

TRABALHO DE GRADUAÇÃO

**CLASSIFICAÇÃO DE MOVIMENTOS
DE MÃO PARA PRÓTESE ROBÓTICA
UTILIZANDO SINAIS MIO-ELÉTRICOS**

Raphael Rodrigues da Costa Barbosa

Brasília, Dezembro de 2021



**ENGENHARIA
MECATRÔNICA**
UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia
Curso de Graduação em Engenharia de Controle e Automação

TRABALHO DE GRADUAÇÃO

CLASSIFICAÇÃO DE MOVIMENTOS
DE MÃO PARA PRÓTESE ROBÓTICA
UTILIZANDO SINAIS MIO-ELÉTRICOS

Raphael Rodrigues da Costa Barbosa

*Relatório submetido como requisito parcial de obtenção
de grau de Engenheiro de Controle e Automação*

Banca Examinadora

Prof. Dr. Roberto Baptista, FGA/UnB

Orientador

Prof. Mariana Costa Bernardes Matias,

FGA/UnB

Examinador interno

Prof. Claudia Patricia Ochoa Diaz, FGA/UnB

Examinador interno

Brasília, Dezembro de 2021

FICHA CATALOGRÁFICA

RODRIGUES DA COSTA BARBOSA, RAPHAEL

Classificação de Movimentos de Mão para Prótese Robótica Utilizando Sinais Mio-elétricos.

[Distrito Federal] 2021.

xiii, 54p., 297 mm (FT/UnB, Engenheiro, Controle e Automação, 2021). Trabalho de Graduação – Universidade de Brasília. Faculdade de Tecnologia.

1.Prótese de Braço

2.Classificação

3.HMM

4.EMG

I. Mecatrônica/FT/UnB

II. Título (Série)

REFERÊNCIA BIBLIOGRÁFICA

BARBOSA, RAPHAEL. R. C., (2021). Classificação de Movimentos de Mão para Prótese Robótica Utilizando Sinais Mio-elétricos. Trabalho de Graduação em Engenharia de Controle e Automação, Publicação FT.TG-*n*°06, Faculdade de Tecnologia, Universidade de Brasília, Brasília, DF, 67p.

CESSÃO DE DIREITOS

AUTOR: Raphael Rodrigues da Costa Barbosa

TÍTULO DO TRABALHO DE GRADUAÇÃO: Classificação de Movimentos de Mão para Prótese Robótica Utilizando Sinais Mio-elétricos.

GRAU: Engenheiro

ANO: 2021

É concedida à Universidade de Brasília permissão para reproduzir cópias deste Trabalho de Graduação e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte desse Trabalho de Graduação pode ser reproduzida sem autorização por escrito do autor.

Raphael Rodrigues da Costa Barbosa.

SQSW 301, Bloco B, apartamento 216, Bairro Sudoeste.

70673-102 Brasília – DF – Brasil.

Dedicatória

Dedico este trabalho à minha família, amigos e a todos que me apoiaram nesta jornada

Raphael Rodrigues da Costa Barbosa

Agradecimentos

Nada nessa vida nada se faz sozinho, e este trabalho de Graduação não é exceção. A ajuda, suporte, apoio de diversas pessoas foi essencial ao desenvolvimento deste trabalho.

Primeiro agradeço à minha mãe, Eliana Rodrigues, que sempre deu apoio e ajudou nos momentos mais difíceis, dando sempre tudo que podia apesar de todos os problemas que viessem acontecer, e pela paciência e compreensão. Um grande exemplo de uma pessoa batalhadora e resiliente. Juntamente, agradeço ao meu pai, Renan Barbosa, e ao meu irmão, Gabriel Rodrigues, por me acompanharem neste momento, e darem todo apoio que foi necessário, e a toda a minha família que estão sempre dando o suporte que podem.

Agradeço aos meus amigos e amigas por compartilharem dos desafios que impõem a vida acadêmica, pelo espírito colaborativo, e por proporcionarem momentos importantes de respiro que foram essenciais para que o foco não fosse perdido.

Também agradeço ao meu orientador, pela paciência e compreensão, pelas valiosas contribuições dadas durante todo o processo, e por dar todo o apoio que foi necessário. Estendo também o agradecimento a todos outros alunos que participam do projeto e do laboratório que deram contribuições, e aconselhamentos essenciais.

A todos aqueles que trabalham em serviços essenciais durante um desafio inédito que é a pandemia de covid-19. Sem essas pessoas, nada seria possível.

E por fim agradeço à Universidade de Brasília, ao seu corpo docente e funcionários todos o do meu curso pela elevada qualidade do ensino oferecido.

Raphael Rodrigues da Costa Barbosa

RESUMO

Apesar dos avanços em tecnologia assistiva pelo mundo, o preço destas tecnologias ainda é uma barreira que os mantém longe da maioria da população. Muitas das próteses tecnológicas disponíveis comercialmente são muito caras para aqueles que necessitam. O desenvolvimento de próteses robóticas visa criar dispositivos protéticos capazes de replicar movimentos de mão de uma pessoa. Por meio de sinais de Eletromiografia de Superfície (sEMG), será proposto um classificador capaz de diferenciar movimentos de mão. Foram pesquisados diversos métodos de classificação existentes na literatura para dar a este trabalho uma boa base de onde começar. Foi escolhido um modelo de classificação baseado nos Modelos Ocultos de Markov, onde, dado um conjunto de sinais, o modelo é capaz de prever qual a sequência de movimentos de mão que foi realizado. Decidiu-se classificar 3 gestos de mão que envolvem contrações estáticas dos músculos envolvidos. Sinais são coletados dos 3 gestos, e deles são extraídas características, a partir do modelo Auto-Regressivo e dos valores RMS do sinal. Um conjunto dos dados é utilizado para treinamento e adequação do modelo. Tendo um modelo estimado, foi possível utilizar outro conjunto de dados para testar e avaliar o desempenho do classificador, que foi acima de 90% para os três gestos. Dessa forma, o classificador foi capaz de separar bem os estados, porém, o universo de movimentos é de apenas 3 gestos, o que é bem reduzido, e trabalhos futuros devem implementar mais gestos incluindo movimentos onde os dados inerciais poderão ser utilizados.

Palavras Chave: Prótese, Sinais Mio-elétricos, Controle. Modelo AutoRegressivo, Modelos Ocultas de Markov, Classificação, Reconhecimento de Padrões.

ABSTRACT

Although is seen through out the world various advancements in assistive technologys, its price still a barrier that keeps these technologies from the majority of the population. Many of the high-tech prosthesis commertially available are to expensive for those who need. The development of robotic prosthetics aims to create prosthetic devices that have the capability of recriating hand movements of somebody. Using myoelectric signals, it's proposed a classifier capable of distinguishing different hand movements. Various existing methods were researched in the literature to give a solid starting point to begin with. It was chosen a classifier model based on Hidden Markov Models, in which, given a set of data signals, the model is capable of predicting which sequence of hand movements were performed. It was decided to classify 3 static

hand gestures. From those signals, features are extracted using the Auto Regressive Models and the RMS value of the data. A set of this data is used for training and fitting of the model. Given this estimated model, it is possible to utilize a dataset for testing and evaluating the classifier performance, that was over 90% for the 3 hand gestures. In this way, the classifier was able to separate the statesm although a small number of hand gestures are being classified. It is inportant in future implementations to classify more movements, including ones that will utilize inertial measurement units.

Keywords: Hand Prosthesis, EMG signal, Control, AutoRegressive Model, Hidden Markov Models, Classifier, Pattern Recognition.

SUMÁRIO

1	Introdução	1
1.1	CONTEXTUALIZAÇÃO.....	1
1.2	DEFINIÇÃO DO PROBLEMA	3
1.3	OBJETIVOS DO PROJETO	3
1.4	RESULTADOS OBTIDOS	4
1.5	APRESENTAÇÃO DO MANUSCRITO.....	4
2	Fundamentos	5
2.1	INTRODUÇÃO.....	5
2.2	REVISÃO BIBLIOGRÁFICA	5
2.2.1	PRÓTESES EXISTENTES NO MERCADO	5
2.2.2	ABORDAGENS EXISTENTES NA LITERATURA	7
2.3	SENSORIAMENTO	11
2.3.1	SENSORES INERCIAIS (IMU)	11
2.3.2	SENSORES DE ELETROMIOGRAFIA DE SUPERFÍCIE (sEMG).....	12
2.4	RECONHECIMENTO DE PADRÕES.....	15
2.4.1	SEGMENTAÇÃO DOS DADOS.....	16
2.4.2	EXTRAÇÃO DE CARACTERÍSTICAS	17
3	Métodos	28
3.1	APARATO INSTRUMENTAL	28
3.1.1	DELSYS - TRIGNO AVANTI.....	28
3.2	IMPLEMENTAÇÃO	30
3.3	AQUISIÇÃO DE DADOS.....	30
3.4	PRÉ-PROCESSAMENTO.....	32
3.5	SEGMENTAÇÃO DOS DADOS.....	32
3.6	EXTRAÇÃO DAS CARACTERÍSTICAS.....	33
3.7	CLASSIFICAÇÃO	34
4	Resultados	36
4.1	RESULTADOS E DISCUSSÃO.....	36
5	Conclusão	44

5.1	CONCLUSÕES E TRABALHOS FUTUROS.....	44
	REFERÊNCIAS BIBLIOGRÁFICAS	46
	Anexos.....	49
I	Códigos e Programas utilizados	50

LISTA DE FIGURAS

2.1	Prótese <i>Hero Arm</i> [1]	6
2.2	Prótese <i>Dextrus Hand</i> [2]	6
2.3	Prótese <i>HackBerry</i>	7
2.4	Esquema Mostrando a posição da IMU no braço, movimentos do braço e ângulos relevantes. Imagem de [3]	8
2.5	Característica de filtro passa-banda com efeito <i>ripple</i> . Imagem adaptada de [4]	15
2.6	Filtro Passa-Baixa de ordem 2. Imagem adaptada de [4]	15
2.7	Sequência de passos para Reconhecimento de Padrões	16
2.8	Separação de uma janela contendo somente uma contração que dura 5s	17
2.9	Esquemas de janelamento com e sem sobreposição. Imagem adaptada de [5]	17
2.10	Cadeia de Markov com dois estados	21
2.11	Ilustração da sequência de eventos levando as condições de 2.28. Imagem obtida de [6]	24
3.1	Equipamentos para aquisição de dados da Delsys Inc. Os equipamentos incluem base de carga, dois sensores e um <i>tablet</i> com aplicações móveis para uso dos sensores.	29
3.2	Posicionamento dos Sensores	30
3.3	Gestos de mão para serem classificados. Respectivamente repouso, mão fechada e extensão do punho	31
3.4	Protocolo para aquisição de sinais	31
3.5	Sinal EMG para procedimento de mão fechada para ambos os canais EMG. Observa-se 10 repetições do gesto	32
3.6	Gestos gerando um sinal eletromiográfico, separado em observações	34
4.1	Sinal EMG para procedimento de mão fechada para ambos os canais EMG. Observa-se 10 repetições do gesto. Canal 1 em azul e Canal 2 em verde	37
4.2	Sinal EMG para procedimento de flexão do punho para ambos os canais EMG. Observa-se 10 repetições do gesto. Canal 1 em azul e Canal 2 em verde	37
4.3	Vetor de características de uma janela de dados para repouso, mão fechada, e punho estendido	37
4.4	Matriz A de transição de estados do modelo HMM	38
4.5	Vetor π de Probabilidade Inicial	38
4.6	log-verossimilhança das observações	39
4.7	Classificação do gesto de fechar a mão	40

4.8	Classificação do gesto de estender o punho	41
4.9	Aquisição de dados de acelerômetro utilizando Trigno Avanti	42
4.10	Aquisição de dados de giroscópio utilizando Trigno Avanti	42
4.11	Aquisição de dados de acelerômetro utilizando Trigno Avanti	43
4.12	Aquisição de dados de giroscópio utilizando Trigno Avanti	43
4.13	Aquisição de dados de acelerômetro utilizando Trigno Avanti	43
I.1	Código para separar dados em conjunto de treinamento, e conjuntos de teste rotulado e não rotulado.....	50
I.2	Código para a extrair características das janelas para cada gesto para cada canal. No pedaço de código visto não inclui o terceiro gesto.....	51
I.3	Código treinamento do modelo HMM.....	52
I.4	Tela inicial do <i>tablet</i> que contém as aplicações móveis da Delsys Inc.	53
I.5	Aplicação <i>EMG Plots</i> disponível lno <i>tablet</i> durante a captura de sinais EMG.....	53
I.6	Programa de conversão de dados	54
I.7	Programa de visualização e análise de dados	54

LISTA DE TABELAS

3.1	Fonte: Manual de usuário Trigno [7]	29
4.1	Acurácia do Classificador	40

LISTA DE SÍMBOLOS

Siglas

IMU	<i>Inertial Measurement Unit</i>
EMG	Eletromiografia - <i>Electromyography</i>
SIHSUS	Sistema de Informações Hospitalares do Sistema Único de Saúde
NinaPro	<i>Non-Invasive Adaptive Hand Prosthetics</i>
WL	<i>Waveform Length</i>
AR	Modelo Auto-Regressivo - <i>Auto-Regressive Model</i>
LogVar	<i>coefficients and log-variance</i>
MV	<i>Mean Value</i>
MAV	Valor Médio Absoluto <i>Mean Absolute Value</i>
RMS	<i>Root Mean Square</i>
MDF	Frequência Mediana - <i>Middle Frequency</i>
MNF	Frequência Média - <i>Mean Frequency</i>
PSR	Densidade Espectral de Potência - <i>Power Spectral Density</i>
STFT	Transformada de Fourier em Curto Termo - <i>Short Term Fourier Transform</i>
LDA	Análise de Discriminantes Lineares <i>Linear Discriminant Analysis</i>
HCI	Interação Humano-Computador - <i>Human-Computer Interaction</i>
HGR	Reconhecimento de Gestos de Mão - <i>Hand Gesture Recognition</i>
OSW	Janela Sobreposta Deslizante - <i>Overlapping Sliding Window</i>
HMM	Modelo Oculto de Markov - <i>Hidden Markov Models</i>
MU	Unidade Motora - <i>Motor Unit</i>
MUAP	Potencial de Ação da Unidade Motora - <i>Motor Unit Action Potential</i>
MMSC	Modelo Matemático Para Contrações Estáticas <i>Mathematic Model for Static Contractions</i>
MMDC	Modelo Matemático Para Contrações Dinâmicas <i>Mathematic Model for Dynamic Contractions</i>

Capítulo 1

Introdução

1.1 Contextualização

Existe no Brasil uma alta prevalência de casos de amputação, que ocorrem principalmente por conta de acidentes de trânsito, acidentes de trabalho ou complicações de doenças crônicas, como por exemplo a diabetes. Porém, há uma escassez de informações e trabalhos de pesquisa envolvendo o assunto o que torna mais difícil trazer dados reais sobre a evolução dos casos no Brasil. Os trabalhos de pesquisa que são feitos aproveitam da base de dados do Sistema de Informações Hospitalares do Sistema Único de Saúde (SIHSUS) [8], uma base pública de dados de caráter administrativo de saúde. Um estudo utilizando essa base analisou que entre 2008 e 2015, foram registrados 361.585 procedimentos de amputação e que, além disso, nesse período a média brasileira de casos de amputação foi de 20,7 casos por 100 mil habitantes [9].

