolvendo-os para evitar o escorregamento. A Figura 34 apresenta a disposição dos eletrodos.

Tabela 4 – Principais características dos bancos de dados do NINAPRO, com resumo das informações disponibilizadas

Nome	Objetivo	Nº de Voluntários	Nº de Poses	Configuração sEMG	Modalidade dos Dado
DB1	Classificação ou Regressão	27 típicos	52	10 Otto Bock	sEMG, cinemáticos
DB2	Classificação ou Regressão	40 típicos	49	12 Delsys Trigno	sEMG, cinemáticos, inercial, força
DB3	Classificação ou Regressão	11 amputados	49	12 Delsys Trigno	sEMG, inercial
DB4	Classificação ou Regressão	10 típicos	52	12 Cometa	sEMG, cinemático
DB5	Classificação ou Regressão	10 típicos	52	2 Thalmic Armbands	sEMG, inercial
DB6	Repetibilidade	10 típicos	7	14 Delsys Trigno	sEMG, rastreamento ocular com câmeras de cena
DB7	Classificação ou Regressão	20 típicos, 2 amputados	40	12 Delsys Trigno	sEMG, inercial, cinemático
DB8	Regressão de Movimentos de dedos	10 típicos, 2 amputados	9	1 Delsys Trigno	sEMG, cinemático
DB9	Análise Cinemática de Grande Escala	77 típicos	40	10 Otto Bock 12 Delsys Trigno 2 Thalmic Armbands	sEMG, cinemático
DB10	Coordenação Visão e Mão, Classificação	30 típicos, 15 amputados	10	12 Delsys Trigno	sEMG, inercial, rastreamento ocular com câmera, comportamentais e clínicos

Fonte: Elaborado pelo autor.

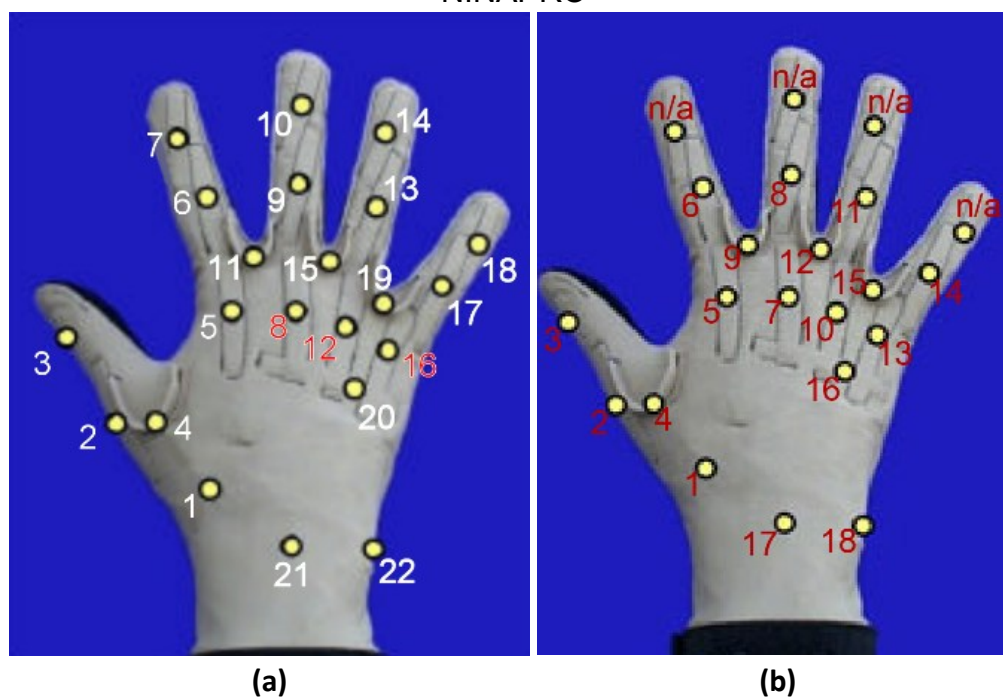
Figura 34 – Disposição dos eletrodos utilizados nos bancos de dados DB2, DB3 e DB7 do NINAPRO durante a coleta de sinais de sEMG



Fonte: KRASOULIS et al., (2017)

ATZORI et al., (2014) utilizou a CyberGlove II com 22 DOF, conforme apresentado na Figura 35a, enquanto KRASOULIS et al., (2017) utilizou a CyberGlove II com 18 DOF, conforme apresentado na Figura 35b, medindo o ângulo de cada articulação com uma frequência amostral de 90 Hz, utilizando sensores flexíveis. A calibração dos dados foi desenvolvida por JARQUE-BOU; ATZORI; MÜLLER, (2020), que tratou os dados das posições coletadas nos DB2 e DB7, e disponibilizou no DB9.

























Figura 35 – Disposição dos sensores flexíveis da CyberGlove II utilizados na aquisição de dados cinemáticos nos bancos de dados DB2 e DB3 (a) e DB7 (b) do NINAPRO



Fonte: NINAPRO, (2017)

Para aquisição dos dados, os participantes executaram 49 movimentos no DB2 e no DB3 e 40 movimentos no DB7 (Figura 36 e Figura 37) selecionados de acordo com os movimentos mais utilizados durante as ATVDs. Os movimentos foram divididos em três classes de exercícios: os movimentos básicos de dedos e punho (Exercício B), os movimentos funcionais de preensão (Exercício C) e padrões de força (Exercício D), sendo esse último utilizado apenas no DB2. Cada voluntário repetiu seis vezes cada movimento durante 5 segundos, com intervalos de 3 segundos de descanso entre cada movimento ou repetição, sendo instruídos através de uma tela de computador indicando o movimento e tempo de descanso. Para os voluntários amputados do DB3 e do DB7 solicitou-se que executassem movimentos bilaterais imaginários espelhados, ou seja, que fosse movimentado o membro não amputado juntamente com o usuário imaginando movimentar o membro amputado, onde estavam posicionados os eletrodos, para auxiliar na execução da contração muscular no coto (ATZORI *et al.*, 2014; KRASOULIS *et al.*, 2017).

Figura 36 – Movimentos realizados pelos participantes durante a aquisição dos bancos de dados DB2, DB3 e DB7 do NINAPRO

Exercício B			Exercício C		
1	Polegar para Cima		18	Preensão de força	
2	Extensão dos dedos indicador e médio com flexão nos demais dedos		19	Preensão de força curta	
3	Flexão dos dedos anelar e mínimo com extensão nos demais dedos		20	Preensão palmar	
4	Oposição de polegar		21	Preensão centrada	
5	Abdução de todos os dedos		22	Preensão cilíndrica	
6	Todos os dedos flexionados juntos		23	Preensão de anel	
7	Apontar o indicador		24	Preensão prismática de quatro dedos	
8	Adução com dedos estendidos		25	Preensão de bastão	
9	Supinação de punho relativo ao dedo médio		26	Pinça trípode	
10	Pronação de punho relativo ao dedo médio		27	Preensão esférica	
11	Supinação de punho relativo ao dedo mínimo		28	Preensão esférica com três dedos	
12	Pronação de punho relativo ao dedo mínimo		29	Preensão esférica de precisão	
13	Flexão de punho		30	Preensão trípode	
14	Extensão de punho		31	Pinça prismática	
15	Desvio radial de punho		32	Pinça bidigital	
16	Desvio ulnar de punho		33	Pinça quadrupla	
17	Extensão de punho com a mão fechada		34	Pinça lateral	
			35	Preensão em extensão	
			36	Preensão em extensão lateral	
			37	Preensão em disco	
			38	Abrir uma garrafa com pinça trípode	
			39	Girar uma chave de fenda	
			40	Cortar alguma coisa	

Fonte: Adaptado de NINAPRO (2017).

Figura 37 – Padrões de força realizados pelos participantes durante a aquisição do banco de dados DB2 do NINAPRO

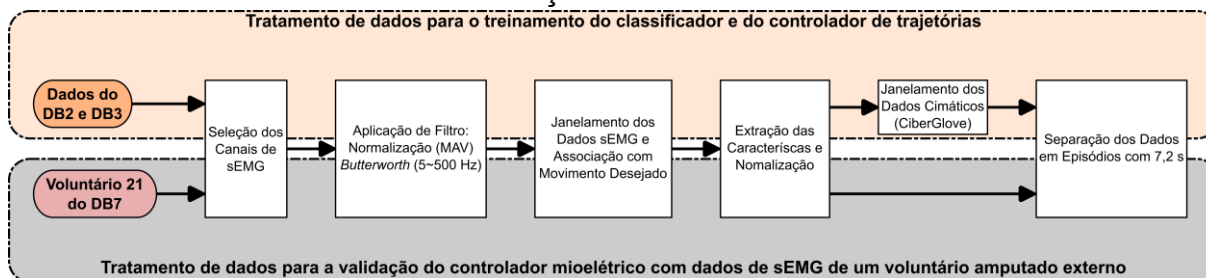
Exercício D		
41	Flexão dedo mínimo	
42	Flexão do dedo anelar	
43	Flexão dedo médio	
44	Flexão do dedo indicador	
45	Abdução de polegar	
46	Flexão de polegar	
47	Flexão do indicador e do dedo mínimo	
48	Flexão do dedo anelar e do dedo médio	
49	Flexão do dedo indicador e polegar	

Fonte: : Adaptado de NINAPRO (2017).

4.1.2 Seleção e Tratamento de Dados do NINAPRO

Para o treinamento e para os testes do classificador de movimentos, foram preparados e selecionados os dados do DB2, do DB3 e do voluntário 21 do DB7. As etapas apresentadas na Figura 38 foram utilizadas para o tratamento dos dados, em que todas as etapas foram realizadas por meio de um algoritmo em linguagem Python™, versão 3.10 (Python Software Foundation, Org, Wilmington, DE, EUA), utilizando as bibliotecas Numpy (HARRIS *et al.*, 2020), Pandas (MCKINNEY, 2010) e SciPy (VIRTANEN *et al.*, 2020).

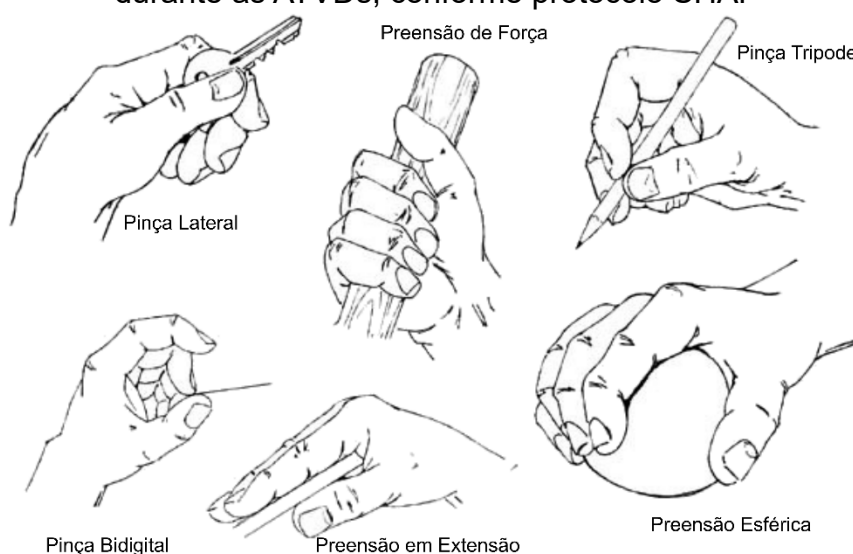
Figura 38 – Fluxo das etapas de tratamento dos dados de sEMG utilizados no treinamento e validação do controlador mioelétrico



Fonte: Elaborado pelo autor.

Inicialmente foi necessário selecionar os movimentos de pinça que serão utilizados, para então determinar as trajetórias que o controlador deverá executar. A seleção dos movimentos de pinça desejados foi baseada nas pinças necessárias para executar o *Southampton Hand Assessment Procedure* (SHAP), que é um protocolo padronizado de avaliação clínica, desenvolvido inicialmente por LIGHT; CHAPPELL; KYBERD, (2002), que busca mensurar de forma objetiva e contextualizada a mão humana, tanto em indivíduos hígidos quanto em pessoas com disfunções ou o desempenho de usuários de prótese. O teste é composto por 12 tarefas com objetos abstratos e 14 tarefas que envolvem ATVDs. Assim o teste consegue avaliar de forma confiável os seis tipos padrão de movimento de mão: a pinça de bidigital, a pinça trípode, a pinça lateral, a preensão esférica, a preensão de força e a preensão em extensão, como apresentadas na Figura 39.

Figura 39 – Tipos de movimentos de pinça e preensão comumente executados durante as ATVDs, conforme protocolo SHAP

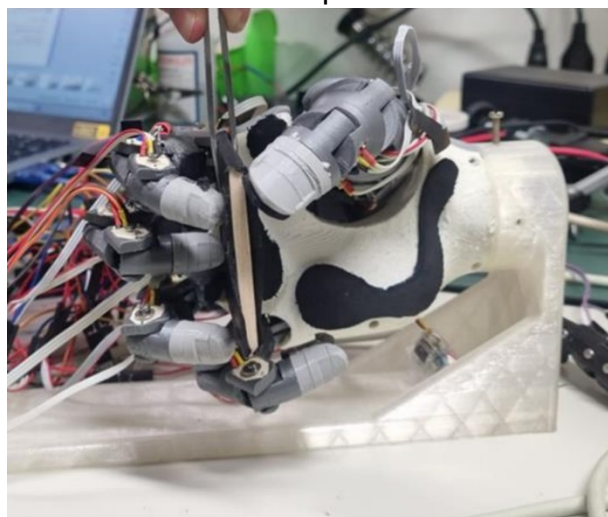


Fonte: Adaptado de LIGHT; CHAPPELL; KYBERD, (2002).

Das pinças definidas pelo SHAP, apenas três foram escolhidas para serem executadas pela prótese, devido às limitações que o dispositivo apresentava, a preensão de força, a pinça trípode e a pinça bidigital, além dos dados da condição de relaxamento. A exclusão da preensão esférica ocorreu porque a prótese não tinha a capacidade de executar a abdução e adução dos dedos 2 a 5, que é a principal diferença entre ela e a preensão de força. A pinça lateral foi excluída pois ela dependia de um movimento de abdução e adução do dedo um de forma independente, o que não era possível executar pela HANDi Hand, uma vez que esse grau de liberdade era

subatuado. Por fim, a preensão em extensão foi excluída, pois a prótese não conseguia executá-la. Como pode ser observado na Figura 40 quando foi reproduzido essa preensão com a prótese os dedos não se adaptavam como esperado no objeto, o que não permitia a execução da trajetória fisiológica esperada.

Figura 40 – Execução da preensão em extensão pela prótese HANDi Hand em bancada experimental



Fonte: Elaborada pelo autor.

Após a seleção dos movimentos, foi feito o condicionamento dos sinais de sEMG. Para isso foram separados os dados dos primeiros 8 canais de sEMG dos três bancos de dados (DB2, DB3 e voluntário 21 do DB7), sendo esses canais referentes aos sinais dos eletrodos posicionados igualmente espaçados em torno do antebraço. Após a seleção, os dados passaram por um filtro do tipo *butterworth* de quarta ordem com frequência de banda de 5 a 500 Hz utilizando a biblioteca SciPy (VIRTANEN *et al.*, 2020). Os filtros do tipo *notch* não foram empregados pois os sensores utilizados por ATZORI *et al.*, (2014) e KRASOULIS *et al.*, (2017) apresentavam isolamento da rede elétrica. Após filtrados, os dados de cada voluntário foram então escalonados para uma faixa de -1 a 1 utilizando o valor máximo absoluto do sinal.

Após o condicionamento, o sinal de sEMG escalonado foi dividido em janela. Para isso, foi estipulado que o tempo máximo de execução do movimento desejado fosse de 300 ms, como indicado no trabalho de MAYOR *et al.*, (2017). Levando esse tempo em consideração foi estipulado que a janela dos dados de sEMG fosse de 200 ms, com uma sobreposição de 160 ms, ou seja as janelas foram espaçadas em incrementos de 40 ms, possibilitando um bom tempo para execução da posição pelo

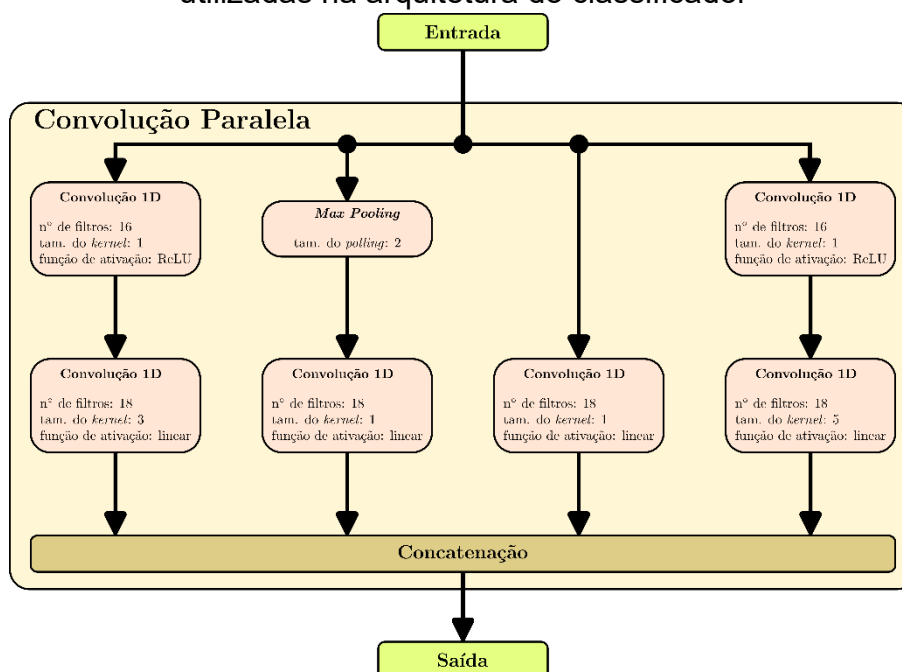
controlador. Para garantir que cada janela tivesse apenas um movimento associado, foi aplicado o valor da moda dos movimentos de cada janela.

Por fim, o sinal de cada janela teve suas características extraídas. Para a seleção das características, foram analisados os trabalhos de ATZORI et al., (2012, 2014) e KRASOULIS et al., (2017) utilizando o NINAPRO, e escolhidas as características comuns entre eles. Foram então utilizadas 6 características do domínio do tempo o RMS (Equação (1)), o desvio padrão (SD), o MAV (Equação (5)), o SSC (Equações (6) e (7)), o ZC (Equação (2)) e o WL (Equação (3)), totalizando 46 entradas para o classificador. Após extraídas, todas as características foram escalonadas para cada voluntário para uma escala de 0 a 1, onde 0 seria relativo ao menor valor da característica para o voluntário e 1 o maior valor.

4.1.3 Arquitetura e Treinamento do Classificador de Movimentos

A arquitetura do classificador foi baseada na arquitetura utilizada por VÁSCO-NEZ et al., (2023), utilizando camadas de convolução em paralelo como apresentado no diagrama da Figura 41. Foram utilizadas as camadas de convolução de uma dimensão ao invés de camadas de convolução de duas dimensões (Figura 14a), uma vez que foram utilizadas apenas características temporais de cada janela.

Figura 41 – Diagrama esquemático das camadas de convolução em paralelo utilizadas na arquitetura do classificador



Fonte: Elaborado pelo autor.

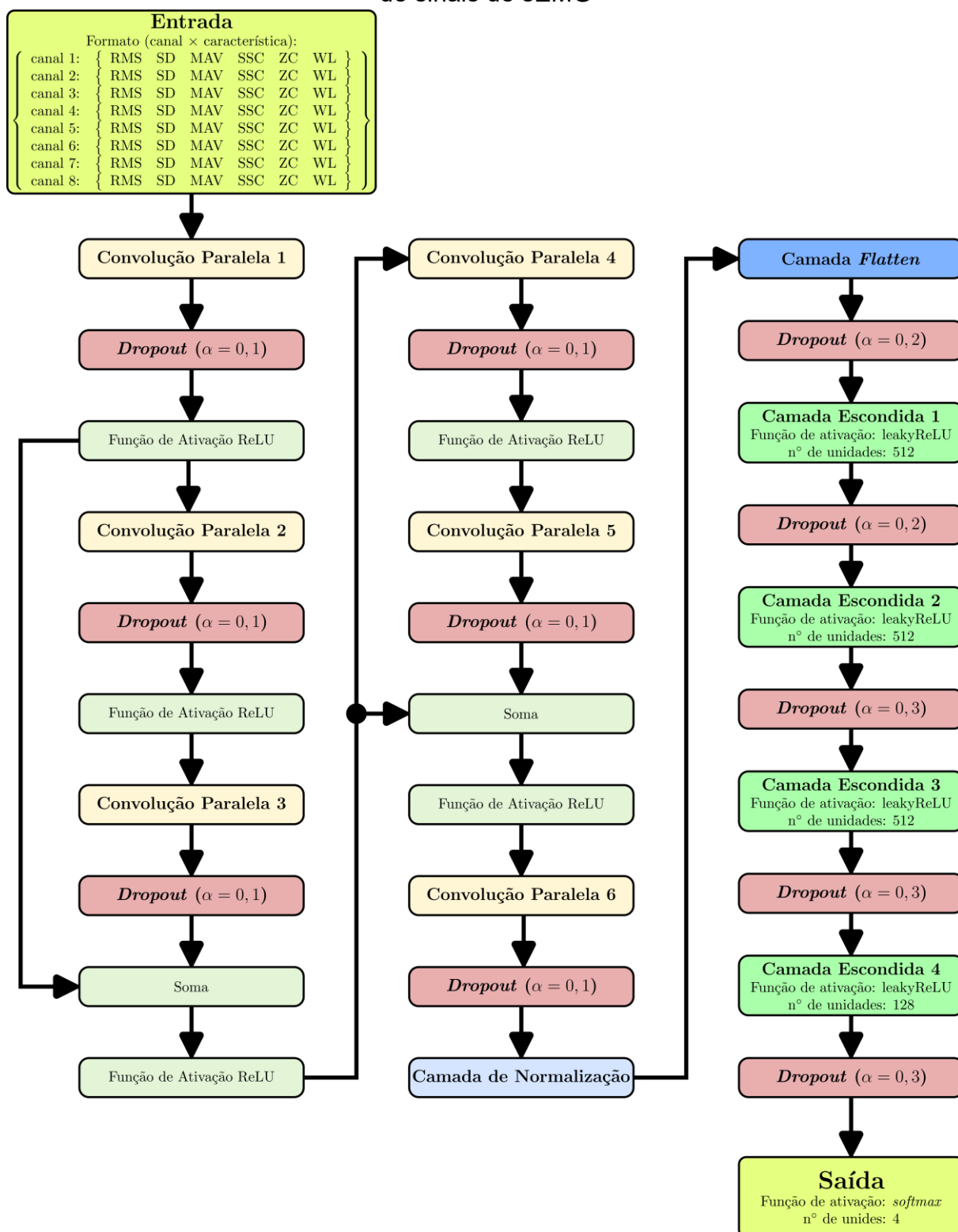
Para a entrada da rede, as características extraídas do sinal foram organizadas de forma que os seis primeiros neurônios de entrada fossem referentes às características extraídas do canal 1, os próximos seis neurônios referentes as características extraídas do canal 2 e assim sucessivamente até o oitavo canal. Várias camadas de convolução paralelas foram combinadas, além da utilização de residuais, para garantir a generalidade do modelo, como demonstrado na Figura 42. As propriedades que foram extraídas das camadas de convolução passaram por três camadas contendo 512 neurônios e uma camada com 128 neurônios, que utilizavam a função leakyReLU como função de ativação, com α de 0,3, valor padrão da biblioteca Keras do Python (CHOLLET, 2015). Para regularizar o treinamento entre os neurônios das camadas ocultas finais foram utilizadas camadas de *dropout*. A Figura 42 apresenta a arquitetura utilizada.

A quantidade de neurônios de saída foi determinada pelo número de movimentos classificados. Como discutido na Sessão 4.1.1, três movimentos serão classificados, assim serão necessários de quatro neurônios de saída, um para representar a posição de repouso e outros três para representar as classificações dos movimentos escolhidos. Os neurônios foram organizados como apresentado na Tabela 5. A função de ativação dos neurônios de saída foi a *softmax* (Equação (15)), ideal para solução de problemas de classificação.

Para o treinamento, 70 % dos dados dos DB2 e do DB3, selecionados aleatoriamente, foram utilizados e a técnica *k-fold* com cinco divisões foi utilizada. Para avaliar o desempenho da RNA foi determinada a acurácia de cada divisão, definida pela Equação (40), e o modelo que apresentou a maior acurácia foi escolhido como modelo final do classificador. Durante o treinamento era esperado que a acurácia obtida com os dados de validação, 30% dos dados do DB2 e DB3, fosse superior a 85 %, a acurácia mínima que o classificador deve ter para não frustrar o usuário (GUO, W. *et al.*, 2017b). Além da acurácia, a RNA escolhida entre as divisões teve sua matriz de confusão calculada para melhor avaliar a discretização do modelo.

$$AC = \frac{n^{\circ} \text{ de movimentos corretos preditos}}{n^{\circ} \text{ total de amostras}} \cdot 100 \% \quad (40)$$

Figura 42 – Arquitetura da RNA utilizada como classificador de movimentos a partir de sinais de sEMG



Fonte: Elaborado pelo autor.

Tabela 5 – Organização dos neurônios de saída do classificador mioelétrico, indicando a correspondência com os movimentos e a condição de repouso

Índice NINAPRO	Movimento	Neurônio na camada de saída
0	Relaxado	1
19	Preensão de Força	2
26	Pinça tripode	3
32	Pinça bidigital	4

Fonte: Elaborado pelo autor.

Como parâmetros de treinamento, foi utilizada a função de custo *cross-entropy* (Equação (19)) e taxa de aprendizado dinâmica, onde o treinamento iniciava-se com 1E-3 de taxa de aprendizado e ela era reduzida em caso de não redução do erro com os dados de validação interna após 50 épocas consecutivas, e tinha o valor mínimo de 1E-5. O número máximo de épocas de treinamento foi de 10.000, com os dados separados em *batches* de 1024 amostra, sendo que 30 % dos dados de cada época utilizados para validação interna do treinamento. O algoritmo de *early stopping* foi utilizado para parar o treinamento caso não ocorresse uma redução no valor da função de custo com os dados de validação após 100 épocas. A Tabela 6 apresenta um resumo dos parâmetros de treinamento utilizados.

Tabela 6 – Resumo dos parâmetros de treinamento do classificador mioelétrico

Parâmetros	Propriedades/Valores
Função de custo	Cross-entropy: $C_{X_k} = \sum_{i=1}^n [Y_i(X_k) \cdot \ln y_i + (1 - Y_i(X_k)) \cdot \ln(1 - y_i)]$
Taxa de aprendizado	Dinâmica: <ul style="list-style-type: none"> • Inicial: 1E-3 • N° máximo de épocas para melhora: 50 • Taxa de redução: 0,5 • Valor mínimo: 1E-5
Épocas	N° máximo de épocas: 10.000 Validação interna: 30 % dos dados Tamanho das <i>batches</i> : 1024
<i>Early stopping</i>	Monitoramento: função de custo da época com os dados de validação interna N° de épocas para melhorar a variável de monitoramento: 100

Fonte: Elaborado pelo autor.

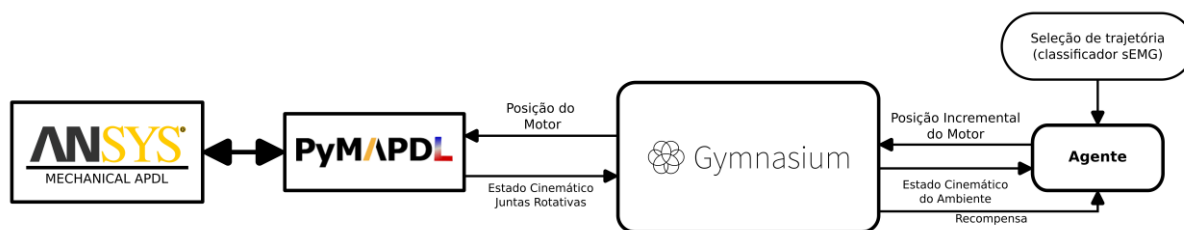
Para a aplicação da técnica do *k-fold*, foi utilizada a biblioteca scikit-learn (PE-DREGOSA *et al.*, 2011) e para elaboração do modelo e o treinamento empregou-se a biblioteca TensorFlow (MARTÍN ABADI *et al.*, 2015).

4.1.4 Desenvolvimento do Ambiente de Treinamento

Para a implementação do controlador de trajetórias, foi necessário construir um ambiente de treinamento possibilitando a implementação do algoritmo de RL que irá realizar o treinamento da RNA, que atuará como controlador de trajetórias. Dada a utilização de um sistema subatuado, utilizando tendões artificiais, não era possível implementar um modelo simples de corpos rígidos para modelar o comportamento dinâmico do dispositivo como realizado no teste piloto. Assim, para garantir que o modelo do dispositivo utilizado no ambiente de treinamento apresente a dinâmica correta de funcionamento do mecanismo de atuação foi optado por construí-lo em um programa de Análise de Elementos Finitos (Finite Element Analysis – FEA), o ANSYS Mechanical APDL® (ANSYS Inc., Lebanon, EUA).

A Figura 43 apresenta o diagrama de funcionamento do ambiente de treinamento. Como pode ser observado a biblioteca Gymnasium continua atuando como ambiente de iteração entre o Agente e o ambiente de treinamento. A biblioteca Gymnasium recebia as ações do Agente que então as convertia nas condições de contorno utilizadas no modelo FEA via a API PyMAPDL, desenvolvida para comunicação do software com o Python.

Figura 43 – Diagrama esquemático de interação com o Ansys Mechanical APDL® para construção do ambiente de treinamento do controlador de trajetórias



Fonte: Elaborado pelo autor.

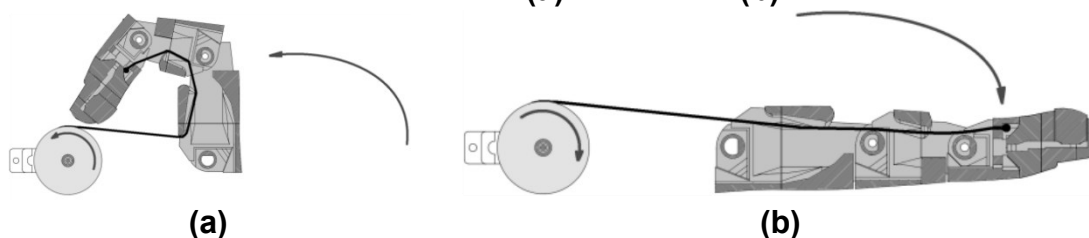
Dada a estrutura de funcionamento do ambiente de treinamento, será inicialmente detalhada a construção do modelo da prótese HANDi Hand no Ansys Mechanical APDL®, bem como os parâmetros de solução utilizados. Por fim, será apresentada a função de recompensa empregada no treinamento do controlador de trajetórias.

4.1.4.1 Criação do Modelo de Prótese de Membro Superior

A função do modelo dinâmico em FEA da prótese utilizada era simular o comportamento do seu mecanismo subatuado para flexão e extensão dos dedos. A Figura

44a representa o movimento de flexão dos dedos, que ocorre quando a polia puxa o tendão, fazendo assim que ele seja tracionado e começa-se a puxar a falange distal causando sua colisão com as paredes das demais falanges. Durante o movimento de extensão (Figura 44b) a polia retorna para posição inicial empurrando o tendão e consequentemente a falange distal, fazendo com que as demais falanges retornem para a posição completamente estendida.

Figura 44 – Mecanismo de atuação do dedo da prótese HANDi Hand durante os movimentos de flexão **(a)** e extensão **(b)** do dedo

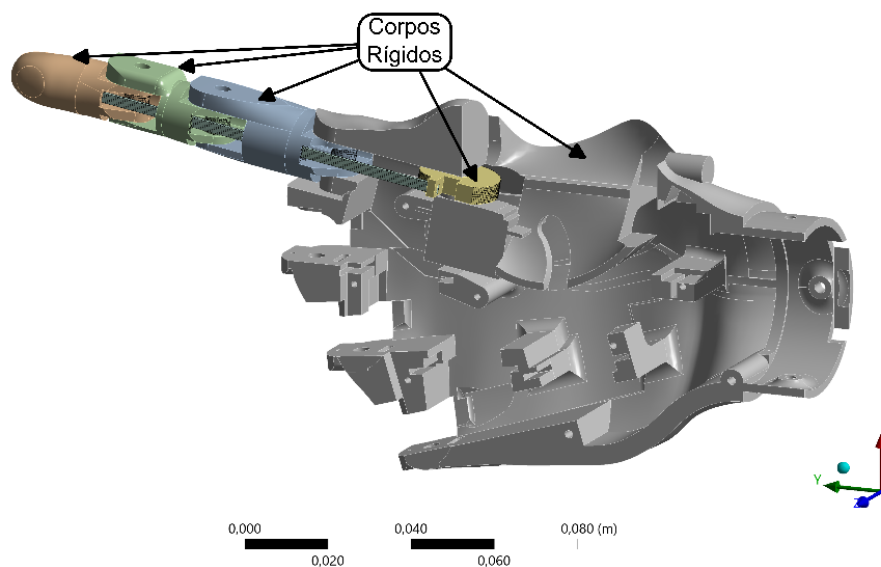


Fonte: Adaptado de BRENNEIS; DAWSON; PILARSKI, (2017).