De acordo com as Diretrizes de Atenção à Pessoa Amputada, podemos definir amputação como o procedimento para a retirada total ou parcial de um membro e é um recurso utilizado para o tratamento de diversas doenças, seu intuito é prover uma melhoria na qualidade de vida de um paciente [10]. Após a amputação de algum membro, um processo de reabilitação deve ser feito para apoio, manutenção da saúde física e mental, adaptação e desenvolvimento de autonomia daquele paciente, inclusão social e retorno às atividades do dia-a-dia. Uma figura importante na reabilitação é o uso de próteses, cujo uso é registrado desde tempos antigos [11].

Próteses podem ser recurso importante para a reabilitação de pacientes, indo além do objetivo de fazer uma reposição estética. Cada vez mais as próteses vêm sendo desenvolvidas para a reposição funcional do membro perdido, aumentando a autonomia de um paciente amputado, ao devolver as funções perdidas pelo procedimento de amputação.

Muitos estudos vêm sendo feitos afim de desenvolver próteses progressivamente mais tecnológicas e acessíveis, visando também aumentar a similaridade entre próteses e os membros amputados. Com os avanços nas áreas de engenharia biomédica e manufatura aditiva, busca-se melhorar as funcionalidades e tornar mais intuitivo o uso das próteses. No caso de próteses de membro superior, existem diversas soluções, sendo elas puramente mecânicas ou que utilizam sensoriamento juntamente microprocessadores ou microcontroladores e com atuação de motores para replicar

movimentos.

Uma das próteses que utilizam desses estudos de engenharia é a prótese mio-elétrica. Nesse tipo de prótese, sinais provenientes da contração muscular do usuário são captados através de sensores de eletromiografia de superfície (sEMG) e têm objetivo de gerar um movimento funcional. Esses sinais são processados e o movimento desejado é replicado na prótese através de atuadores. O tipo de movimento, sua força, precisão, e qualidade em geral vão depender dos equipamentos, sensores e na estratégia de controle utilizado.

Em estudos mais recentes, vêm se utilizando de sensores inerciais (IMUs) para aumentar a precisão e acurácia do controle desenvolvido para próteses [12], [3]. Esses sensores são dispositivos eletrônicos que combinam acelerômetro, giroscópio e, em alguns casos, magnetômetros, para cada 1 dos 3 eixos ortogonais, fornecendo um total de até 9 medidas em um único dispositivo. Além disso, medidas como aceleração e velocidade angular podem ser usadas para estimar orientação espacial. Porém existe uma dificuldade em utilizar esses sensores por apresentarem um acúmulo de erro ao longo do tempo, o que leva à um *drift*, uma diferença entre a posição real e a posição medida, o que pode aumentar a complexidade do controlador.

Existem muitas iniciativas com o esforço de desenvolver próteses tecnológicas para o mercado, mirando no baixo custo e design moderno e intuitivo. Um exemplo é o *Hero Arm* produzido pela empresa *Open Bionics*, uma prótese mio-elétrica de forma personalizada e única para cada usuário ao escanear o membro do paciente. Além disso, possui uma parceria com a Disney e assim oferece as próteses com designs baseados em personagens famosos mundialmente. Fundada em 2014, a empresa britânica já ganhou várias premiações e hoje sua prótese variam entre R\$1400,00 e R\$4300,00 [1].

Também existem iniciativas brasileiras com o intuito de desenvolver próteses de mão a baixo custo. Entre eles, por exemplo, o projeto Mão 3D da Universidade Federal de São Paulo, que procura doar próteses de membro superior impressas em 3D para pessoas de todo Brasil, fazendo pesquisa e desenvolvendo a partir de projetos de código aberto [13].

O desenvolvimento de próteses de mão robóticas é um tema atual e no Brasil são poucos os estudos e projetos sobre. A maior complexidade deste tema está em criar uma prótese que consiga replicar os movimentos de uma mão de forma precisa e eficiente, que seja intuitiva e de fácil adaptação para então aumentar a qualidade de vida dos usuários.

Logo, para que próteses de alta tecnologia sejam desenvolvidas, além de todo o modelo mecânico envolvido, deve-se começar a explorar aspectos eletrônicos, que envolvem sensoriamento, estratégias de controle e atuação em uma prótese. Para se iniciar esse processo, este trabalho visa em reconhecer padrões de um sinal eletromiográfico e classificar os gestos executados pela mão. O resultado da classificação poderá ser usado para identificar qual o gesto que a prótese deve replicar.

1.2 Definição do problema

A diferença de uma prótese robótica para próteses puramente mecânicas é a capacidade de se replicar os movimentos de mão. Para isso, é necessário saber qual o movimento que um indivíduo está tentando reproduzir, para que a prótese possa replicar o mesmo movimento. Existem diversas maneiras de se alcançar este objetivo, e para este trabalho, deseja-se utilizar sensores capazes de obter sinais eletromiográficos de superfície (sEMG) e sinais inerciais. É possível, a partir desses sinais, utilizar algoritmos de reconhecimento de padrão para saber qual o movimento está sendo reproduzido, de forma que uma prótese possa identificar essa intenção de movimento e então replicar o movimento desejado.

Foram utilizados sensores capazes de coletar sinais inerciais e sinais eletromiográficos ao mesmo tempo, de maneira *offline*. Os dados são coletados e carregados no computador para que então possam ser utilizados pelo programa de reconhecimento de Padrões.

Este é um trabalho inicial de classificação de movimentos, e portanto, longe de se utilizar o resultado da classificação para controle de próteses. Porém, é o primeiro passo nessa direção

1.3 Objetivos do projeto

Se existe uma perspectiva de se controlar próteses robóticas no futuro, o objetivo deste trabalho é fechar o ciclo de procedimentos para reconhecimento de padrões utilizando Modelos Ocultos de Markov, ou seja, coletar sinais mio-elétricos através de sensores, separar e extrair as características do sinal e então realizar a classificação dos movimentos que foram realizados. Não chegaram a ser usados os dados inerciais para classificação por motivos que serão explicados nas próximas sessões, mas estes também fazem parte da discussão. Para realizar a classificação de movimentos de mão, portanto, definem-se alguns **objetivos específicos**.

- Selecionar um conjunto de gestos de mão para serem reproduzidos e classificados
- Coletar dados eletromiográficos utilizando o sensor *Trigno Avanti* da *Delsys Inc.*
- Separar os dados coletados em janelas de análise utilizando métodos de segmentação de dados
- Extrair as características dos dados a partir do modelo estatístico Auto-Regressivo de ordem 4, e dos valores *RMS* dos sinal.
- Utilizar o conjunto de características para se estimar os parâmetros dos Modelos Ocultos de Markov
- A partir do modelo estimado, utilizar algoritmos para classificar os movimentos a partir das características do sinal gerado.
- Avaliar o desempenho da Classificação.
- Discutir o uso dos sinais inerciais na classificação.

1.4 Resultados obtidos

A partir de um protocolo experimental, foram coletados dados eletromiográficos de 3 posições distintas de mão. São elas mão em repouso, mão fechada e extensão do punho. A repetição desses 3 gestos gerou um conjunto de dados, que foram segmentados. A partir dessa segmentação, se extrai dos dados algumas características, são elas os coeficientes do modelo Auto Regressivo de ordem 4, e o valor RMS da janela. Essas características serviram para o treinamento de um modelo HMM, utilizado para a classificação de movimentos. O classificador foi testado e teve desempenho acima de 90% para os gestos utilizados para os testes.

1.5 Apresentação do manuscrito

Este trabalho está dividido em 5 capítulos e organizados da seguinte maneira:

- Capítulo 1 – Introdução: Possui uma contextualização do cenário e do problema que se deseja resolver, além de definir os objetivos para o trabalho
- Capítulo 2 – Fundamentos teóricos e Revisão Bibliográfica: Neste capítulo é explorada a literatura envolvendo o presente problema, além de dar toda os conceitos e fundamentos teóricos que serão necessários no desenvolvimento do trabalho.
- Capítulo 3 – Métodos: Nesse capítulo são descritos os métodos utilizados para resolver o problema, além de todo o procedimento experimental envolvido.
- Capítulo 4 – Resultados: Neste capítulo são apresentados os resultados e apresentada discussões em cima deles, avaliando o desempenho do modelo criado para classificar os estados.
- Capítulo 5 – Conclusão: O último capítulo que resume o desenvolvimento do trabalho, e apresenta perspectivas para o futuro do trabalho e suas melhoras.

Capítulo 2

Fundamentos

2.1 Introdução

O primeiro passo foi fazer uma profunda pesquisa bibliográfica buscando uma fundamentação teórica sólida e que dê os panoramas da literatura existente sobre o assunto deste trabalho. Dessa forma, foi possível obter uma boa perspectiva dos estudos feitos sobre o assunto, as soluções mais comuns utilizadas e os temas mais recentes. Além disso, buscou-se uma base sobre os conceitos necessários para o desenvolvimento desta e das próximas partes deste trabalho, que envolvem o sensoriamento e condicionamento de sinais, e todo o processo para o reconhecimento de padrões.

2.2 Revisão Bibliográfica

2.2.1 Próteses Existentes no Mercado

2.2.1.1 Hero Arm

Desenvolvida pela empresa Open Bionics, a prótese Hero Arm vista na figura 2.1 é uma das mais conhecidas mundialmente. Feita a partir de impressão 3D o que diminui o custo de fabricação e possui um visual baseado em super-heróis e outros personagens bastante conhecidos. É feito um scan em 3D do braço para que a prótese seja feita de forma mais confortável ao se adaptar à anatomia do usuário [1].

Um de seus diferenciais é ser uma prótese mioelétrica e possuir um controle proporcional que dá a capacidade de reproduzir 6 posições com a mão o que torna possível que o usuário consiga realizar tarefas delicadas do dia-a-dia, como segurar um ovo.

2.2.1.2 Open Hand Project – Dextrus Hand

A *Dextrus Hand* é um projeto de código aberto desenvolvido pelo Open Hand Project. O objetivo deste projeto é fazer próteses de mão mais acessíveis para amputados. Uma diferença



Figura 2.1: Prótese *Hero Arm* [1]

deste projeto é ser apenas a mão como mostrado em 2.2, podendo ser encaixada em outras próteses utilizando um conector padrão. Ainda, é uma prótese mio-elétrica, com um sistema de controle que é implementado com apenas dois eletrodos para captação de sinais EMG. A prótese é confeccionada em plástico e feito a partir de manufatura aditiva, com impressão 3D [2].

A *Dextrus Hand* possui um design pensado para ao agarrar um objeto os dedos se ajustam em torno do formato do objeto para se ter uma pegada firme, se adaptando para se ajustar às formas que a mão agarra. A mão faz isso por ter um tendão para cada dedo, que percorre todas as articulações até a ponta.



Figura 2.2: Prótese *Dextrus Hand* [2]

2.2.1.3 HackBerry

Outra prótese de código aberto, cujas peças podem ser manufaturadas a partir de impressão 3D, gerenciado atualmente pela Mission Arm Japan, uma instituição sem fins lucrativos. O objetivo é de se desenvolver um braço artificial mas também se tornar uma plataforma de onde desenvolvedores e usuários de braços artificiais do mundo inteiro possam usar para construir da forma que desejarem [14].

Sensores mio-elétricos são utilizados para captar sinais de controle, os dedos são independentes, se movem por meio de atuadores, possuem dois graus de liberdade para realizar movimentos de flexão, e o punho possui um grau de liberdade, podendo estender ou flexionar.



Figura 2.3: Prótese *HackBerry*

2.2.2 Abordagens Existentes na Literatura

Fez-se uma extensa pesquisa sobre os estudos e técnicas já desenvolvidas sobre o tema em questão. Utilizar sensores para controlar próteses de braço é algo relativamente recente e por isso muito ainda está sendo estudado e desenvolvido. O objetivo destes trabalhos é utilizar os sinais obtidos através de sensores para poder controlar a prótese, mais especificamente, definir posições com a mão a partir dos movimentos feitos pelo usuário e sua intenção de movimento. Nota-se que grande parte dos estudos utiliza principalmente de eletromiografia de superfície e que a maioria trabalhos envolvendo IMUs os utilizam para complementar análises e técnicas feitas com sEMGs, ou servir de base de comparação.

2.2.2.1 Controle de Rotação de punho

Uma das principais referências utilizadas para este trabalho é o artigo de D. Bennet e M. Goldfarb [3]. Neste artigo, o objetivo é propor um método de controle de rotação de punho para que se possa obter próteses com uma melhor funcionalidade, mais intuitiva, efetiva e com controle mais coordenado ao se reduzir os movimentos que são feitos com o torso e com a parte superior do braço para compensar a falta de controle de movimento do punho.

Se baseando na sinergia de movimentos que existe entre a abdução/adução do ombro com a supinação/pronação do punho, é proposto em [3] uma estratégia de controle baseado nos ângulos medidos a partir de uma IMU localizada no antebraço, a fim de relacionar o movimento de abdução/adução ombro à velocidade angular de supinação/rotação do punho.

A entrada do controlador é, portanto, o ângulo de abdução/adução do ombro, que é aproximado a partir do ângulo do braço em relação ao plano vertical frontal θ_a como visto na figura 2.4. Para os casos em que o torso se mantém vertical, esse é o mesmo valor do ângulo de abdução/adução.

A velocidade angular de rotação do punho $\dot{\theta}_w$ será diretamente proporcional à θ_a , medido a partir de uma posição neutra onde o braço está levemente abduzido do torso. Mas o plano definido pelo braço e antebraço se mantém paralelo ao plano médio sagital. Um valor positivo de θ_a indica abdução do ombro resultando em uma rotação do punho na direção de pronação, enquanto valores negativos de θ_a resultam em uma velocidade angular na direção de supinação. Ao retornar o braço para uma pequena faixa ao redor da posição neutra ($\theta_{0lb} \leq \theta_a \leq \theta_{0ub}$) resultam em velocidade angular nula. A ligação entre esses ângulos e velocidade de rotação é ajustada especificando os ganhos de velocidade para adução e abdução respectivamente.