Como observado na Figura 44, a dinâmica de funcionamento dos dedos da prótese depende do modelo de “colisão” entre o tendão e as falanges, além do modelo de flexão do tendão, que no caso da HANDi Hand era feito com uma abraçadeira de Nylon 6.6. Exatamente por ter que simular o comportamento de colisão e de flexão do tendão, foi escolhido o uso do software comercial de FEA Ansys Mechanical APDL®, devido à sua capacidade de simular modelos dinâmicos complexos, com comportamento não linear e além de possuir modelos de contato de colisão. O grande problema ao se utilizar esse tipo de modelo é o tempo necessário para sua solução, devido à grande quantidade de elementos de malha envolvidos no processo. Assim, para viabilizar o treinamento do controlador optou-se, a priori, por modelar apenas o dedo indicador (dedo 2), apresentado na Figura 45.

Para otimização da solução, foram combinados no modelo elementos de corpos rígidos com elementos flexíveis. Como pode ser observado na Figura 45, as falanges, a palma da prótese e a polia do mecanismo foram considerados como elementos rígidos, enquanto o tendão artificial foi selecionado como um elemento flexível, construído utilizando um elemento de viga retangular com 1 mm de altura por 2,5 mm de largura. Para os elementos rígidos foi adotado o Ácido Polilático (PLA) como material e para o tendão artificial o Nylon 6.6, com as características da Tabela 7.

Figura 45 – Modelo do dedo indicador da HANDi Hand construído no *software* ANSYS Mechanical APDL®



Fonte: Elaborado pelo autor.

Tabela 7 – Propriedades dos materiais utilizados na construção do modelo

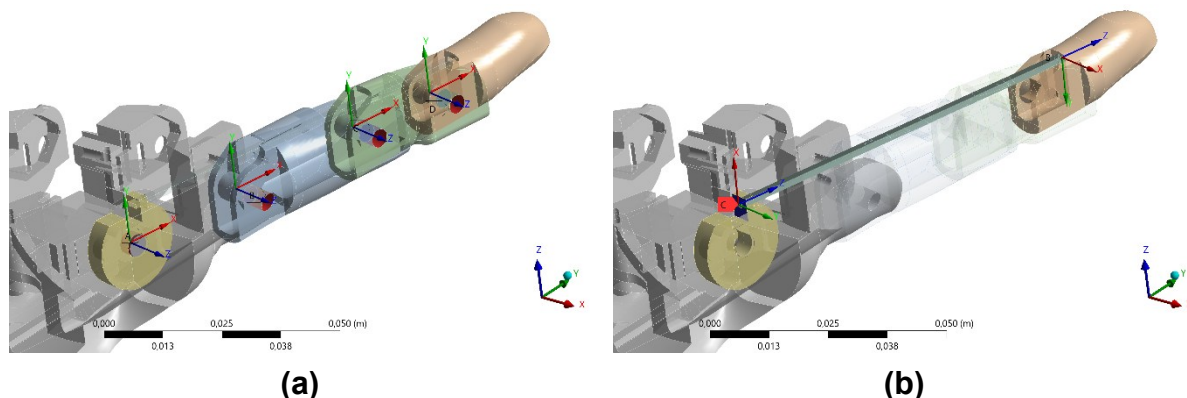
Propriedade [Unidade]	Material	
	Ácido Polilático (PLA)	Nylon 6.6
Densidade [kg/m ³]	1244	1130
Módulo de Young [GPa]	2,939	1,153
Coefficiente de Poisson	0,33	0,34
Resistencia ao Escoamento [Mpa]	-	37,08
Resistencia a Tração [MPa]	56,92	45,83

Fonte: ANSYS INC., (2025)

Conectando os componentes da prótese, foram definidos elementos de junta. Para conexão da polia e da flange proximal com a palma, bem como a ligação entre as falanges foram utilizadas juntas de revolução com eixo z conforme apresentadas na Figura 45a. Para as juntas utilizadas na ligação entre a flange proximal e a palma e as juntas interfalangeanas, foram definidas restrições do tipo *stop*, garantindo que a faixa de movimento da junta entre de 0° a 90° , representando os limitadores de movimento presentes. Apesar de as restrições do modelo simplificarem a dinâmica de colisão das falanges, foi identificado um problema durante a solução do modelo, dado que ao atingir as posições limitantes, as falanges passaram a apresentar pequenas oscilações. Assim, para solucionar esse problema foram definidos como valores de amortecimento de 5×10^{-5} Nms/ $^\circ$ e de rigidez de 0 Nm/ $^\circ$, garantindo que as oscilações sejam reduzidas e que a dinâmica do movimento não fosse modificada. Para todas as demais juntas de revolução, os valores de amortecimento e rigidez foram

zerados. Para conexão do tendão artificial com a polia e com a falange distal foram utilizadas juntas fixas, com eixos coordenados conforme apresentado na Figura 46b. Por fim a palma foi considerada fixa ao solo.

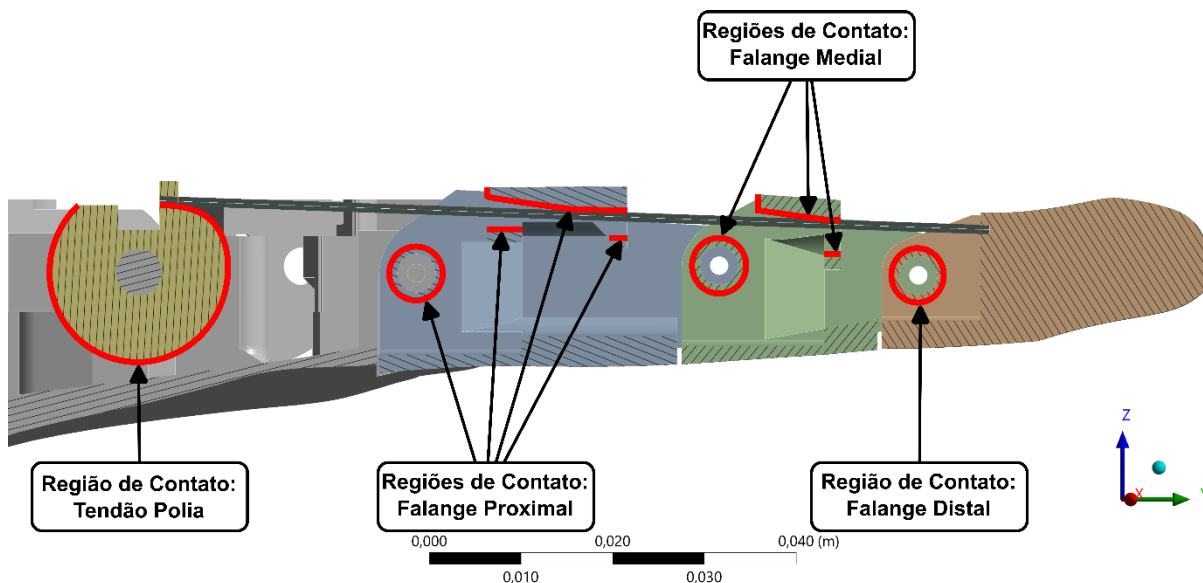
Figura 46 – Juntas de revolução (a) e juntas fixas (b) definidas na construção do modelo do mecanismo da prótese



Fonte: Elaborado pelo autor.

Para modelar o contato entre o tendão artificial e as falanges, bem como a flexão do tendão na polia, foram definidos contatos sem fricção nas regiões apresentadas na Figura 47. Todos os contatos foram ajustados de modo que não ocorresse complicações na convergência do modelo matemático e interferência significativa nos resultados, como a garantia de que a energia de amortecimento de estabilização fosse significativamente menor que as energias de deformação do sistema. Os parâmetros de contato utilizados estão apresentados na Tabela 8.

Figura 47 – Regiões de contato entre as falanges, a polia e o tendão artificial



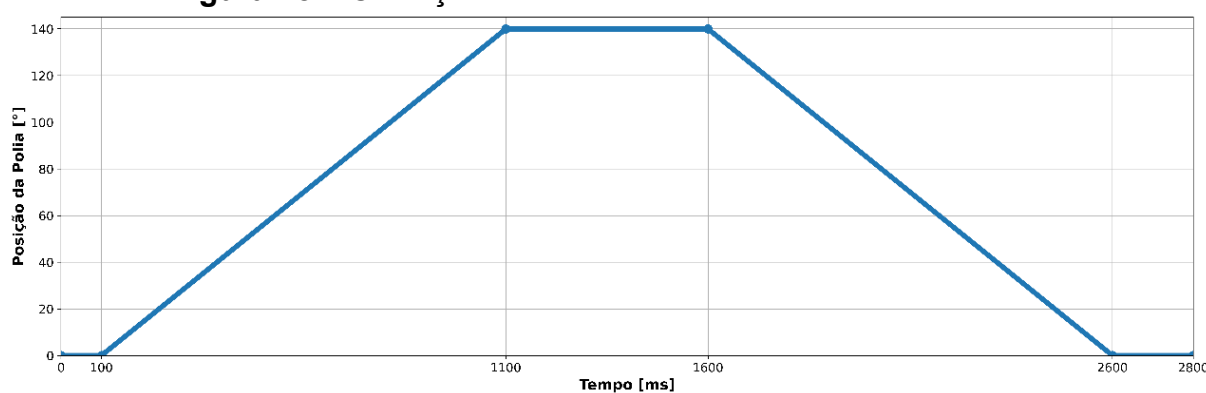
Fonte: Elaborado pelo autor.

Tabela 8 – Resumo dos parâmetros utilizados na definição dos contatos do modelo

Parâmetro	Configuração/Valor
	Formulação <i>Augmented Lagrange</i>
Pequenos Escorregamentos	Adaptativo
Tolerância de Penetração	5E-5 m
Fator de Rigidez Normal	0,5
Atualização de Rigidez	Toda iteração
Fator Estabilizador de Amortecimento	0
Raio da Região de Contato	2E-3 m
Controles de Passo Temporal	Bissecção Automática
Tratamento de Interface	<i>Offset</i> de 4E-4 m, com efeito de rampa

Fonte: Elaborado pelo autor.

Como o método FEA é influenciado pela malha, foi realizado um teste de malha para assegurar que a discretização não interferisse os resultados. Para isso, foi configurada uma condição de contorno padrão para simular a operação do mecanismo da prótese e avaliar a malha utilizada. O mecanismo flexiona e estende o dedo através da polia que está acoplada a um servo motor modelo HS-40 (Hitec Group, San Diego, California, EUA) que possui uma velocidade angular máxima de $600\text{ }^\circ/\text{s}$ e torque máximo de 73,6 Nmm. Como o comando de atuação, o servo motor utilizava a posição do eixo, sendo assim escolhido como condição de contorno de entrada do modelo a posição da polia relativa à posição inicial conforme apresentado na Figura 48. Para prevenir acelerações excessivas, optou-se no teste de malha por utilizar uma velocidade angular inferior à máxima do motor ($140\text{ }^\circ/\text{s}$), para a realização dos movimentos de flexão e extensão do dedo.

Figura 48 – Condição de contorno utilizada no teste de malha

Fonte: Elaborado pelo autor.

Após definir a condição de contorno, foram definidos os parâmetros comuns de todas as malhas, sendo o único parâmetro alternado entre uma malha e outra o tamanho máximo dos elementos. Todas as malhas possuíam elementos de primeira ordem, sem tamanho adaptativo. Nas falanges proximal e medial, foram permitidos elementos quadriláteros e triangulares, já na falange distal e na polia foram utilizados somente elementos quadriláteros. Com os parâmetros em comum estabelecidos, as malhas testadas foram definidas reduzindo o tamanho máximo permitido ao elemento a partir de uma malha base com tamanho máximo de elemento de 1 mm, resultando em cinco configurações de malha, apresentadas na Tabela 9. Para garantir a qualidade dos elementos, foi definido que o valor de máximo de *skewness* para cada malha não ultrapassasse 0,5.

Tabela 9 – Malhas Utilizadas durante o teste de malha

	Tamanho Máximo do Elemento [mm]	Número de Nós	Número de Elementos	Skewness Máximo
Malha 1	1	1093	822	0,3654
Malha 2	0,75	1735	1373	0,2104
Malha 3	0,5	3416	2887	0,2925
Malha 4	0,35	6692	5943	0,3852
Malha 5	0,25	12437	11407	0,4528

Fonte: Elaborado pelo autor.

Após a realização das simulações dinâmicas com todas as malhas, os ângulos máximos obtidos em cada junta são apresentados na Tabela 10. Observa-se que, na maioria das malhas, os ângulos máximos atingidos foram semelhantes, com uma variação percentual máxima de 8,23% entre as malhas 2 e 3, referente ao valor de rotação relativa máxima na junta distal.

Tabela 10 – Diferenças entra rotação angular relativa das juntas da prótese em cada uma das malhas testadas

Rotação Relativa Máxima	Malha 1	Malha 2	Malha 3	Malha 4	Malha 5
Junta Proximal [°]	75,175	72,960	70,200	69,607	69,163
Varição Relativa Junta Proximal [%]	–	3,121	3,932	0,8519	0,6420
Junta Medial [°]	64,310	62,691	62,905	62,923	63,056
Varição Relativa Junta Medial [%]	–	2,583	0,3402	2,860E-2	0,2109
Junta Distal [°]	49,404	51,854	56,504	56,570	57,153
Varição Relativa Junta Distal [%]	–	4,725	8,230	0,1167	1,020

Fonte: Elaborado pelo autor.

A última validação realizada no modelo FEA foi garantir que era possível utilizar o modelo como um ambiente de treinamento. Para isso, condições de contorno semelhantes com às apresentada na Figura 48 foram utilizadas, porém com velocidades de $100\text{ }^\circ/\text{s}$, de $200\text{ }^\circ/\text{s}$, de $300\text{ }^\circ/\text{s}$ e de $400\text{ }^\circ/\text{s}$, utilizando dois passos temporais de solução: 1 ms e 2 ms, e foi observado se o modelo era capaz de convergir com as características da malha e passo temporal. Nessa validação foi observado que apenas as Malhas 3, 4 e 5 apresentaram convergência para todas as velocidades e passos temporais. Assim, devido à sua quantidade inferior de elementos e sua capacidade de ser utilizada em diferentes velocidades a Malha 3 foi escolhida. Além disso, para otimizar o tempo de solução, foi adotado o passo temporal de 2 ms.

Com a malha definida, foi possível concluir a integração entre a biblioteca Gymnasium e o Ansys Mechanical APDL® para a construção do ambiente de treinamento. Como todo o mecanismo de acionamento do dedo dependia da polia acoplada ao servo motor, a entrada do modelo correspondia à posição do motor. Como o passo de solução da simulação foi definido em 2 ms, o Agente teria a oportunidade de atualizar a posição do motor nesse mesmo espaço de tempo, porém dada a velocidade máxima de $600\text{ }^\circ/\text{s}$, não era possível movimentar o eixo de 0° a 180° em apenas um passo temporal. Dessa forma, definiu-se que o Agente determinaria incrementos de posição em uma faixa de $-0,8^\circ$ a $0,8^\circ$, limitando a velocidade máxima do motor a $400\text{ }^\circ/\text{s}$, assegurando a estabilidade do funcionamento. Assim, como indicado na Figura 43, a cada passo de temporal o Agente definia o incremento de posição, que era somado a posição anterior e enviado ao modelo numérico. Este, por sua vez, retornava à posição e a velocidade angular de cada junta, que eram retornados ao Agente para determinação da próxima ação.

4.1.4.2 Função de Recompensa

Com o modelo em FEA da prótese definido, a etapa final para elaboração do ambiente de treinamento foi determinar a função de recompensa. Diferentemente do apresentado no teste piloto (Seção 3.1.1), para o treinamento do controlador algumas mudanças foram necessárias, pois ao utilizar um modelo de FEA para simular a dinâmica do dispositivo, o custo computacional foi elevado, necessitando de uma redução no número de episódios para atingir a convergência do treinamento. Dessa forma a

função de recompensa foi dividida em quatro parcelas: erro de posição, bônus direcional, bônus de estabilidade e penalidade.

O erro de posição (Equação (41)) tem como objetivo mapear a diferença entre a posição espacial da ponta do dedo com relação a posição alvo do movimento executado. Para isso, utilizou-se uma função exponencial para valorizar ações que se aproximassem progressivamente da posição esperada. Para mensurar a distância, foi utilizado o erro absoluto (E) apresentado na Equação (42), onde E_x representa a diferença na coordenada x da posição alvo com a posição executada e E_y representa a diferença na coordenada y da posição alvo com a posição executada. O erro então foi normalizado pelo valor de 0,15, que representa o erro máximo que poderia ser obtido, uma vez que é o diâmetro do espaço de ações do dedo da prótese. Por fim a constante -3 foi utilizada para garantir o bom espaçamento entre as recompensas obtidas em posições próximas e distantes. Assim, o erro de posição resultava em uma recompensa mínima no valor de -0,45 e máxima de 0,50.

$$R_p = e^{-3\frac{E}{0,15}} - 0,5 \quad (41)$$

$$E = \sqrt{(E_x)^2 + (E_y)^2} \quad (42)$$

O bônus direcional (Equação (43)) foi utilizado para auxiliar o Agente no reconhecimento de ações que o aproximem da posição-alvo. Para determinar o bônus, foi calculada a melhora relativa entre o erro do passo anterior (E_{i-1}) e o erro do passo atual (E_i) utilizando a Equação (42). O bônus direcional só foi utilizado quando a ponta do dedo estava a uma distância grande da posição desejada (0,01 m) e caso houvesse uma redução no erro. Outra característica importante é que a melhora relativa do erro era limitada à faixa de 0 a 1, fazendo o bônus variar em uma faixa de recompensa entre 0 e 0,1.

$$B_d = \begin{cases} 0, & E_i < 0,01 \text{ m} \\ 0, & E_{i-1} - E_i \leq 0 \\ 0,1 \text{clip}\left(\frac{E_{i-1} - E_i}{E_i}, 0,1\right), & \text{caso contrário} \end{cases} \quad (43)$$

O bônus de estabilidade (Equação (44)) foi atribuído ao Agente quando a posição executada estava próxima da posição desejada e tinha como objetivo beneficiá-lo quando não eram realizadas ações abruptas. Assim, quando o erro era menor que

0,01 m, quanto menor fosse a ação tomada (a) pelo Agente, maior era o bônus. A ação foi normalizada pelo incremento máximo de posição ($0,8^\circ$) garantindo que o bônus variava entre 0 e 0,2.

$$B_e = \begin{cases} 0, & E > 0,01 \text{ m} \\ 0,2 \left(1 - \frac{|a|}{0,8}\right), & E \leq 0,01 \text{ m} \end{cases} \quad (44)$$

A penalidade (Equação (45)) era atribuída ao Agente quando ele tomava ações que levavam o motor para posições além dos limites permitidos. Os limites de posição da polia foram definidos pelos limites de operação do motor, sendo definidos como 0° a menor posição permitida e 160° a maior. A escolha do limite superior de posição da polia ser 160° foi estabelecida através de observações das simulações do modelo, em que se observou que nessa posição o dedo já havia executado a flexão completa. Assim, a posição da polia (θ) foi comparada com o limite inferior ou superior, dependendo do limite extrapolado, e então normalizada pelo incremento máximo permitido ao Agente de $0,8^\circ$. Como pode ser observado a recompensa variava de -0,5 a -0,75 proporcionalmente com a posição extrapolada.

$$P = \begin{cases} -0,5 - \left(\frac{\theta}{0,8}\right)^2, & -0,40^\circ \leq \theta < 0^\circ \\ -0,5 - \left(\frac{\theta - 160}{0,8}\right)^2, & 160^\circ < \theta \leq 160,4^\circ \\ 0, & \text{caso contrário} \end{cases} \quad (45)$$

Outra diferença adotada nesse ambiente de treinamento foi que o episódio não era interrompido quando as posições limites eram extrapoladas. Essa característica do ambiente de treinamento foi adotada para possibilitar ao Agente retornar à posição permitida, assegurando consistência durante o treinamento. Assim, o episódio só era interrompido caso a ação fizesse a polia a extrapolar mais de $0,4^\circ$ a posição mínima ou máxima, em qualquer outra condição a posição da polia era mantida no limite e o episódio prosseguia. Quando interrompida, a função de penalidade foi substituída pela função de interrupção, conforme apresentado na Equação (46).

$$P_{\text{interrupção}} = \begin{cases} -1,5 - \left(\frac{\theta}{0,8}\right)^2, & -0,80^\circ \leq \theta < -0,4^\circ \\ -1,5 - \left(\frac{\theta - 160}{0,8}\right)^2, & 160,4^\circ < \theta \leq 160,8^\circ \end{cases} \quad (46)$$

Uma outra condição de interrupção do episódio ocorreria em caso de divergência na solução numérica do modelo em FEA da prótese. Dadas as características numéricas do método FEA, algumas ações poderiam levar a estrutura do dedo a colapsar, ou a condições em que a solução numérica se tornava inviável. Para essas ações adotou-se uma penalidade fixa de -3. Considerando todas as parcelas, a função de recompensa foi representada pela Equação (47).

$$R = \begin{cases} R_p + B_d + B_e + P, & -0,4^\circ \leq \theta \leq 160,4^\circ \\ -3, & \text{caso ocorra a divergência da solução numérica} \\ P_{\text{interrupção}}, & \text{demais posições} \end{cases} \quad (47)$$

4.1.5 Definição das Trajetórias-alvo

As trajetórias alvo utilizadas para o treinamento do controlador de trajetórias foram definidas com base nos dados coletados das posições das articulações da mão dos voluntários no DB2 e calibradas no DB9 do NINAPRO, durante a execução dos movimentos selecionados para o classificador. Inicialmente os dados foram tratados para serem compatíveis com as janelas do sEMG, ou seja, foram separados em janelas de 200 ms com sobreposição de 160 ms, ou seja, espaçadas em 40 ms. Para cada janela utilizou-se o último valor medido como a posição alvo de cada articulação.

Com os valores das janelas definidas, os dados foram separados em episódios. Nesta etapa, cada intervalo de movimento teve seus dados separados e adicionados no início e no final com dados referentes à condição de relaxamento. Para garantir que todos os episódios tivessem o mesmo tamanho, o tempo na condição de relaxamento foi dinamicamente ajustado, assim todos os episódios apresentaram sempre 180 amostras (7,2 segundos) na seguinte sequência de eventos: relaxamento; movimento; relaxamento. O valor de 180 amostras foi definido de forma que a quantidade mínima de amostras em condição de relaxamento fosse 20 amostras, garantindo o menor tamanho de episódio possível.

Dos episódios definidos foram separados os episódios referentes aos movimentos desejados de preensão de força, pinça trípode e pinça bidigital. Como discutido na Seção 4.1.4.2, para o cálculo da função de recompensa era necessária a definição da posição da ponta do dedo. Para extrapolar os dados à posição desejada da prótese, uma simplificação foi feita considerando que os ângulos das articulações deveriam ser os ângulos esperados de cada junta do dedo protético. Assim, ao aplicar

modelo cinemático direto do dedo protético apresentado na Equação (48), foram determinadas as coordenadas x e y esperadas da ponta do dedo, onde l_i representava o comprimento de cada falange da prótese e θ_i o ângulo relativo de cada uma das articulações. Como sistema de referência, foi utilizado o eixo da junta de revolução que representava a articulação metacarpofalangeana na prótese com eixo x na extensão do dedo.

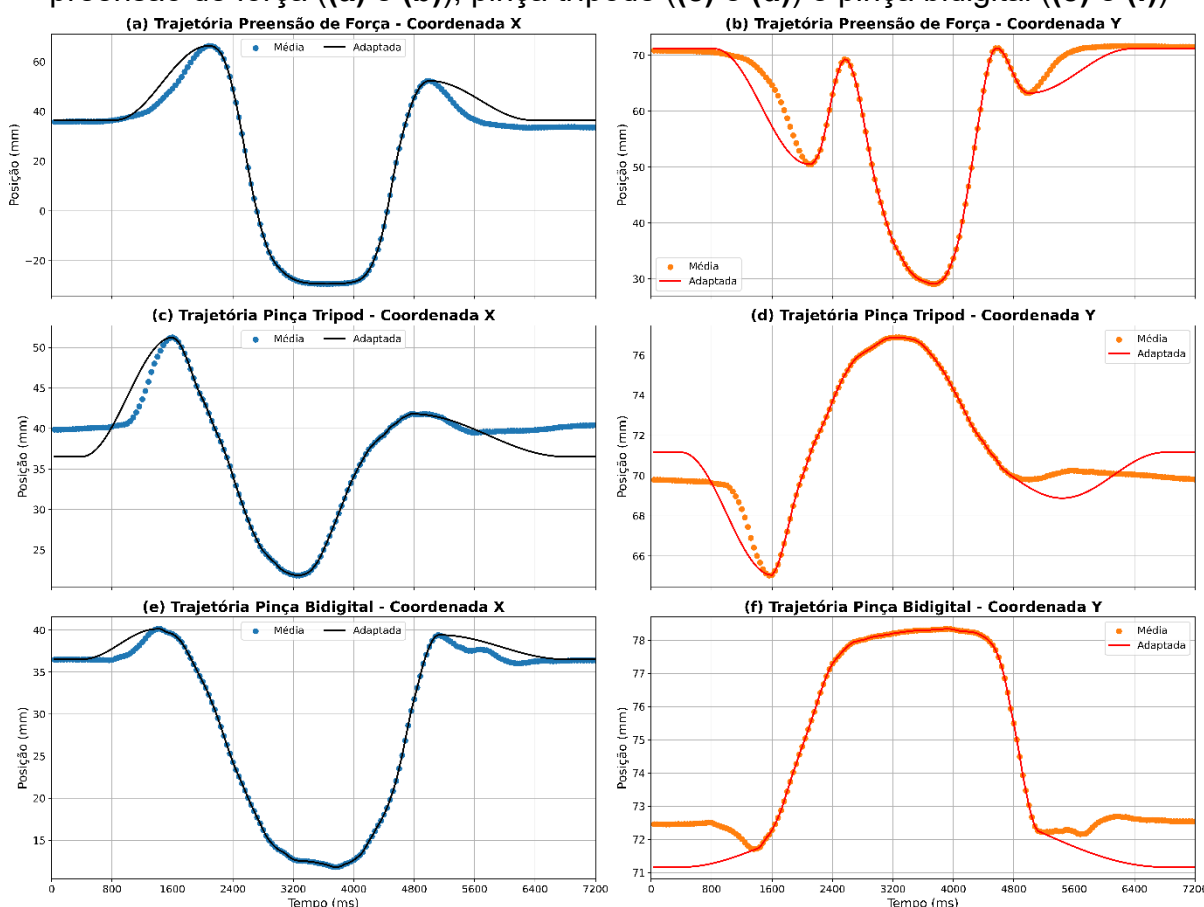
$$\begin{Bmatrix} x \\ y \\ z \end{Bmatrix} = \begin{Bmatrix} l_1 \cos(\theta_1) + l_2 \cos(\theta_1 + \theta_2) + l_3 \cos(\theta_1 + \theta_2 + \theta_3) \\ l_1 \sin(\theta_1) + l_2 \sin(\theta_1 + \theta_2) + l_3 \sin(\theta_1 + \theta_2 + \theta_3) \\ 0 \end{Bmatrix} \quad (48)$$

Uma vez que as trajetórias de cada movimento para cada usuário foram definidas, foi então possível extrair as trajetórias “ideais” para cada movimento. Inicialmente, as trajetórias médias de cada movimento foram definidas a partir dos ângulos médios de cada articulação em cada passo de tempo de todos os usuários, apresentadas na Figura 49. Observou-se que a posição no estado de relaxamento, apesar de próxima, não era comum para todas as trajetórias, logo, para definir uma pose padrão para o relaxamento, foi calculada a média de todas as posições angulares quando o estado de relaxamento era o desejado na janela. Apesar de solucionar o problema, estipulando uma posição comum de relaxamento, a transição entre o relaxamento e o movimento precisou ser redefinida. Para conectar a pose de relaxamento com o início e o final do movimento executado todas as janelas definidas com a condição de relaxamento foram fixadas em uma posição constante. Durante a execução do movimento, foi preservada a trajetória entre os dois maiores picos de posição, observando o eixo x . A transição entre a condição de relaxamento e os picos foi realizada por meio de funções senoidais que tangenciavam as duas curvas como demonstrado na Figura 48, resultando em uma transição suave.

Por fim, foram definidas as entradas de trajetória para cada passo de tempo, para o treinamento. Para a condição de relaxamento a entrada de trajetória foi $\{1 \ 0 \ 0 \ 0\}^T$, para a preensão de força foi utilizada a entrada $\{0 \ 1 \ 0 \ 0\}^T$, para a pinça trípode adotou-se a entrada $\{0 \ 0 \ 1 \ 0\}^T$ e para pinça bidigital foi atribuída a entrada $\{0 \ 0 \ 0 \ 1\}^T$. Todas as entradas de trajetórias foram definidas utilizando a mesma técnica. A entrada inicial de relaxamento foi utilizada até o início de abertura do dedo. Durante a transição até o ponto mínimo da coordenada x foi determinada a entrada de trajetória referida ao movimento executado. Do ponto mínimo até o fim do

movimento, a entrada retornava à condição de relaxamento. Essa definição levou como princípio o funcionamento da mão durante a execução dos movimentos. Inicialmente, parte-se da posição de relaxamento estendendo completamente os dedos e depois os flexionando na posição desejada. Para retornar à posição de relaxamento, os dedos são estendidos novamente retornando à condição de relaxamento.

Figura 49 – Coordenadas médias (x e y) e trajetórias esperadas dos episódios: prensão de força ((a) e (b)), pinça tripode ((c) e (d)) e pinça bidigital ((e) e (f))



Fonte: Elaborado pelo autor.

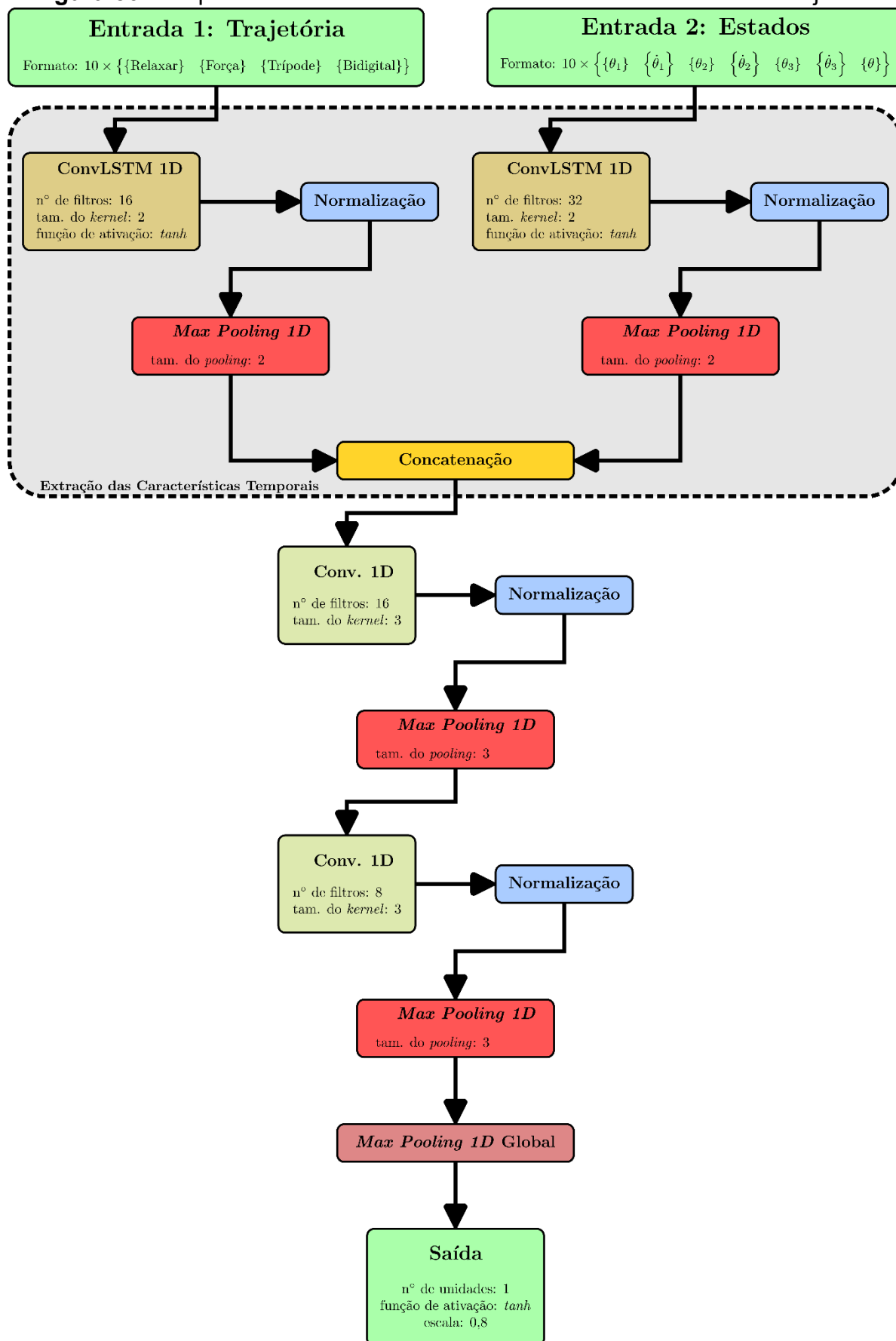
4.1.6 Arquitetura e Treinamento do Controlador de Trajetórias

Para efetuar o treinamento da RNA, que atuou como controlador de trajetórias da prótese, foi utilizado o algoritmo TD3. A mudança do DDPG para o TD3 foi necessária devido ao uso do modelo em FEA da prótese. Como o custo computacional para solução do modelo era elevado, fez-se necessário escolher um algoritmo que fosse próximo ao DDPG, utilizado no teste piloto, porém que tivesse uma tendência de convergência mais rápida de treinamento. Devido a essas necessidades, o algoritmo TD3

foi utilizado, pois ele é uma atualização do algoritmo DDPG que tende a apresentar uma convergência com menos iterações.