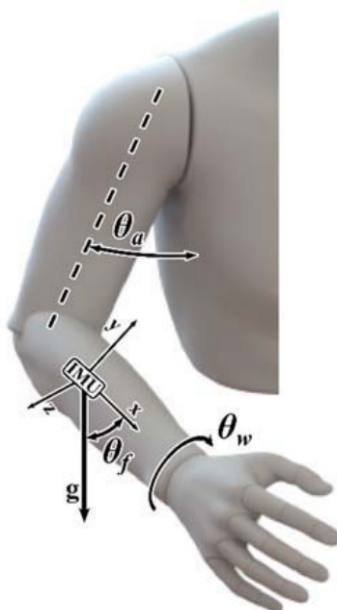


Figura 2.4: Esquema Mostrando a posição da IMU no braço, movimentos do braço e ângulos relevantes. Imagem de [3]

Porém, no caso de próteses transradiais, onde o IMU está posicionado no antebraço, o ângulo θ_a do plano frontal do braço, que é utilizado para a rotação do punho, não pode ser medido diretamente. Por isso o ângulo é calculado com base no vetor de gravidade sendo projetado sobre os planos principais.

Este controlador baseado em IMU foi comparado com outra abordagem para controle de rotação de punho, a interface de controle sequencial em dois pontos. Nesta técnica, um único par de eletrodos são utilizados para controlar um Grau de Liberdade mio-elétrico em qualquer momento do tempo. A velocidade do componente utilizado é proporcional à amplitude do sinal EMG, associado à direção do movimento.

Neste artigo [3], é abordado somente o controle de rotação de punho pelos métodos descritos acima. O controle e classificação das posições de mãos são feitos a partir de medidas de sEMG

que serão descritas a seguir.

2.2.2.2 Métodos de Controle e Classificação de Posições de Mão

A abordagem mais comum encontrada para se fazer o controle de próteses é a partir de sinais mio-elétricos. Uma grande variedade de estratégias de controle foram desenvolvidas para traduzir a informação dos sinais EMG e a maioria das estratégias aplicadas utilizam controle sequencial, enquanto pesquisas buscam formas de implementar controle simultâneo da mão. No controle sequencial, os sinais EMG podem ser traduzidos utilizando diversas formas de controle, tais como Controle Proporcional, Controle por Máquina de Estados Finitos, Controle por Reconhecimento de Padrões, entre outros [15].

Será visto com mais atenção duas estratégias de controle estudadas, que se baseiam em Controle por Máquina de Estados Finitos e Controle por Reconhecimento de Padrões.

2.2.2.3 Controle Por Máquina de Estados Finitos

Neste método, posições de mão são pré-definidas como estados, e as transições entre estados também são definidos ou decodificados a partir da entrada. É um método bem utilizado para um número limitado de posições de mão mas não tanto para multifuncionalidade. A mudança (transição) ocorre a partir do sinal EMG, e continua até a posição desejada for selecionada [15].

Esta abordagem foi utilizada em [3] e descrita em [16], sendo projetada para atender um certo conjunto de posturas de mão utilizando uma interface de dois eletrodos EMG como interface de entrada.

Presume-se que para o controlador proposto em [16], alguns critérios sejam atendidos. Os critérios envolvem extensão, flexão, reposição e oposição do dedão, flexão e extensão do indicador, flexão e extensão simultânea dos outros 3 dedos, também espera-se que o deslocamento e as forças desses dedos seja mensuráveis ou conhecidas.

A troca de estado é decidida então a partir de uma função do estado atual e das entradas do controlador, que envolvem o sinal EMG, a configuração atual da mão e a informação de força do aperto dos dedos. Os estados são organizados de forma progressiva de acordo com o grau de fechamento da mão e também agrupando posições de mão de acordo com a posição do dedão. De acordo com os sinais obtidos através dos eletrodos, o sinal pode ser dividido em 4 entradas, flexão, extensão, co-contração e repouso. As transições de estados são baseados em condições lógicas que são decididas a partir do deslocamento e força dos dedos e níveis do sinal EMG.

O comportamento da mão em cada estado é determinado por um controlador de coordenação que define qual subgrupo de atuadores estão ativos na mão. Se uma transição ocorre e o estado da mão muda, um novo subgrupo de atuadores é ativado pelo controlador [16].

2.2.2.4 Controle Por Reconhecimento de Padrões

Esta abordagem de controle consiste na utilização de técnicas para extração e classificação de *features*, características, dos dados segmentados no processamento de sinais para comandar o controlador dos atuadores. Normalmente, essas *features* podem ser extraídas em tempo, frequências e tempo-frequência para identificar a informação do sinal mio-elétrico e a partir das características coletadas, o treinamento de algoritmos para identificação de intenção do movimento é feita para se classificar as posições de mão durante o movimento.

O primeiro passo para se aplicar este método de controle é a aquisição dos dados para análise e extração das *features*. Para isso, em geral, essa coleta de dados é feita de forma *offline* e consiste em repetir alguns movimentos pré-determinados. Um protocolo experimental em geral é criado para esta coleta de dados de acordo com cada autor, mas um exemplo disso é o protocolo do projeto NinaPro [17] (*Non-Invasive Adaptive Hand Prosthetics*). Este projeto construiu uma base de dados e algoritmos e protocolos com o objetivo de aumentar a destreza e reduzir tempo de treinamento para próteses controladas por sinais mio-elétricos, incluindo um conjunto de posições de mão que podem ser usados para se fazer a coleta de dados e o treinamento de algoritmos de reconhecimento de padrões [18].

Seguindo o protocolo NinaPro, e utilizando Reconhecimento de Padrões para montar um algoritmo de controle para Próteses, o trabalho [12] propõe complementar o uso de sinais mio-elétricos com sinais obtidos com IMU. Utiliza-se um total de 12 sensores capazes de captar tanto os sinais mio-elétricos quanto os sinais inerciais e com isso, foram extraídos *Waveform Length* (WL), *4th-order auto-regressive* (AR), *coefficients and log-variance* (LogVar) como características do sinal EMG, e *Mean Value* (MV) como característica para ambos os sensores.

Para decodificação de intenção de movimento a partir das características extraídas, foi aplicado o classificador de análise de discriminantes lineares (LDA), um classificador probabilístico que estima a distribuição de probabilidade sobre as classes. Este classificador e suas variantes são muito usadas no contexto de controle mio-elétrico por poderem obter alta performance de decodificação com requisitos mínimos [12]. Este trabalho foi a principal referência ao qual este trabalho se baseou, a ideia inicial consistia em, mesmo utilizando um classificador diferente, tentar reproduzir uma classificação como a descrita no trabalho, mas com uma quantidade reduzida de sensores, considerando a disponibilidade destes no laboratório.

Portanto, classificação por Reconhecimento de Padrões será a estratégia utilizada neste trabalho. Serão coletados sinais a partir de uma série de repetições de movimentos, esses sinais terão suas características extraídas para o treinamento de um classificador por reconhecimento de padrões. Porém, diferentemente dos trabalhos citados acima, serão utilizados Modelos Ocultos de Markov (HMM) para a classificação dos estados, baseando-se em trabalhos como [19], [20]. Todo o processo de reconhecimento de padrões será descrito com mais detalhes na sessão 2.4. É importante ressaltar que o escopo do trabalho foi reduzido ao longo de seu desenvolvimento para a classificação de apenas 3 gestos utilizando somente sensores eletromiográficos, incluindo os sinais inerciais como parte da discussão mas não como entrada para o classificador.

2.3 Sensoriamento

2.3.1 Sensores Inerciais (IMU)

Inertial Measurement Units ou Unidade de Medição Inercial, é um dispositivo que contém um conjunto de sensores capaz de mensurar grandezas inerciais de um corpo. Em geral, é composto por um acelerômetro e um giroscópio, podendo também incluir magnetômetros e outros sensores. Esses sensores são medidos nos 3 eixos ortogonais podendo obter medidas com 6 ou 9 Graus de Liberdade, dependendo da presença do magnetômetro. Com a informação destes sensores, pode-se fazer uma estimativa sobre a orientação do corpo medido.

2.3.1.1 Giroscópio

Este tipo de sensor mede movimento angular, de forma a indicar mudanças na orientação de um corpo em movimento. Não mede orientação absoluta, mas é possível obter a velocidade angular do movimento de um corpo, e é muito útil para corrigir a orientação durante o movimento. Além disso é possível fazer uma estimativa da orientação através da integração de suas medidas ao longo do tempo, mas isso pode levar a erros acumulados e por isso deve ser corrigida periodicamente.

2.3.1.2 Acelerômetro

Este sensor é utilizado para medir as acelerações que estão em atuação sobre o corpo medido, sejam elas acelerações estáticas (aceleração gravitacional) ou dinâmicas (movimentação, perturbação e vibração). A direção do campo gravitacional obtida desse sensor é usada como referência absoluta para a estimação da orientação espacial. Com isso em mente, outras acelerações além da gravitacional podem piorar essa estimativa, e por isso o uso de outros sensores além do acelerômetro contribuem na estimativa.

2.3.1.3 Magnetômetro

O magnetômetro mede direção e magnitude do campo magnético ao qual o corpo está submetido. Detectando o campo magnético da Terra, este, sendo perpendicular ao vetor gravitacional, serve como a segunda referência absoluta para estimação de orientação espacial. Porém suas medidas podem ser afetadas por materiais ferrosos ou magnéticos, dependendo mais uma vez da composição dos outros sensores para manter uma boa estimativa da orientação.

2.3.1.4 Orientação: ângulos de Euler e Quaternions

A estimativa de orientação por ângulos de Euler é a mais comum. Os ângulos são dados pela orientação em relação à um sistema de coordenadas fixo. Elas podem ser definidas pela rotação de rolagem (*roll*), de arfagem (*pitch*) e de guinada *yaw* a partir da posição inicial. Geralmente

essas rotações são definidas ao redor dos eixos x , y e z , porém estes eixos podem ser alterados caso haja necessidade.

Já os Quaternions são uma outra forma de representação de orientação, capaz de representar rotações tri-dimensionais sobre um eixo arbitrário, utilizado em diversas aplicações como robótica, mecânica orbital, tecnologias aeroespaciais, gráficos de computador e programação de jogos [21]. Quaternions são uma extensão em 4 dimensões de números complexos e sua notação matemática é vista na equação 2.1, onde q_0 é a parte real, ou componente escalar de q , enquanto $w = \{q_1, q_2, q_3\}$ formam um vetor de 3 eixos imaginários.

$$\begin{aligned} q &= (q_0, q_1, q_2, q_3) \\ &= (q_0, w) \\ &= q_0 + iq_1 + jq_2 + kq_3 \end{aligned} \tag{2.1}$$

Quaternions possuem um potencial de ser uma poderosa ferramenta para um operador de rotação e diversas APIs possuem funções de operação com quaternions [21].

É possível transitar entre a representação em ângulos de Euler e representação em Quaternions. A conversão de Euler para Quaternion ocorre pela multiplicação matricial mostrada pela equação 2.2, onde ψ é a rotação em torno do eixo z , θ a rotação em torno de y e ϕ a rotação em torno de x enquanto o contrário é mostrado na equação 2.3.

$$q = \begin{bmatrix} \cos(\psi/2) \\ 0 \\ 0 \\ \sin(\psi/2) \end{bmatrix} \begin{bmatrix} \cos(\theta/2) \\ 0 \\ \sin(\theta/2) \\ 0 \end{bmatrix} \begin{bmatrix} \cos(\phi/2) \\ \sin(\phi/2) \\ 0 \\ 0 \end{bmatrix} \tag{2.2}$$

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \arctan\left(\frac{2(q_0q_1 + q_2q_3)}{1 - 2(q_1^2 + q_2^2)}\right) \\ \arcsin(2(q_0q_2 - q_3q_1)) \\ \arctan\left(\frac{2(q_0q_3 + q_1q_2)}{1 - 2(q_2^2 + q_3^2)}\right) \end{bmatrix} \tag{2.3}$$

2.3.2 Sensores de Eletromiografia de Superfície (sEMG)

Eletromiografia é a técnica de aquisição de sinais elétricos que são gerados pelas fibras musculares ao responder à uma atividade muscular. Este sinal, chamado de eletromiograma (EMG) pode ser medido de forma invasiva, ou seja, através de uma agulha que é inserida no músculo, ou de forma não-invasiva através de eletrodos posicionados na superfície da pele na área a ser estudada. Este segundo método é chamado de Eletromiografia de Superfície (sEMG) e é o sensoriamento que geralmente é feito para a aplicação de próteses mio-elétricas.

Porém, a aquisição do sinal por meio da superfície da pele não é tão seletiva e é suscetível à atividade de um ou mais músculos vizinhos e seus sinais podem ter interferência com o sEMG do músculo desejado [22]. Apesar disso, eletrodos de superfície, em geral, são utilizados para a avaliação de aspectos de comportamento, padrões de atividade, fadiga do músculo ou de um

grupo de músculos [23]. Uma razão significativa para se utilizar esse tipo de sinal para controle de próteses se dá pelo fato desses sinais conterem informações de intenção de movimento, na forma da contração muscular que é detectada em sinais de eletromiografia de superfície [24].

Uma unidade motora (MU) consiste em fibras musculares e um motoneurônio, e os sinais sEMG são coletados do potencial elétrico que ocorre em uma ou mais unidades motoras quando o músculo esquelético é contraído. Já, enquanto o músculo esquelético está em repouso, o potencial elétrico das fibras musculares é aproximadamente $-80mV$ [25]. O sinal sEMG coletado será uma soma entre vários potenciais de ação das fibras musculares, conhecido como Potencial de Ação da Unidade Motora (MUAP), ou seja, o sinal será o somatório dos MUAP de todas as unidades motoras recrutadas [26].

Consideram-se dois tipos de contração muscular, isométricas e isotônicas. As contrações isométricas, também chamadas de estáticas, são aquelas que não causam mudanças no tamanho das fibras musculares, e que não causam movimento articular. Já as contrações dinâmicas, também conhecidas como isotônicas, causam movimento articular e o comprimento das fibras musculares varia, se alongando ou encurtando. A modelagem matemática dos sinais EMG como um modelo estocástico vai depender desses dois tipos de contração [25]:

- **Contrações Estáticas:** O modelo matemático para contrações estáticas (MMSC) é um processo estacionário e o modelo mais básico do sinal durante uma contração isométrica é a soma de um número de MUAPs.

$$EMG(t) = \sum_{i=1}^N s_i(t) * m_i(t) \quad (2.4)$$

Na equação 2.4, N representa o número de MUs ativas, $s_i(t)$ é um trem de impulsos que representa as ativações da i -ésima MU, e $m_i(t)$ representa a MUAP da i -ésima MU. Portanto o termo $s_i(t)*m_i(t)$ se torna uma representação de um trem de MUAP. O operador $*$ significa a convolução entre os sinais.

- **Contrações Dinâmicas:** O modelo matemático para contrações dinâmicas (MMDC) é feito com o seguinte modelo não-estacionário descrito na equação 2.5.

$$EMG(t) = a(t)w(t) + n(t) \quad (2.5)$$

Na equação, $a(t)$ indica a intensidade do sinal EMG, ou seja, a informação do sinal. $w(t)$ é a variância-unitária do processo Gaussiano, representando o aspecto estocástico do modelo EMG, e $n(t)$ é o ruído dos sensores e sinais biológicos.