Para a RNA utilizada como Agente, foi adotada uma arquitetura que combinava RNNs e camadas convolucionais, visando obter uma convergência mais rápida. Dentre várias arquiteturas testadas a que apresentou os melhores resultados é a arquitetura apresentada na Figura 50. Nessa arquitetura foram novamente separadas as entradas referentes à seleção de trajetórias das entradas referentes aos estados da prótese, juntamente com a posição polia. Para as entradas de trajetórias, foi utilizada uma camada convolucional LSTM com 16 filtros e tamanho 2 de *kernel*, enquanto para os estados da prótese adotou-se uma camada convolucional LSTM com 32 filtros e tamanho 2 de *kernel*. Após as camadas de LSTM, foi então utilizada uma camada de normalização e uma camada de *pooling* de valor máximo com tamanho 2 de *kernel* em ambas as entradas. Após a camada de *pooling* as entradas foram concatenadas e passaram por duas camadas de convolução simples utilizando 16 e 8 filtros com tamanho 3 de *kernel*. Após cada camada de convolução, foi aplicada uma camada de normalização e uma camada de *pooling* máximo com tamanho de *kernel* 3. Antes da camada de neurônios de saída foi aplicada uma camada de *pooling* máximo global. Por fim, a camada de saída um neurônio com função de ativação de tangente hiperbólica, para reduzir a possibilidade de ações com valor terminal, sendo escalonada por um fator de 0,8, garantindo assim que o espaço das ações fosse contínuo, com valores dentro da faixa de -0,8 e 0,8, limites incrementais da posição do motor a cada passo de tempo. Todas as camadas de convolução utilizaram a função de ativação *leaky ReLU*, com α igual a 0,3.

Figura 50 – Arquitetura da RNA utilizada como Ator no controlador de trajetória



Fonte: Elaborado pelo autor.

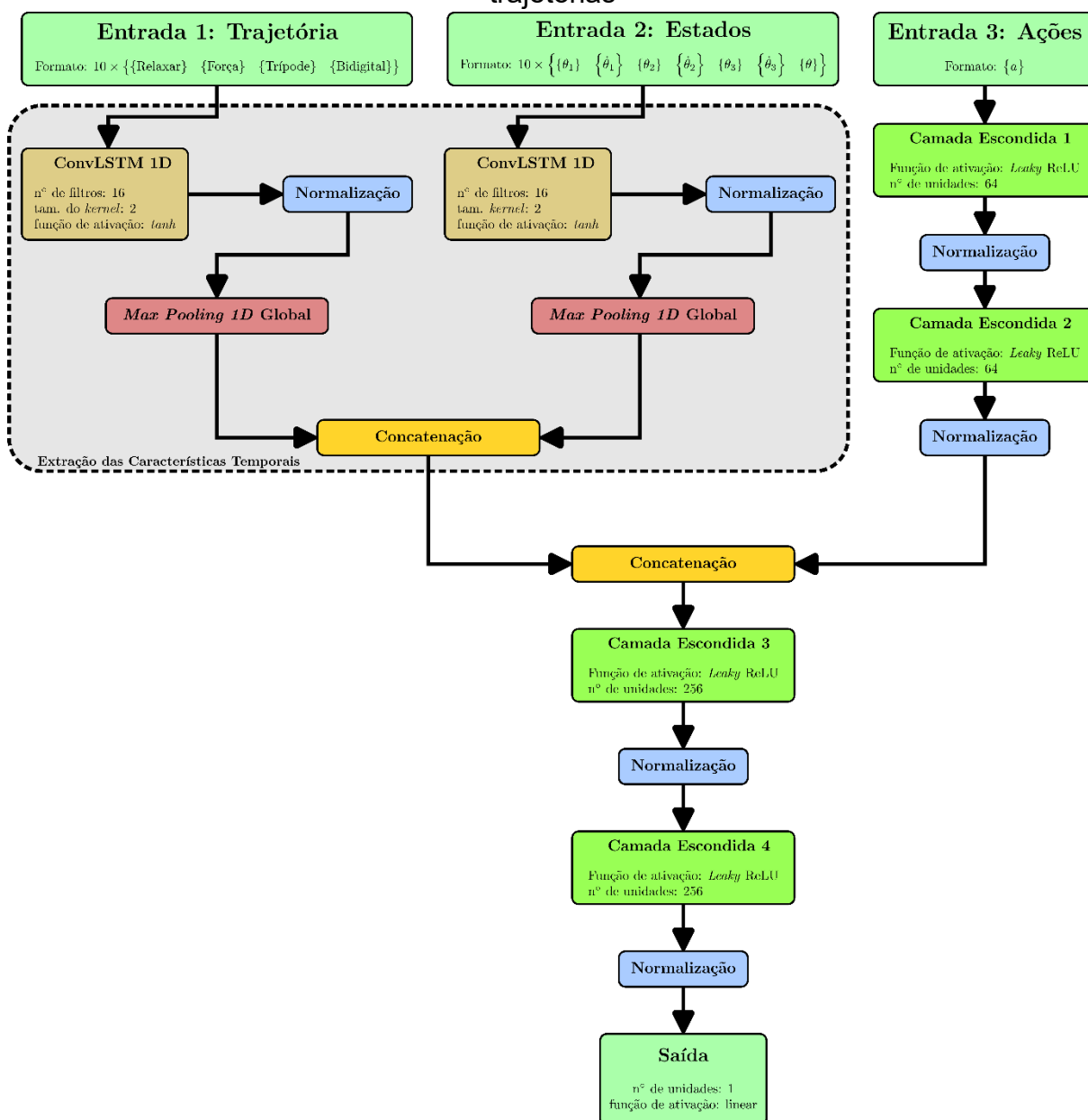
Assim como o DDPG, o TD3 também faz parte da classe de algoritmos ator-crítico, ou seja, depende de uma RNA auxiliar adicional para determinar a função valor utilizada para treinar o Agente, o Crítico. Apesar de utilizar duas RNAs auxiliares para a determinação da função valor (ou dois Críticos), ambas possuem a mesma arquitetura, mas com parâmetros distintos e independentes. Assim, para realização do treinamento, foi definida a arquitetura apresentada na Figura 51 como a RNA utilizada para ambos os Críticos. Para avaliar a ação do Agente três entradas foram utilizadas, a entrada de seleção de trajetória, a entrada dos estados da prótese e a entrada das ações do Ator/Agente. As entradas de seleção de trajetória e dos estados cinemáticos passaram cada uma por uma camada convolucional LSTM com 16 filtros e com tamanho 2 de *kernel* seguida por uma camada de normalização e uma camada de *pooling* máximo global. Já as entradas referentes as ações do Ator passaram por duas camadas ocultas com 64 neurônios com função de ativação *leaky* ReLU, com α igual a 0,3, seguidas por camadas de normalização. As saídas da última camada oculta ligada a cada entrada foram concatenadas e então ligadas a duas camadas ocultas com 256 neurônios com função de ativação *leaky* ReLU, com α de 0,3. Por fim, a camada de saída com um neurônio com função de ativação linear foi utilizada, prevendo a função valor de cada um dos Críticos que será usado no algoritmo.

Para as entradas de seleção de trajetória e de estados, foram utilizados os valores referentes aos últimos dez passos temporais, uma vez que utilização da célula LSTM se deu para o Agente prever as características temporais das entradas. Além disso, os estados da prótese foram normalizados para a faixa de 0 a 1 proporcionalmente aos valores máximos e mínimos de cada variável. Para as posições das juntas de revolução, o valor mínimo utilizado foi de 0° enquanto o máximo foi de 90° . Para as velocidades das juntas de revolução os valores máximo e mínimo foram idênticos às velocidade máximas e mínimas permitidas ao motor, ou seja, de $-400^\circ/\text{s}$ a $400^\circ/\text{s}$. Por fim, para a polia a posição mínima de 0° e máxima de 160° .

Adequando melhor o ambiente para o treinamento, algumas mudanças na posição inicial do dedo foram necessárias. De acordo com as trajetórias definidas (Figura 49), o dedo apresenta uma flexão inicial, enquanto no modelo FEA era iniciado com o dedo na posição de extensão máxima. Para auxiliar o Agente, ao iniciar o episódio, a polia foi rotacionada em 60° por 150 ms e deixada por 50 ms em repouso para então

o episódio ser iniciado, garantindo assim uma maior proximidade da posição inicial do dedo com a condição de relaxamento.

Figura 51 – Arquitetura utilizada nos Críticos para o treinamento do controlador de trajetórias



Fonte: Elaborado pelo autor.

Assim como ocorreu na adaptação do algoritmo de treinamento do teste piloto para o controlador de trajetórias, algumas modificações foram necessárias, conforme apresentadas no Algoritmo 4. Para garantir a apresentação de todas as trajetórias ao Agente, os episódios foram sequencialmente executados, sendo definida uma época de treinamento como uma sequência dos três episódios apresentados. Além disso, antes de se iniciar o treinamento 18.000 amostras foram armazenadas por meio de

entradas aleatórias no ambiente de treinamento. As taxas de aprendizado dos Críticos e do Ator foram de $3E-4$ com uma taxa de suavização das redes alvo (τ) de $5E-3$ e taxa de desconto de ações futuras (γ) de 0,99. O tamanho máximo de memória de repetição utilizado foi de $1E6$ amostras com 512 amostras utilizadas em cada iteração de treinamento. Garantindo que tanto amostras de falha como de não falha sejam apresentadas, foi definido que 20% das amostras de treinamento fossem de passos de falha do Agente (102 amostras). Como ruído de extrapolação e exploração do ambiente foi utilizado ruído Gaussiano com desvio padrão decrescente a uma taxa de 0,002, valor inicial de 0,16 e valor mínimo de 0,02. Por fim como otimizador foi utilizado o AdamW com decaimento de $1E-4$ e limitação de gradiente em 1. O treinamento foi executado até que o número de épocas fosse igual a 200 (ou 600 episódios fossem concluídos) ou que o valor da recompensa de todos os episódios da época fosse maior que 1400 pontos, o que representaria um erro médio de trajetória de 4 mm.

Assim como no desenvolvimento classificador todas as etapas de treinamento e preparação do treinamento, foram desenvolvidas em Python™ com o treinamento sendo executado pela API TensorFlow (MARTÍN ABADI et al., 2015).

Algoritmo 4: TD3 utilizado no treinamento do controlador de trajetória

- 1 Inicialize os críticos $Q_1(i, s, a | \theta^{Q_1})$ e $Q_2(i, s, a | \theta^{Q_2})$ e o ator $\mu(i, s | \theta^\mu)$
- 2 Inicialize as redes alvo Q'_1, Q'_2 e μ' com os parâmetros $\theta^{Q'_1} \leftarrow \theta^{Q_1}, \theta^{Q'_2} \leftarrow \theta^{Q_2}$ e $\theta^{\mu'} \leftarrow \theta^\mu$
- 3 Inicialize a memória de repetição B
- 4 Inicialize o contador de iterações de treinamento $it \leftarrow 0$
- 5 **para época de 1 até 200 faça:**
- 6 Inicie a memória das recompensas da época E_R
- 7 Inicie a memória de passos da época E_S
- 8 **para episódio em [força, trípode, bidigital] faça:**
- 9 Inicialize a recompensa do agente $R \leftarrow 0$
- 10 Inicialize o ruído com $\sigma \leftarrow 0,16$
- 11 Inicialize o contador de passos do episódios $S \leftarrow 0$
- 12 Receba os estados iniciais s_1 do ambiente
- 13 **para T de 1 até 180 faça:**
- 14 Selecione e entrada de trajetória m_t para os próximos 20 passos
- 15 **para t de 1 até 20 faça:**
- 16 Selecione uma ação de acordo com a política atual e o ruído:

$$a_t \leftarrow \text{clip}(\mu(i_T, s_t | \theta^\mu) + \epsilon, a_{min}, a_{max}), \quad \epsilon \approx \mathcal{N}(0, \sigma)$$
- 17 Execute a ação a_t e observe a recompensa r_t , o próximo estado s_{t+1} e se a ação teve falha ou não (f_t)

Algoritmo 4 (continuação): TD3 utilizado no treinamento do controlador de trajetória

18	Atualize a recompensa do agente $R \leftarrow R + r_t$
19	Atualize o contador e passos $S \leftarrow S + 1$
20	se $t < 20$: Armazene $(i_T, s_t, a_t, r_t, i_{T+1}, s_{t+1}, d_t, f_t)$ em R
21	se $t = 20$: Armazene $(i_T, s_t, a_t, r_t, i_{T+1}, s_{t+1}, d_t, f_t)$ em R
22	Atualize as iterações de treinamento $it \leftarrow it + 1$
23	Pegue uma miniamostra aleatória com 204 passos de falha e 820 passo sem falha da transição $(i_n, s_n, a_n, r_n, i_{n+1}, s_{n+1}, d_n, f_n)$ de R
24	Determine a ação alvo suavizada: $a'(s) = clip(\mu'(i_{n+1}, s_{n+1} \theta^{\mu'}) + clip(\epsilon; 0,4; 0,4); 0,8; 0,8), \quad \epsilon \sim \mathcal{N}(0; 0,08)$
25	Determine o valor alvo da função valor $\delta_{n+1} = r_n + \gamma(1 - d_n) \min_{i=1,2} (Q'_i(i_{n+1}, s_{n+1}, a'(s_{n+1}) \theta^{Q'_i}))$
26	Atualize separadamente os críticos com um passo de gradiente descendente usando: $\nabla_{\theta^{Q_i}} \frac{1}{N} \sum_n (Q_i(i_n, s_n, a_n \theta^{Q_i}) - \delta_{n+1})^2, \quad i = 1,2$
27	se $it/2$ tem resto 0 então :
28	Atualize a política com um passo de método de gradiente ascendente usando: $\nabla_{\theta^\mu} \frac{1}{N} \sum_n Q_1(s_n, \mu(s_n \theta^\mu) \theta^{Q_1})$
29	Atualize as redes alvo individualmente: $\theta^{Q'_i} \leftarrow \tau \theta^{Q_i} + (1 - \tau) \theta^{Q'_i}, \quad i = 1,2$ $\theta^{\mu'} \leftarrow \tau \theta^\mu + (1 - \tau) \theta^{\mu'}$
30	fim-se
31	se f_t é terminal: Interrompa episódio
32	se $S = 3600$ e R é a maior de todos: Armazene os parâmetros do ator
33	fim-para
34	fim-para
35	Atualize o ruído de extrapolação/exploração $\sigma \leftarrow \min(0,998\sigma; 0,02)$
36	Armazene R em E_R
37	Armazene S em E_S
38	fim-para
39	se todos $E_r \geq 1400$ e todos $E_s = 3600$: Interrompa o treinamento
40	fim-para

4.1.7 Avaliação do Controlador

Para avaliar o desempenho do controlador após os treinamentos testes virtuais, foram executados utilizando o ambiente do treinamento desenvolvido. No primeiro teste, foram utilizadas as entradas de trajetórias binárias, avaliando as respostas do controlador sem o ruído. Observou-se qual era o erro absoluto obtido entre a trajetória esperada e a trajetória executada, servindo como base para avaliação do treinamento do controlador.

Para avaliação do desempenho do controlador com dados de sEMG foram feitas duas validações, uma com dados de um indivíduo amputado do DB3, selecionado aleatoriamente e outra com voluntário 21 do DB7. A avaliação com o dado do voluntário do DB3 teve como objetivo avaliar como o controlador se comportava com dados utilizados no treinamento do controlador de trajetórias e do classificador, avaliando o desempenho do controlador em alternar entre as trajetórias selecionadas e executar uma trajetória próxima da esperada.

A avaliação com um indivíduo amputado do DB7 (voluntário 21) teve como objetivo simular o uso do sistema em uma condição real, pois o DB7 não foi utilizado em nenhum momento para treinar ambas as redes. Assim, antes de iniciar o teste, um processo de calibração do classificador foi necessário utilizando a técnica de transferência de treinamento, muito utilizada para melhoria do desempenho de redes de classificação de sinais (FAN *et al.*, 2023). Essa técnica consiste no retreinamento da RNA utilizando apenas os dados do usuário avaliado no teste. Para isso novamente 70% dos dados foram utilizados para a calibração. Finalizada a calibração, a avaliação completa do controlador mioelétrico foi realizada, utilizando como entradas de trajetórias as saídas do classificador calibrado. Foi então avaliada a diferença entre as trajetórias executadas, tanto com os dados internos do treinamento como com um dado externo de um indivíduo amputado.

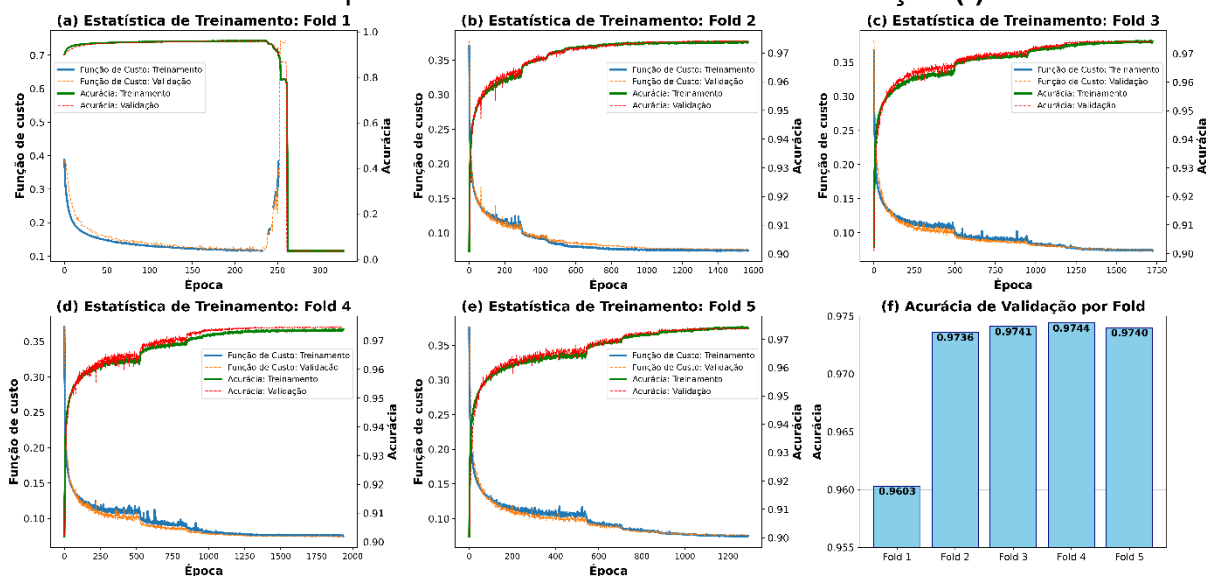
4.2 Resultados e Discussão

Os resultados referentes à avaliação do controlador mioelétrico proposto nesse trabalho foram divididos em duas seções. Primeiramente serão apresentados os resultados referentes ao desempenho do classificador de sinais mioelétricos, demonstrando sua acurácia e sua capacidade de discretização das classes, relacionados aos movimentos pretendidos pelo usuário, tanto com os dados de validação do DB3 e do DB2, como com os sinais do voluntário 21 do DB7. Posteriormente serão apresentados os resultados do treinamento do controlador de trajetórias utilizando o algoritmo TD3. Por fim, o comportamento do controlador operando com os dados de sEMG de m usuário do DB3 selecionado aleatoriamente e o voluntário 21 do DB7, em ambiente virtual e no protótipo em bancada serão analisados.

4.2.1 Classificador de Sinais de sEMG

Para a escolha do modelo de RNA utilizado no classificador, foi avaliado o desempenho durante o treinamento, em cada uma das *folds*. Da Figura 52a à Figura 52e são apresentadas as curvas de evolução da acurácia e da função de custo para cada uma delas, com os dados de validação interna. Observa-se que, com exceção da primeira *fold*, todas apresentaram um comportamento consistente com a convergência esperada, demonstrando a convergência do treinamento entre 1250 e 2000 épocas.

Figura 52 – Evolução da acurácia e da função de custo durante o treinamento do classificador nas *folds*: 1 (a), 2 (b), 3 (c), 4 (d) e 5 (e), bem como as acurácias obtidas para cada *fold* com os dados de validação (f)



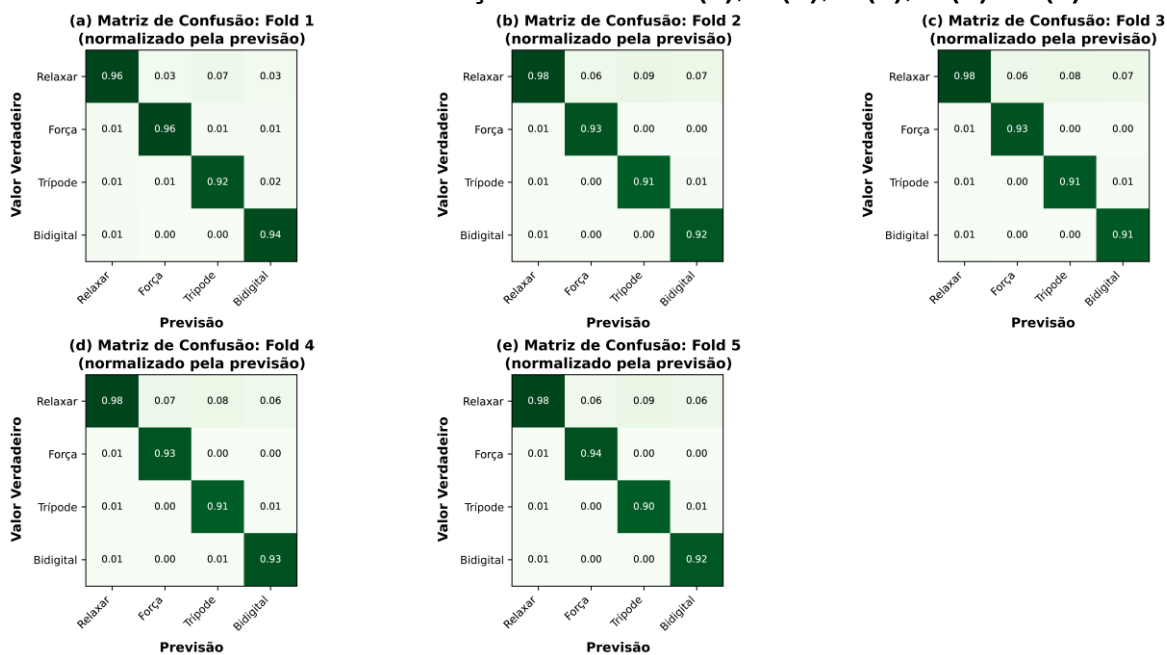
Fonte: Elaborado pelo autor.

Pode ser observado, pela evolução do treinamento, que a aplicação da taxa de aprendizado dinâmica foi crucial para o bom desempenho da rede. Observa-se que quando a função de custo se estabilizava, a mudança de taxa de aprendizado fez uma rápida redução no seu valor auxiliando na convergência do treinamento. Outro ponto observado foi que com a utilização do algoritmo de parada *early stop*, em nenhum momento ocorreu um aumento exacerbado nos valores da função de custo de validação, garantindo que o treinamento não apresentasse *overfitting*.

Com os dados de validação da rede, foi avaliada a acurácia obtida em cada uma das *folds*. Como indicado na Figura 52f as acurácias obtidas para cada uma delas foram de 96,03%; 97,36%; 97,41%; 97,44% e 97,40% respectivamente. Assim, a acurácia média entre as *folds* foi de $97,13 \pm 0,55\%$, sendo o maior valor obtido na *fold* 4. É importante notar que todas as *folds* obtiveram uma acurácia de classificação superior ao mínimo para não comprometer a experiência do usuário de, 85%, como sugerido por GUO, W. et al., (2017a), indicando que todas as RNAs poderiam ser utilizadas. Outro ponto que pode ser observado foi que o uso da rede *Inception* combinado com as características do tempo do sinal foram capazes de produzir resultado com acurácia mínima, entre as *folds*, de 96,03% e máxima de 97,44%. Os altos valores encontrados correspondem aos observados na literatura para esse tipo de rede, como apresentado por VÁSCONEZ et al., (2023), que obteve uma acurácia de 90,49%, para 6 movimentos classificados utilizando características do domínio da frequência.

Foi também avaliada a matriz de confusão de cada uma das *folds*, apresentadas na Figura 53. Em todas as redes o melhor desempenho obtido foi na separação da condição de relaxamento, obtendo entre 96% e 98% de assertividade. Apesar das assertividades serem menores, as demais classes também obtiveram um valor satisfatório de acurácia, variando entre 91% e 94%. Avaliando os demais movimentos, a maior confusão ocorreu entre os movimentos e a condição de relaxamento para todas as *folds*. Outro ponto importante foi observado na separação entre os movimentos de pinça trípode e a pinça bidigital, que foram confundidas entre si pelas RNAs de todas as *folds*. Avaliando a *fold* 4, onde foi observada a maior acurácia, sua matriz de confusão corrobora com o bom desempenho, apresentando a maior assertividade, de 98%, em identificar o relaxamento e a menor, de 91%, para identificar o movimento de pinça trípode. Devido a esses resultados a RNA treinada na *fold* 4 foi selecionada.

Figura 53 – Matrizes de confusão normalizadas em relação às previsões, obtidas com os dados de validação nas *fold*s 1 (a), 2 (b), 3 (c), 4 (d) e 5(e)



Fonte: Elaborada pelo autor.

Verificou-se que a classe de maior facilidade de classificação foi a de estado de relaxamento, cuja assertividade variou entre 96% e 98%. Alguns fatores podem explicar essa observação. O primeiro foi a maior quantidade de dados de relaxamento presentes no banco de dados, pois antes da execução de cada movimento o voluntário ficava em estado de relaxamento para depois efetuá-lo, contribuindo assim para a maior quantidade de dados. Outro fator foi que durante o relaxamento existe pouquíssima atividade muscular, fazendo com que algumas características extraídas tendessem a valores próximos a zero, facilitando essa distinção.

O segundo movimento com maior grau de assertividade foi o de prensão de força, com assertividade mínima de 93% e máxima de 96%. Essa classificação foi observada pelo fato de que, dos três movimentos escolhidos, esse era o movimento com o padrão mais distinto, no qual se realizou a flexão completa de todos os dedos, como pode ser observado na Figura 48. Assim, durante essa atividade os valores das características foram os máximos, facilitando a distinção pela rede.

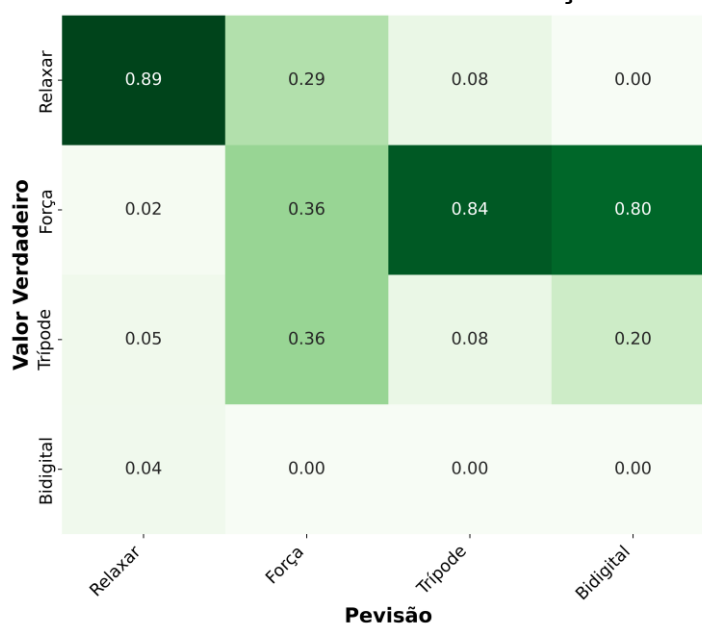
Observando os movimentos de pinça trípode e bidigital, ambos apresentaram os menores valores de assertividade (90% e 91% respectivamente). Duas observações relevantes sobre essas classes de movimento, a primeira foi que elas apresentaram um pequeno grau de confusão entre si, o que pode ser explicado pela similaridade dos dois movimentos, o que dificultou a distinção entre eles pelo classificador.

Outro ponto foi a confusão com o estado de relaxamento, que pode ter ocorrido devido a atividade muscular não ser tão intensa quanto as observadas na preensão de força.

O último ponto que foi observado nas matrizes de confusão da Figura 53 foi que o maior grau de confusão de todas as classes se deu com o estado de relaxamento. Uma explicação para essa observação foi que a transição entre o estado de relaxamento para os demais estados de movimento é o maior gargalo nos controladores mioelétricos baseados apenas em classificadores do sinal. Como apontado por JIANG et al., (2025) o grande gargalo desse tipo de sistema são os movimentos transitórios, pois as mudanças suaves nos sinais podem dificultar a correta classificação, levando a movimentos não naturais. Assim essa mudança suave de sinal pode ser um fator que explica o grau de confusão observável.

Para o uso efetivo da rede, com dados externos, fez-se necessário o processo de calibração do classificador, ajustando-o para melhor classificar os dados do voluntário 21 do DB7. Antes de realizar a calibração, foi observado que o desempenho do classificador caiu drasticamente, apresentando uma acurácia de 88,49%. Avaliando a matriz de confusão, apresentada na Figura 54, foi possível ver que a única classe identificada pelo sistema era a condição de relaxamento, enquanto os demais movimentos não foram devidamente classificados, demonstrando a necessidade de calibração do sistema, utilizando o processo de transferência de aprendizado.

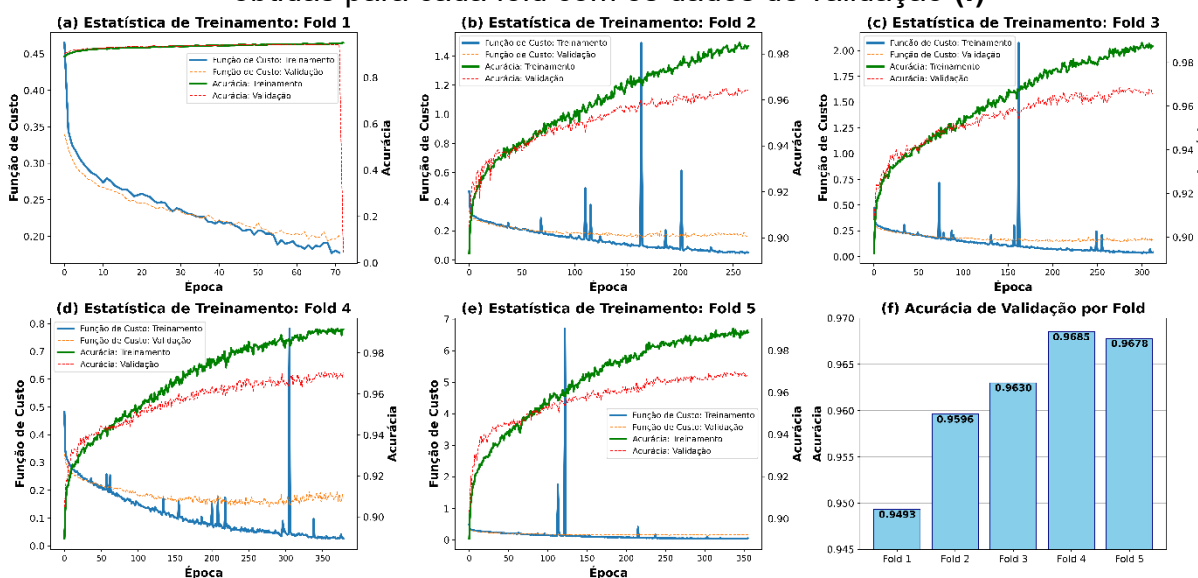
Figura 54 – Matriz de confusão normalizada em relação às previsões, obtida com os dados do voluntário 21 do DB7 antes da calibração do classificador



Fonte: Elaborado pelo autor.

A calibração do classificador, utilizando a RNA da *fold* 4, foi realizada com os dados de sEMG do voluntário 21 do DB7, coletados em 70% dos episódios. Assim, a rede foi retreinada, apresentando a evolução do treinamento mostrada na Figura 55a a Figura 55e. Com exceção da primeira *fold*, todas obtiveram um comportamento semelhante com o observado na Figura 52, entretanto ocorreram mais instabilidades durante o ajuste. Apesar disso, a rede foi capaz de alcançar acurácias, com os dados de validação, de 94,93%; 95,96%; 96,30%; 96,85% e 96,78% para cada uma das *fold*s respectivamente (Figura 55f), e uma acurácia média de $96,16 \pm 0,70\%$.

Figura 55 – Evolução da acurácia e da função de custo durante a calibração do classificador nas *fold*s: 1 (a), 2 (b), 3 (c), 4 (d) e 5 (e), bem como as acurácias obtidas para cada *fold* com os dados de validação (f)

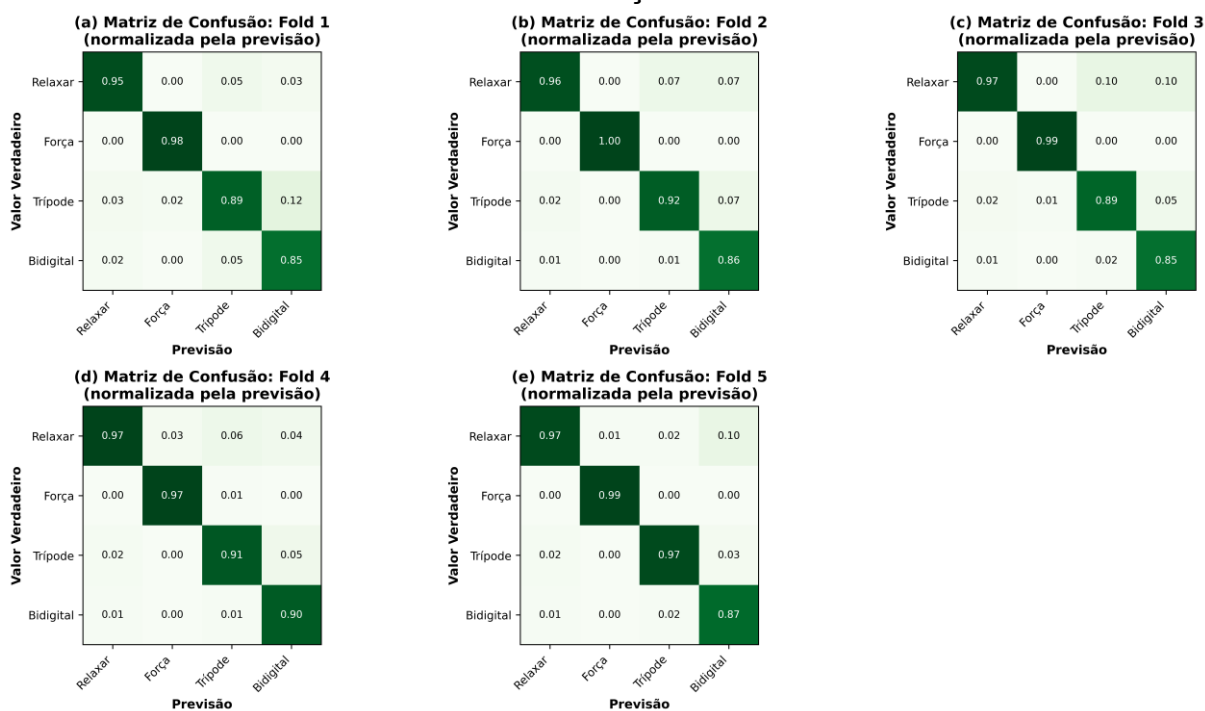


Fonte: Elabora pelo autor.

O desempenho da calibração também foi avaliado por meio da matriz de confusão de cada *fold* utilizando os dados dos episódios que ficaram fora da calibração. Com as matrizes de confusão apresentadas na Figura 56, foi observado que em todas as *fold*s o desempenho foi inferior ao observado no treinamento (Figura 53). Apesar de inferior, o desempenho foi satisfatório, com a maior assertividade de predição foi obtida na condição de relaxamento de 97% e a menor na predição da pinça bidigital, com 85%, considerando todas as *fold*s. Analisando a matriz de confusão da *fold* 4, que obteve a maior acurácia, observa-se que o classificador apresenta a melhor divisão de classes, obtendo como menor assertividade de predição a classe bidigital, com 90%. Um aspecto relevante foi que as maiores assertividades, de 97%, ocorreram para predição do estado de relaxamento e na predição do movimento de prensão de

força. A maior dificuldade, entretanto, ocorreu na predição das pinças trípode e bidigital, que além de apresentarem as menores assertividades apresentaram confusão entre si. Assim, após a realização da calibração, a RNA obtida no *fold* 4 foi escolhida para ser utilizada no classificador com os dados de sEMG do voluntário 21 do DB7.

Figura 56 – Matrizes de confusão normalizadas em relação às previsões, obtidas com os dados de validação nas *folders* 1 (a), 2 (b), 3 (c), 4 (d) e 5(e), após a calibração



Fonte: Elaborado pelo autor.

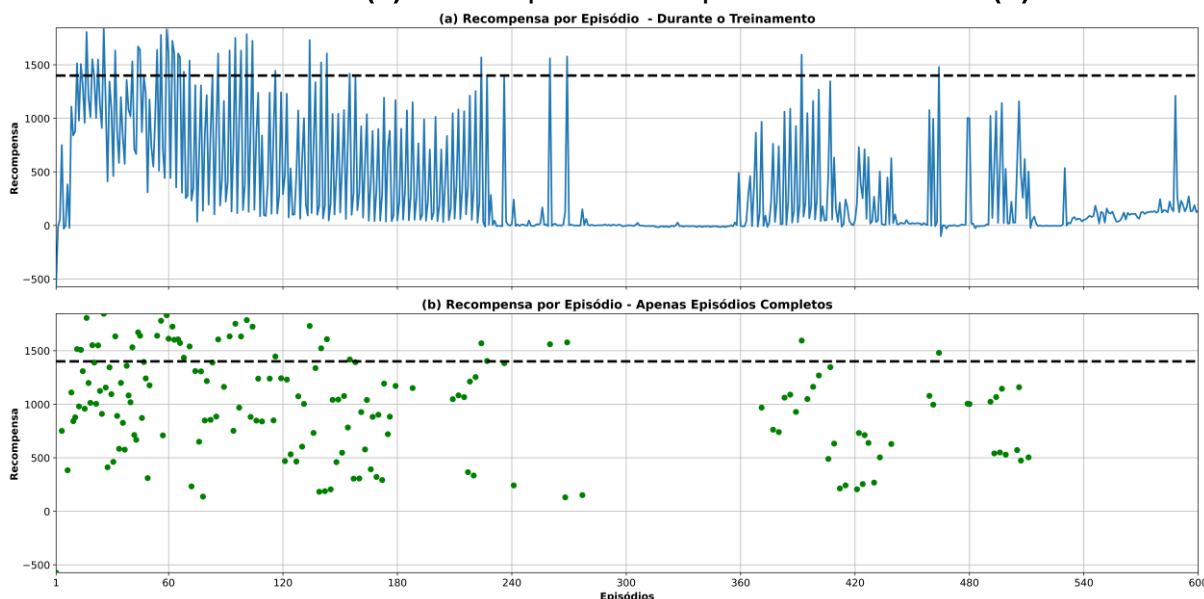
Os resultados apresentados após a calibração do classificador demonstram a importância do processo para o uso efetivo do classificador, aumentando significativamente a acurácia do classificador, para o voluntário 21 do DB7, de 88,49% para 96,45%. A melhora de desempenho era esperada uma vez que o processo de calibração proporciona um ajuste fino dos parâmetros da rede apenas para um usuário específico e as condições de utilização da prótese, conforme apresentado em outros trabalhos da literatura (MOHAMMADREZA; FONSECA; JIANG, 2025; YANG; GU; *et al.*, 2017). Outro ponto importante foi a confirmação de como o uso do processo de transferência de aprendizado se mostra adequado para a calibração de classificadores de sEMG, melhorando significativamente seu desempenho, como proposto no trabalho de FAN *et al.*, (2023).

Observando as matrizes de confusão de cada *fold* (Figura 56) observa-se que a discretização de cada uma das classes foi melhorada, sendo mais uma vez a condição de relaxamento com maior assertividade observada. Observou-se ainda que houve uma melhora na classificação do movimento de preensão de força, tendo assertividades parecidas com a condição de relaxamento. Como o movimento de preensão de força se apresenta como uma das atividades com a maior atividade muscular, é menos complexo para o usuário amputado executá-lo, pois uma simples contração dos músculos remanescentes no coto faria o sistema classificá-lo como preensão de força. A confusão entre as pinças trípede e bidigital foi aumentada, porém como esses movimentos são mais finos o processo de amputação pode ter levado a dificuldade do usuário em diferenciar as contrações dos grupos musculares utilizados em cada ação. Assim, com o bom desempenho do classificador após a calibração, fica evidente que a estratégia de uso do sistema para condições reais depende do uso do aprendizado *online*, em que, após cada sessão o classificador deveria ser retreinado para que seu desempenho seja aprimorado a cada sessão, devido ao aumento da diversificação dos dados em variadas condições de uso do sistema, o que é essencial para o bom desempenho em tempo real (MOHAMMADREZA; FONSECA; JIANG, 2025; YANG; GU; *et al.*, 2017). É importante salientar que outras técnicas podem ser utilizadas para melhorar o desempenho do classificador, como por exemplo o voto majoritário, que é amplamente utilizado na literatura para esse fim (AL-TIMEMY *et al.*, 2016; AL-TIMEMY; KHUSHABA; ESCUDERO, 2016; LIU *et al.*, 2016).

4.2.2 Controlador de Trajetória

Para avaliar o desempenho do controlador de trajetórias, inicialmente foi observada a evolução da recompensa obtida pelo Ator durante o treinamento, apresentada na Figura 57. Como pode ser observado, o treinamento não convergiu satisfatoriamente, chegando ao final de 600 episódios (200 épocas) sem completar nenhum episódio. No início do treinamento observou-se que muitos episódios foram concluídos (Figura 57b), alguns superaram a recompensa-alvo (1400 pontos, representada pela linha horizontal tracejada no gráfico), entretanto sem nenhuma época superando a recompensa-alvo.

Figura 57 – Evolução da recompensa obtida pelo Ator a cada episódio de treinamento **(a)** e recompensa nos episódios concluídos **(b)**



Fonte: Elaborado pelo autor.

A evolução do treinamento indica que 200 épocas continuam um número reduzido de episódios para convergência do treinamento, mesmo após as mudanças feitas entre o treinamento do controlador do teste piloto e do controlador de trajetórias. Como observado por VÁSZONEZ et al., (2023) o treinamento utilizando RL é muito sensível aos parâmetros de treinamento, demandando muitos ajustes para atingir um resultado satisfatório. Com essa característica intrínseca, atrelada ao uso do modelo FEA, ficou impossibilitado o treinamento com mais episódios, visto que durante o treinamento foram gastos em média 45 minutos para a execução de um episódio de treinamento o que levou mais de 15 dias para completar os 600 episódios, inviabilizando o treinamento com vários parâmetros, ou a execução de mais épocas de treinamento, pois com o alto custo computacional, muito tempo foi gasto sem saber se de fato o treinamento tendia a convergência.

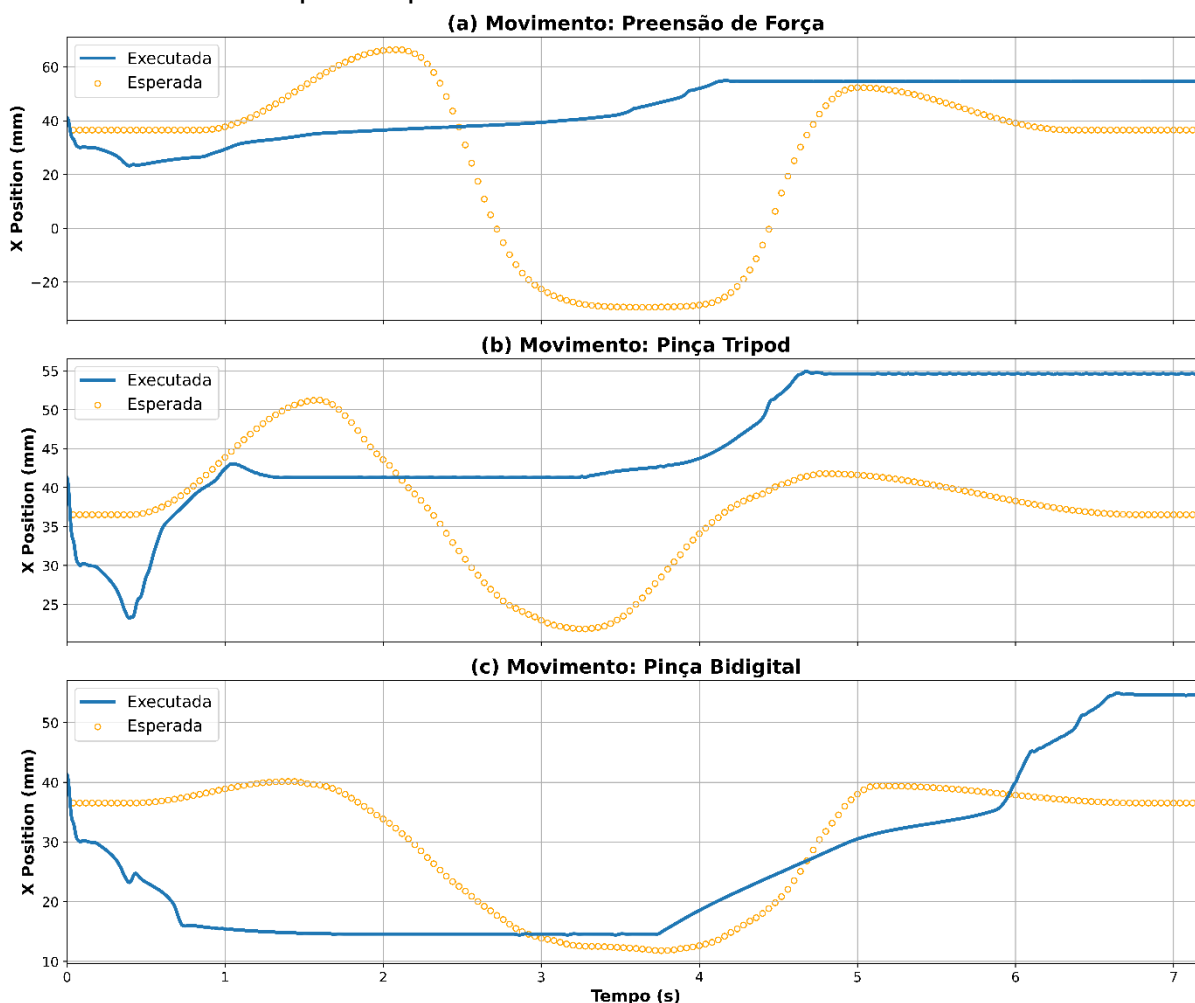
Comparando com o número de episódios gastos durante o teste piloto (79496 episódios) foi evidente que mesmo alterando o algoritmo para o TD3, e utilizando arquiteturas mais complexas, as células LSTM e as camadas convolucionais, eram necessários mais episódios do que os 600 utilizados, dada as limitações computacionais impostas com o uso do modelo FEA no ambiente de treinamento. Evitando aumentar o recurso computacional e visando possibilitar o uso de técnicas de RL no desenvolvimento de controladores de trajetórias real, faz-se necessário investigar outros meios para simular o mecanismo subatacado de dedos. Uma solução poderia ser o uso de

outras plataformas, como o MuJoCo Physics Simulator, utilizado nos trabalhos de MONTGOMERY; ALLEN; ELBASIOUNY, (2021) e de ZHANG et al., (2024), que possui um modelo simplificado de atuadores utilizando tendões artificiais. Uma outra solução seria o uso do modelo FEA como fonte de dados para criação de RNAs ou funções de regressão, que simplificam e reduzem o custo computacional da simulação do dispositivo. Apesar viabilizarem a execução do treinamento RL, essas soluções podem apresentar uma discrepância elevada entre o comportamento dinâmico real do dispositivo e o comportamento do ambiente de treinamento, necessitando de avaliações dos impactos dessas simplificações no uso real do controlador de trajetórias.

Apesar de o treinamento não ter convergido para recompensa alvo, foi avaliado o comportamento do controlador utilizando os parâmetros de RNA que completou todos os movimentos de uma época, obtendo a maior média de recompensa. Três avaliações foram realizadas, onde na primeira foi observada a resposta do controlador com as entradas de trajetória binárias. Posteriormente foi avaliada a resposta com as entradas de trajetórias provenientes do classificador de sEMG com os dados de um voluntário do DB3. Por fim, foi avaliada a resposta utilizando como seletor de trajetórias o classificador calibrado do voluntário 21 do DB7.

Na Figura 58 é apresentada a evolução da coordenada x ao longo da execução dos três movimentos de pinça utilizando as entradas de treinamento sem ruído. A resposta apresentada ficou distante da esperada, apresentando um erro médio absoluto de $30,85 \pm 25,85$ mm para preensão de força (Figura 58a), de $14,63 \pm 6,16$ mm para a pinça trípode (Figura 58b) e de $12,84 \pm 6,90$ mm para pinça bidigital (Figura 58c). Apesar de distante do comportamento esperado, algumas observações podem ser feitas sobre o comportamento do sistema. Comparando os três comportamentos, observa-se que existe uma dificuldade do sistema em perceber o movimento de extensão do dedo, aumento da coordenada x , para a posterior flexão. Essa dificuldade foi o que impediu o sistema em executar a trajetória. Outro ponto observável é que em todos os movimentos o sistema finalizou aproximadamente no mesmo ponto 54,5 mm.

Figura 58 – Comparação da variação da coordenada x ao longo do tempo nos movimentos de prensão de força **(a)**, pinça trípode **(b)** e pinça bidigital **(c)**, executada e esperada para as entradas binárias utilizadas no treinamento

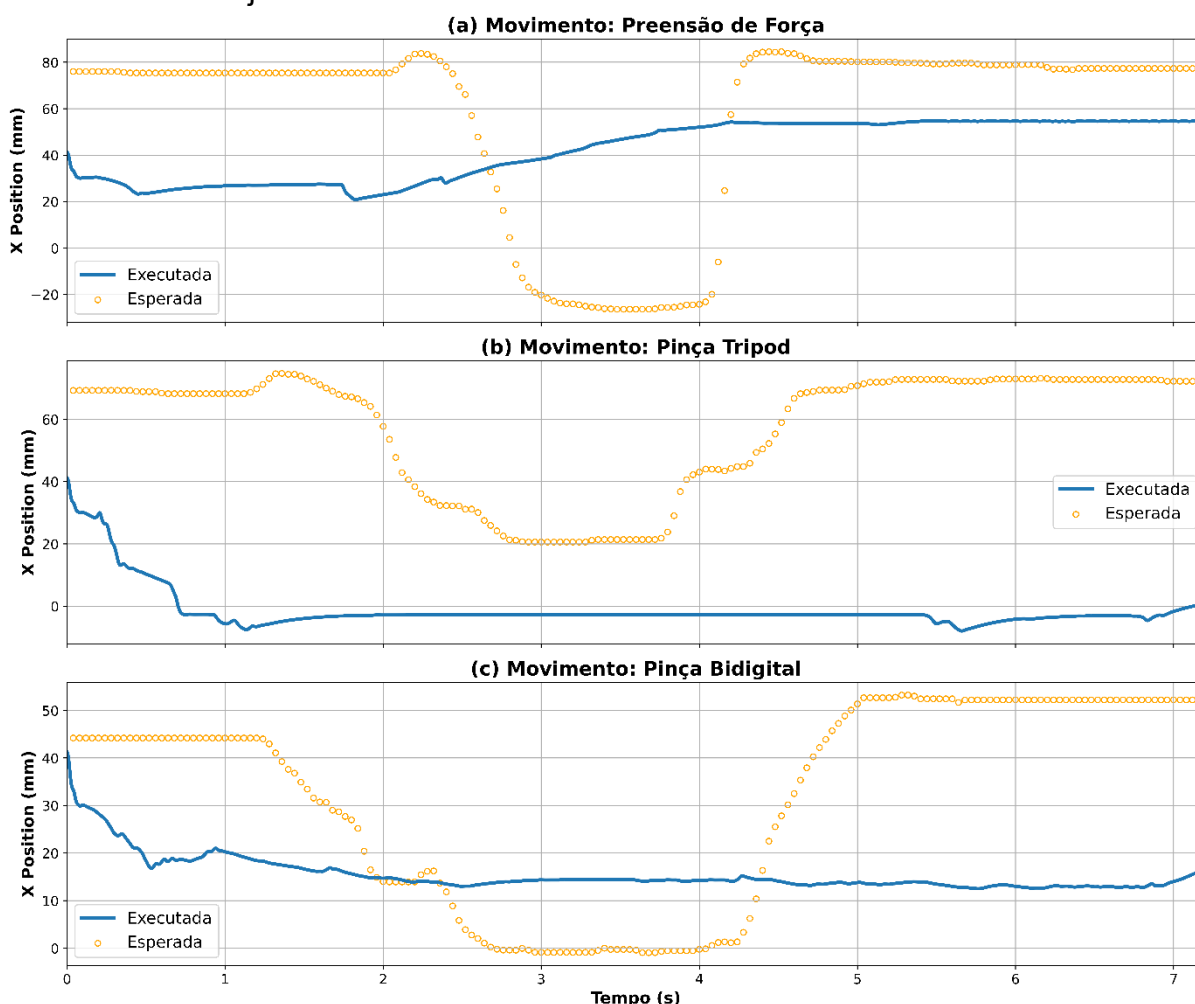


Fonte: Elaborado pelo autor

Observando a diferenciação entre movimentos, observou-se que para cada trajetória o sistema apresentou um padrão de movimento diferente. Na prensão de força (Figura 58a) o dedo foi flexionado e, posteriormente, estendido de forma contínua até a posição final. Para a pinça trípode (Figura 58b) foi observado que no início do movimento o dedo foi flexionado e então quando a seleção de trajetória mudou, da condição de relaxamento para pinça trípode (aumento inicial da coordenada x) o sistema começou a acompanhar a abertura do movimento, entretanto se estabilizou na posição aberta sem efetuar o movimento de fechamento esperado. Para o movimento de pinça bidigital (Figura 58c) observa-se que o sistema executou o fechamento do dedo até uma posição próxima da posição final do movimento da pinça bidigital (14,45 mm), e no retorno para condição de relaxamento foi realizada apenas a abertura do dedo.

Na Figura 59 demonstrou a resposta do controlador mioelétrico proposto, utilizando como dados de entrada de sEMG de um voluntário aleatoriamente selecionado do DB3. Comparando as respostas do controlador com o observado com as entradas binárias (Figura 58) pode-se observar um padrão distinto de movimento. Os erros médios absolutos observados foram de $49,86 \pm 17,14$ mm para a preensão de força (Figura 59a), de $60,88 \pm 21,57$ mm para a pinça trípode (Figura 59b) e de $24,06 \pm 12,88$ mm para a pinça bidigital (Figura 59c).

Figura 59 – Comparação da variação da coordenada x ao longo do tempo nos movimentos de preensão de força (a), pinça trípode (b) e pinça bidigital (c), executada e esperada, utilizando o classificador associado ao controlador de trajetória com dados de sEMG de um voluntário do DB3

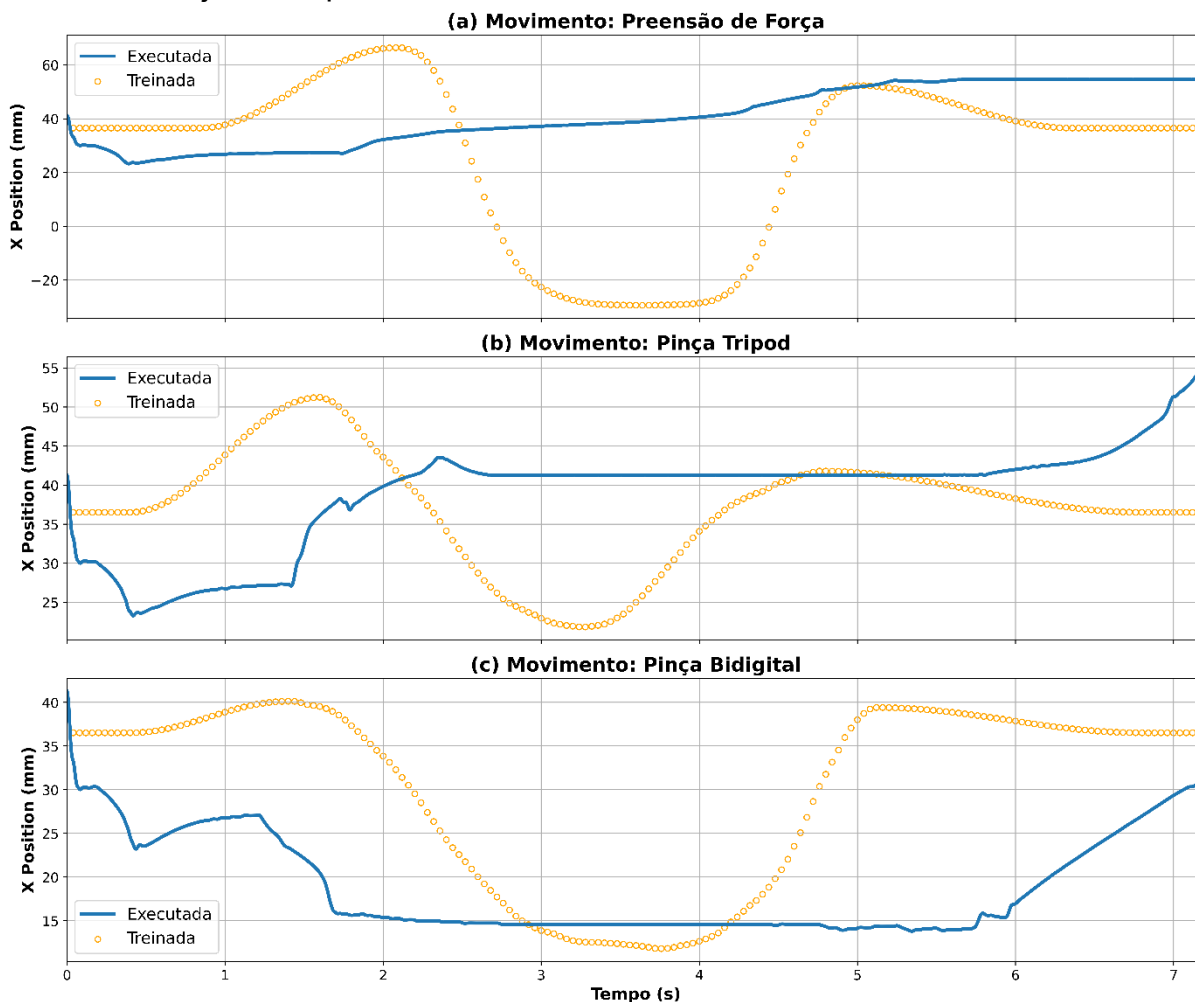


Fonte: Elaborado pelo autor.

Quanto ao padrão dos movimentos, observou-se que para cada episódio o ponto final foi distinto. Para a preensão de força (Figura 59a), inicialmente foi realizado o movimento de flexão, porém apresentando maiores instabilidades, seguido por uma extensão até estabilizar na posição final de 54,63 mm. Dos três movimentos o de preensão de força foi o que apresentou a maior similaridade com o observado com as entradas binárias (Figura 58a). No movimento de pinça trípode (Figura 59b) o padrão de movimento foi diferente do observado com as entradas binárias (Figura 58b). Com o classificador observou-se que o movimento utilizado pelo sistema foi apenas de flexão, sem realizar a extensão e estabilizando na posição final de 0,7104 mm. Para o movimento de pinça bidigital (Figura 59c), o movimento executado também foi apenas o de flexão e estabilização na posição final de 16,50 mm. Para as entradas referentes ao episódio de pinça trípode e de pinça bidigital, o sistema não foi capaz de fazer a abertura do dedo como observado com as entradas binárias (Figura 58b e Figura 58c).

O último teste virtual realizado, com os dados do voluntário 21 do DB7, teve a coordenada x dos movimentos executados como apresentado na Figura 60. Observando a resposta obtida, vê-se que as trajetórias obtidas também ficaram longe das esperadas, entretanto as respostas do controlador foram diferentes das observadas com as entradas binárias (Figura 60) e do voluntário do DB3 (Figura 59). Nesse teste o controlador levou a ponta do dedo para mesma posição final (aproximadamente 55 mm) nos movimentos de preensão de força (Figura 60a) e de pinça trípode (Figura 60b), enquanto a posição final para o movimento de pinça bidigital (Figura 60c) foi interrompido em uma posição inferior. Quanto ao padrão de movimento, foi observado que para preensão de força a resposta obtida foi parecida, onde inicialmente o sistema flexionou o dedo e posteriormente o estendeu até a posição final, como observado nos demais testes (Figura 58a e Figura 59a). Para o movimento de pinça trípode o comportamento foi mais parecido com o observado nos testes com os dados de treinamento (Figura 58b), onde o dedo fez uma flexão inicial e depois o estendeu nos primeiros momentos do movimento até atingir uma posição constante de 41,29 mm para então finalizar o movimento de abertura até a posição final. Para a pinça bidigital, o movimento foi iniciado com uma flexão até a posição mínima de aproximadamente 13 mm, para então, no final, abrir o dedo com um comportamento aproximadamente linear. O comportamento apresentado em todos os casos foi distinto dos testes anteriores, apresentando oscilações.

Figura 60 – Comparação da variação da coordenada x ao longo do tempo nos movimentos de prensão de força **(a)**, pinça trípode **(b)** e pinça bidigital **(c)**, executada e esperada, utilizando o classificador associado ao controlador de trajetória a partir dos dados de sEMG do voluntário 21 do DB7



Fonte: Elaborado pelo autor.

Comparando as respostas do controlador com várias entradas de movimento observa-se que a maior dificuldade do controlador foi a de identificar que o movimento possui 2 etapas, uma abertura inicial do dedo seguida de seu fechamento. Entretanto, a rede apenas levou o dedo diretamente a uma posição e acordo com a entrada de seleção de trajetórias determinada. Com as entradas do voluntário do DB3 (Figura 59) observa-se que o controlador tendeu a levar o dedo para uma posição e o manteve durante todo o movimento, não identificando o retorno necessário à posição de relaxamento. Para o voluntário 21 do DB7 (Figura 60) o comportamento foi semelhante ao observado com os dados binários de treinamento (Figura 58), em que, para cada etapa do movimento, o sistema tendeu a levar o dedo a uma posição.

Apesar dos resultados insatisfatórios, observa-se que o sistema tomou ações diferentes para cada movimento, o que demonstra que ele teve condições de identificar as mudanças de seleção de trajetória (Figura 58). Além da diferenciação das trajetórias, o controlador foi capaz de completar o episódio para entradas diferentes provenientes do classificador de movimentos, como observado nas respostas com as entradas do voluntário do DB3 (Figura 59) e do voluntário 21 do DB7 (Figura 60). Essa observação indica que, se o treinamento tivesse convergido para a recompensa alvo, provavelmente a diferenciação de trajetórias entre usuários seria observada.

4.2.3 Limitações

É importante salientar que o trabalho apresentou algumas limitações que impactaram os resultados obtidos. O primeiro deles foi que para o classificador de movimentos foram utilizadas apenas análises em ambiente simulado e o desempenho em tempo real não foi avaliado. Durante o uso em indivíduos amputados, em tempo real, alguns problemas iriam interferir diretamente no desempenho do sistema, pois, como apresentado por MOHAMMADREZA; FONSECA; JIANG, (2025), vários fatores podem influenciar no desempenho do controlador, como peso da prótese, suor, deslocamento dos eletrodos, fatores esses que impactam o sinal e conseqüentemente o desempenho do controlador. Assim, para avaliar o desempenho em aplicações reais do sistema, faz-se necessária a realização de testes clínicos com usuários amputados. Entretanto, com a metodologia de calibração apresentada, espera-se que após as primeiras sessões de treinamento, o classificador consiga ser ajustado ao usuário.