No contexto de reconhecimento de padrões, os modelos matemáticos do EMG em geral não são utilizados, dado a dificuldade de estimação em processos não-estacionários [26]. Para simplificar o escopo do trabalho, decidiu-se classificar apenas contrações estáticas. Porém, é importante que futuramente mais contrações sejam abordadas. A ausência de contrações dinâmicas neste trabalho faz com que leituras vindas dos sensores inerciais não tenham muita utilidade, dado que

pouco ou nenhum movimento do braço e antebraço será detectado, já que apenas a mão realizará movimentos.

2.3.2.1 Condicionamento de Sinais EMG

Para se extrair informações úteis do sinal EMG, deve-se considerar algumas características do sinal. O Sinal EMG é estocástico e pode ser bem representado por uma função de distribuição Gaussiana. Sua amplitude pode variar de 0 a $10mV$ (pico a pico) ou de 0 a $1.5mV$ (RMS). A energia útil do sinal é limitada na faixa de frequência de 0 a $500Hz$ com sua energia dominante estando na faixa de $50 - 150Hz$ [27] [28]. Como o sinal é analógico e contínuo, deve ser convertido para um sinal digital e amostrado. Considerando essa faixa de frequência útil entre 0 e $500Hz$, para satisfazer ao teorema de Nyquist, a taxa de amostragem do sinal deve ser de pelo menos $1000Hz$.

Um problema que pode ocorrer durante a digitalização do sinal é o problema de *Alias*, que ocorre quando frequências indesejadas mais altas no espectro do sinal ($> 500Hz$), que fogem da regra do teorema de Nyquist, causem interferências no espectro de frequência que deseja ser analisado. Portanto, deve-se utilizar um filtro passa-baixas que limite a frequência do sinal à metade da frequência de amostragem. antes mesmo do sinal ser coletado. Esses filtros são conhecidos como filtros *Anti-Aliasing* e estão implementados no sistemas da Delsys com frequências de corte em $500Hz$ [4].

O sinal EMG coletado é um sinal ruidoso, que pode sofrer interferências em diversas fases do processo, seja por *crosstalk* entre músculos, ou por ruídos inerentes aos componentes eletrônicos, interferências eletromagnéticas do ambiente, entre outras [27]. Por isso além do filtro *Anti-Aliasing*, outros filtros podem ser usados para condicionamento de sinal e remoção de ruído.

O filtro de Butterworth é um filtro analógico bem difundido e bem sucedido para aplicações de circuitos analógicos [29]. É também um filtro bastante usado para eliminar informações irrelevantes do sinal EMG [28]. O filtro é usado por maximizar a resposta de forma plana, fazendo que a resposta em frequência não possua ondulações ou efeitos de *ripple* (ilustrado pela figura 2.5) para transmissão da banda passante. Além disso, é o filtro mais indicado para preservar a linearidade da amplitude da região de banda passante, fato que o torna um candidato ideal para o condicionamento do sinal EMG [4].

O filtro de Butterworth passa-baixas de ordem N pode ser definido matematicamente pela seguinte fórmula.

$$|H(j\omega)| = \frac{1}{\sqrt{1 + (j\omega/j\omega_c)^{2N}}} \quad (2.6)$$

A figura 2.6, mostra um filtro passa-baixas de polo duplo (dois polos simples em cascata) facilitado por um amplificador operacional. Com a correta combinação dos valores de capacitância C e resistência R podem levar o filtro a produzir respostas como a de Butterworth.

O sistema Delsys inclui filtros analógicos de alta performance para as finalidades descritas. As bandas de transição para sistemas EMG são geralmente de $-80dB/década$ e os *ripples* de banda

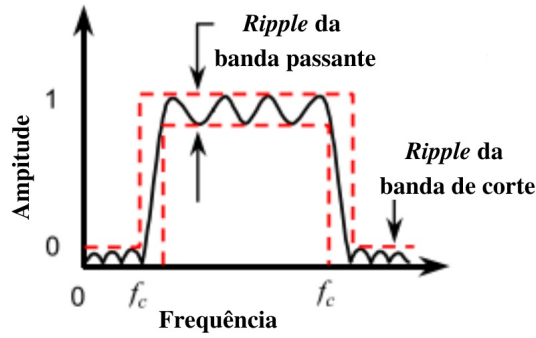


Figura 2.5: Característica de filtro passa-banda com efeito *ripple*. Imagem adaptada de [4]

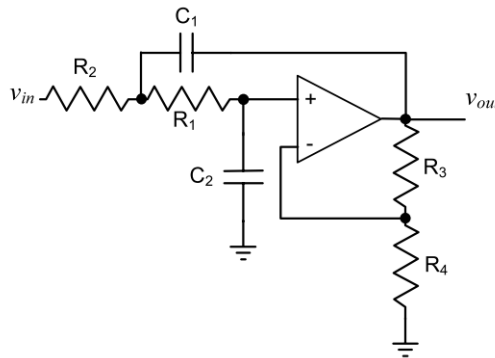


Figura 2.6: Filtro Passa-Baixa de ordem 2. Imagem adaptada de [4]

passante estão dentro de um desvio de 0.5%. De acordo com o manual de usuário, os sensores usados possuem 2 pólos passa alta e 4 polos passa baixa, nas frequências 20 – 450Hz [4], [7].

2.4 Reconhecimento de Padrões

O campo de estudo de interação entre humanos e computadores (*Human-Computer Interaction* - HCI) é uma área de estudo que visa em facilitar a comunicação entre humanos e computadores, sendo uma dessas formas de interação, o Reconhecimento de Gestos de Mão (*Hand Gesture Recognition* - HGR) [26]. Modelos HGR são sistemas que vão determinar intenção de movimento a partir do reconhecimento do gesto que está sendo reproduzido e podem fazê-lo através de diversas formas de aquisição dos dados do movimento tais como câmeras para visão computacional, sensores inerciais e sensores eletromiográficos. Como também são usados para diversas aplicações, tais como reconhecimento de linguagem de sinais, controle de dispositivos e o controle de próteses robóticas [26]. Neste trabalho, o reconhecimento de gestos de mão será feito através de aquisição de dados eletromiográficos, explicados nas sessões anteriores.

Para se reconhecer os gestos reproduzidos pela mão, algoritmos de reconhecimento de padrão podem ser aplicados. Um processo de classificação pode ser feitos através de diversas abordagem diferentes mas existem pontos fundamentais para o reconhecimento de padrões de sinais EMG [30].

Essa sequência de passos que são seguidas quando se deseja fazer classificação por reconhecimento de padrão é ilustrada na imagem 2.7. O método de reconhecimento de padrões em sinais EMG consistem em extrair características que representam a atividade muscular durante um movimento. Essas características serão divididas em classes de movimento. Neste trabalho a segmentação foi feita em duas etapas, na primeira manualmente e na segunda utilizando o método de *Overlapping Sliding Window* (OSW). Para se extrair características se utilizou o modelo AutoRegressivo (AR) e o valor *RMS*. Para a classificação foram utilizados os Modelos Ocultos de Markov (HMM).

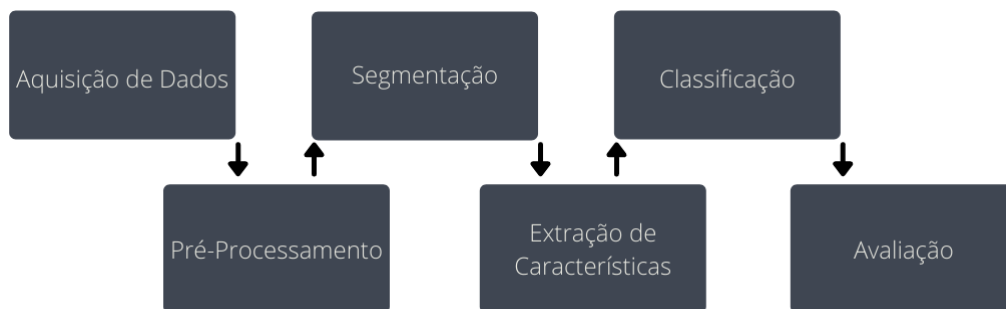


Figura 2.7: Sequência de passos para Reconhecimento de Padrões

2.4.1 Segmentação dos Dados

Segmentação de dados é o processo de particionar um segmento de dados em diferentes janelas de análise para assim, um vetor de características ser extraída em cada uma.

Existem diversas formas de se dividir uma sequência de dados em janelas. Duas serão exploradas neste trabalho. A primeira, chamada de Detecção de Gesto, consiste em dado um sinal EMG, separar apenas a área do sinal que corresponde à uma contração, criando um segmento para cada contração no sinal envolvido. Os tamanhos do segmento podem variar dependendo da duração da contração. Neste trabalho, durante o procedimento experimental, a instrução era de se manter as contrações por 5 segundos. A separação dos gestos foi feita manualmente a partir do *Software EMGWorks Analysis* do Delsys, e para cada contração, uma nova sequência apenas com 5 segundos daquela contração foi criada. A figura 2.8 ilustra a Detecção do Gesto feito manualmente.

O esquema de OSW (janela sobreposta deslizante) consiste em dividir um sinal dinâmico em janelas, de tamanho fixo, que vão se movimentando ao longo do sinal, com uma pequena sobreposição entre elas. Um esquema mostrando como funciona esse tipo de segmentação pode ser vista na figura 2.9, mostrando a diferença com o esquema de janelamento sem sobreposição.

Deve-se então decidir o tamanho da janela e da sobreposição a serem usadas. Esse fator é muito importante pois afeta o tempo de resposta de um controlador quando se trata de implementação em tempo real, como também afeta a acurácia do classificador.

Estudos mostram que o tamanho da janela deve ser menor que $300ms$ para que o controle em tempo real tenha um atraso aceitável [31], ao mesmo tempo que janelas de tamanho entre $150ms$ e $250ms$ possuem desempenho ótimo [32].

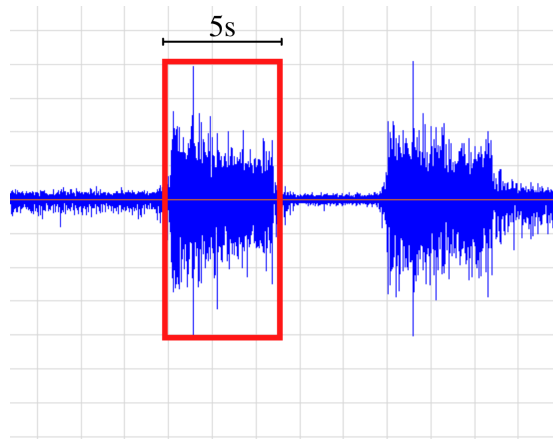


Figura 2.8: Separação de uma janela contendo somente uma contração que dura 5s

Seguindo esse esquema, pode-se calcular a quantidade de janelas a partir da fórmula 2.7

$$N \text{ janelas} = \frac{\text{tamanho dos dados} - \text{tamanho da janela}}{\text{incremento da janela}} + 1 \quad (2.7)$$

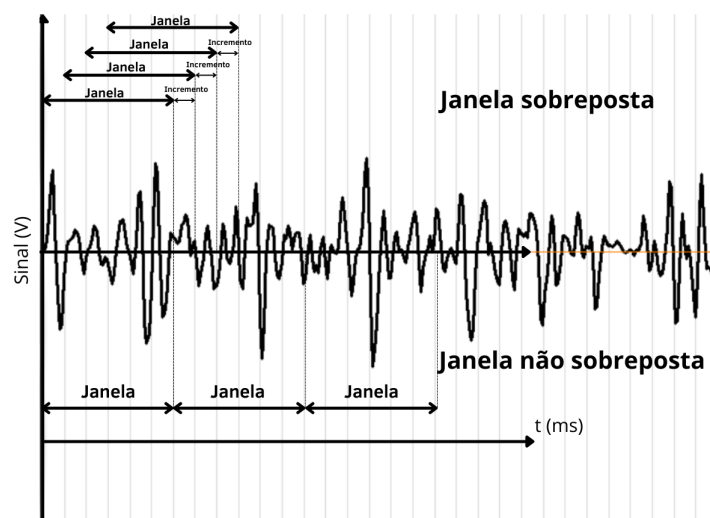


Figura 2.9: Esquemas de janelamento com e sem sobreposição. Imagem adaptada de [5]

2.4.2 Extração de Características

A Extração de Características corresponde a um processo de mapear o sinal em um conjunto de características que vão de alguma forma descrever o sinal original. Ao invés de se utilizar o sinal EMG da forma em que é coletado, são essas características, organizados em um vetor, que serão dados como entrada para o classificador. Este vetor é conhecido como vetor de características.

Em sinais EMG, as características podem ser extraídas de atividades musculares para então serem atribuídos à classes de movimentos.

Existe um grande número de possíveis características que podem ser extraídas de um sinal eletro-miográfico, e podem ser divididas em três categorias: domínio do tempo, domínio da frequência, domínio do tempo-frequência [24].

- **Domínio do Tempo:** Características no domínio do tempo são calculados a partir de séries temporais dos sinais EMG e são principalmente baseados na amplitude do sinal. Essas características podem indicar a energia do sinal, duração, força, e pode ser proporcional ao número de unidades motoras ativas, e sua taxa de ativação [24].

Em sua maioria, as características neste domínio permitem cálculos relativamente simples e possuem boas propriedades de classificação [5]. Algumas das características podem incluir: Média Quadrática (*Root Mean Square* - RMS), Valor Médio Absoluto (*Mean Absolute Value* - MAV), coeficientes do modelo AutoRegressivo (*AutoRegressive Model*- AR), entre diversos outros. Para este trabalho, foram utilizadas características extraídas do modelo AR, juntamente com os valores *RMS*.

- **Domínio da Frequência:** As características no domínio da frequência descrevem a fadiga muscular além de diferenciar mudanças no recrutamento de unidades motoras (MU). Algumas características nesse domínio são: Frequência Média (*Mean Frequency* - MNF), Frequência Mediana (*Middle Frequency* - MDF), Densidade Espectral de Potência (*Power Spectrum Ratio* - PSR), entre outras na literatura [26].
- **Domínio do Tempo-Frequência:** Uma transformada de Fourier de curto termo (STFT) pode mapear um sinal em uma função de 2 dimensões, de tempo e frequência. Esse mapeamento é útil para que não se perca informação temporal ao se analisar o espectro da frequência de um sinal, porém possui uma precisão limitada, dado o tamanho da janela de análise [24].

Em geral, Transformadas *Wavelet* são utilizadas para analisar características neste domínio, mas envolvem cálculos relativamente complexos para tal. [5].

2.4.2.1 Valor *RMS*

O Valor *RMS* (*Root Mean Square*) é o valor dado pela raiz quadrada das médias quadráticas da janela de dados, e é dada pela fórmula 2.8, onde N é a quantidade de amostras na janela e x_i o valor da amostra.

$$RMS = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2} \quad (2.8)$$

2.4.2.2 Modelos Auto-Regressivos

Além do valor *RMS*, as outras características extraídas do sinal são os coeficientes do modelo Auto Regressivo. Estes modelos se baseiam em tentar prever uma série temporal somente pelos valores passados da série, chamados de atrasos.