Outras limitações do processo têm relação ao modelo dinâmico utilizado no ambiente de treinamento. Todos os componentes, com exceção do tendão artificial, foram considerados corpos rígidos, ou seja, as deformações dos componentes foram desprezadas. Como as forças envolvidas são baixas em comparação com os parâmetros de resistência do material, essa consideração pouco deve ter impactado nos resultados. Outra simplificação foi a não utilização de dados reais de rigidez e amortecimento das juntas. Essa consideração pode impactar no resultado em comparação com o movimento real do dedo. Entretanto, dadas as condições construtivas da prótese, os parâmetros dinâmicos não eram constantes. Foi feita uma tentativa de ajustar o aperto dos parafusos e conexões dos dedos, contudo, após breve utilização do dispositivo, o sistema já apresentava variações no funcionamento. Assim, a

determinação desses parâmetros era complexa e fugia do escopo do trabalho. Apesar de não considerar os parâmetros reais, caso o modelo tivesse obtido um desempenho satisfatório, era esperado que o controlador se adaptasse às novas condições ambientais e compensasse essa discrepância como observado no teste piloto (Figura 30).

A última limitação do estudo foi o uso de apenas um algoritmo de treinamento para a construção do controlador de trajetórias. A escolha do TD3 para treinar a rede do controlador se deu pelo fato da ampla aplicação desses algoritmos em muitos controladores robóticos apresentados na literatura, e da sua tendência de convergência mais rápida, quando comparado com o DDPG. O uso do SAC ou do PPO poderia auxiliar na obtenção de uma melhor resposta do controlador. Um estudo comparativo dos algoritmos poderia ser realizado a fim de avaliar o desempenho de cada um deles no treinamento do controlador de trajetória. Devido ao tempo necessário para treinar a rede, essa comparação não foi realizada, ficando como sugestão para futuros trabalhos que podem ser realizados. Além disso, dadas as limitações de tempo de execução de treinamento, apenas um dedo foi utilizado, não sendo possível em tempo hábil executar o treinamento com todos os dedos da prótese, pois um aumento muito grande no tempo de cada episódio foi observado.

CAPÍTULO 5: CONCLUSÃO

Na presente tese foi avaliado, computacionalmente, o uso de técnicas de RL para construção de um novo controlador inteligente para dedos de próteses de membro superior combinando um classificador de sinais de sEMG com um controlador de múltiplas trajetórias. Inicialmente realizou-se um teste piloto, treinando uma RNA para atuar como controlador de torque em cada uma das articulações de um modelo biomecânico de dedo simplificado. O objetivo desse teste foi avaliar como o algoritmo de RL DDPG poderia ser utilizado no desenvolvimento de um controlador que combinasse a classificação de movimentos de sEMG com os estados cinemáticos, executando múltiplas trajetórias e se adaptando a variações ambientais. Foi observado que o controlador obteve um erro médio de $3,98 \pm 2,90$ mm quando a entrada de trajetória senoidal binária era utilizada e de $3,22 \pm 1,42$ mm com relação as trajetórias-alvo de treinamento. Observando os sinais de atuação, observa-se que a rede criou duas estratégias de controle distintas, um padrão oscilatório para altos valores de excitação do neurônio de entrada trajetória linear, e uma atuação suave de pequenas correções quando o neurônio da trajetória senoidal era excitado, indicando a diferenciação das trajetórias pelo controlador. Além disso, com as entradas proporcionais, as trajetórias executadas foram modificadas, apresentando outros padrões de atuação. Avaliando a adaptabilidade do sistema ao ambiente, foi observado que o modelo percebeu as mudanças nos parâmetros inerciais do dedo e adaptou suas estratégias de controle para cada tamanho de dedo. Dessa forma o teste piloto mostrou que a arquitetura de controlador proposta poderia criar controladores de trajetórias para dedos de próteses de membro superior utilizando dados de entrada de trajetórias provenientes de um classificador de movimentos, combinados com o estado atual da prótese, garantindo que o sistema se adaptasse às mudanças ambientais e nas classificações.

Para verificar se os resultados do teste piloto seriam reproduzíveis em próteses reais, foi aplicada a arquitetura de controlador proposto em um controlador mioelétrico para uma prótese real, a HANDi Hand. Foram então utilizados dados de sEMG do banco de dados público NINAPRO para construir o seletor de trajetórias e o controlador de trajetórias foi treinado para executar trajetórias baseados nos movimentos fisiológicos no dedo indicador da prótese. O classificador de trajetórias utilizou-se de uma CNN do tipo *Inception* obtendo uma acurácia superior a 97% com os dados de validação. Apesar do bom desempenho, fez-se necessário a calibração do sistema para o

uso com um voluntário externo aos dados do treinamento fazendo com que a acurácia obtida fosse de 96%. Utilizando essa arquitetura de RNA ficou demonstrado sua alta capacidade em classificar movimentos, obtendo valores elevados de acurácia essenciais no uso do sistema de classificação pretendido. Já na construção com controlador de trajetória, não foi possível obter a recompensa alvo estipulada. Tal dificuldade de convergência se deu pelo fato da necessidade do uso de um modelo em FEA para simular a dinâmica de atuação subatuada do dedo da prótese, fazendo assim com que o tempo de treinamento para cada episódio fosse de 45 minutos. Devido a essa alta demanda computacional apenas 600 episódios de treinamento foram utilizados, o que não foi suficiente para obter a convergência de treinamento do controlador. Apesar de não se obter as respostas esperadas, pode-se observar que alguns fatores favorecem o uso da técnica proposta no desenvolvimento de controladores de trajetória mioelétricos. Foi observado que, utilizando o controlador com o melhor desempenho em uma época de treinamento, para cada entrada, sejam as de treinamento ou as provenientes do classificador de movimentos, o modelo proposto foi capaz de determinar padrões de movimento diferentes, o que era esperado com o desenvolvimento do sistema proposto.

Assim, foi demonstrado que o uso de técnicas de RL pode ser uma boa alternativa para construção de controladores de trajetórias de prótese, utilizando como referência a seleção de trajetórias proveniente de um classificador mioelétrico. O teste piloto demonstra que quando o controlador atingiu a convergência de treinamento o sistema foi capaz de modificar as trajetórias e adaptar os sinais de controle dadas as modificações nas condições da prótese. Apesar do bom desempenho, o trabalho demonstrou que a aplicabilidade da metodologia em uma prótese real ainda demanda de um alto custo computacional. O uso de mecanismos subatuados dificulta a simulação da dinâmica do movimento da prótese, fazendo assim que técnicas computacionalmente pesadas fossem utilizadas. Nesse trabalho, optou-se por utilizar o FEA para simulações dinâmicas do ambiente de treinamento. Tal ferramenta não se mostrou adequada devido ao tempo necessário para a simulação, demonstrando assim que o uso do RL em condições reais, apesar de promissor, ainda necessita de uma otimização dos modelos físicos utilizados no ambiente de treinamento, onde devem ser utilizadas técnicas computacionalmente leves, sem a perda da fidedignidade do comportamento dinâmico da prótese.

CAPÍTULO 6: TRABALHOS FUTUROS

Com o trabalho proposto, algumas lacunas ficaram em aberto para serem preenchidas em trabalhos futuros. A seguir são apresentadas as sugestões:

- 1.** Avaliação de algoritmos de aprendizado por reforço para o controle de múltiplas trajetórias.
- 2.** Avaliação comparativa do desempenho de modelos simplificados de mecanismos subatuados de próteses robóticas no treinamento de controladores utilizando aprendizado por reforço.
- 3.** Avaliação do impacto do poder computacional no desempenho do treinamento de aprendizado por reforço combinado com modelos dinâmicos de próteses usando elementos finitos.
- 4.** Investigação do impacto dos fatores da função de recompensa no desempenho do treinamento de um controlador de trajetórias usando aprendizado por reforço.
- 5.** Criação de modelos dinâmicos simplificados para ambientes de treinamento de uma prótese completa subatuada de membro superior.
- 6.** Comparação entre o uso de características no domínio da frequência e do tempo na classificação dos sinais de eletromiografia.
- 7.** Desenvolvimento de um modelo de controlador de trajetória para as seis poses padrão de uma prótese completa de membro superior.
- 8.** Avaliação do desempenho do classificador desenvolvido em usuários amputados por meio de testes clínicos.
- 9.** Avaliação do desempenho do controlador mioelétrico de trajetórias proposto em indivíduos amputados.
- 10.** Investigação do uso de outros bio-sinais na classificação de movimentos desejados combinados ao classificador proposto

REFERÊNCIAS BIBLIOGRÁFICAS

AHMADIZADEH, C. *et al.* Toward Intuitive Prosthetic Control: Solving Common Issues Using Force Myography, Surface Electromyography, and Pattern Recognition in a Pilot Case Study. *IEEE Robotics & Automation Magazine*, v. 24, n. 4, p. 102–111, 1 dez. 2017. Disponível em: <<https://ieeexplore.ieee.org/document/8098553/>>. Acesso em: 6 out. 2022.

ALMAHAMID, F.; GROLINGER, K. Autonomous Unmanned Aerial Vehicle navigation using Reinforcement Learning: A systematic review. *Engineering Applications of Artificial Intelligence*, v. 115, p. 105321, 1 out. 2022. Disponível em: <<https://www.sciencedirect.com/science/article/abs/pii/S095219762200358X>>. Acesso em: 21 jul. 2025.

AL-TIMEMY, A. H. *et al.* Improving the Performance Against Force Variation of EMG Controlled Multifunctional Upper-Limb Prostheses for Transradial Amputees. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, v. 24, n. 6, p. 650–661, 1 jun. 2016. Disponível em: <<https://ieeexplore.ieee.org/document/7131549/>>. Acesso em: 7 nov. 2022.

AL-TIMEMY, A. H.; KHUSHABA, R. N.; ESCUDERO, J. A comparison of post-processing techniques on the performance of EMG based pattern recognition system for the transradial amputees. 15 out. 2016, [S.l.]: IEEE, 15 out. 2016. p. 46–49. Disponível em: <<http://ieeexplore.ieee.org/document/7745405/>>. Acesso em: 7 nov. 2022.

ALY, H. I.; YOUSSEF, S.; FATHY, C. Hybrid Brain Computer Interface for Movement Control of Upper Limb Prostheses. 24 jul. 2018, [S.l.]: IEEE, 24 jul. 2018. p. 1–6. Disponível em: <<https://ieeexplore.ieee.org/document/8471729/>>. Acesso em: 3 out. 2022.

AMSUSS, S. *et al.* Long term stability of surface EMG pattern classification for prosthetic control. jul. 2013, [S.l.]: IEEE, jul. 2013. p. 3622–3625. Disponível em: <<http://ieeexplore.ieee.org/document/6610327/>>. Acesso em: 10 nov. 2022.

AMSUSS, S. *et al.* Self-Correcting Pattern Recognition System of Surface EMG Signals for Upper Limb Prosthesis Control. *IEEE Transactions on Biomedical Engineering*, v. 61, n. 4, p. 1167–1176, abr. 2014. Disponível em: <<http://ieeexplore.ieee.org/document/6692872/>>. Acesso em: 10 nov. 2022.

ANSYS INC. *Mechanical User's Guide*. . Canonsburg, PA: [s.n.], 2025

ARANCETA-GARZA, A.; LAKANY, H.; CONWAY, B. A. An Investigation into Thumb Rotation Using High Density Surface Electromyography of Extrinsic Hand Muscles. out. 2013, [S.l.]: IEEE, out. 2013. p. 3751–3755. Disponível em: <<http://ieeexplore.ieee.org/document/6722392/>>. Acesso em: 10 nov. 2022.

ATZORI, M. *et al.* Building the Ninapro database: A resource for the biorobotics community. jun. 2012, [S.I.]: IEEE, jun. 2012. p. 1258–1265. Disponível em: <<http://ieeexplore.ieee.org/document/6290287/>>. Acesso em: 10 nov. 2022.

ATZORI, M. *et al.* Electromyography data for non-invasive naturally-controlled robotic hand prostheses. *Scientific Data*, v. 1, n. 1, p. 140053, 9 ago. 2014. Disponível em: <<http://www.nature.com/articles/sdata201453>>.

AZAB, A. M.; ARVANCH, M.; MIHAYLOVA, L. S. Estimation of joint angle based on surface electromyogram signals recorded at different load levels. 13 jul. 2017, [S.I.]: IEEE, 13 jul. 2017. p. 2538–2541. Disponível em: <<https://ieeexplore.ieee.org/document/8037374/>>. Acesso em: 12 out. 2022.

BAKSHI, K.; MANJUNATHA, M.; KUMAR, C. S. Estimation of continuous and constraint-free 3 DoF wrist movements from surface electromyogram signal using kernel recursive least square tracker. *Biomedical Signal Processing and Control*, v. 46, p. 104–115, 1 set. 2018. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S1746809418301642>>. Acesso em: 3 out. 2022.

BERTOS, G. A.; PAPADOPOULOS, E. G. Lower-Limb Prosthetics. *Handbook of Biomechanics*. [S.I.]: Elsevier, 2019. p. 241–282. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/B9780128125397000076>>.

BETTHAUSER, J. L. *et al.* Limb-position robust classification of myoelectric signals for prosthesis control using sparse representations. 13 ago. 2016, [S.I.]: IEEE, 13 ago. 2016. p. 6373–6376. Disponível em: <<http://ieeexplore.ieee.org/document/7592186/>>. Acesso em: 7 nov. 2022.

BI, L.; FELEKE, A.-->GENETU; GUAN, C. A review on EMG-based motor intention prediction of continuous human upper limb motion for human-robot collaboration. *Biomedical Signal Processing and Control*, v. 51, p. 113–127, maio 2019. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S1746809419300473>>.

BRAGA, A. DE P.; CARVALHO, A. P. DE L. F. DE; LUDERMIR, T. B. *Redes neurais artificiais teoria e aplicações*. [Rio de Janeiro: Livros Técnicos e Científicos 2000., 2000.

BRENNEIS, D. J. A.; DAWSON, M. R.; PILARSKI, P. M. DEVELOPMENT OF THE HANDI HAND: AN INEXPENSIVE, MULTI-ARTICULATING, SENSORIZED HAND FOR MACHINE LEARNING RESEARCH IN MYOELECTRIC CONTROL. 2017, Fredericton, New Brunswick: [s.n.], 2017. . Acesso em: 20 ago. 2025.

BROCKMAN, G. *et al.* OpenAI Gym. 5 jun. 2016. Disponível em: <<http://arxiv.org/abs/1606.01540>>.

BUENO, D. R.; MONTANO, L. Neuromusculoskeletal model self-calibration for on-line sequential bayesian moment estimation. *Journal of Neural Engineering*, v. 14, n. 2, p. 026011, 1 abr. 2017. Disponível em: <<https://iopscience.iop.org/article/10.1088/1741-2552/aa58f5>>. Acesso em: 7 nov. 2022.

CHEN, M.; ZHOU, P. A Novel Framework Based on FastICA for High Density Surface EMG Decomposition. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, v. 24, n. 1, p. 117–127, 1 jan. 2016. Disponível em: <<https://ieeexplore.ieee.org/document/7058391/>>. Acesso em: 11 set. 2023.

CHEN, P.; HAN, D. Reward adaptive wind power tracking control based on deep deterministic policy gradient. *Applied Energy*, v. 348, p. 121519, 15 out. 2023. Disponível em: <<https://www.sciencedirect.com/science/article/abs/pii/S0306261923008838?via%3Dihub>>. Acesso em: 9 ago. 2025.

CHEN, Z. *et al.* Optimization of Hydraulic Robot Arm Motion Control Algorithm Based on Neural Network Algorithm. *2024 IEEE 4th International Conference on Electronic Technology, Communication and Information, ICETCI 2024*, p. 1330–1335, 2024. Acesso em: 9 ago. 2025.

CHOLLET, F. AND OTHERS. *Keras*. [S.l.: s.n.], 2015

DA SILVA, I. N. *et al.* *Artificial Neural Networks*. Cham: Springer International Publishing, 2017.

DAI, C.; HU, X. Finger Joint Angle Estimation Based on Motoneuron Discharge Activities. *IEEE Journal of Biomedical and Health Informatics*, v. 24, n. 3, p. 760–767, 1 mar. 2020. Acesso em: 13 jul. 2025.

DE LUCA, C. J. *et al.* Decomposition of Surface EMG Signals. *Journal of Neurophysiology*, v. 96, n. 3, p. 1646–1657, set. 2006. Disponível em: <<https://www.physiology.org/doi/10.1152/jn.00009.2006>>.

DING, S. *et al.* Movement Identification Based on Transient sEMG for Control of Prosthesis. *Advanced Materials Research*, v. 971–973, p. 1651–1654, jun. 2014. Disponível em: <<https://www.scientific.net/AMR.971-973.1651>>. Acesso em: 10 nov. 2022.

DONG, R. *et al.* An enhanced deep deterministic policy gradient algorithm for intelligent control of robotic arms. *Frontiers in Neuroinformatics*, v. 17, p. 1096053, 23 jan. 2023. Acesso em: 21 jul. 2025.

DUTRA, B. G.; SILVEIRA, A. DA S. Multivariable grasping force control of myoelectric multi-fingered hand prosthesis. *International Journal of Dynamics and Control*, v. 11, n. 6, p. 3145–3158, 3 dez. 2023. Disponível em: <<https://link.springer.com/10.1007/s40435-023-01130-8>>. Acesso em: 11 fev. 2025.

FAJARDO, J. *et al.* An Affordable open-source multifunctional upper-limb prosthesis with intrinsic actuation. 1 mar. 2017, [S.l.]: IEEE, 1 mar. 2017. p. 1–6. Disponível em: <<http://ieeexplore.ieee.org/document/8025206/>>. Acesso em: 12 out. 2022.

FAJARDO, J.; LEMUS, A.; ROHMER, E. Galileo bionic hand: sEMG activated approaches for a multifunction upper-limb prosthetic. 7 nov. 2015, [S.l.]: IEEE, 7 nov.

2015. p. 1–6. Disponível em: <<https://ieeexplore.ieee.org/document/7428468/>>. Acesso em: 7 nov. 2022.

FAN, J. *et al.* Improving sEMG-based motion intention recognition for upper-limb amputees using transfer learning. *Neural Computing and Applications*, v. 35, n. 22, p. 16101–16111, 25 ago. 2023. Disponível em: <<https://link.springer.com/10.1007/s00521-021-06292-0>>. Acesso em: 25 set. 2022.

FUJIMOTO, S.; VAN HOOFF, H.; MEGER, D. Addressing Function Approximation Error in Actor-Critic Methods. 2018. Disponível em: <<https://github.com/>>. Acesso em: 11 ago. 2025.

GALLIVAN, J. P. *et al.* Decision-making in sensorimotor control. *Nature Reviews Neuroscience*, v. 19, n. 9, p. 519–534, 8 set. 2018. Disponível em: <<https://www.nature.com/articles/s41583-018-0045-9>>.

GHAZALI, R. *et al.* Intelligent Controller Design for Multifunctional Prosthetics Hand. *International Journal of Mechanical Engineering and Robotics Research*, v. 6, n. 6, p. 495–501, 2017. Disponível em: <<http://www.ijmerr.com/index.php?m=content&c=index&a=show&catid=152&id=890>>. Acesso em: 11 fev. 2025.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. Cambridge, MA: MIT Press, 2017.

GREGORI, V.; CAPUTO, B.; GIJSBERTS, A. The Difficulty of Recognizing Grasps from sEMG during Activities of Daily Living. 9 ago. 2018, [S.l.]: IEEE, 9 ago. 2018. p. 583–588. Disponível em: <<https://ieeexplore.ieee.org/document/8487966/>>. Acesso em: 3 out. 2022.

GUO, S. *et al.* Comparison of sEMG-Based Feature Extraction and Motion Classification Methods for Upper-Limb Movement. *Sensors*, v. 15, n. 4, p. 9022–9038, 16 abr. 2015. Disponível em: <<http://www.mdpi.com/1424-8220/15/4/9022>>. Acesso em: 10 nov. 2022.

GUO, W. *et al.* Mechanomyography Assisted Myoelectric Sensing for Upper-Extremity Prostheses: A Hybrid Approach. *IEEE Sensors Journal*, v. 17, n. 10, p. 3100–3108, 15 maio 2017a. Disponível em: <<http://ieeexplore.ieee.org/document/7874103/>>. Acesso em: 7 nov. 2022.

GUO, W. *et al.* Toward an Enhanced Human–Machine Interface for Upper-Limb Prosthesis Control With Combined EMG and NIRS Signals. *IEEE Transactions on Human-Machine Systems*, v. 47, n. 4, p. 564–575, 1 ago. 2017b. Disponível em: <<http://ieeexplore.ieee.org/document/7805220/>>. Acesso em: 7 nov. 2022.

HAARNOJA, T. *et al.* Soft Actor-Critic Algorithms and Applications. 12 dez. 2018. Disponível em: <<http://arxiv.org/abs/1812.05905>>. Acesso em: 27 fev. 2025.

HARRIS, C. R. *et al.* Array programming with NumPy. *Nature*, v. 585, n. 7825, p. 357–362, 17 set. 2020.

HERLE, S. Movement intention detection from SEMG signals using time-domain features and discriminant analysis classifiers. 3 maio 2018, [S.l.]: IEEE, 3 maio 2018. p. 1–6. Disponível em: <<https://ieeexplore.ieee.org/document/8402774/>>. Acesso em: 3 out. 2022.

HUANG, G. *et al.* Divide-and-conquer muscle synergies: A new feature space decomposition approach for simultaneous multifunction myoelectric control. *Biomedical Signal Processing and Control*, v. 44, p. 209–220, 1 jul. 2018. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S1746809418300880>>. Acesso em: 3 out. 2022.

HUDGINS, B.; PARKER, P.; SCOTT, R. N. A new strategy for multifunction myoelectric control. *IEEE Transactions on Biomedical Engineering*, v. 40, n. 1, p. 82–94, 1993. Disponível em: <<http://ieeexplore.ieee.org/document/204774/>>.

JARQUE-BOU, N. J.; ATZORI, M.; MÜLLER, H. A large calibrated database of hand movements and grasps kinematics. *Scientific Data*, v. 7, n. 1, p. 1–10, 1 dez. 2020. Disponível em: <<https://www.nature.com/articles/s41597-019-0349-2>>. Acesso em: 24 ago. 2025.

JARRASSE, N. *et al.* Voluntary phantom hand and finger movements in transhumeral amputees could be used to naturally control polydigital prostheses. 11 jul. 2017, [S.l.]: IEEE, 11 jul. 2017. p. 1239–1245. Disponível em: <<https://ieeexplore.ieee.org/document/8009419/>>. Acesso em: 12 out. 2022.

JIANG, H. *et al.* TF2AngleNet: Continuous finger joint angle estimation based on multidimensional time–frequency features of sEMG signals. *Biomedical Signal Processing and Control*, v. 107, p. 107833, 1 set. 2025. Disponível em: <<https://www.sciencedirect-com.ez27.periodicos.capes.gov.br/science/article/pii/S1746809425003441?via%3Dihub>>. Acesso em: 25 jun. 2025.

KAWAMOTO, H. *et al.* Voluntary motion support control of Robot Suit HAL triggered by bioelectrical signal for hemiplegia. *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC'10*, p. 462–466, 2010. Acesso em: 5 dez. 2023.

KRASOULIS, A. *et al.* Improved prosthetic hand control with concurrent use of myoelectric and inertial measurements. *Journal of NeuroEngineering and Rehabilitation*, v. 14, n. 1, p. 71, 11 dez. 2017. Disponível em: <<http://jneuroengrehab.biomedcentral.com/articles/10.1186/s12984-017-0284-4>>. Acesso em: 7 nov. 2022.

KULWA, F. *et al.* A Multidataset Characterization of Window-Based Hyperparameters for Deep CNN-Driven sEMG Pattern Recognition. *IEEE Transactions on Human-Machine Systems*, v. 54, n. 1, p. 131–142, 1 fev. 2024. Disponível em: <<https://ieeexplore.ieee.org/document/10360331/>>. Acesso em: 15 set. 2024.

KUMAR, P. *et al.* Proximal Policy Optimization based computations offloading for delay optimization in UAV-assisted mobile edge computing. 15 dez. 2023, [S.I.]: IEEE, 15 dez. 2023. p. 3355–3364. Disponível em: <<https://ieeexplore.ieee.org/document/10386861/>>. Acesso em: 27 fev. 2025.

LI, C. *et al.* PCA and deep learning based myoelectric grasping control of a prosthetic hand. *BioMedical Engineering OnLine*, v. 17, n. 1, p. 107, 6 dez. 2018. Disponível em: <<https://biomedical-engineering-online.biomedcentral.com/articles/10.1186/s12938-018-0539-8>>. Acesso em: 3 out. 2022.

LI, K. *et al.* A review of the key technologies for sEMG-based human-robot interaction systems. *Biomedical Signal Processing and Control*, v. 62, p. 102074, set. 2020. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S1746809420302305>>.

LI, W.; SHI, P.; YU, H. Gesture Recognition Using Surface Electromyography and Deep Learning for Prostheses Hand: State-of-the-Art, Challenges, and Future. *Frontiers in Neuroscience*, v. 15, 26 abr. 2021. Acesso em: 25 set. 2022.

LI, Z. *et al.* Motion Planning of Six-DOF Arm Robot Based on Improved DDPG Algorithm. jul. 2020, [S.I.]: IEEE, jul. 2020. p. 3954–3959. Disponível em: <<https://ieeexplore.ieee.org/document/9188521/>>.

LIANG PENG *et al.* Combined use of sEMG and accelerometer in hand motion classification considering forearm rotation. jul. 2013, [S.I.]: IEEE, jul. 2013. p. 4227–4230. Disponível em: <<http://ieeexplore.ieee.org/document/6610478/>>. Acesso em: 10 nov. 2022.

LIGHT, C. M.; CHAPPELL, P. H.; KYBERD, P. J. Establishing a standardized clinical assessment tool of pathologic and prosthetic hand function: Normative data, reliability, and validity. *Archives of Physical Medicine and Rehabilitation*, v. 83, n. 6, p. 776–783, 1 jun. 2002. Acesso em: 30 jan. 2024.

LILLICRAP, T. P. *et al.* Continuous control with deep reinforcement learning. 9 set. 2015. Disponível em: <<http://arxiv.org/abs/1509.02971>>.

LIU, J. *et al.* Continuous Recognition of Multifunctional Finger and Wrist Movements in Amputee Subjects Based on sEMG and Accelerometry. *The Open Biomedical Engineering Journal*, v. 10, n. 1, p. 101–110, 30 nov. 2016. Disponível em: <<https://openbiomedicalengineeringjournal.com/VOLUME/10/PAGE/101/>>. Acesso em: 7 nov. 2022.

LIZHI PAN *et al.* A linear model for simultaneously and proportionally estimating wrist kinematics from emg during mirrored bilateral movements. jul. 2013, [S.I.]: IEEE, jul. 2013. p. 4593–4596. Disponível em: <<http://ieeexplore.ieee.org/document/6610570/>>. Acesso em: 10 nov. 2022.

LONG, Y. *et al.* A Transfer Learning Based Cross-Subject Generic Model for Continuous Estimation of Finger Joint Angles From a New User. *IEEE Journal of Biomedical and Health Informatics*, v. 27, n. 4, p. 1914–1925, 1 abr. 2023. Disponível em: <<https://ieeexplore.ieee.org/document/10008406/>>. Acesso em: 13 jul. 2025.

MARTÍN ABADI *et al.* *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. [S.l.: s.n.]. Disponível em: <<https://www.tensorflow.org/>>. , 2015

MATOS, J.; CAPSI-MORALES, P.; PIAZZA, C. Autonomous slip control inspired by human physiology for improved shared control strategy. *Wearable Technologies*, v. 6, p. e22, 9 jun. 2025. Disponível em: <https://www.cambridge.org/core/product/identifier/S2631717625100078/type/journal_article>. Acesso em: 16 jul. 2025.

MAYOR, J. J. V. *et al.* Dexterous hand gestures recognition based on low-density sEMG signals for upper-limb forearm amputees. *Research on Biomedical Engineering*, v. 33, n. 3, p. 202–217, 30 set. 2017. Acesso em: 6 out. 2022.

MCKINNEY, W. *Data Structures for Statistical Computing in Python*. 2010, [S.l.: s.n.], 2010. p. 56–61.

MERLETTI, R.; FARINA, D. *Surface Electromyography: Physiology, Engineering, and Applications*. Piscataway, NJ: Hoboken, New Jersey: Wiley, 2016. Disponível em: <<https://onlinelibrary.wiley.com/doi/book/10.1002/9781119082934>>. (IEEE Press series in biomedical engineering).

MEYES, R.; SCHEIDERER, C.; MEISEN, T. Continuous Motion Planning for Industrial Robots based on Direct Sensory Input. *Procedia CIRP*, v. 72, p. 291–296, 2018.

MNIH, V. *et al.* Playing Atari with Deep Reinforcement Learning. 19 dez. 2013.

MOHAMMADREZA, E.; FONSECA, V. P. DA; JIANG, X. Investigating the Impact of Training Protocols on Myoelectric Pattern Recognition Control in Upper-Limb Amputees. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, v. 33, p. 1338–1348, 2025. Disponível em: <<https://ieeexplore.ieee.org/document/10942474/>>.

MONTGOMERY, A. E.; ALLEN, J. M.; ELBASIOUNY, S. M. Adaptive Neural Decoder for Prosthetic Hand Control. *Frontiers in Neuroscience*, v. 15, p. 590775, 8 abr. 2021. Disponível em: <www.frontiersin.org>. Acesso em: 9 dez. 2025.

MORALES, MIGUEL. *Grokking Deep Reinforcement Learning*. New York: Manning, 2020.

MOSLEMI, M.; SADEDEL, M.; MOGHADAM, M. M. Squat and tuck jump maneuver for single-legged robot with an active toe joint using model-free deep reinforcement learning. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, v. 46, n. 8, p. 456, 17 ago. 2024. Disponível em: <<https://link.springer.com/10.1007/s40430-024-05028-0>>. Acesso em: 27 fev. 2025.

NAIK, G. R.; AL-TIMEMY, A. H.; NGUYEN, H. T. Transradial Amputee Gesture Classification Using an Optimal Number of sEMG Sensors: An Approach Using ICA Clustering. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, v.

24, n. 8, p. 837–846, 1 ago. 2016. Disponível em: <<https://ieeexplore.ieee.org/document/7271103/>>. Acesso em: 7 nov. 2022.

NASTARIN, A.; AKTER, A.; AWAL, MD. A. Robust Control of Hand Prostheses from Surface EMG Signal for Transradial Amputees. set. 2019, Dhaka, Bangladesh: IEEE, set. 2019. p. 143–148. Disponível em: <<https://ieeexplore.ieee.org/document/8975630/>>.

NAZEMI, A.; MALEKI, A. Artificial neural network classifier in comparison with LDA and LS-SVM classifiers to recognize 52 hand postures and movements. 18 out. 2014, [S.l.]: IEEE, 18 out. 2014. p. 18–22. Disponível em: <<http://ieeexplore.ieee.org/document/6993343/>>. Acesso em: 10 nov. 2022.

NIELSEN, M. A. *Neural Networks and Deep Learning*. [S.l.]: Determination Press, 2019.

NINAPRO. *DB7 Guidelines*. [S.l: s.n.], 2017.

NOVAK, D.; RIENER, R. A survey of sensor fusion methods in wearable robotics. *Robotics and Autonomous Systems*, v. 73, p. 155–170, nov. 2015. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S0921889014001705>>.

NUGROHO, L. *et al.* Optimization of reward shaping function based on genetic algorithm applied to a cross validated deep deterministic policy gradient in a powered landing guidance problem. *Engineering Applications of Artificial Intelligence*, v. 120, p. 105798, abr. 2023. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S0952197622007886>>.

PAN, L.; HARMODY, A.; HUANG, H. A Reliable Multi-User EMG Interface Based on A Generic-Musculoskeletal Model against Loading Weight Changes. 26 jul. 2018, [S.l.]: IEEE, 26 jul. 2018. p. 2104–2107. Disponível em: <<https://ieeexplore.ieee.org/document/8512685/>>. Acesso em: 3 out. 2022.

PEDREGOSA, F. *et al.* Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.

PHILIPSON, L. Adaptable myoelectric prosthetic control with functional visual feedback using microprocessor techniques. *Medical & Biological Engineering & Computing*, v. 23, n. 1, p. 8–14, jan. 1985. Disponível em: <<http://link.springer.com/10.1007/BF02444020>>. Acesso em: 10 nov. 2022.

POWAR, O. S.; CHEMMANGAT, K. Feature selection for myoelectric pattern recognition using two channel surface electromyography signals. 19 nov. 2017, [S.l.]: IEEE, 19 nov. 2017. p. 1022–1026. Disponível em: <<http://ieeexplore.ieee.org/document/8228007/>>. Acesso em: 28 mar. 2023.

RAHMATILLAH, A.; SALAMAT, L.; SOELISTIONO, S. Design and Implementation of Prosthetic Hand Control Using Myoelectric Signal. *International Journal on Advanced Science, Engineering and Information Technology*, v. 9, n. 4, p. 1231, 18 ago. 2019. Disponível em:

<http://ijaseit.insightsociety.org/index.php?option=com_content&view=article&id=9&Itemid=1&article_id=4887>. Acesso em: 3 out. 2022.

RAZZAGHI, P. *et al.* A survey on reinforcement learning in aviation applications. *Engineering Applications of Artificial Intelligence*, v. 136, p. 108911, 1 out. 2024. Disponível em: <<https://www.sciencedirect.com/science/article/abs/pii/S0952197624010698>>. Acesso em: 21 jul. 2025.

REATEGUI, J. *et al.* Handgrip estimation based on total variation denoising filtering for control applications. nov. 2013, [S.l.]: IEEE, nov. 2013. p. 1–5. Disponível em: <<http://ieeexplore.ieee.org/document/6701587/>>. Acesso em: 10 nov. 2022.

REHMAN, M. Z. UR *et al.* Performance of Combined Surface and Intramuscular EMG for Classification of Hand Movements. 26 jul. 2018, [S.l.]: IEEE, 26 jul. 2018. p. 5220–5223. Disponível em: <<https://ieeexplore.ieee.org/document/8513480/>>. Acesso em: 3 out. 2022.

RÚBIO, G. DE P.; COSTA, M. C. B.; VIMIEIRO, C. B. S. A New Proposal for Intelligent Continuous Controller of Robotic Finger Prostheses Using Deep Deterministic Policy Gradient Algorithm Through Simulated Assessments. *Robotics*, v. 14, n. 4, p. 49, 1 abr. 2025. Disponível em: <<https://www.mdpi.com/2218-6581/14/4/49/htm>>. Acesso em: 17 ago. 2025.

RYAIT, H. S.; ARORA, A. S.; AGARWAL, R. Realisation of SEMG-based multifunctional prototype elbow prosthesis. *International Journal of Biomedical Engineering and Technology*, v. 9, n. 1, p. 72, 2012. Disponível em: <<http://www.inderscience.com/link.php?id=47372>>. Acesso em: 10 nov. 2022.

SADIKOGLU, F.; KAVALCIOGLU, C.; DAGMAN, B. Electromyogram (EMG) signal detection, classification of EMG signals and diagnosis of neuropathy muscle disease. *Procedia Computer Science*, v. 120, p. 422–429, 1 jan. 2017. Acesso em: 5 dez. 2023.

SCHEME, E.; ENGLEHART, K. Electromyogram pattern recognition for control of powered upper-limb prostheses: state of the art and challenges for clinical use. *Journal of rehabilitation research and development*, v. 48, n. 6, p. 643–660, 2011. Disponível em: <<https://pubmed.ncbi.nlm.nih.gov/21938652/>>. Acesso em: 5 dez. 2023.

SCHULMAN, J. *et al.* Proximal Policy Optimization Algorithms. 20 jul. 2017. Disponível em: <<https://arxiv.org/abs/1707.06347v2>>. Acesso em: 12 fev. 2025.

SEGIL, J. L.; WEIR. Novel postural control algorithm for control of multifunctional myoelectric prosthetic hands. *Journal of Rehabilitation Research and Development*, v. 52, n. 4, p. 449–466, 29 ago. 2015. Disponível em: <<http://www.rehab.research.va.gov/jour/2015/524/pdf/JRRD-2014-05-0134.pdf>>. Acesso em: 7 nov. 2022.

SIMAO, M. *et al.* A Review on Electromyography Decoding and Pattern Recognition for Human-Machine Interaction. *IEEE Access*, v. 7, p. 39564–39582, 2019. Disponível em: <<https://ieeexplore.ieee.org/document/8672131/>>.

SPONG, M. W.; HUTCHINSON, S.; VIDYASAGAR, M. *Robot modeling and control*. 2. ed. Hoboken, NJ: John Wiley & Sons, Inc., 2020.

STRAZZULLA, I. *et al.* Online Bimanual Manipulation Using Surface Electromyography and Incremental Learning. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, v. 25, n. 3, p. 227–234, 1 mar. 2017. Disponível em: <<https://ieeexplore.ieee.org/document/7460959/>>. Acesso em: 7 nov. 2022.

STURMA, A. *et al.* A surface EMG test tool to measure proportional prosthetic control. *Biomedical Engineering / Biomedizinische Technik*, v. 60, n. 3, p. 207–213, 1 jan. 2015. Disponível em: <<https://www.degruyter.com/document/doi/10.1515/bmt-2014-0022/html>>. Acesso em: 10 nov. 2022.

SUMIEA, E. H. *et al.* Deep deterministic policy gradient algorithm: A systematic review. *Heliyon*, v. 10, n. 9, p. e30697, maio 2024. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S2405844024067288>>. Acesso em: 27 fev. 2025.

SUN, M. *et al.* DDPG-Based Decision-Making Strategy of Adaptive Cruising for Heavy Vehicles Considering Stability. *IEEE Access*, v. 8, p. 59225–59246, 2020. Acesso em: 9 ago. 2025.

SUTTON, R. S.; BARTO, A. G. *Reinforcement learning: an introduction*. Second edition ed. Cambridge, MA: The MIT Press, 2018.

TANG, Z.; YU, H.; CANG, S. Impact of Load Variation on Joint Angle Estimation From Surface EMG Signals. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, v. 24, n. 12, p. 1342–1350, 1 dez. 2016. Disponível em: <<https://ieeexplore.ieee.org/document/7332977/>>. Acesso em: 7 nov. 2022.

TELLO, R. M. G. *et al.* Towards semg classification based on Bayesian and k-NN to control a prosthetic hand. fev. 2013, [S.l.]: IEEE, fev. 2013. p. 1–6. Disponível em: <<http://ieeexplore.ieee.org/document/6487520/>>. Acesso em: 10 nov. 2022.

THOMAS, S. J.; ZENI, J. A.; WINTER, D. A. *Winter's biomechanics and motor control of human movement*. 5. ed. Hoboken, NJ: Wiley, 2023.

VÁSCONEZ, J. P. *et al.* A comparison of EMG-based hand gesture recognition systems based on supervised and reinforcement learning. *Engineering Applications of Artificial Intelligence*, v. 123, p. 106327, 1 ago. 2023. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S0952197623005110>>. Acesso em: 3 nov. 2024.

VEER, K. An analytical approach to test and design upper limb prosthesis. *Journal of Medical Engineering & Technology*, v. 39, n. 6, p. 328–330, 18 ago. 2015. Disponível em: <<http://www.tandfonline.com/doi/full/10.3109/03091902.2015.1059506>>. Acesso em: 10 nov. 2022.

VERGARA, M.; AGOST, M. J.; GRACIA-IBÁÑEZ, V. Dorsal and palmar aspect dimensions of hand anthropometry for designing hand tools and protections. *Human Factors and Ergonomics in Manufacturing & Service Industries*, v. 28, n. 1, p. 17–28, 1 jan. 2018. Disponível em: <<https://onlinelibrary.wiley.com/doi/full/10.1002/hfm.20714>>. Acesso em: 25 fev. 2024.

VERGARA, M.; AGOST, M. J.; GRACIA-IBÁÑEZ, V. Dorsal and palmar aspect dimensions of hand anthropometry for designing hand tools and protections. *Human Factors and Ergonomics in Manufacturing & Service Industries*, v. 28, n. 1, p. 17–28, 9 jan. 2018.

VIRTANEN, P. *et al.* SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, v. 17, n. 3, p. 261–272, 2 mar. 2020. Disponível em: <<https://www.nature.com/articles/s41592-019-0686-2>>.

WANG, Q.; HONG, Z.; ZHONG, Y. Learn to swim: Online motion control of an underactuated robotic eel based on deep reinforcement learning. *Biomimetic Intelligence and Robotics*, v. 2, n. 4, p. 100066, 1 dez. 2022. Acesso em: 11 fev. 2025.

WARIS, M. A. *et al.* Classification of functional motions of hand for upper limb prosthesis with surface electromyography. *International Journal of Biology and Biomedical Engineering*, v. 8, p. 15–20, 2014. Acesso em: 10 nov. 2022.

YANG, D.; YANG, W.; *et al.* Classification of Multiple Finger Motions During Dynamic Upper Limb Movements. *IEEE Journal of Biomedical and Health Informatics*, v. 21, n. 1, p. 134–141, 1 jan. 2017. Disponível em: <<http://ieeexplore.ieee.org/document/7298404/>>. Acesso em: 7 nov. 2022.

YANG, D.; GU, Y.; *et al.* Dynamic training protocol improves the robustness of PR-based myoelectric control. *Biomedical Signal Processing and Control*, v. 31, p. 249–256, 1 jan. 2017. Disponível em: <<https://linkinghub.elsevier.com/retrieve/pii/S1746809416301185>>. Acesso em: 7 nov. 2022.

YING, ZHENZHI. *et al.* Real-time Dexterous Prosthesis Hand Control by Decoding Neural Information Based on EMG Decomposition. 13 maio 2024, [S.l.]: IEEE, 13 maio 2024. p. 966–972. Disponível em: <<https://ieeexplore.ieee.org/document/10611356/>>. Acesso em: 11 fev. 2025.

YOUNG, A. J. *et al.* Classification of Simultaneous Movements Using Surface EMG Pattern Recognition. *IEEE Transactions on Biomedical Engineering*, v. 60, n. 5, p. 1250–1258, maio 2013. Disponível em: <<http://ieeexplore.ieee.org/document/6377275/>>. Acesso em: 10 nov. 2022.

YOUNG, AARON J. *et al.* A new hierarchical approach for simultaneous control of multi-joint powered prostheses. jun. 2012, [S.l.]: IEEE, jun. 2012. p. 514–520. Disponível em: <<http://ieeexplore.ieee.org/document/6290709/>>. Acesso em: 10 nov. 2022.

ZHANG, Z. *et al.* Closed-Loop Force Control by Biorealistic Hand Prosthesis With Visual and Tactile Sensory Feedback. *IEEE transactions on neural systems and rehabilitation engineering: a publication of the IEEE Engineering in Medicine and Biology Society*, v. 32, p. 2939–2949, 2024. Disponível em: <<https://pubmed.ncbi.nlm.nih.gov/39110556/>>. Acesso em: 9 dez. 2025.

ZHOU, H.; ALICI, G. Non-Invasive Human-Machine Interface (HMI) Systems With Hybrid On-Body Sensors for Controlling Upper-Limb Prosthesis: A Review. *IEEE Sensors Journal*, v. 22, n. 11, p. 10292–10307, 1 jun. 2022. Disponível em: <<https://ieeexplore.ieee.org/document/9761870/>>. Acesso em: 25 set. 2022.

APÊNDICE A – RESULTADOS COMPLEMENTARES DO TESTE PILOTO

A.1) INTRODUÇÃO

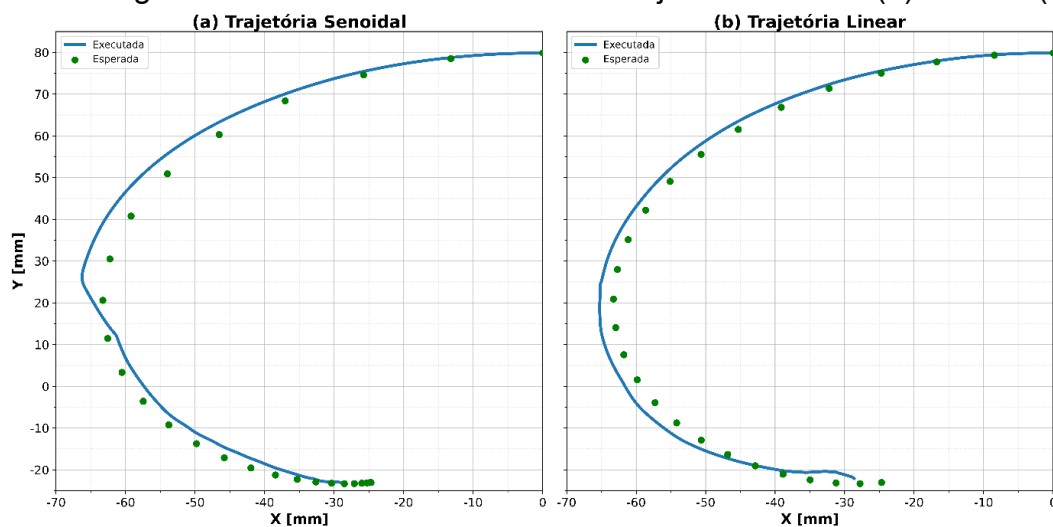
Neste apêndice serão apresentados os resultados complementares da avaliação da rede neural artificial (RNA), treinada utilizando o algoritmo *Deep Deterministic Policy Gradient* (DDPG), utilizada para controlar o dedo indicador a realizar diferentes trajetórias. Aqui, serão apresentados os resultados referentes aos teste do comportamento de RNA em ambientes com diferentes escalas (90%, 110% e 125% do volume original). Nas próximas sessões serão apresentadas o torque em cada junta, o valor das variáveis de estado e a trajetória desenvolvida quando a entrada de trajetória utilizava as entradas binárias de trajetória linear e senoidal e quando foram utilizadas as entradas proporcionais com 25% de seleção senoidal (M2575), 50 % de seleção senoidal (M5050) e 75% de seleção senoidal (M7525).

A.2) RESPOSTA DO CONTROLADOR ATUANDO O DEDO COM 90% DO VOLUME ORIGINAL

Nessa sessão são apresentados os resultados obtidos com o controlador treinado atuando o dedo com 90% do volume original. Na Figura A.1 são apresentadas as trajetórias desenvolvidas pela ponta do dedo utilizando as entradas binárias das trajetórias senoidal e linear. O torque aplicado em cada uma das juntas é apresentado na Figura A.3 demonstrando o padrão oscilatório para as trajetórias binárias. As variáveis de estado para cada uma das entradas binárias são apresentadas na Figura A.4 e na Figura A.5. Os resultados apresentados são todos comparados com as respostas esperadas para as trajetórias linear e senoidal.

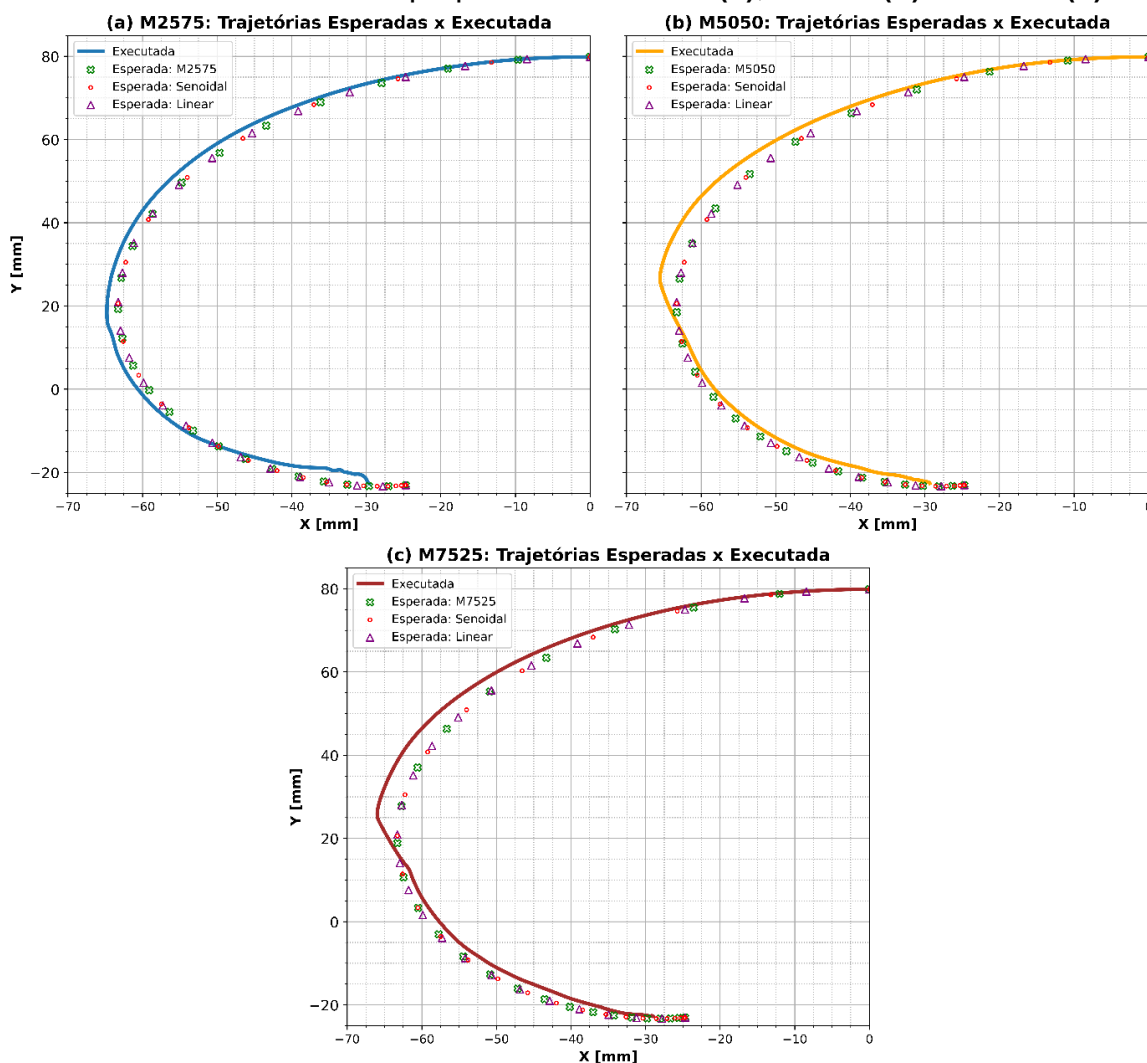
Foi também avaliado as respostas do controlador para as entradas proporcionais M2575, M5050 e M7525. As trajetórias desenvolvidas são apresentadas na Figura A.2, o valor do torque empregado em cada junta na Figura A.6 e as variáveis de estado na Figura A.7. Foi também apresentado as coordenadas X e Y individualmente no tempo para cada uma das entradas em comparação com as coordenadas esperadas nas trajetória linear e senoidal apresentado na Figura A.8. Por fim é apresentado na Tabela A.1 um comparativo entre a magnitude do erro médio de trajetória do modelo utilizando as entradas proporcionais e as trajetórias de treinamento utilizadas.

Figura A.1 – Trajetória executado pelo controlador treinado com o dedo com 90% do volume original com as entradas binárias de trajetória senoidal (a) e linear (b).



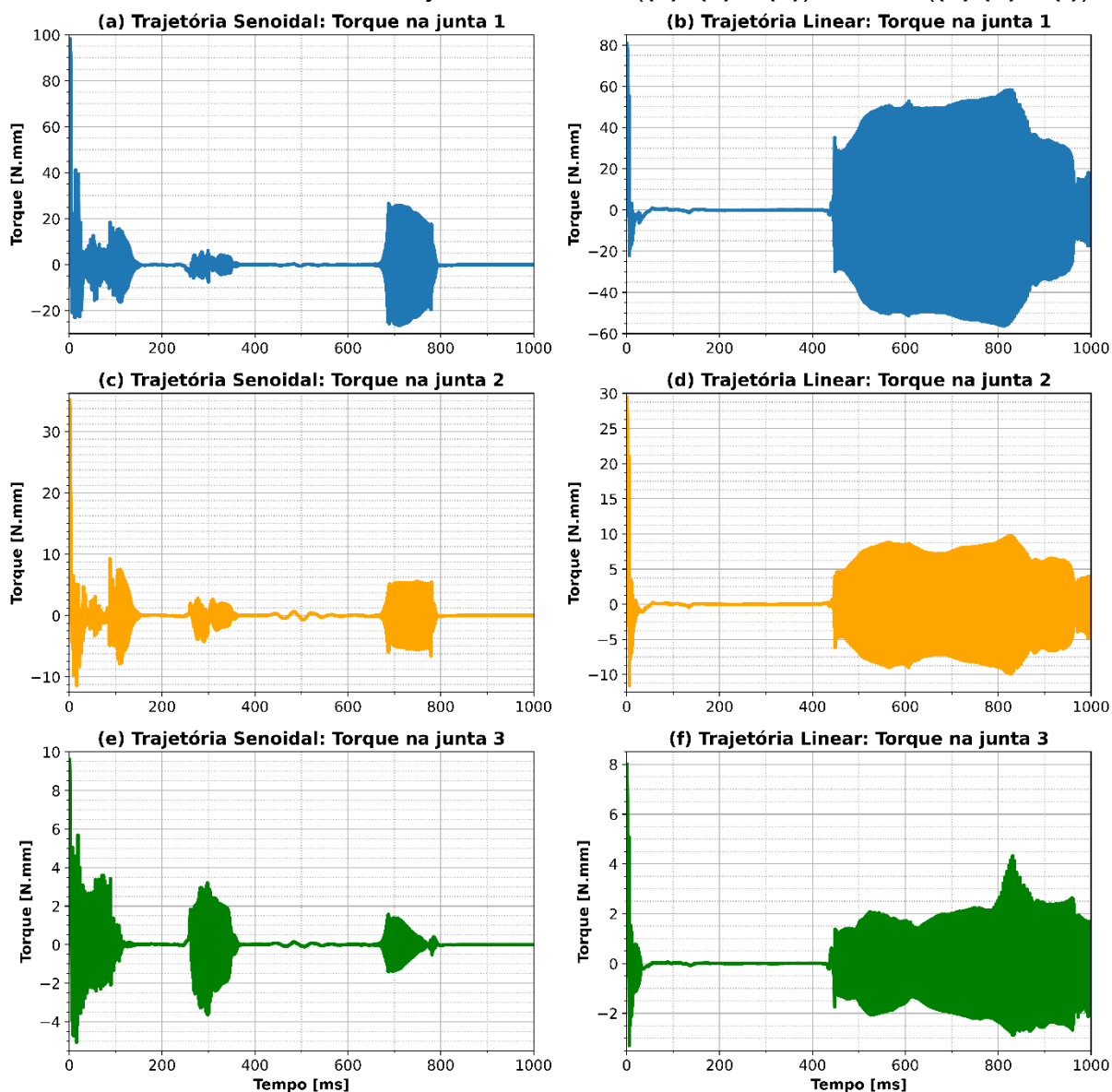
Fonte: Elaborado pelo autor

Figura A.2 – Trajetória executada pelo controlador treinado com o dedo com 90% do volume com as entradas proporcionais M2575 (a), M5050 (b) e M7525 (c).



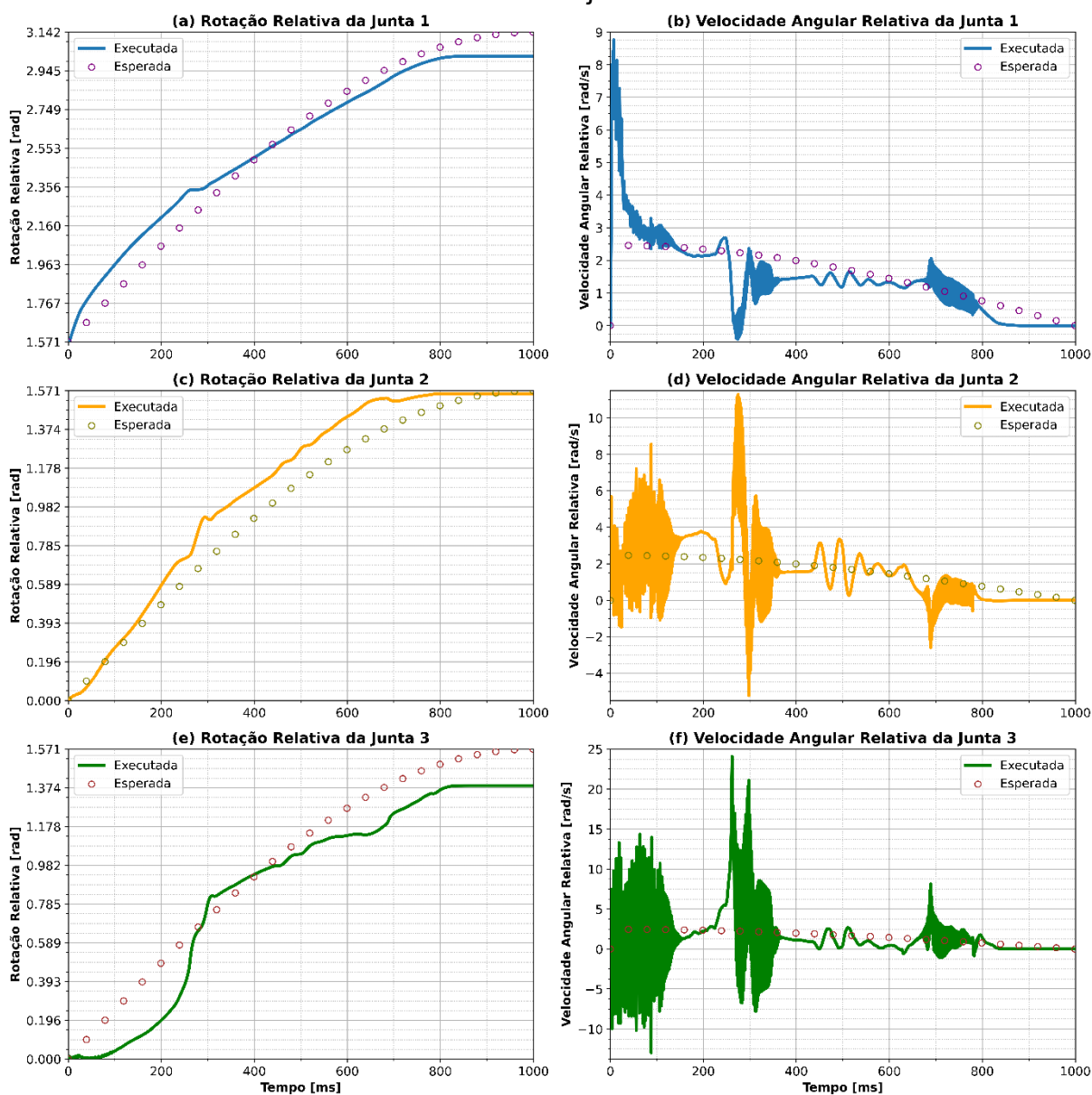
Fonte: Elaborado pelo autor.

Figura A.3 – Torque aplicados em cada uma das juntas do dedo com 90% do volume com as entradas de trajetória senoidal ((a), (c) e (e)) e linear ((b),(d) e (f)).



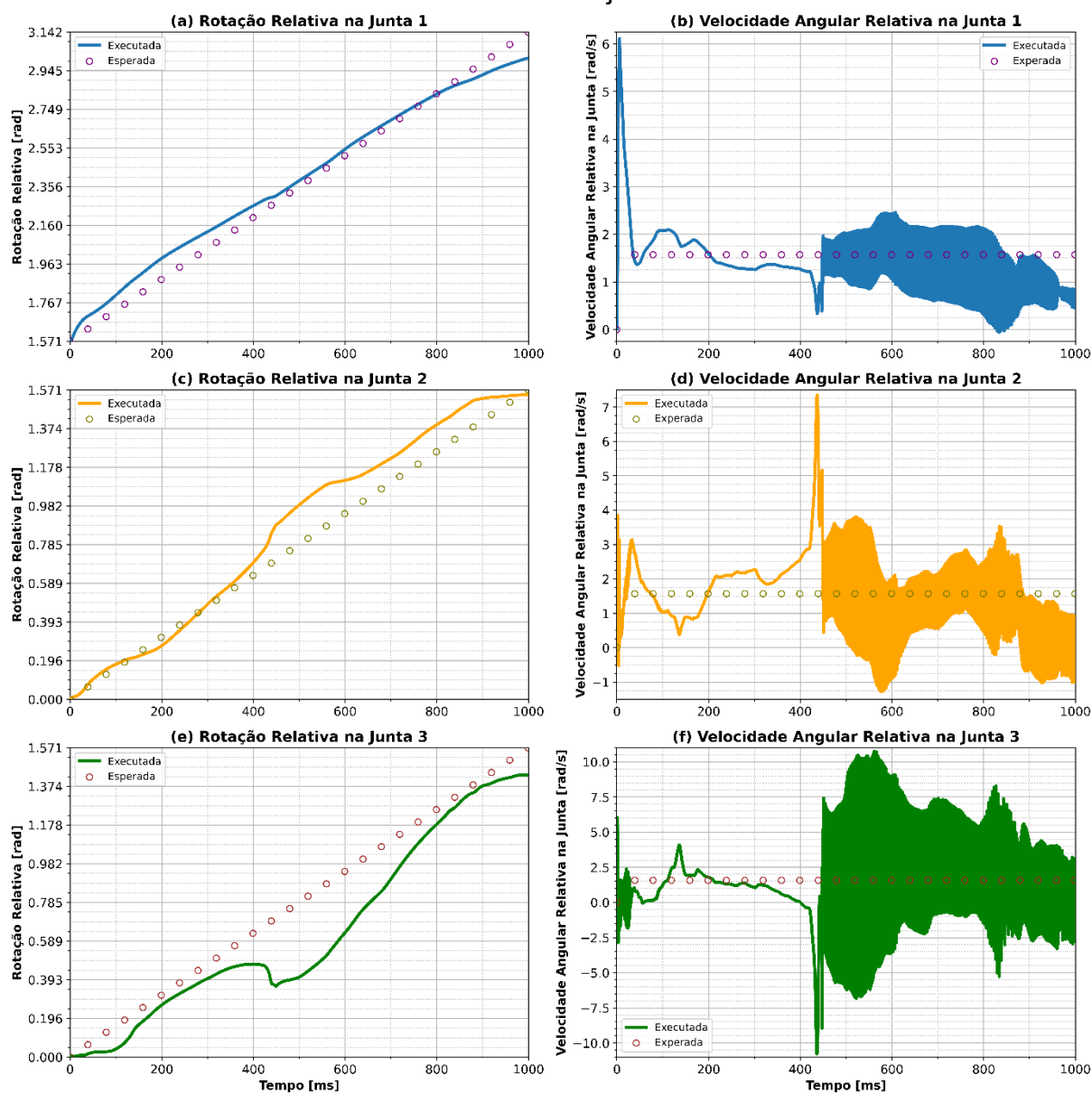
Fonte: Elaborado pelo autor.

Figura A.4 – Rotação angular relativa de cada uma das juntas ((a), (c) e (e)) e velocidades angulares relativas ((b), (d) e (f)) com o dedo com 90% do volume com a entrada binária de trajetória senoidal.



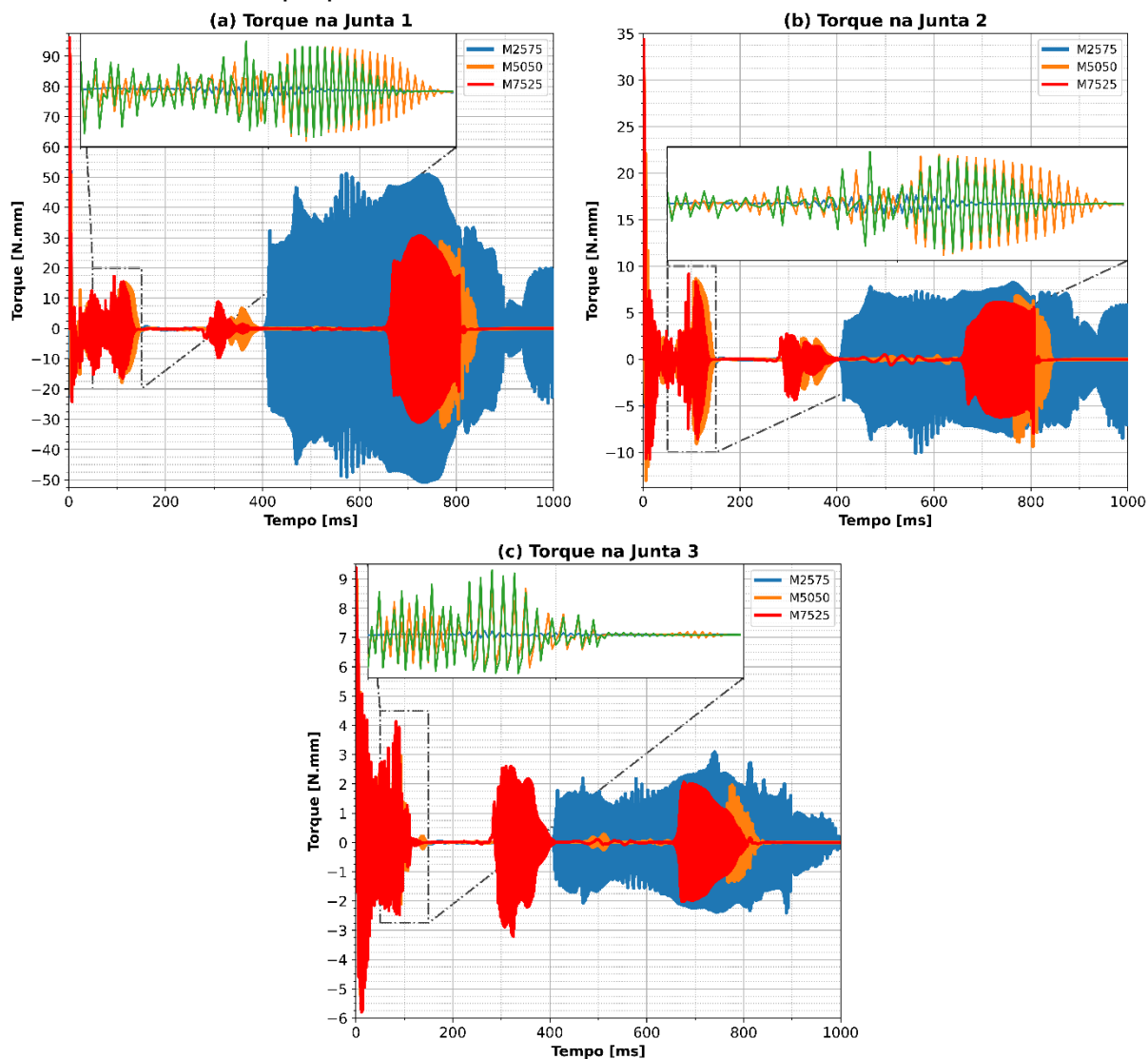
Fonte: Elaborado pelo autor.