Assumindo uma série temporal $Y(t)$ uma sequência estacionária, o seu respectivo modelo AR pode ser dado pela fórmula 2.9. Supondo que a ordem seria 1, o modelo teria o seguinte formato:

$$y_k = a_1 y_{k-1} + n \quad (2.9)$$

Onde y_k é a observação no momento atual, a_1 é o coeficiente do modelo, e n um termo para representar o resíduo do ruído branco. Portanto, estendendo a fórmula para um modelo de ordem P , obtém-se a fórmula

$$y_k = \sum_{i=1}^P a_i y_{k-i} + n_i \quad (2.10)$$

De acordo com [33], uma série de experimentos indica que quando a ordem do modelo AR é $P = 4$, a análise do sinal e performance do reconhecimento é a melhor, por isso é a ordem escolhida para o modelo neste trabalho. É necessário então calcular os coeficientes do modelo. Para isso utilizam-se conceitos estatísticos de autocorrelação e auto covariância de uma série temporal.

Covariância e Correlação são conceitos semelhantes, e medem a relação entre como duas variáveis variam juntas. A covariância é uma medida não-padronizada do grau em que duas variáveis se movem juntas e é estimada a partir da fórmula 2.11, onde x_i e y_i são as variáveis a serem analisadas no instante i , μ_x e μ_y as suas médias e n o número de amostras.

$$Cov(X, Y) = \gamma(x, y) = \frac{\sum_{i=1}^n (x_i - \mu_x)(y_i - \mu_y)}{n} \quad (2.11)$$

Correlação é uma medida padronizada que vai indicar tanto a intensidade quanto a direção desse relacionamento entre duas variáveis e é obtida a partir da covariância entre duas variáveis divididos pelos seus respectivos desvios padrões.

$$\rho(x, y) = \frac{\gamma(x, y)}{\sigma_x \sigma_y} \quad (2.12)$$

Já autocovariância e autocorrelação dizem respeito às relações de covariância e correlação, mas ao invés de ser sobre duas variáveis diferentes, consiste entre a relação entre uma variável com ela mesma em instantes anteriores, ou seja, com um atraso. Se x_t indica uma série temporal no instante t , x_{t-k} a mesma série atrasada em k instantes, então a Função de Autocovariância pode ser vista como uma função do atraso k e é definida na fórmula 2.13, sendo que o desvio padrão de

uma variável atrasada não muda, portando o denominador se torna o quadrado do desvio padrão que é a própria variância de x_t .

$$\rho(k) = \frac{Cov(x_t, x_{t-k})}{\sigma_x^2} = \frac{Cov(x_t, x_{t-k})}{Var(x_t)} \quad (2.13)$$

Com estes coeficientes definidos, para se finalmente calcular os coeficientes do modelo AR, utiliza-se as chamadas Equações de Yule-Walker, que vão relacionar os valores de autocorrelação, que podem ser calculados dado uma série de dados, com os coeficientes do modelo. Considerando a expressão 2.14.

$$\rho(k) = a_1\rho(k-1) + a_2\rho(k-2) + \dots + a_p\rho(p) \quad (2.14)$$

Sendo $k = 1, 2, 3, \dots$. Essas são as equações de Yule-Walker, pois para cada valor de k , obtém-se uma equação. Assim, essa equação pode ser representada em forma matricial. Essa expressão, para ser usada para estimar o modelo, pois uma vez que se tem uma série temporal observada, é possível calcular os valores de autocorrelação, e basta resolver um sistema de equações. Escrita em forma matricial para um modelo de ordem $p = 4$:

$$\begin{bmatrix} \rho(1) \\ \rho(2) \\ \rho(3) \\ \rho(4) \end{bmatrix} = \begin{bmatrix} \rho(0) & \rho(1) & \rho(2) & \rho(3) \\ \rho(1) & \rho(0) & \rho(1) & \rho(2) \\ \rho(2) & \rho(1) & \rho(0) & \rho(1) \\ \rho(3) & \rho(2) & \rho(1) & \rho(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} \quad (2.15)$$

Que também pode ser vista como $R = PA$, considerando R o vetor de autocorrelações, P uma matriz quadrada de autocorrelações e A o vetor de coeficientes que deseja-se descobrir. É possível ver que uma vez que se tenha os coeficientes da autocorrelação, basta que $A = P^{-1}R$.

2.4.2.3 HMM - Modelos Ocultos de Markov

O Modelo Oculto de Markov é uma abordagem probabilística para reconhecimento de padrões de sinais EMG, que se baseia na topologia das cadeias de Markov, possuindo estados e matrizes de probabilidade [24].

A cadeia de Markov, por sua vez é um caso particular de um processo estocástico que para uma sequência de eventos, a probabilidade de um evento futuro depende apenas do presente, e não de eventos passados, ou seja, é um processo "sem memória". Portanto, considerando um processo estocástico $\{X_n, n \in \mathbb{N}\}$, com estados $S_i, i = 0, 1, 2, \dots, n$, sendo n o número total de estados e S_i o i -ésimo estado, o processo é dito como uma Cadeia de Markov se:

$$P(X_{n+1} = S_j | X_0 = S_0, X_1 = S_1, \dots, X_n = S_i) = P(X_{n+1} = S_j | X_n = S_i) \quad (2.16)$$

Para todo estado S_0, S_1, \dots, S_n . Isto significa dizer que no passo $n + 1$, a chance de X_{n+1} estar no estado S_j depende apenas de estado atual S_i de X_n .

Considerando como exemplo, que os estados da cadeia de Markov são posições realizadas por uma mão, ou seja, considerando os estados $S_0 = \text{mão aberta}$ e $S_1 = \text{mão fechada}$. Existe uma probabilidade de se transicionar entre estados, a mão fecha (estado S_0 para estado S_1), ou a mão abre (estado S_1 para estado S_0), e suas probabilidades são, respectivamente p_{01} e p_{10} . E existe uma probabilidade da se manter no estado atual, a mão aberta continua aberta (S_0 continua em S_0) ou a mão fechada permanece fechada (S_1 continua em S_1), cujas probabilidades são p_{00} e p_{11} respectivamente. Esses estados e probabilidades podem ser representados por uma máquina de estados finitos como ilustrado pela figura 2.10.

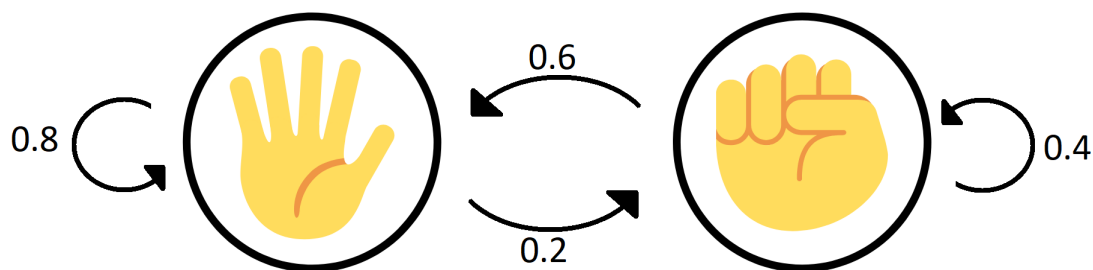


Figura 2.10: Cadeia de Markov com dois estados

Essas probabilidades podem ser representadas através de uma matriz, cujas linhas representam o estado atual e as colunas representam o próximo estado, e portanto. Os elementos dessa matriz são as probabilidades de se sair de um estado para o outro. Essa matriz é conhecida como a Matriz de Transição de Estados e para o exemplo em questão, essa matriz é dada por:

$$P = \begin{bmatrix} p_{00} & p_{01} \\ p_{10} & p_{11} \end{bmatrix} = \begin{bmatrix} 0.8 & 0.2 \\ 0.6 & 0.4 \end{bmatrix} \quad (2.17)$$

Se estendermos isso para um modelo com k estados. P é dado pela matriz na fórmula 2.17.

$$P = \begin{bmatrix} p_{00} & p_{01} & p_{02} & \cdots & p_{0k} \\ p_{10} & p_{11} & p_{12} & \cdots & p_{1k} \\ p_{20} & p_{21} & p_{22} & \cdots & p_{2k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{n0} & p_{n1} & p_{n2} & \cdots & p_{nk} \end{bmatrix} \quad (2.18)$$

Sendo cada elemento $p_{i,j}$, definido por:

$$p_{i,j} = P(X_{n+1} = S_j | X_n = S_i). \quad (2.19)$$

Este exemplo corresponde a um modelo onde os estados correspondem a eventos observáveis, e não é aplicável para o problema de reconhecimento de gestos de mão. Por isso se usa uma extensão deste modelo, chamados de Modelos Ocultos de Markov (HMM). Neste caso, a Observação é

uma função probabilística do estado, resultando em um modelo que é um processo estocástico duplamente embutido. Isso significa dizer que um processo não pode ser observado (oculto), e só pode ser observado a partir de outro conjunto de processos estocásticos que produzem a sequência de observações [34]. Assim, tem-se uma sequência de dados observados e uma sequência de estados ocultos [20].

Isso posto, dado uma sequência observada de sinais eletromiográficos, o HMM poderá classificar os movimentos realizados pela mão, sendo cada movimento um estado oculto para o modelo. A partir disso, pode se definir que as características extraídas do sinal são a sequência de observações do modelo, enquanto as posições de mão são os estados ocultos, dois dos parâmetros para a criação de um modelo HMM.

O Modelo HMM possui alguns elementos e parâmetros que são definidos a seguir [19]:

- Uma sequência de N estados ocultos $S = (s_t), t = 1, 2, \dots, N$. Esses estados são representados pelos gestos de mão que deseja-se classificar.
- Uma sequência de observações $O_T = o_1, o_2, \dots, o_t$ onde o_t representa um vetor de valores de valores na janela t , e T o tamanho da sequência de observações. Esse vetor de valores é dado pelas características do sinal observado na janela t , com M símbolos de observação diferentes.
- Vetor de probabilidade inicial $\pi = (\pi_i)$ que representa a probabilidade de estar no estado i , no primeiro momento do tempo.
- Matriz de Probabilidade de Observação $B = [b_i(o_t)]_{N \times M}$, sendo $b_i(o_t)$ a distribuição de probabilidade de observação no estado i . Em outras palavras, $b_i(o_t)$ é a probabilidade de o_t ser observado, dado o estado atual s_i

$$b_i(o_t) \stackrel{\text{def}}{=} P(o_t | s(t) = s_i). \quad (2.20)$$

- A matriz de probabilidade de transição de estados $A = [a_{i,j}]_{N \times N}$, com $a_{i,j}$ representando a probabilidade de transição do estado i para o estado j , da mesma forma que definido na equação 2.19.

$$a_{i,j} \stackrel{\text{def}}{=} P(s(t+1) = s_j | s(t) = s_i). \quad (2.21)$$

O modelo é representado pela notação compacta em 2.22 [6], contudo, os parâmetros do modelo, A , B e π , ainda não tem seus valores conhecidos. A única informação conhecida são um conjunto de observações, que neste trabalho é dado pelas características do sinal EMG.

$$\lambda = (A, B, \pi) \quad (2.22)$$

Será necessário então, a partir desse conjunto de observações, estimar quais são os parâmetros do modelo HMM. Esse é o chamado Problema de Aprendizagem do modelo HMM, um dos problemas chave resolvidos pelo HMM. Este problema é resolvido através do algoritmo de Baum-Welch, um caso especial do Algoritmo de Maximização de Expectativa (EM), usado para a estimação do modelo λ a partir das observações.

Estimando os Parâmetros do HMM - Algoritmo de Maximização de Expectativa

O Algoritmo EM é principalmente usado para encontrar a verossimilhança local máxima dos parâmetros de um modelo estatístico no caso em que existem variáveis ocultas ou dados incompletos. Primeiramente, se considera um conjunto de parâmetros iniciais. Em seguida, vem o passo de expectativa (Passo E). Esse passo é usado para estimar os valores dos variáveis ocultas no modelo e envolve os dados observados para se adivinhar os valores dos dados que faltam. O próximo passo é o de maximização (Passo M), que completa os dados após o passo E e atualiza os valores dos dados ocultos. Os passos E e M são repetidos até os valores convergirem.

Este algoritmo utiliza de outros parâmetros e ferramentas intermediárias para a estimação do modelo, que serão definidas a seguir. Do Procedimento *Forward-Backward*, a variável "para frente" (*forward*), $\alpha_t(i)$ é definida como:

$$\alpha_t(i) = P(O_1 O_2 \dots O_t, q_t = S_i | \lambda) \quad (2.23)$$

Isso significa a probabilidade de estar em um dado estado S_i no momento parcial t , dado todas as observações que vieram nos momentos anteriores. O que o $\alpha_t(i)$ tenta representar não é o caminho exato que se leva ao estado S_i , mas sim a probabilidade em geral de se chegar ao estado S_i , dado as observações passadas, independente do caminho. É um parâmetro "para frente" pois o estado está a frente no tempo das observações.

De maneira similar, $\beta_t(i)$ é o parâmetro "para trás" (*backward*) que significa a probabilidade de estar em um dado estado S_i no momento parcial T , dado todas as observações nos momentos que virão após t , como mostrado em 2.24. É um parâmetro "para trás" pois o estado está atrás das observações, no tempo.

$$\beta_t(i) = P(O_{t+1} O_{t+2} \dots O_T, q_t = S_i | \lambda) \quad (2.24)$$

$\gamma_t(i)$ é definido como a probabilidade de se encontrar no estado S_i , no momento t , sabendo todas a sequência de observações.

$$\gamma_t(i) = P(q_t = S_i | O | \lambda) \quad (2.25)$$

Considerando que $\gamma_t(i)$ inclui todas as observações antes e após o estado S_i , é possível ver que essa função é a junção de $\alpha_t(i)$ e de $\beta_t(i)$, de forma que $\gamma_t(i)$ pode ser reescrito na seguinte forma, normalizada:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^N \alpha_t(i)\beta_t(i)} \quad (2.26)$$

Somando $\gamma_t(i)$ pelo tempo t , obtém-se uma quantidade que pode ser interpretada como o número esperado de vezes que o estado S_i é visitado, ou o número esperado de transições feitas

do estado S_i .

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{número de transições esperadas de } S_i \quad (2.27)$$

E por fim, $\xi_t(i, j)$ captura a probabilidade de se encontrar no estado S_i no momento t e de se encontrar no estado S_j no momento $t + 1$, dado o modelo e as observações. Um diferencial de ξ , se dá por ele envolver 2 estados diferentes e afirmando que não deseja-se saber apenas se está no estado S_i ou S_j , mas qual a probabilidade de estar em um e então transicionar para o outro.

$$\xi_t(i, j) = P(q_t = S_i, q_{t+1} = S_j | O, \lambda) \quad (2.28)$$

Observando a figura em 2.11, será possível entender com mais facilidade como calcular ξ , além de também ilustrar α e β . Observando o lado esquerdo da figura, é possível ver uma sequência de eventos que leva ao evento S_i . Portanto $\alpha_t(i)$ fica explícito como a probabilidade de estar em S_i , dado as observações dos eventos anteriores. Da mesma forma, do lado direito da figura, os eventos futuros que seguem de S_j , ilustrando $\beta_t(j)$ a probabilidade de se estar no estado j , dado toda a sequência de observações dos eventos futuros.

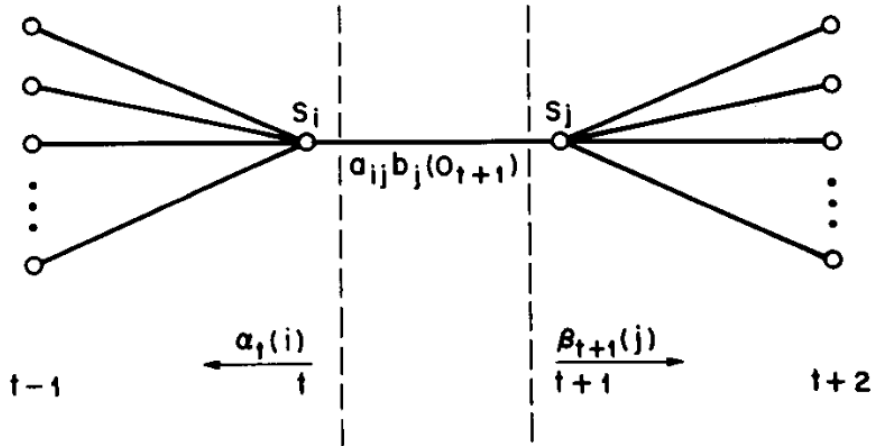


Figura 2.11: Ilustração da sequência de eventos levando as condições de 2.28. Imagem obtida de [6]

Agora seguindo para a parte central da figura 2.11, segue da equação 2.19 que para se transicionar do estado S_i para o estado S_j é $a_{i,j}$ multiplicado pela probabilidade de se ver a observação no momento $t + 1$, dado que o estado é S_j .

Enfim, a forma de se transicionar de S_i para S_j em termos de probabilidade é multiplicar os três momentos da figura, formando toda a trajetória do caminho. Em outras palavras, o que acontece é que existem dois pontos (estados) no espaço, e se olha para tudo que vem antes, tudo que virá depois, e conectar os pontos juntos. Logo, é possível escrever $\xi_t(i, j)$ pela fórmula 2.29. O termo do numerador é simplesmente uma representação da equação 2.28, enquanto o divisor é um fator

de normalização, dado que se tenta calcular uma probabilidade e não apenas a verossimilhança, e é dado pela soma de todas as probabilidades possíveis.

$$\begin{aligned}\xi_t(i, j) &= \frac{\alpha_t(i)a_{i,j}b_j(O_{t+1})\beta_{t+1}(j)}{P(O|\lambda)} \\ &= \frac{\alpha_t(i)a_{i,j}b_j(O_{t+1})\beta_{t+1}(j)}{\sum_{i=1}^N \sum_{i=1}^N \alpha_t(i)a_{i,j}b_j(O_{t+1})\beta_{t+1}(j)}\end{aligned}\tag{2.29}$$

Da mesma forma que acontece em 2.27, a soma de $\xi_t(i, j)$, pode ser interpretada como o número esperado de transições do estado S_i para o estado S_j .

$$\sum_{t=1}^{T-1} \xi_t(i, j) = \text{número de transições esperadas de } S_i \text{ para } S_j\tag{2.30}$$

Como $\gamma_i(t)$ foi definido como a probabilidade do estado S_i em um dado tempo, se for feita a iteração de $\xi_t(i, j)$ para todo j , é possível relacionar $\gamma_i(t)$ e $\xi_t(i, j)$.

$$\gamma_i(t) = \sum_{j=1}^N \xi_t(i, j)\tag{2.31}$$

Tendo definido todos os fatores importantes, finalmente é possível utilizar as equações de re-estimação definidas para o algoritmo de Baum-Welch.

$$\begin{aligned}\bar{\pi}_i &= \gamma_1(i) = \text{frequência esperada do estado } S_i \text{ no momento } (t = 1) \\ \bar{a}_{i,j} &= \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} = \frac{\text{número esperado de transições do estado } S_i \text{ para o estado } S_j}{\text{número esperado de transições do estado } S_i} \\ \bar{b}_j(k) &= \frac{\sum_{i=0}^T \gamma_t(i)}{\sum_{t=1}^T \gamma_t(i)} = \frac{\text{número esperado de vezes no estado } S_j \text{ observando } v_k}{\text{número esperado de vezes no estado } S_j}\end{aligned}\tag{2.32}$$

O modelo re-estimado é, portanto, definido como $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$. Portanto, para resumir, dado um modelo $\lambda = (A, B, \pi)$, é possível produzir um conjunto de parâmetros $\alpha_t(i)$, $\beta_t(i)$, $\gamma_t(i)$, $\xi_t(i, j)$. Ao mesmo tempo que dado um conjunto de parâmetros $\alpha_t(i)$, $\beta_t(i)$, $\gamma_t(i)$, $\xi_t(i, j)$ é possível produzir $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$. E esses são justamente os passos do Algoritmo EM, o passo E é dado

pela estimação dos parâmetros $\alpha_t(i)$, $\beta_t(i)$, $\gamma_t(i)$, $\xi_t(i, j)$, já o passo M é dado pela maximização do modelo, resultando em $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\pi})$

Todo o procedimento pode ser visto com mais detalhes em [34], que diz que o modelo inicial λ ou define um ponto crítico de verossimilhança, no caso onde $(\bar{\lambda} = \lambda)$ ou o modelo $\bar{\lambda}$ é mais provável ou verossímil. Outro ponto é que a re-estimação por esse algoritmo encontra apenas um máximo local.

Voltando ao problema deste trabalho, a partir da sequência de observações dadas pelas características do sinal EMG, seguindo o algoritmo descrito, é possível então estimar os parâmetros do modelo HMM. O processo de classificação agora se dá por, a partir de um conjunto de observações, descobrir qual o gesto que foi realizado, ou seja, qual o estado, ou quais estados, geraram aquela sequência de observações. Esse é outro dos problemas chave resolvidos pelo HMM, chamado de Problema de Decodificação. Este problema é resolvido pelo algoritmo de Viterbi.

Decodificando uma Sequência de Estados

O objetivo deste algoritmo é encontrar a sequência de caminhos que maximiza uma sequência de estados que são possíveis de se transitar, dado as observações e o modelo. É o equivalente de se maximizar a probabilidade de uma sequência de estados e de observações, dado o modelo.

$$P(s_1, s_2, \dots, s_T, O_1, O_2, \dots, O_T | \lambda) \quad (2.33)$$

Define-se uma nova quantidade δ_t que significa a maior probabilidade em um caminho $(s_1, s_2, \dots, s_{t-1})$ no tempo t , que considera as primeiras t observações e termina no estado S_i .

$$\delta_t(i) = \max_{s_1, s_2, \dots, s_{t-1}} P(s_1, s_2, \dots, s_T, O_1, O_2, \dots, O_T | \lambda) \quad (2.34)$$

É possível resolver isso usando por indução. O passo de indução segue:

$$\delta_{t+1}(i) = [\max_j \delta_t(j) a_{j,i}] b_j(O_{t+1}) \quad (2.35)$$

Deseja-se encontrar a melhor sequência de estados, estendendo de t para $t+1$, portanto procura-se o estado que maximizou $\delta_t(i)$ no momento anterior t , ou seja, faz-se uma busca entre todos os estados, por aquele que maximiza a transição do estado no momento anterior, para o estado no momento atual $t+1$. Então, $\delta_{t+1}(i)$ é essa probabilidade, multiplicado pela probabilidade $a_{i,j}$ de se transicionar para o próximo estado (ou seja, tomando o próximo passo), multiplicado pela probabilidade de se ver a observação emitida no passo seguinte $b_j(O_{t+1})$.

Mas não se deseja saber apenas um estado, mas sim todo o caminho de estados, portanto, deve-se manter salvo qual o estado que maximizou o resultado em cada passo de tempo, para que o caminho possa ser recriado posteriormente. Para isso, é utilizada a variável $\psi_t(i)$. Dessa forma, o algoritmo Viterbi executa os seguintes passos:

- Passo de Inicialização

$$\delta_1(i) = \pi_i b_i(O_1) \quad (2.36a)$$

$$\psi_1(i) = 0 \quad (2.36b)$$

No primeiro passo do tempo, $\delta_1(i)$ é inicializado como sendo a probabilidade (π_i) do primeiro estado ser o estado S_i , e observar a primeira observação $b_i(O_1)$. Como este é o primeiro momento, dado que não existem estados anteriores, $\psi_1(i)$ não guarda nenhum valor.

- Passo de Indução

$$\delta_t(j) = \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{i,j}] b_j(O_t) \quad 2 \leq t \leq T \quad (2.37a)$$

$$\psi_t(j) = \arg \max_{1 \leq i \leq N} [\delta_{t-1}(i) a_{i,j}] \quad 1 \leq j \leq N \quad (2.37b)$$

Neste passo, o objetivo é determinar δ_t no estado S_j , portanto qual é a probabilidade da melhor sequência de estados que chega no estado S_j . Para fazer isso deve-se olhar todos os estados de 1 a N que podem vir antes e olhar a máxima probabilidade de estar naquele estado no momento anterior, dado uma sequência válida de estados, e multiplica-se por $a_{i,j}$ e $b_j(O_t)$ da mesma forma que na definição em 2.35. Para todos os possíveis estados, o valor é salvo em $\delta_t(j)$. Logo, são salvos todos os possíveis melhores caminhos de estar sobre o próximo estado. E para salvar o caminho que pode ser recriado, $\psi_t(j)$ salva qual o foi estado que levou para esse caminho.

- Passo de Terminação

$$P^* = \max_{1 \leq i \leq N} [\delta_T(i)] \quad (2.38a)$$

$$s_T^* = \arg \max_{1 \leq i \leq N} [\delta_T(i)] \quad (2.38b)$$

$$s_t^* = \psi_{t+1}(s_{t+1}^*) \quad (2.38c)$$

No passo de terminação, descobre-se a probabilidade ótima P^* , ou a maior probabilidade da sequência de estados que foi possível transitar, que foi uma sequência válida de estados, será aquela sequência de estados que tiver $\delta_T(i)$. Ou seja, aquele $\delta_T(i)$ que tiver o maior valor, aquele é o estado final, que teve a maior probabilidade e teve uma sequência de estados válida que levou a esse caminho. E o Estado que é o final, é qualquer estado que corresponde ao maior $\delta_T(i)$.

Já, para descobrir recursivamente de quais os estados que vieram antes, utiliza-se a variável ψ_{t+1} para se fazer o caminho de volta. Portanto, qualquer estado em particular que se deseja descobrir s_t^* , que é o estado ótimo que veio antes, será o próximo estado que se obtém da variável ψ_{t+1} . Mais detalhes sobre o algoritmo e sobre os problemas chave do HMM podem ser lidos em [34].

Finalmente, agora é possível descobrir qual a sequência de gestos de mão (estados) que gerou o sinal EMG que foi obtido pelos sensores (observações). E assim, é possível classificar os gestos, que é o objetivo deste trabalho.

Capítulo 3

Métodos

O objetivo final deste trabalho é ter um classificador capaz de identificar, a partir de sinais eletromiográficos, qual o gesto realizado por uma mão. Portanto, nesta seção é descrita desde o aparato instrumental utilizado, até o método utilizado para o procedimento experimental e seus resultados, a partir de toda fundamentação teórica dada na sessão anterior.

3.1 Aparato Instrumental

3.1.1 Delsys - Trigno Avanti

O *Trigno Wireless Biofeedback System* é um dispositivo feito para deixar a detecção de sinais eletromiográficos (EMG) e de bio feedback fácil e confiável. O sistema transmite os sinais dos sensores Trigno Avanti para uma estação base usando protocolos de comunicação com sincronização temporal, ou, para aplicações móveis, os sensores podem se comunicar a partir de bluetooth BLE 4.2 com os dispositivos hospedeiros. Além disso, o sistema é capaz de integrar com equipamentos *3rd party* através de algumas interfaces e integrações, utilizando a Trigno API (*Application Program Interface*), ou com o Trigno SDK (*Software Development Kit*).

Algumas configurações podem ser selecionadas com relação à aquisição de dados dos sensores, como mostrado na tabela 3.1, tirada do manual de usuário [7].

Da tabela, é importante destacar que

1. A Faixa de medição, largura de banda e a taxa de amostragem para o sinal EMG e para sinais inerciais são configuráveis por *software*.
2. A orientação é calculada a bordo com algoritmos de fusão de dados
3. Um cálculo RMS pode ser feito a bordo pode ser utilizado para reduzir taxas de transmissão de dados e maximizar recursos de banda

O equipamento utilizado neste trabalho consiste em dois sensores *Trigno Avanti*, juntamente

Faixa de Medição Eletromiográfico (EMG)							
Faixa de Entrada ¹	11mV	ou	22mV				
Largura de Banda ¹	20 – 450Hz	ou	10 – 850Hz				
Janela RMS (opcional) ³	100ms						
Faixa de Medição da IMU							
Acelerômetro ¹	±2g	ou	±4g	ou	±8g	ou	±16g
Largura de Banda do Acelerômetro ¹	24Hz – 470Hz						
Giroscópio ¹	±250dps	ou	±500dps	ou	±1000dps	ou	±2000dps
Largura de Banda do Giroscópio ¹	24Hz – 360Hz						
Orientação ²	10Hz						

Tabela 3.1: Fonte: Manual de usuário Trigno [7]

de uma base para carregar, e um tablet que se conecta aos sensores para a aquisição dos dados através de aplicativo. Todo o sistema pode ser visto na figura 3.1.



Figura 3.1: Equipamentos para aquisição de dados da Delsys Inc. Os equipamentos incluem base de carga, dois sensores e um *tablet* com aplicações móveis para uso dos sensores.

Além dos sensores, o *tablet* possui aplicações móveis diversas para aquisição e análise dos sinais recebidos pelos sensores, sejam eles eletromiográficos ou inerciais. Foi utilizada principalmente a aplicação *EMG Plots* para coletar, plotar e salvar os dados. A figura I.4 no anexo I mostram as aplicações móveis disponíveis, a figura I.5 mostra o programa *EMG Plots* durante a aquisição de dados EMG.

Outros softwares importantes são o *Delsys File Utility* que converte os dados do formato cole-

tado para outros formatos de arquivo, incluindo o formato lido pelo outro software EMG Works, que serve para plotar, analisar e realizar cálculos em cima dos dados coletados. Imagens representando ambos os programas estão apresentadas no anexo I, nas imagens I.6 e I.7 respectivamente.