Figura A.5 – Rotação angular relativa de cada uma das juntas ((a), (c) e (e)) e velocidades angulares relativas ((b), (d) e (f)) com o dedo com 90% do volume com a entrada binária de trajetória linear.



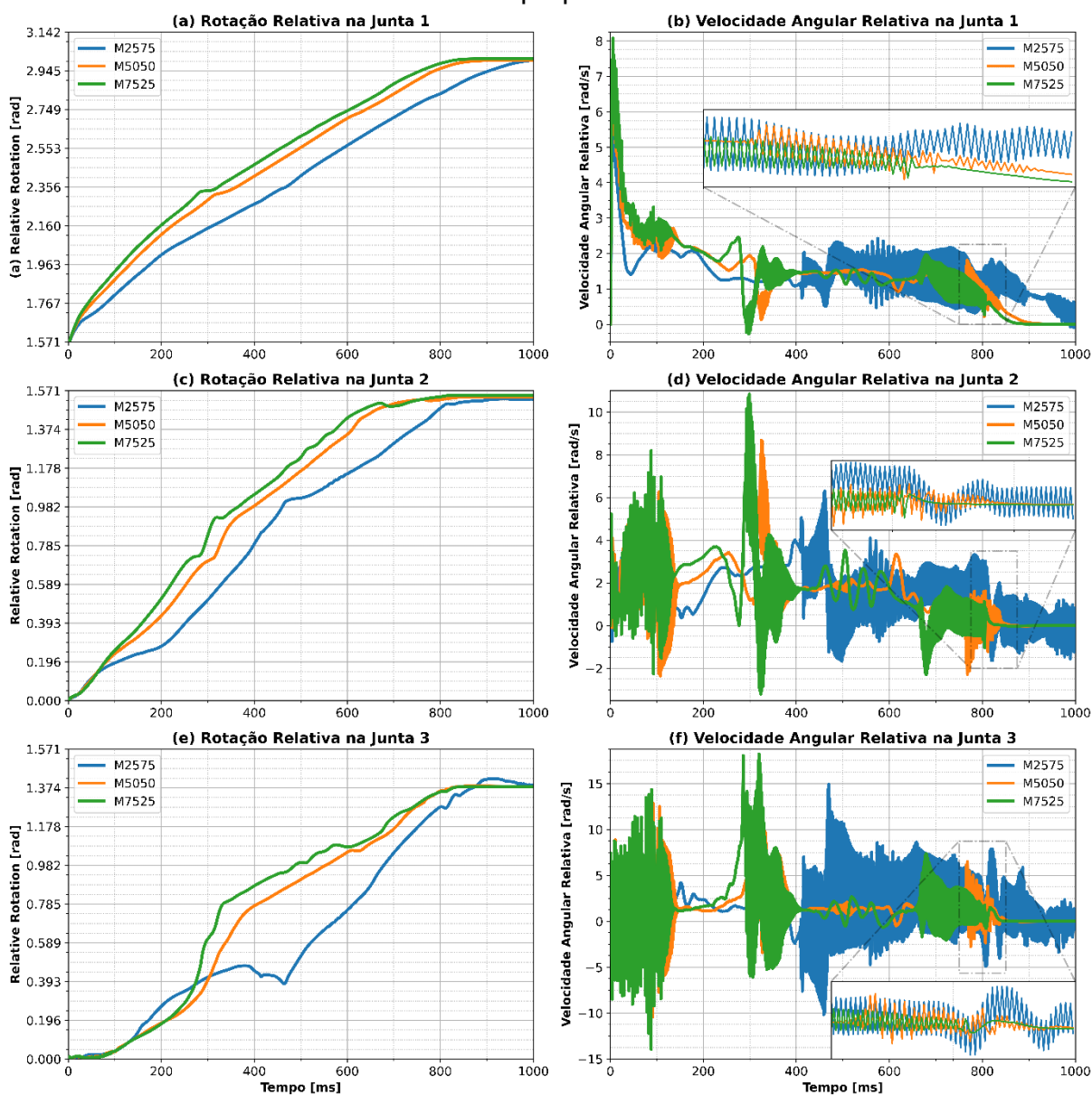
Fonte: Elaborado pelo autor.

Figura A.6 – Torque aplicado na junta 1 (a), 2 (b) e 3 (c) com as entradas proporcionais no dedo com 90% do volume.

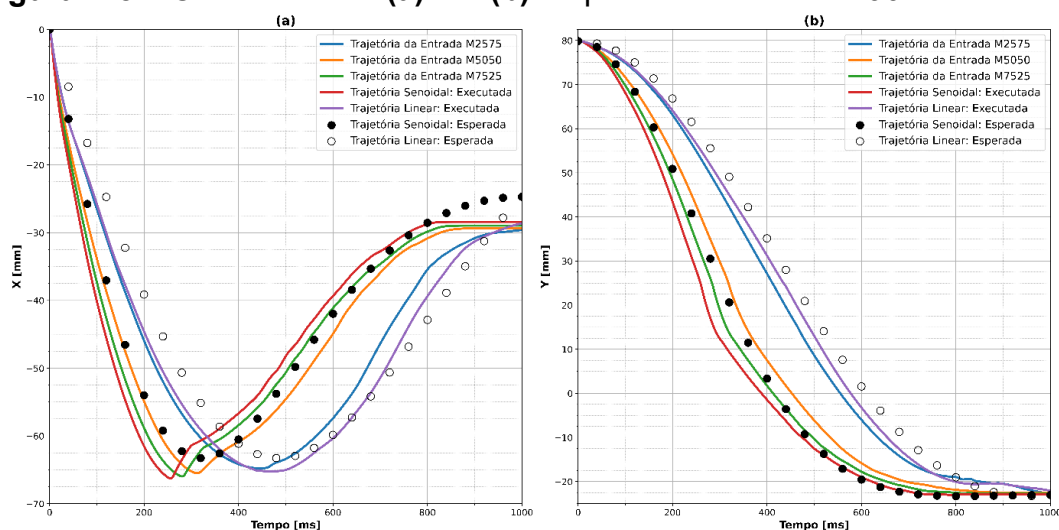


Fonte: Elaborado pelo autor.

Figura A.7 – Rotação angular relativa de cada uma das juntas ((a), (c) e (e)) e velocidades angulares relativa ((b), (d) e (f)) no dedo com 90% do volume com as entradas proporcionais.



Fonte: Elaborado pelo autor.

Figura A.8 – Coordenada X (a) e Y (b) da ponta do dedo com 90% do volume.

Fonte: Elaborado pelo autor.

Tabela A.1 – Magnitude média do erro em milímetros entre as trajetórias esperadas e as entradas proporcionais no dedo com 90% do volume.

Entrada	Trajetória				
	M2575	M5050	M7525	Linear	Senoidal
M2575	$3,345 \pm 1,690$	–	–	$4,431 \pm 1,949$	$17,03 \pm 8,398$
M5050	–	$5,026 \pm 2,480$	–	$14,73 \pm 7,657$	$6,193 \pm 2,924$
M7525	–	–	$4,396 \pm 2,741$	$18,50 \pm 10,05$	$3,356 \pm 1,529$

Fonte: Elaborado pelo autor.

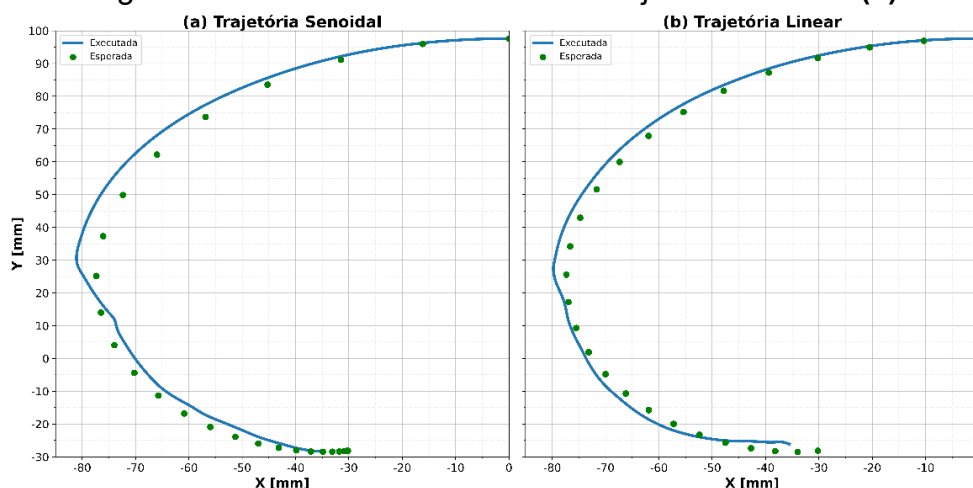
A.3) RESPOSTA DO CONTROLADOR ATUANDO O DEDO COM 110% DO VOLUME ORIGINAL

Os resultados obtidos com o controlador atuando no dedo com 110% do volume original são apresentados nessa sessão. Na Figura A.9 são apresentadas as trajetórias desenvolvidas pela ponta do dedo utilizando as entradas binárias senoidal e linear. Na Figura A.11 é apresentado o torque aplicado em cada uma das juntas, demonstrando o padrão oscilatório reduzido apenas para a entrada linear, como observado no modelo original apresentado na Sessão 3.2.1. As variáveis de estado para cada uma das entradas binárias são apresentadas na Figura A.12 e na Figura A.13.

Foi também avaliado as respostas do controlador para as entradas proporcionais M2575, M5050 e M7525. As trajetórias desenvolvidas são apresentadas na Figura A.10, o valor do torque empregado em cada junta na Figura A.14 e as variáveis de estado na Figura A.15. Foi também apresentado as coordenadas X e Y individualmente no tempo para cada uma das entradas em comparação com as coordenadas esperadas nas trajetória linear e senoidal apresentado na Figura A.16. Por fim é

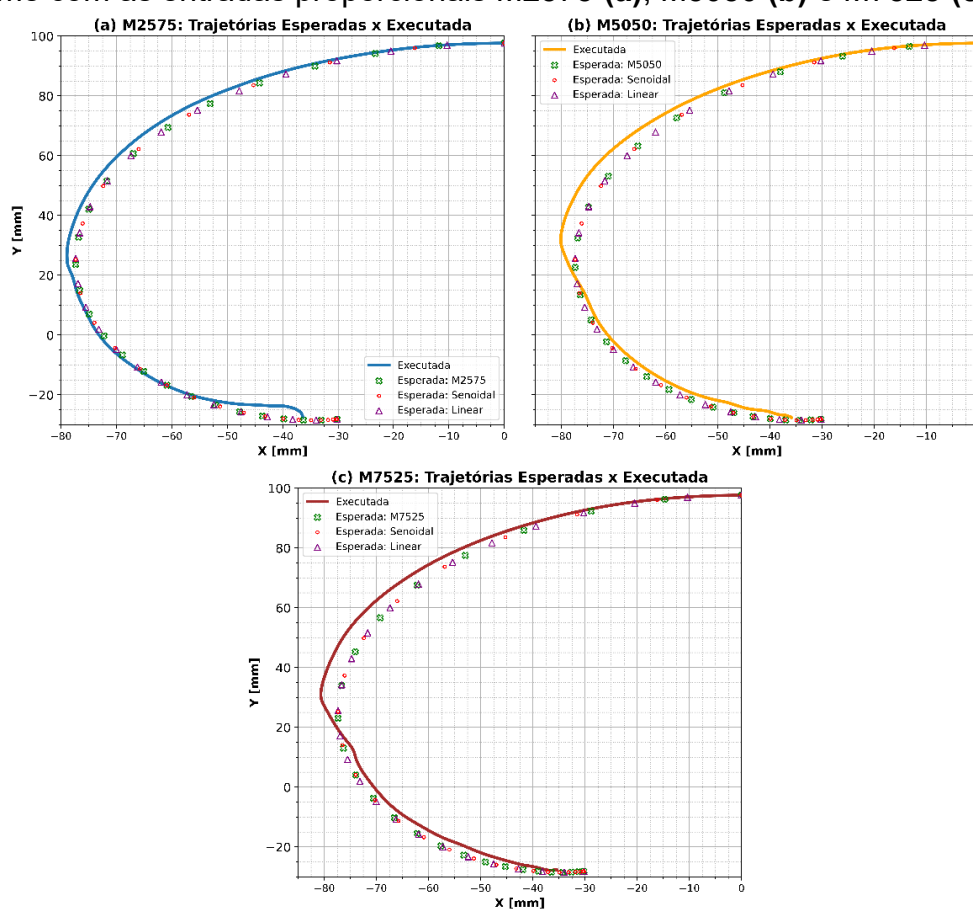
apresentado um comparativo entre a magnitude do erro médio de trajetória do modelo utilizando as entradas proporcionais e as trajetórias de treinamento (Tabela A.2).

Figura A.9 – Trajetória executado pelo controlador treinado com o dedo com 110% do volume original com as entradas binárias de trajetória senoidal (a) e linear (b).



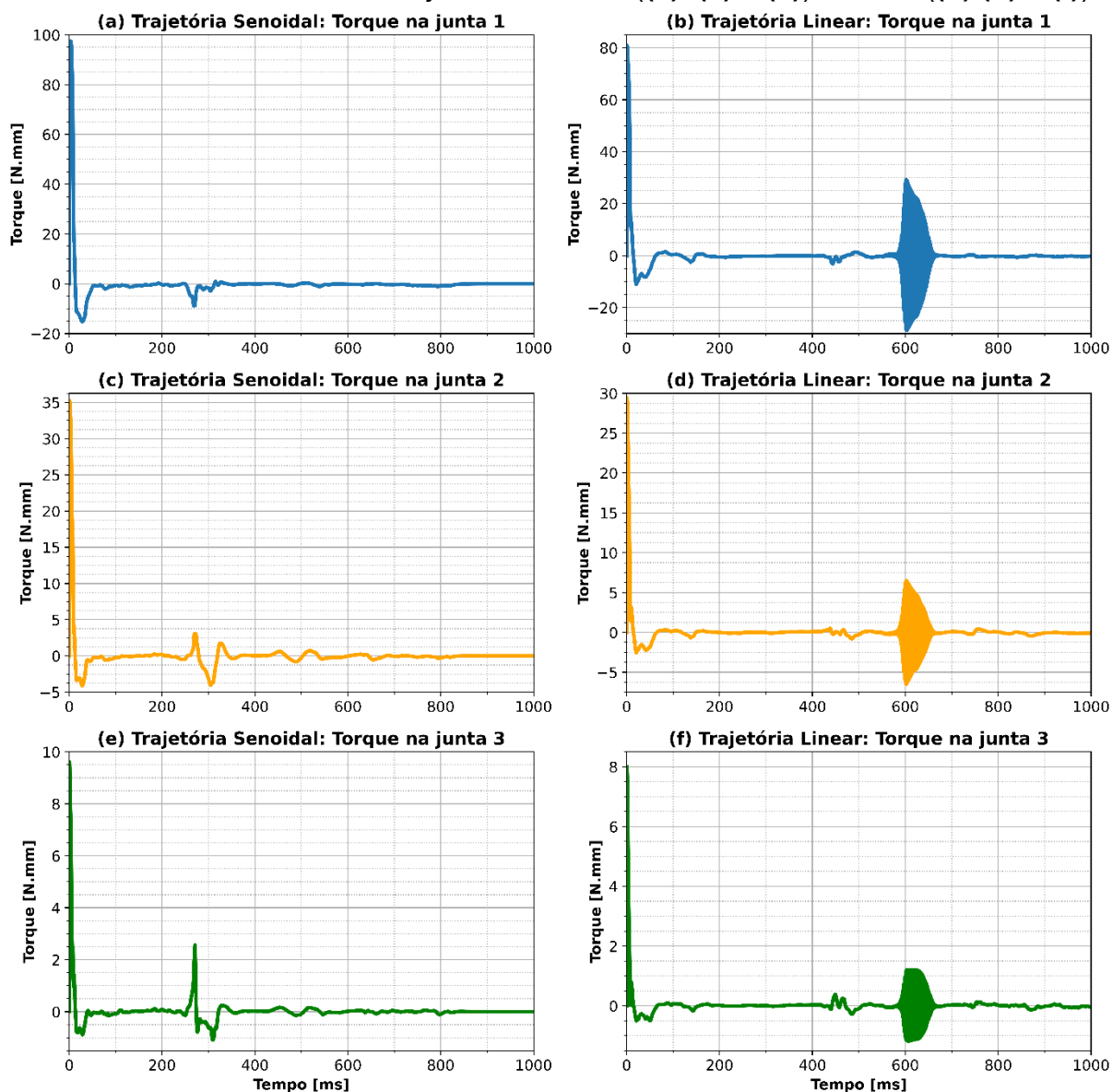
Fonte: Elaborado pelo autor.

Figura A.10 – Trajetória executada pelo controlador treinado com o dedo com 110% do volume com as entradas proporcionais M2575 (a), M5050 (b) e M7525 (c).



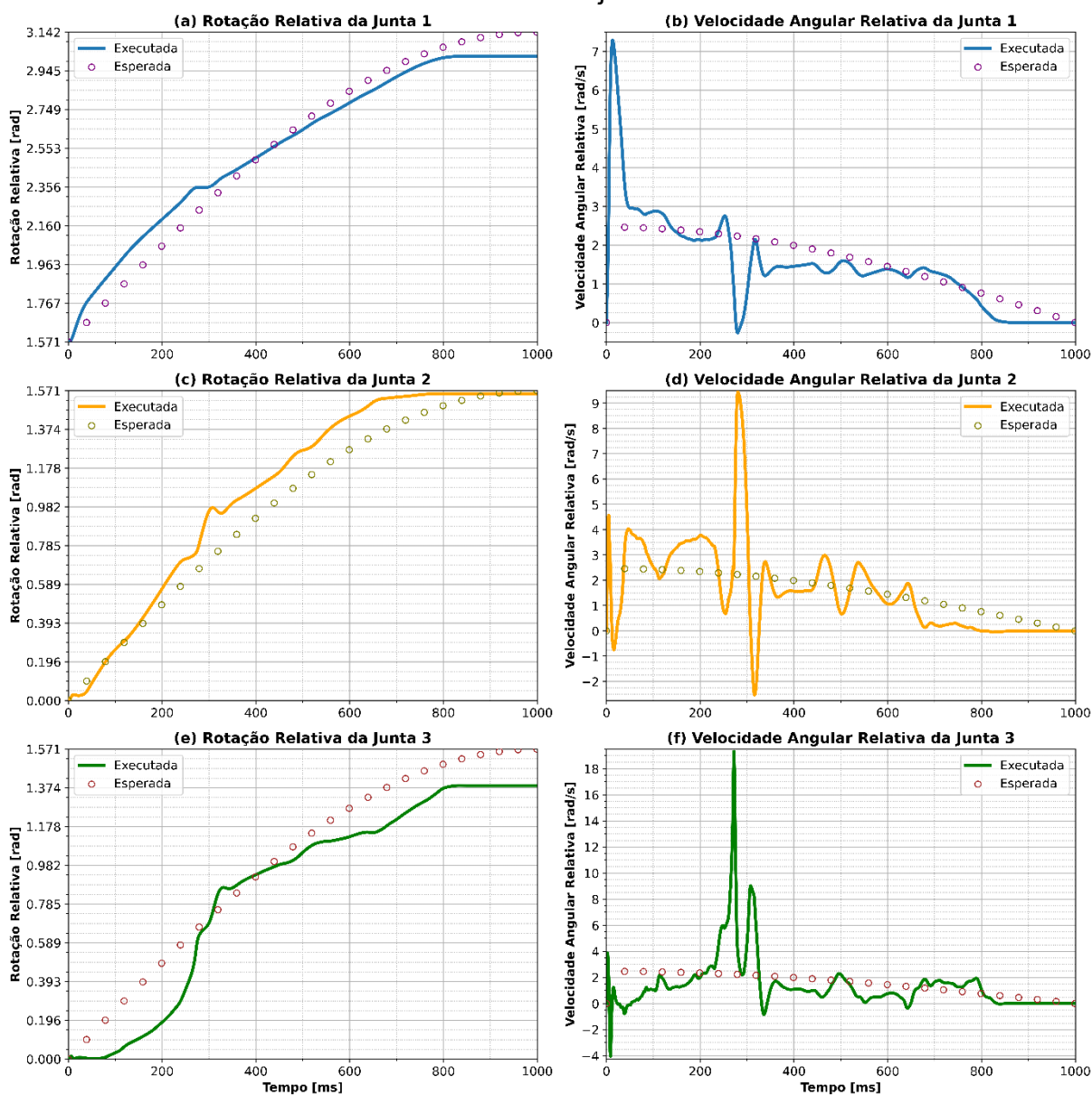
Fonte: Elaborado pelo autor.

Figura A.11 – Torque aplicados em cada uma das juntas do dedo com 110% do volume com as entradas de trajetória senoidal ((a), (c) e (e)) e linear ((b),(d) e (f)).



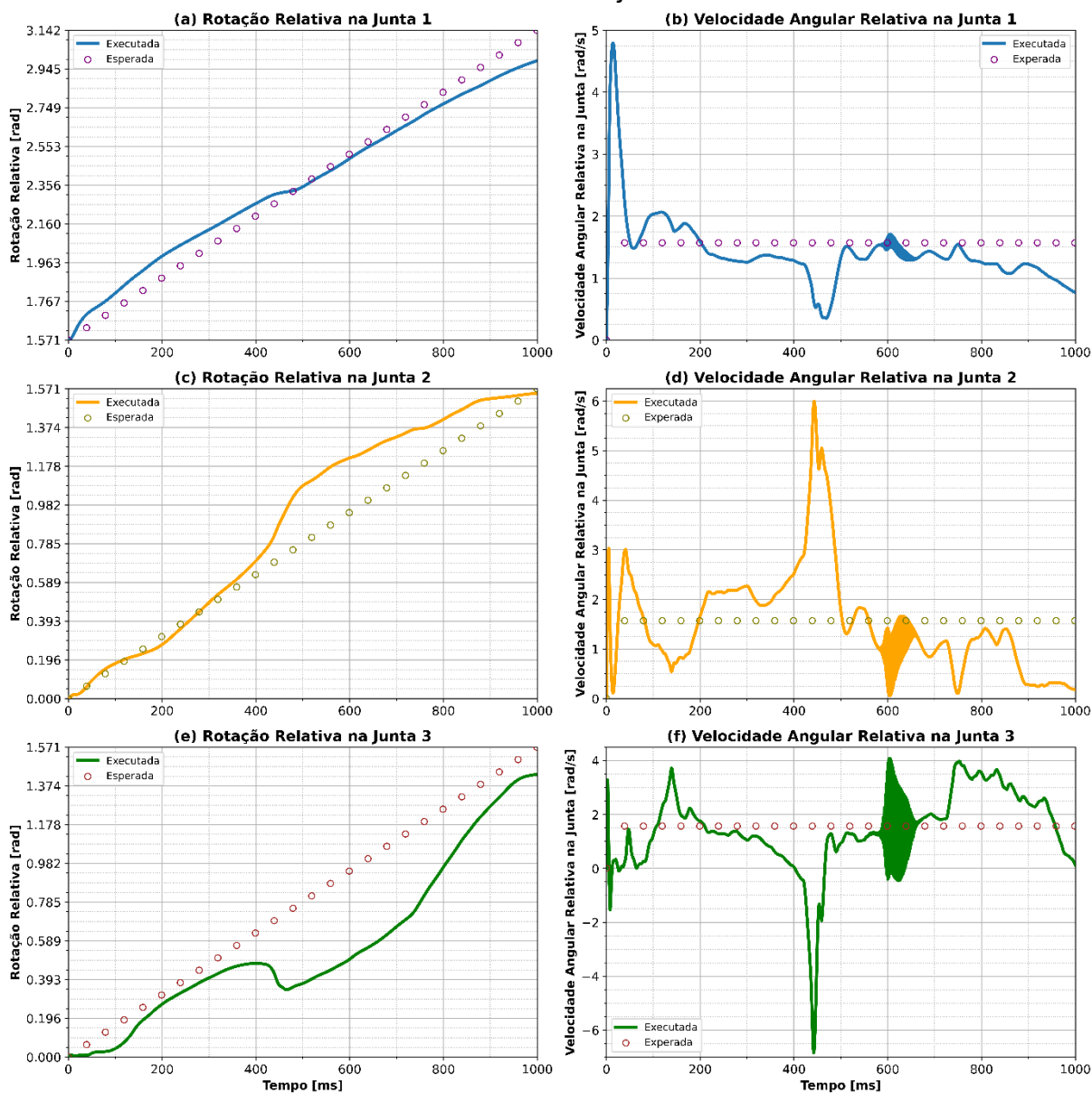
Fonte: Elaborado pelo autor.

Figura A.12 – Rotação angular relativa de cada uma das juntas ((a), (c) e (e)) e velocidades angulares relativas ((b), (d) e (f)) com o dedo com 110% do volume com a entrada binária de trajetória senoidal.



Fonte: Elaborado pelo autor.

Figura A.13 – Rotação angular relativa de cada uma das juntas ((a), (c) e (e)) e velocidades angulares relativas ((b), (d) e (f)) com o dedo com 110% do volume com a entrada binária de trajetória linear.



Fonte: Elaborado pelo autor.

Figura A.14 – Torque aplicado na junta 1 (a), 2 (b) e 3 (c) com as entradas proporcionais no dedo com 110% do volume.

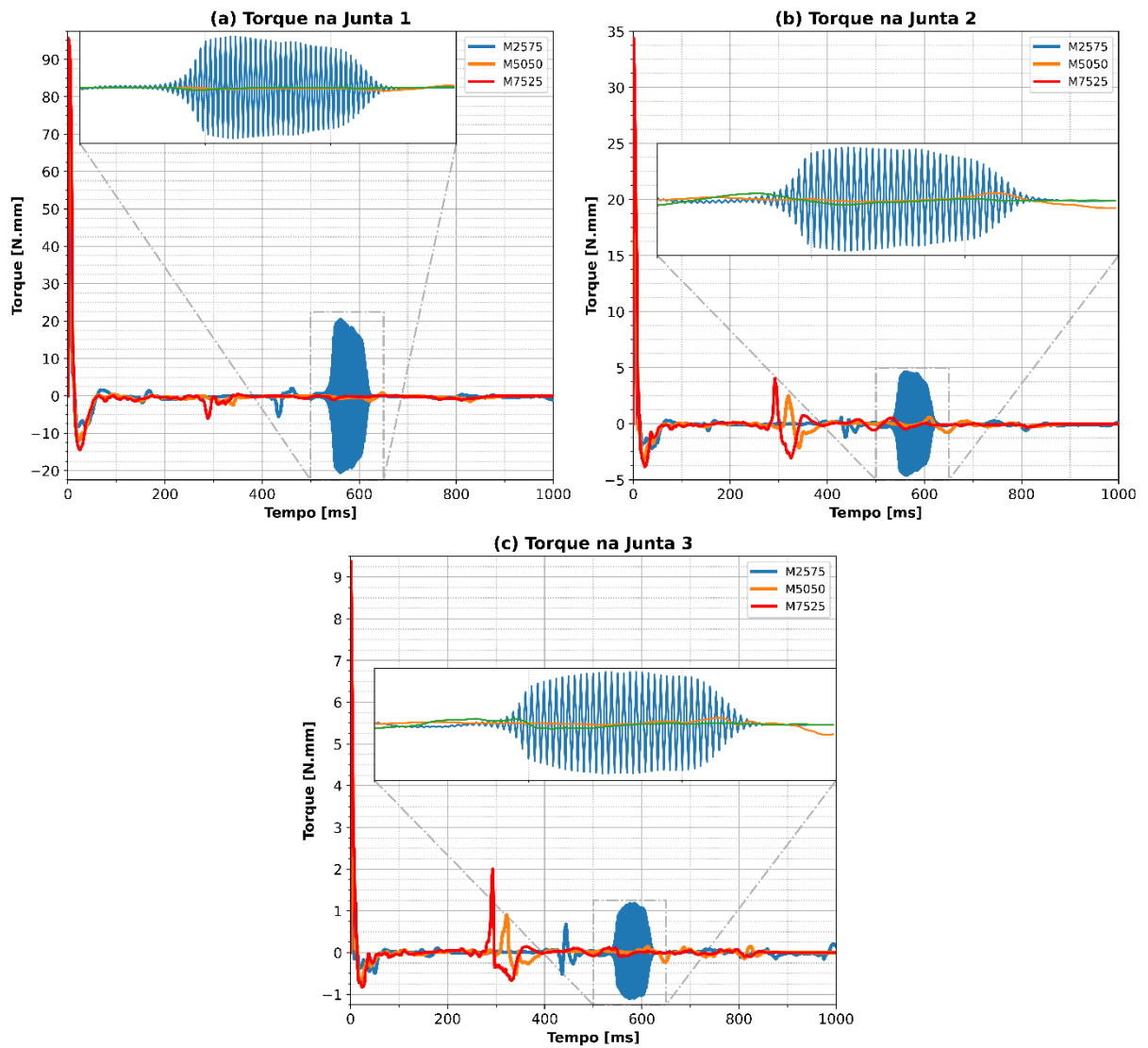
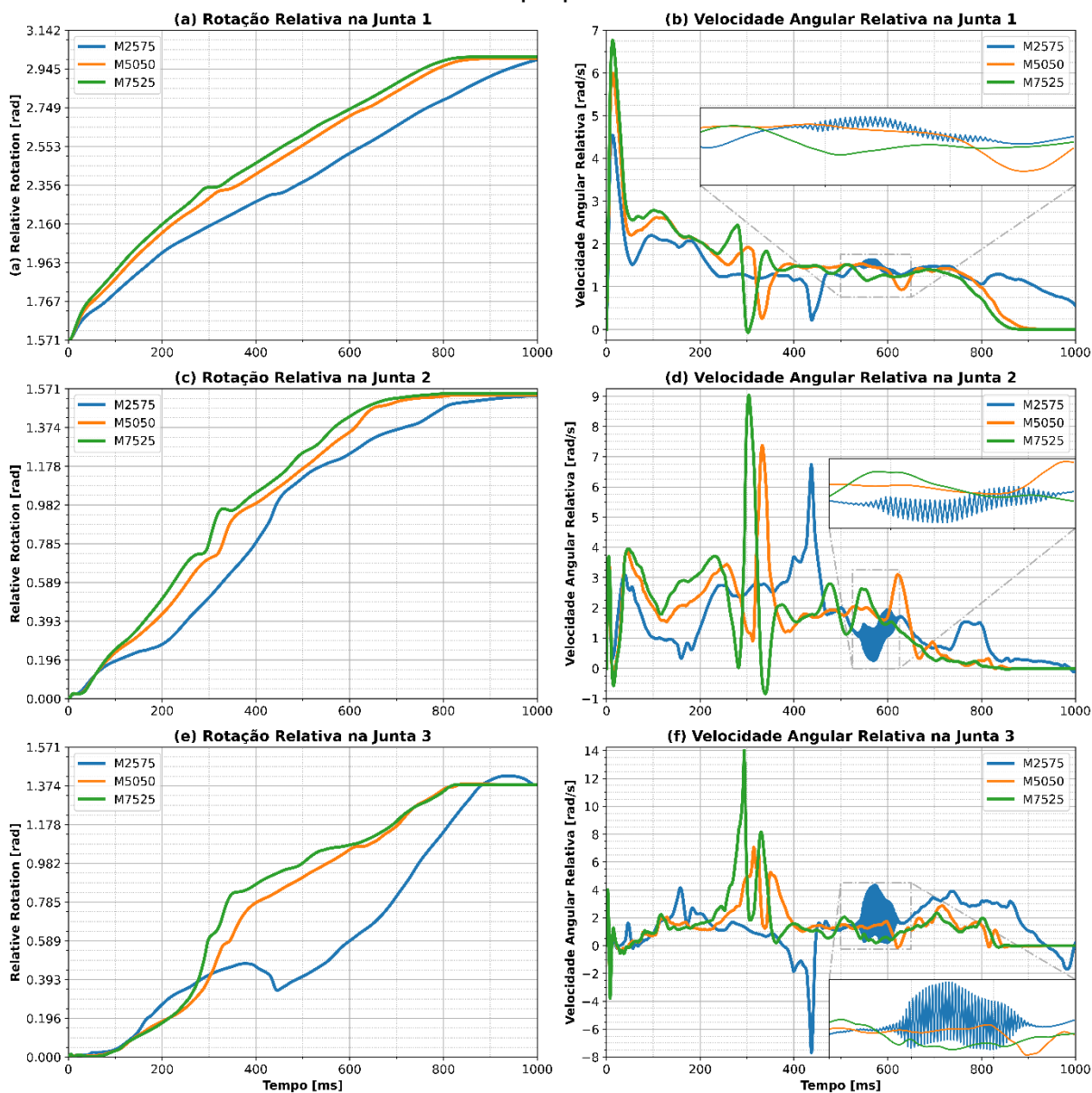
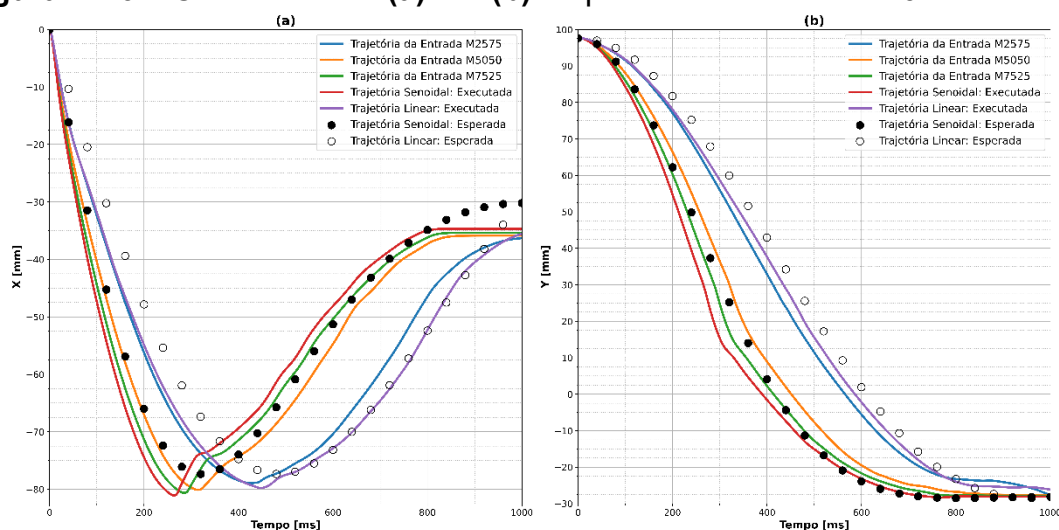


Figura A.15 – Rotação angular relativa de cada uma das juntas ((a), (c) e (e)) e velocidades angulares relativa ((b), (d) e (f)) no dedo com 110% do volume com as entradas proporcionais.