3.2 Implementação

Além dos *softwares* citados, foi criado um código em Python para todo o processo de classificação. O código foi desenvolvido em um ambiente virtual em Python com módulo *venv*. No módulo foram instaladas as dependências e pacotes necessários, tais como *matplotlib*, *pandas*, *numpy*, *statsmodels*, *hmmlearn*. Também foi utilizado o *JupyterNotebook* para auxiliar no desenvolvimento do código. Trechos do código estão anexados em I.

3.3 Aquisição de Dados

Como dito anteriormente, foram utilizados dois sensores *Trigno Wireless Biofeedback System* para a aquisição dos dados EMG. O posicionamento dos sensores no braço deve ser feito com muito cuidado para se adquirir o sinal da melhor maneira possível. Para isso, o sensor deve ser posicionado sobre os principais grupos musculares e de forma aos contatos ficarem paralelos às fibras musculares, a área em que os sensores serão posicionados também teve o excesso de pelos retirado e foi limpa com álcool momentos antes para reduzir a oleosidade da região. Os sensores foram posicionados sobre o braço na área afim de coletar o sinal dos músculos extensores e flexores da mão no braço.



Figura 3.2: Posicionamento dos Sensores

Com o objetivo de se classificar contrações estáticas, foram escolhidos 2 movimentos distintos da mão mais a posição de repouso para o reconhecimento. São eles (1) repouso, (2) mão fechada, (3) extensão do punho (mão levemente aberta nesse gesto). Sendo este o primeiro trabalho de reconhecimento de padrões, foram escolhidos gestos mais simples, por isso esses movimentos foram escolhidos por serem posições básicas, por diferenciarem extensão dos dedos e do punho e ativarem diferentes músculos.

O procedimento experimental foi realizado da seguinte maneira. Os sensores são ligados posici-



Figura 3.3: Gestos de mão para serem classificados. Respectivamente repouso, mão fechada e extensão do punho

onados num indivíduo, sobre os músculos especificados anteriormente. Os sensores são conectados à um *tablet* com o *software* de aquisição da *Delsys Inc.* e assim a aquisição pode ser iniciada. Os movimentos de mão são feitos para testar os dados recebidos.

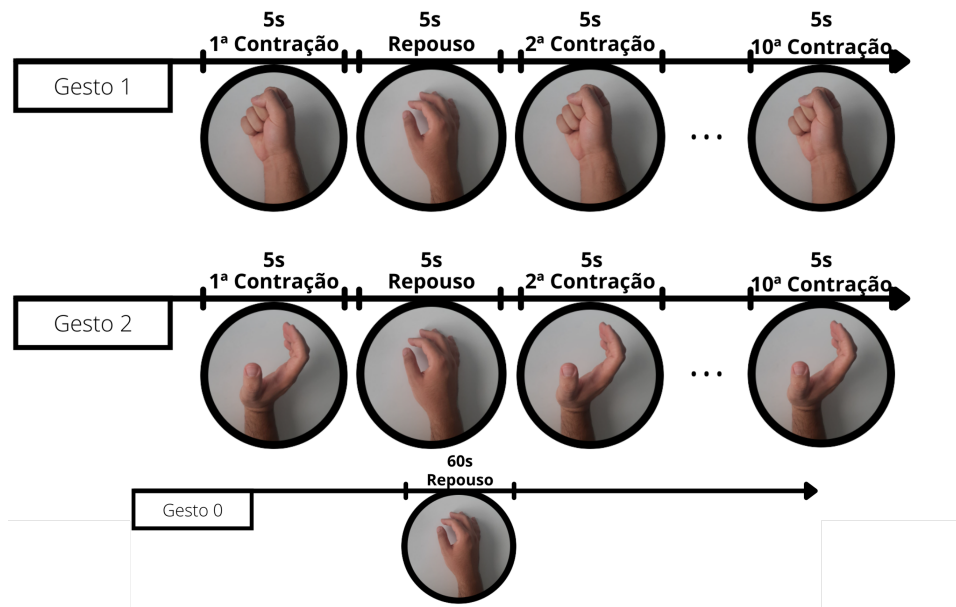


Figura 3.4: Protocolo para aquisição de sinais

Durante o procedimento, como ilustrado na figura 3.4, o indivíduo repete sequencialmente cada posição de mão por aproximadamente 5 segundos por um total de 10 repetições cada, com um intervalo de repouso de aproximadamente 4 segundos entre cada repetição. Ao fim da aquisição de um dos gestos, se faz um repouso de 1 minuto para evitar fadiga muscular, e então é realizada a próxima rodada para o próximo movimento. Também foram coletados algumas sequências apenas de repouso (sem nenhuma contração) para também ser separado e classificado. Assim, coletou-se em dois canais, uma sequência de dados para cada uma das posições que foi escolhida.

Foram coletados ao fim do experimento, os sinais EMG de 3 gestos/posições (repouso, mão fechada e punho flexionado) utilizando 2 canais de dados para cada. Cada gesto foi coletado 10 vezes por uma duração de 5s, totalizando uma coleção de 30 gestos para cada canal. A figura 3.5

ilustra a aquisição dos dados para a mão fechada.

Apesar de não serem utilizados para a classificação, ao fim do experimento, foram realizados alguns movimentos com o braço com a finalidade de se obter alguns dados inerciais, para demonstrar a capacidade do sensor. O resultado dessas medidas se encontrará no final de 4.1.

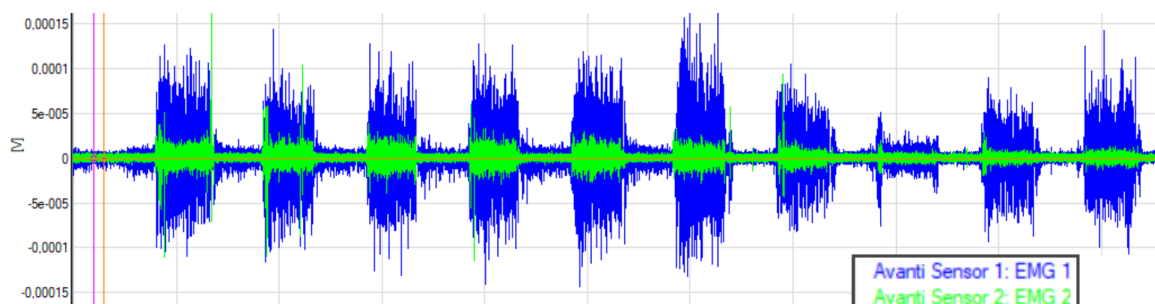


Figura 3.5: Sinal EMG para procedimento de mão fechada para ambos os canais EMG. Observa-se 10 repetições do gesto.

3.4 Pré-Processamento

Seguindo o que foi apresentado na seção 2.3.2.1, os sensores são configurados para aquisição de dados com uma frequência de amostragem de $1000Hz$. E antes mesmo da conversão do sinal para digital, o sinal analógico passa por filtros anti-aliasing e Butterworth passa-faixas na frequência de $20 - 450Hz$ a fim de se extrair as informações úteis do sinal EMG, e dispensar ruídos e interferências. Os sinais de ambos os canais são sincronizados e salvos.

Antes de partir para a segmentação dos dados, eles são visualizados através do software *EMGWorks Analysis* e então é feita uma compensação no deslocamento DC do sinal, para que o sinal EMG possua média 0.

3.5 Segmentação dos Dados

A primeira separação de dados feita é manual, como ilustrado na figura 2.8, utilizando o software *EMGWorks Analysis*. Para cada contração, e para o repouso, foram então separadas amostras de $5s$ que, considerando a taxa de $1000Hz$, resulta em 5000 pontos de dados. Cada amostra desta é salva em um arquivo ".csv" diferente. Como foram coletado 3 gestos diferentes, cada gesto foi realizado 10 vezes, e este procedimento é feito para ambos os canais, resulta-se então em 30 janelas por canal, totalizando 60 arquivos de dados diferentes.

Logo depois, o código para classificação é iniciado e todos os arquivos são carregados em vetores de séries temporais que são separados por gesto e por canal.

Em seguida é feita a segunda segmentação dos dados através do esquema OSW de janelas sobrepostas. Foi escolhido um tamanho de janela de $150ms$ com incremento de $50ms$, para cada amostra de gesto. Considerando então que cada amostra possui $5000ms$ e seguindo a fórmula 2.7,

calcula-se que cada amostra é separada em 98 janelas. O algoritmo em 1 é seguido, para todos os gestos, para ambos os canais.

Se são 98 janelas por gesto, e cada gesto é repetido 10 vezes, então se obtém 980 janelas por gesto. Define-se que as características extraídas de 1470 (490×3) delas serão usadas para treinamento do modelo enquanto a outra metade será usado para testar o modelo. Apesar de serem dois canais, como ambos os canais representam uma mesma amostra, então o número de janelas consideradas é o mesmo, a diferença se dá ao extrair as características.

Algorithm 1 Segmentação por OSW

```

tamanho_dados ← tamanho(serie_gestos)
tamanho_janela ← 150
incremento ← 50
n_janelas ← 1 + (tamanho_dados − tamanho_janela)/incremento
for j in range(n_repeticoes) do
    for i in range(n_janelas) do
        janela_gesto_c1 ← gesto_c1[j][(i*incremento) : (i*incremento) + tamanho_janela]
        janela_gesto_c2 ← gesto_c2[j][(i*incremento) : (i*incremento) + tamanho_janela]
    end for
end for

```

3.6 Extração das Características

As características extraídas do sinal são as observações do modelos HMM. São extraídas características de todas as janelas de ambos os canais, e então as características são concatenadas em um único vetor chamado de vetor de características. Para se descobrir os coeficientes do modelo AutoRegressivo, utilizou-se o módulo *statsmodels* em Python, como visto no código em I.2, no anexo I. Esse módulo possui classes e funções para estimação de diversos modelos estatísticos.

A implementação do cálculo do valor *RMS* é simples e feita aplicando a série temporal da janela na fórmula 2.8.

O pseudocódigo em 2 demonstra como é construído o vetor de características.

São obtidos então um vetor de características para cada uma das 980 janelas de cada gesto, para todos os 3 gestos, gerando 2940 vetores de características. As características extraídas de cada janela são, para os dois canais, o Modelo AutoRegressivo de ordem 4, e o valor RMS. Assim, o vetor de características é composto pelos coeficientes do modelo AR e pelo valor *RMS* como mostrado em 3.1, onde $a_{i,j}$ é o i -ésimo coeficiente do modelo AR para o canal j de dados, e RMS_j é o valor *RMS* calculado para aquela janela, para o canal j .

$$V = [a_{1,1} \ a_{2,1} \ a_{3,1} \ a_{4,1} \ a_{1,2} \ a_{2,2} \ a_{3,2} \ a_{4,2} \ RMS_1 \ RMS_2] \quad (3.1)$$

Algorithm 2 Gerando o vetor de características

```
vetor_janelas_1 ← janelas_canal_1           ▷ índice 1 se refere ao canal 1
vetor_janelas_2 ← janelas_canal_2           ▷ índice 2 se refere ao canal 2
for  $i \leq n\_janelas \times n\_repeticoes$  do
   $coefs\_AR\_1 \leftarrow parametros\_ar \leftarrow modelo\_AR(vetor\_janelas\_1[i], ordem = 4)$ 
   $coefs\_AR\_2 \leftarrow parametros\_ar \leftarrow modelo\_AR(vetor\_janelas\_2[i], ordem = 4)$ 
   $RMS\_1 \leftarrow RMS(vetor\_janelas\_1[i])$ 
   $RMS\_2 \leftarrow RMS(vetor\_janelas\_2[i])$ 
   $vetor\_caracteristicas[i] \leftarrow [coefs\_AR\_1, coefs\_AR\_2, RMS\_1, RMS\_2]$ 
end for
```

3.7 Classificação

A figura 3.6 ilustra a sequência de passos dados até o momento, uma sequência de movimentos gera o sinal visto, e a partir do sinal são extraídos um conjunto de observações. Possuindo todas as observações necessárias, I.1 mostra como é feita a separação entre o conjunto que será utilizado para o treinamento do modelo HMM e o conjunto que será usado para teste. São concatenados os primeiros 490 vetores de características para cada gesto, gerando 1470 vetores de características separados em repouso (1 – 490), mão fechada (491 – 980) e punho flexionado (981 – 1470). O mesmo é feito para teste, com a segunda metade dos dados para cada gesto.

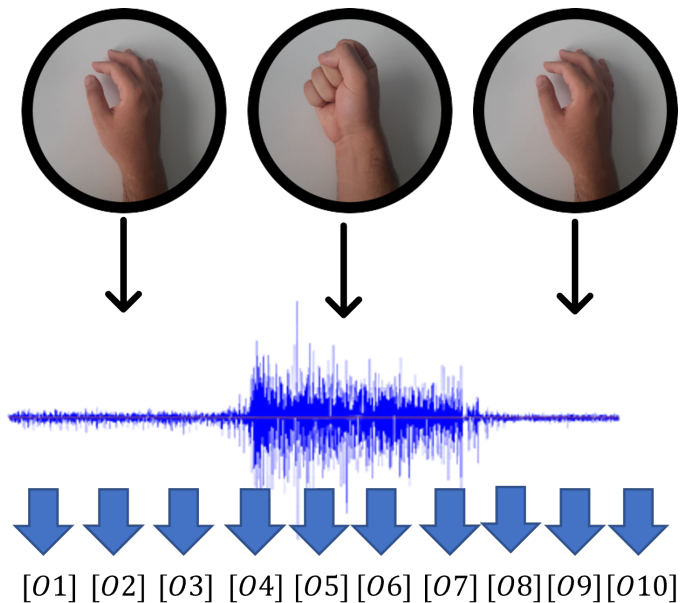


Figura 3.6: Gestos gerando um sinal eletromiográfico, separado em observações

Tendo os dados separados, utiliza-se o vetor de característica de dados para treinamento como a sequência de observações utilizada pelo algoritmo EM para se estimar o modelo HMM. O modelo recebe o vetor de característica como parâmetro e ajusta o modelo. Vale dizer que a implementação da biblioteca utiliza um modelo de Misturas Gaussianas para se estimar a distribuição de probabilidades do vetor de características, com matrizes de covariância entrelaçadas. Porém esse

modelo não faz parte do escopo deste trabalho e por isso não foi aprofundado.

Após a aprendizagem ter sido realizada, obtém-se um modelo HMM com todos os parâmetros estimados, portanto deve-se utilizar o restante dos dados para o teste do modelo. Os dados separados para teste são a sequência de observações e deseja-se descobrir, dado o modelo estimado, qual a sequência de estados que gerou aquelas observações. Isso é feito através do Algoritmo de Viterbi.

Após o treinamento, um gráfico com os valores logaritmos da verossimilhança calculados mostrando os valores para o treinamento e o para o teste é gerado e pode ser visto em 4.6. Além disso, sabendo de qual gesto o teste foi feito, é possível avaliar se o estado resultante da classificação do modelo HMM é o estado real ou não, tudo isso será mostrado e discutido na próxima sessão.