Fonte: Elaborado pelo autor.

Figura A.16 – Coordenada X (a) e Y (b) da ponta do dedo com 110% do volume.



Fonte: Elaborado pelo autor.

Tabela A.2 – Magnitude média do erro em milímetros entre as trajetórias esperadas e as entradas proporcionais no dedo com 110% do volume.

Entrada	Trajetória				
	M2575	M5050	M7525	Linear	Senoidal
M2575	$4,507 \pm 2,083$	–	–	$4,772 \pm 2,363$	$21,68 \pm 9,775$
M5050	–	$6,250 \pm 3,076$	–	$18,02 \pm 9,479$	$7,575 \pm 3,622$
M7525	–	–	$5,257 \pm 3,141$	$22,35 \pm 12,20$	$4,152 \pm 2,103$

Fonte: Elaborado pelo autor.

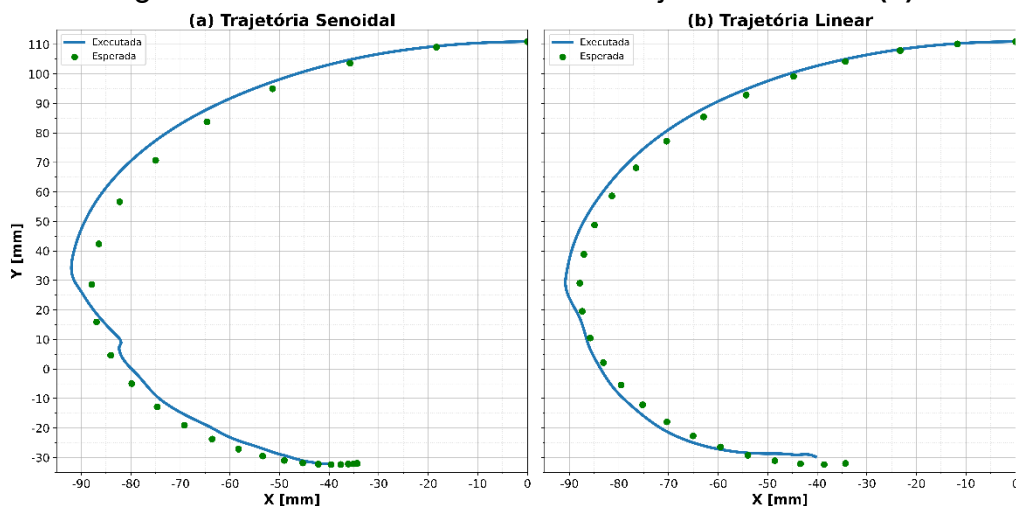
A.4) RESPOSTA DO CONTROLADOR ATUANDO O DEDO COM 125% DO VOLUME ORIGINAL

Os resultados obtidos com o controlador atuando no dedo com 125% do volume original são apresentados nessa sessão. Na Figura A.17 são apresentadas as trajetórias desenvolvidas pela ponta do dedo utilizando as entradas binárias senoidal e linear. Na Figura A.19 é apresentado o torque aplicado em cada uma das juntas, demonstrando que com o aumento da massa o padrão oscilatório não é mais utilizado. As variáveis de estado para cada uma das entradas binárias são apresentadas na Figura A.20 e na Figura A.21.

Foi também avaliado as respostas do controlador para as entradas proporcionais M2575, M5050 e M7525. As trajetórias desenvolvidas são apresentadas na Figura A.18, o valor do torque empregado em cada junta na Figura A.22 e as variáveis de estado na Figura A.23. Foi também apresentado as coordenadas X e Y individualmente no tempo para cada uma das entradas em comparação com as coordenadas esperadas nas trajetória linear e senoidal apresentado na Figura A.24. Por fim é

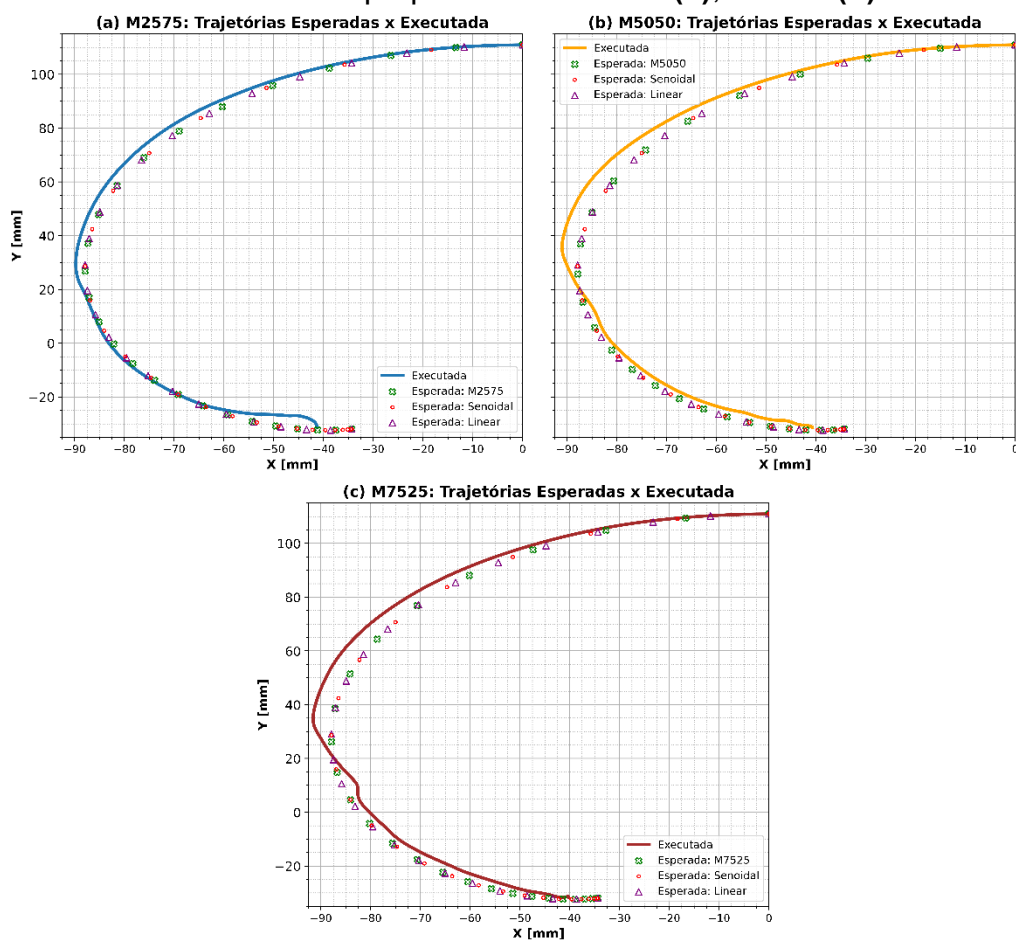
apresentado um comparativo entre a magnitude do erro médio de trajetória do modelo utilizando as entradas proporcionais e as trajetórias de treinamento (Tabela A.3).

Figura A.17 – Trajetória executado pelo controlador treinado com o dedo com 125% do volume original com as entradas binárias de trajetória senoidal (a) e linear (b).



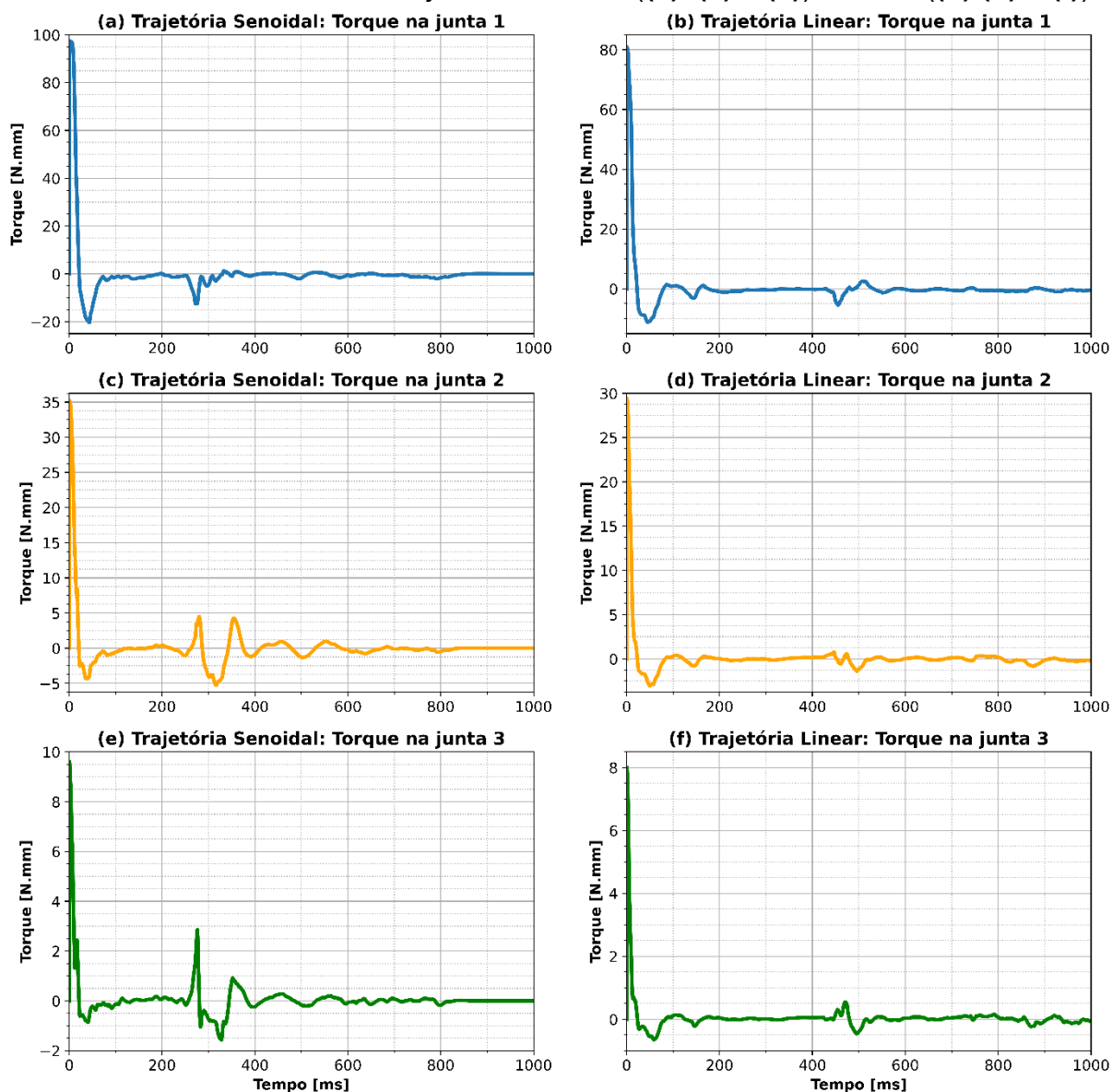
Fonte: Elaborado pelo autor.

Figura A.18 – Trajetória executada pelo controlador treinado com o dedo com 125% do volume com as entradas proporcionais M2575 (a), M5050 (b) e M7525 (c).



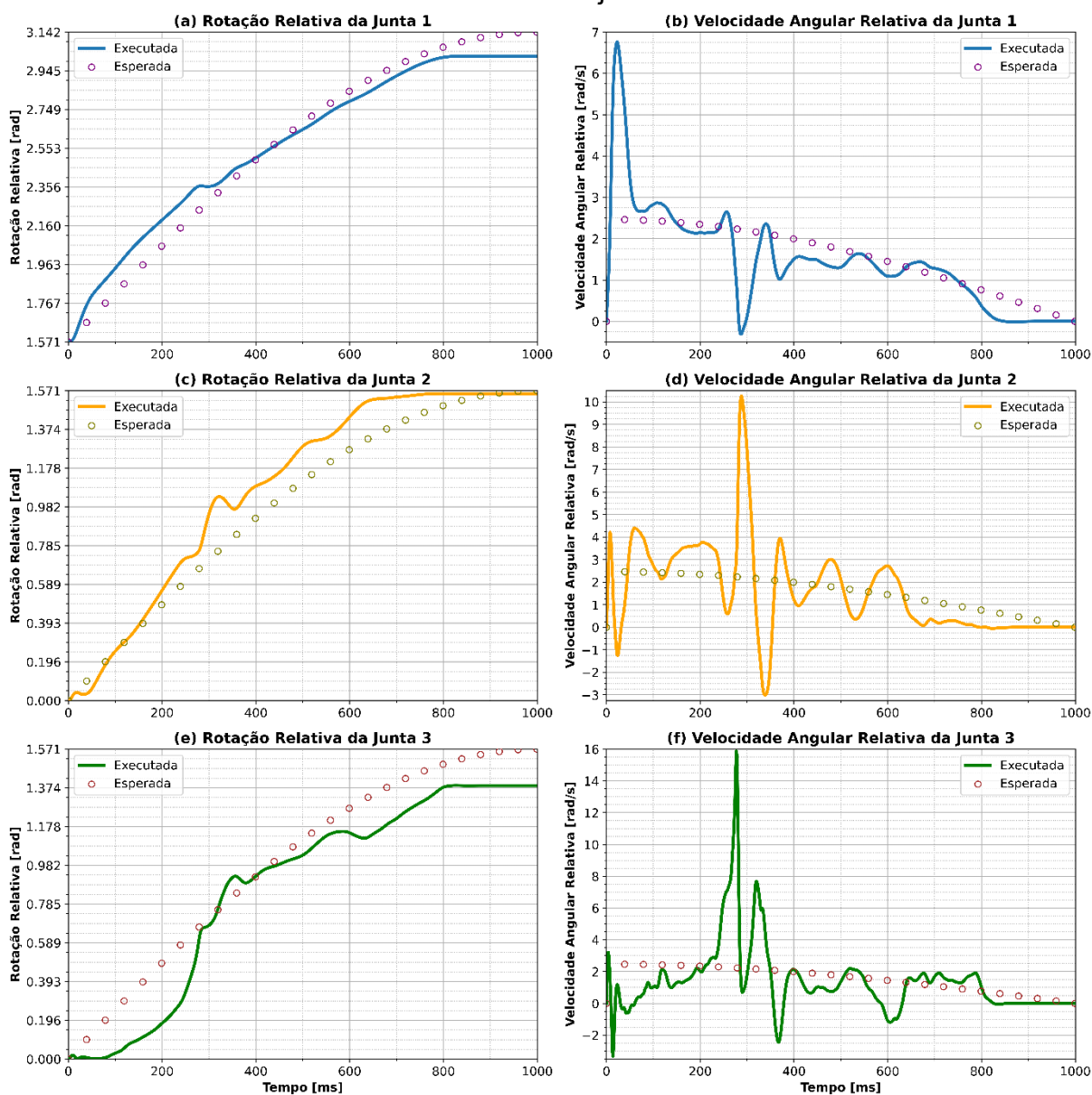
Fonte: Elaborado pelo autor.

Figura A.19 – Torque aplicados em cada uma das juntas do dedo com 125% do volume com as entradas de trajetória senoidal ((a), (c) e (e)) e linear ((b),(d) e (f)).



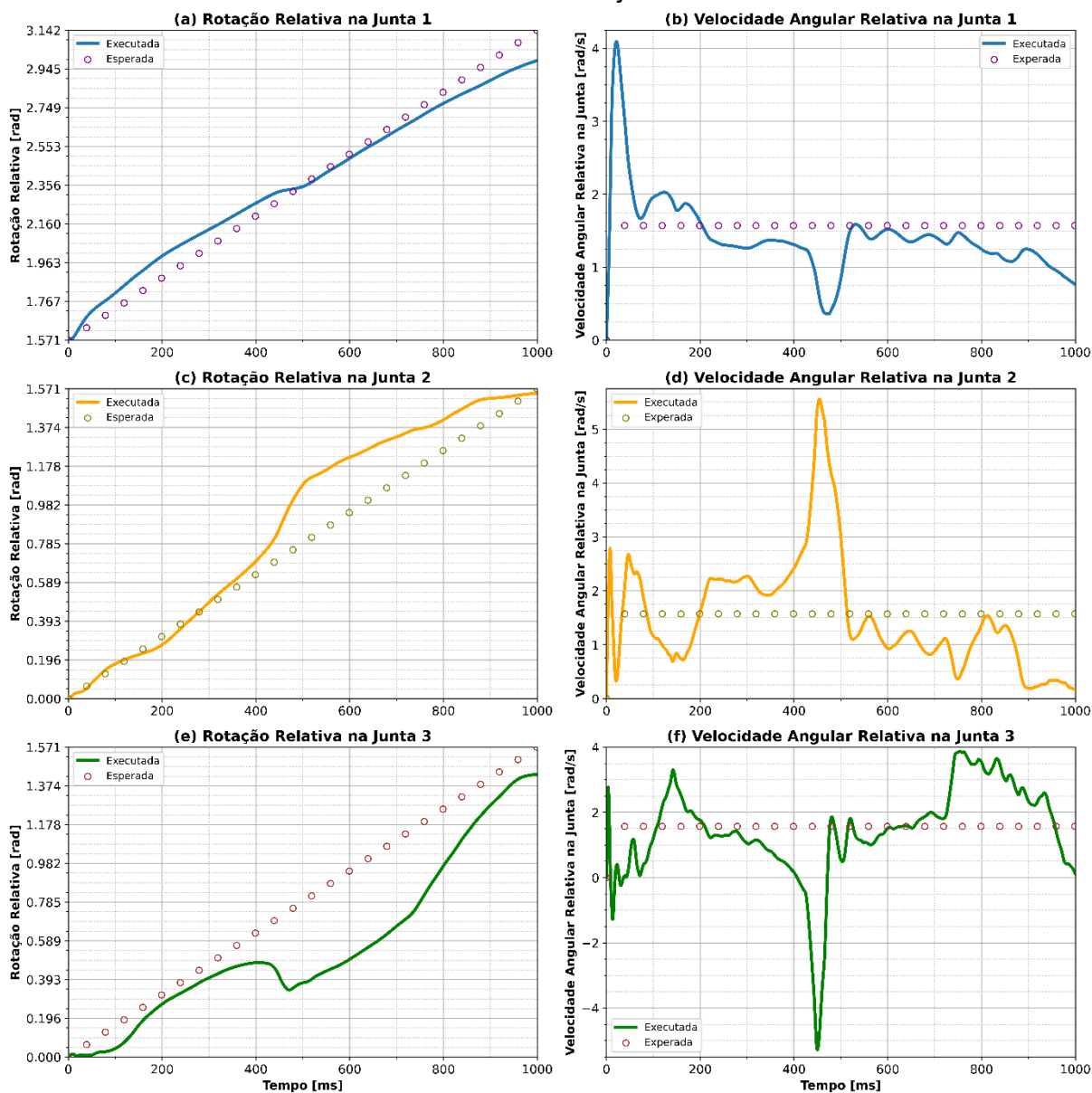
Fonte: Elaborado pelo autor.

Figura A.20 – Rotação angular relativa de cada uma das juntas ((a), (c) e (e)) e velocidades angulares relativas ((b), (d) e (f)) com o dedo com 125% do volume com a entrada binária de trajetória senoidal.



Fonte: Elaborado pelo autor.

Figura A.21 – Rotação angular relativa de cada uma das juntas ((a), (c) e (e)) e velocidades angulares relativa ((b), (d) e (f)) com o dedo com 125% do volume com a entrada binária de trajetória linear.



Fonte: Elaborado pelo autor.

Figura A.22 – Torque aplicado na junta 1 (a), 2 (b) e 3 (c) com as entradas proporcionais no dedo com 125% do volume.

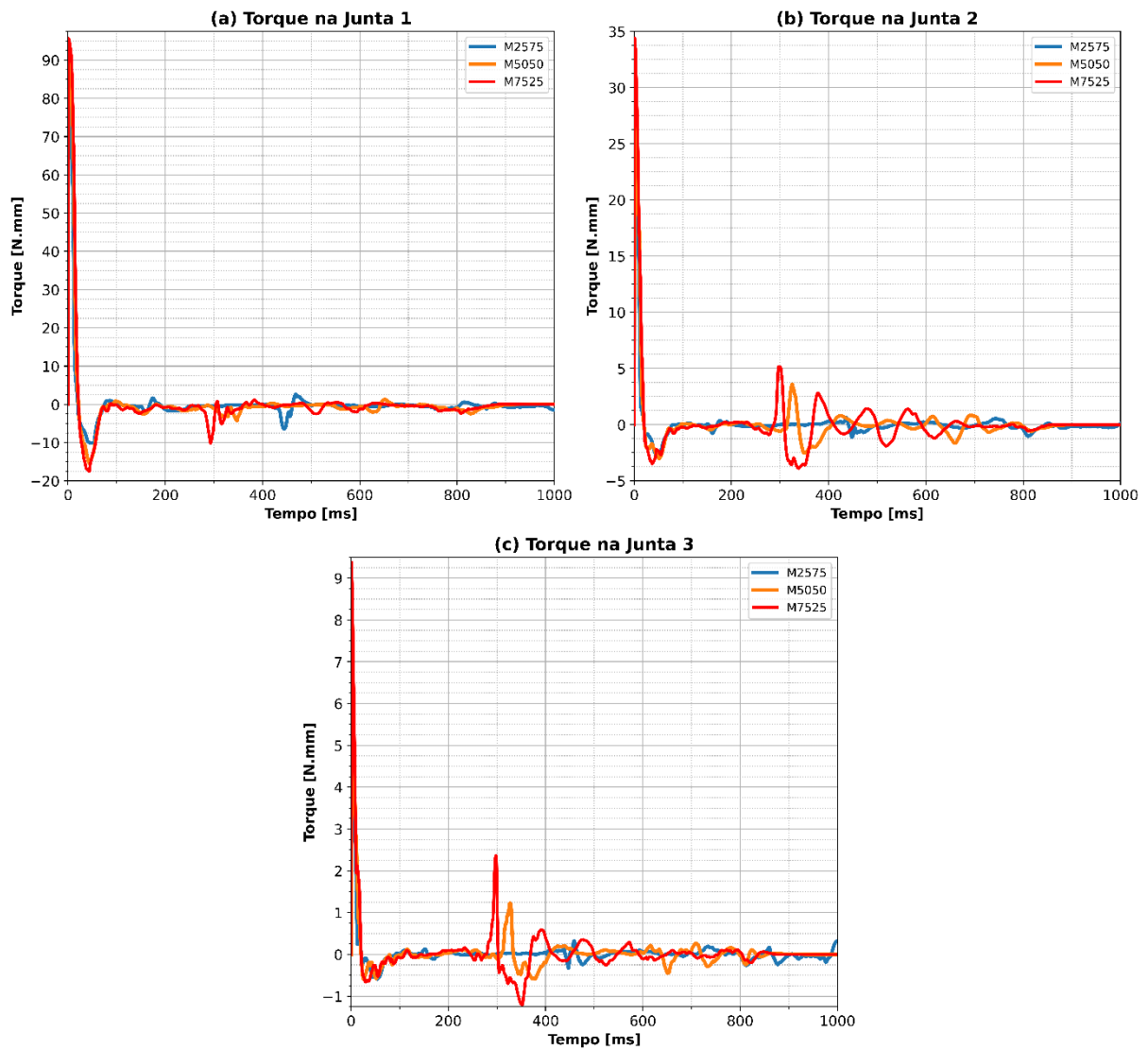
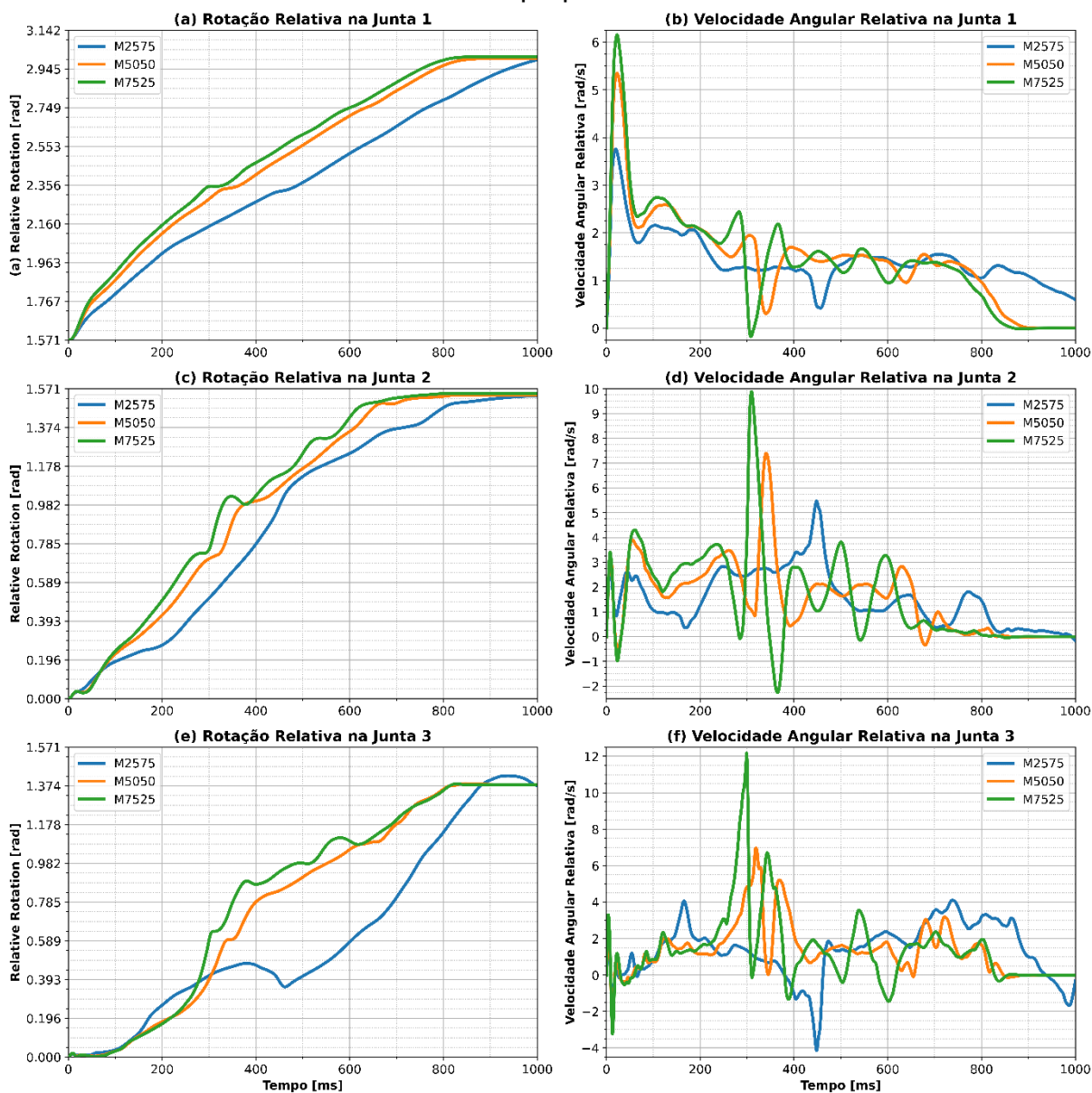
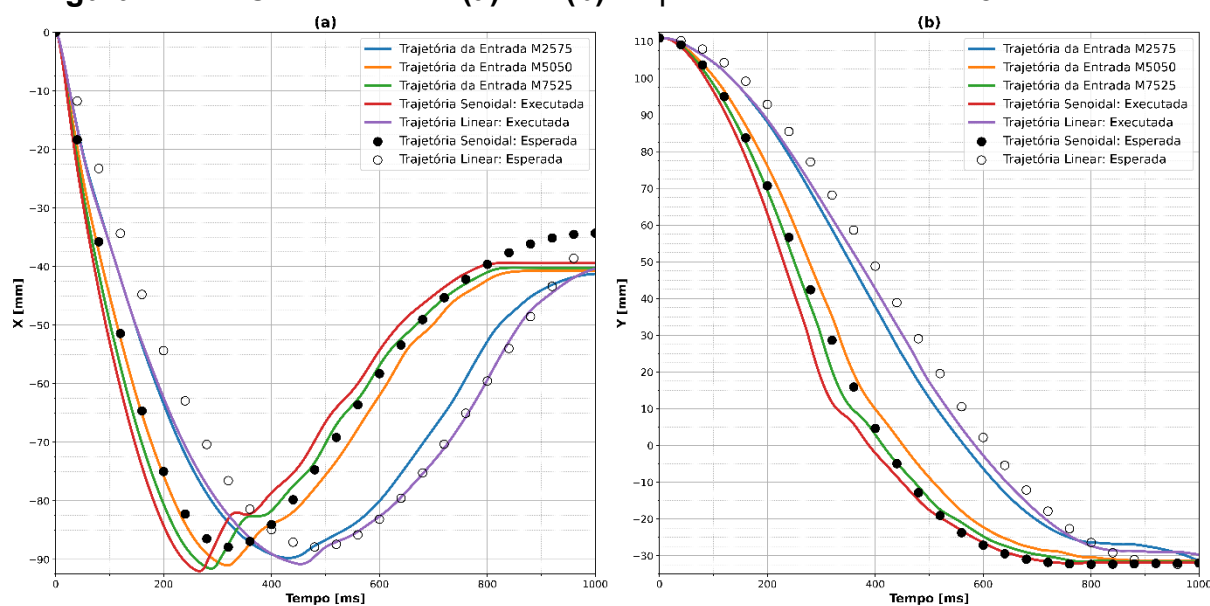


Figura A.23 – Rotação angular relativa de cada uma das juntas ((a), (c) e (e)) e velocidades angulares relativa ((b), (d) e (f)) no dedo com 125% do volume com as entradas proporcionais.



Fonte: Elaborado pelo autor.

Figura A.24 – Coordenada X (a) e Y (b) da ponta do dedo com 125% do volume.



Fonte: Elaborado pelo autor.

Tabela A.3 – Magnitude média do erro em milímetros entre as trajetórias esperadas e as entradas proporcionais no dedo com 125% do volume.

Entrada	Trajetória				
	M2575	M5050	M7525	Linear	Senoidal
M2575	$2,298 \pm 2,471$	–	–	$5,311 \pm 2,263$	$24,93 \pm 11,04$
M5050	–	$7,056 \pm 3,462$	–	$20,27 \pm 10,81$	$8,873 \pm 4,305$
M7525	–	–	$5,965 \pm 3,511$	$25,18 \pm 13,91$	$4,844 \pm 2,693$

Fonte: Elaborado pelo autor.