Capítulo 4

Resultados

4.1 Resultados e Discussão

O primeiro passo foi a coleta de dados, utilizando todo o aparato instrumental descrito na sessão 3.1 e o procedimento descrito em 3.3. Nas figuras 4.1 e 4.2, encontram-se os gráficos para as aquisições dos movimentos da mão fechada e extensão do punho.

Observando ambos os gráficos, nota-se que o sinal do canal 1 (em azul) para o punho estendido, para grande parte das repetições, possui amplitude maior que a do sinal para a mão fechada. A amplitude do sinal em 4.2 chega a ultrapassar $0,15mV$ na maioria de suas contrações, possuindo picos acima de $0,25mV$ enquanto a amplitude em 4.1 em poucas ocasiões ultrapassa o valor de $0,15mV$.

Já para o canal 2, é evidente que para ambas as contrações, a amplitude do sinal é mais baixa que do canal 1. Ainda, diferentemente do canal 1, a amplitude do canal 2 para a contração de fechar a mão é maior que a amplitude do canal 2 para a contração de se estender o punho, que não teve variação significativa. A dissemelhança entre os canais mostra a diferença de unidades motoras sendo ativadas para estes movimentos, para cada uma das regiões monitoradas pelos sensores, o que já era esperado. O sensor do canal 1 monitora músculos extensores, que possuem um sinal muito mais significativos para a extensão do punho, enquanto os músculos flexores, monitorados pelo canal 2, possuem quase nenhuma. Já para o movimento de fechar a mão, os sinais evidenciam a co-contracção dos músculos flexores e extensores, mesmo com a diferença de amplitude.

Essa diferença de músculos ativados faz parte da diferença de intenção identificada para cada gesto realizado, e uma forma clara de se diferenciar os movimentos. Para estes movimentos, esta diferença no sinal leva a crer que soluções mais simples, como um detector de limiar de amplitude, seria o suficiente para classificar os movimentos em questão. Para outros tipos de gestos, o sinal poderia não ser tão distinto em termos de amplitude, de forma que o reconhecimento de padrões poderia diferenciar enquanto detectores de limiar teria dificuldade em fazê-lo, ao mesmo tempo que para uma sequência aleatória de gestos, o classificador é mais robusto para decodificar os gestos do que soluções mais simples.

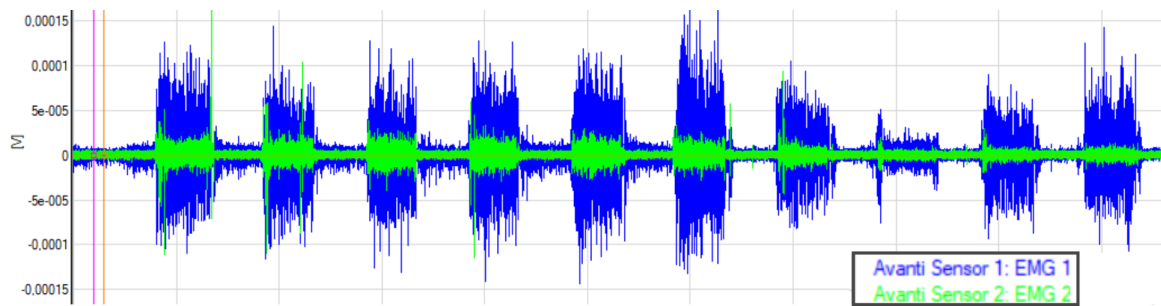


Figura 4.1: Sinal EMG para procedimento de mão fechada para ambos os canais EMG. Observa-se 10 repetições do gesto. Canal 1 em azul e Canal 2 em verde

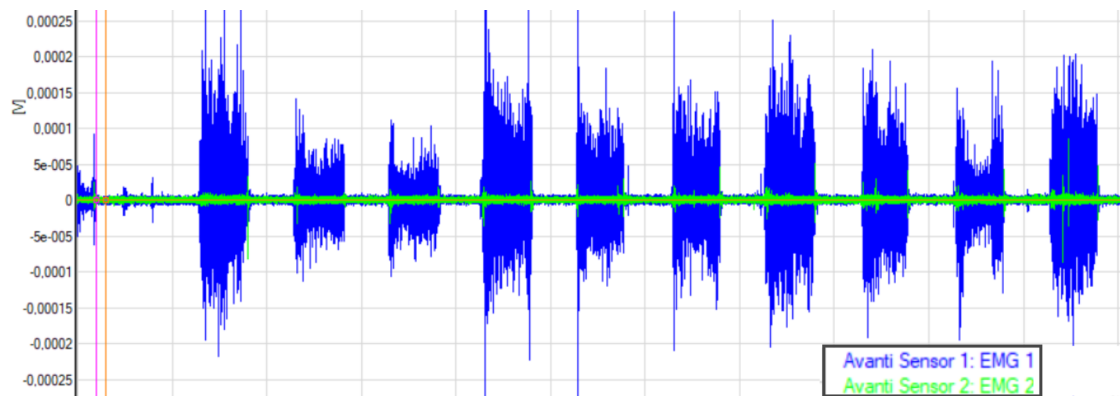


Figura 4.2: Sinal EMG para procedimento de flexão do punho para ambos os canais EMG. Observa-se 10 repetições do gesto. Canal 1 em azul e Canal 2 em verde

Ao fim da coleta dos dados, os dados são separados e as características são extraídas. É gerado então um vetor de características para cada janela e para cada movimento. O vetor de características possui os 4 coeficientes do modelo Auto Regressivo e o valor *RMS* para os dois canais, totalizando um vetor de 10 valores distintos. A imagem 4.3 mostra os vetores para uma das janelas dos gestos de repouso, mão fechada, e punho flexionado, respectivamente. Lembrando que a organização do vetor (os significados de cada valor) segue a equação 3.1.

```
[ 5.86983637e-01  3.55973539e-02  3.00673839e-02 -2.21214684e-02
 4.53140819e-01  1.58819946e-01  5.33832578e-03 -3.04636856e-02
 2.57769624e-06  2.02378113e-06]
[ 1.27016483e+00 -9.45589777e-01  4.28657152e-01 -2.23926463e-01
 7.09330561e-01 -2.93962776e-01  2.10806899e-01 -1.06227058e-01
 2.21831940e-05  3.16097496e-06]
[ 1.16902462e+00 -6.67025142e-01  2.28040106e-01 -1.90743401e-01
 5.72940624e-01  1.22637400e-02  1.04236994e-01  5.53325930e-02
 3.04238366e-05  2.75158450e-06]
```

Figura 4.3: Vetor de características de uma janela de dados para repouso, mão fechada, e punho estendido

Então, tendo esses vetores para todas as janelas, cada vetor de características é uma observação para o modelo. As observações são organizadas para serem dadas como entrada para o treinamento

do HMM, e assim, os parâmetros do modelo são estimados. Dos 3 parâmetros, vale a pena destacar dois, a matriz A e o vetor π . Na figura 4.4 é visto a matriz A de transição de estados estimada após o treinamento. Nota-se que a diagonal principal da matriz A , os elementos $a_{i,i}$, que são as probabilidades de se manter no estado em que se encontra, possuem os valores mais altos em suas respectivas linhas. Isso pode ser explicado pelo fato de que, dado um gesto, o modelo estima que a probabilidade de se manter no mesmo gesto é muito maior do que a de se trocar para outro gesto.

```
[[9.88118743e-001 5.05067113e-185 1.18812572e-002]
 [8.55853033e-239 1.00000000e+000 3.33217312e-155]
 [1.26851319e-002 2.24768641e-145 9.87314868e-001]]
```

Figura 4.4: Matriz A de transição de estados do modelo HMM

Já na imagem 4.5, é possível observar o vetor de Probabilidade Inicial π . A estimação desse modelo resulta que a probabilidade de se iniciar em um estado é aproximadamente o mesmo para todos.

```
[0.33333333 0.33333333 0.33333333]
```

Figura 4.5: Vetor π de Probabilidade Inicial

O primeiro teste do modelo se dá pela geração de um gráfico que mostra uma "pontuação" dada pelo log da probabilidade sobre o modelo para cada um dos conjuntos de observações que são fornecidos para o modelo, sendo os conjuntos de observações as características extraídas para cada gesto. Esse gráfico pode ser visto na figura 4.6. Em azul, são observados os resultados apenas para os dados de treinamento, em vermelho para o conjunto de observações apenas do repouso, em verde para a mão fechada, e em amarelo para a extensão do punho. Esses valores são calculados por uma função objetiva do GMM do modelo.

No eixo horizontal, a separação se dá apenas pelo índice que foi atribuído à observação, já no eixo vertical, a altura do ponto é o valor do log da verossimilhança daquela observação. É notável que existe uma área específica para cada um dos três gestos, mesmo que alguns pontos ainda possuam alturas similares no gráfico, é possível traçar uma linha horizontal que irá separar a maioria dos pontos para cada gesto.

Contudo essa informação não é a melhor para ser usada para avaliar o modelo, e nem é por esses valores que a classificação é feita. O próximo passo é utilizar uma função de predição que utiliza o algoritmo Viterbi para se encontrar a sequência de estados que melhor descreve um dado conjunto de observações. Ou seja, dado as características do sinal, descobrir quais os gestos que geraram tal observação.

Para cada gesto, ao fim do treinamento, ainda existem sequências de 490 observações que vão servir para testar o classificador. Ao fornecer uma sequência, é conhecido qual o gesto que foi realizado para toda a sequência, porém, essa informação o classificador não possui e irá retornar

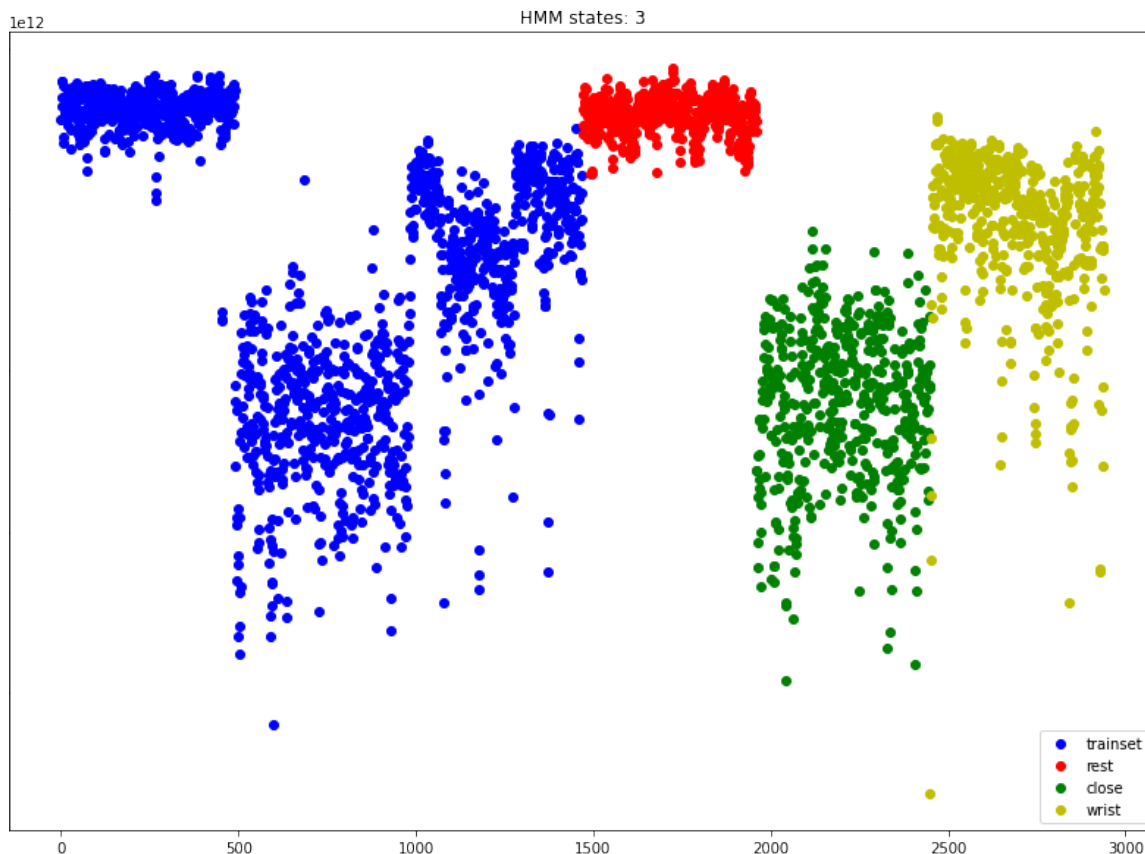


Figura 4.6: log-verossimilhança das observações

com a sequência de estados que é a mais provável que tenha sido realizada pelo conjunto de observações dados. Portanto, se é rotulada a sequência, é possível avaliar se a classificação acertou ou não o estado comparando com o rótulo e , conseqüentemente, avaliar a acurácia do classificador.

Uma sequência de testes rotulados é importante pois a biblioteca utilizada possui um problema conhecido como *label-switching problem*, onde ao treinar um modelo, os índices atribuídos a cada estado podem variar de uma execução para outra, portanto, fazer testes onde se conhece a sequência de estados antes de se utilizar o classificador é uma forma de relacionar o índice que o modelo atribui com o estado correto.

Esse procedimento então foi realizado, e foi computado quantos acertos e erros teve o classificador. na tabela 4.1, é possível ver os acertos e os erros para cada um dos gestos classificados. Nota-se que, pelo conjunto de teste para repouso, o classificador acertou todas as vezes para este gesto. O repouso possui um sinal sem muita variação de amplitude e sempre muito próximo de 0, portanto não é surpresa que tenha sido um sinal fácil de se classificar. Já o gesto da mão fechada teve uma acurácia de 95.51% para o conjunto utilizado para testes, enquanto o punho estendido teve uma acurácia de 90.82%. Todas tiveram uma acurácia acima de 90%, o que é um bom indicativo do desempenho do modelo, capaz de classificar os gestos em questão com um boa taxa de sucesso. Porém, é reconhecido que apenas 3 estados estão sendo classificados, o que está longe de ser o ideal, dado existem diversos outros gestos diferentes. Existe muito a ser feito futuramente,

	Gesto Correto	Gesto Errado	Acurácia
Repouso	490	0	100.00%
Mão Fechada	468	22	95.51%
Punho Flexionado	445	45	90.82%

Tabela 4.1: Acurácia do Classificador

e a inclusão de mais gestos para a classificação é essencial.

Para demonstrar o funcionamento *offline* do classificador, um novo teste foi realizado. Se separou um segmento de 20 segundos dos dados coletados, criando uma sequência que alterna entre um gesto, e repouso. Com essa nova sequência, repete-se a segmentação e a extração de características, mas com o modelo já treinado pela sequência de treinamento dos passos anteriores, se utiliza desse novo conjunto de dados para se obter os estados das mãos. O resultado da classificação é mostrado num gráfico juntamente com o sinal, para facilitar a visualização. Esse teste foi feito para os dois gestos e o resultado pode ser vistos na figura 4.7 para a mão fechada e na figura 4.8 para a extensão do punho.

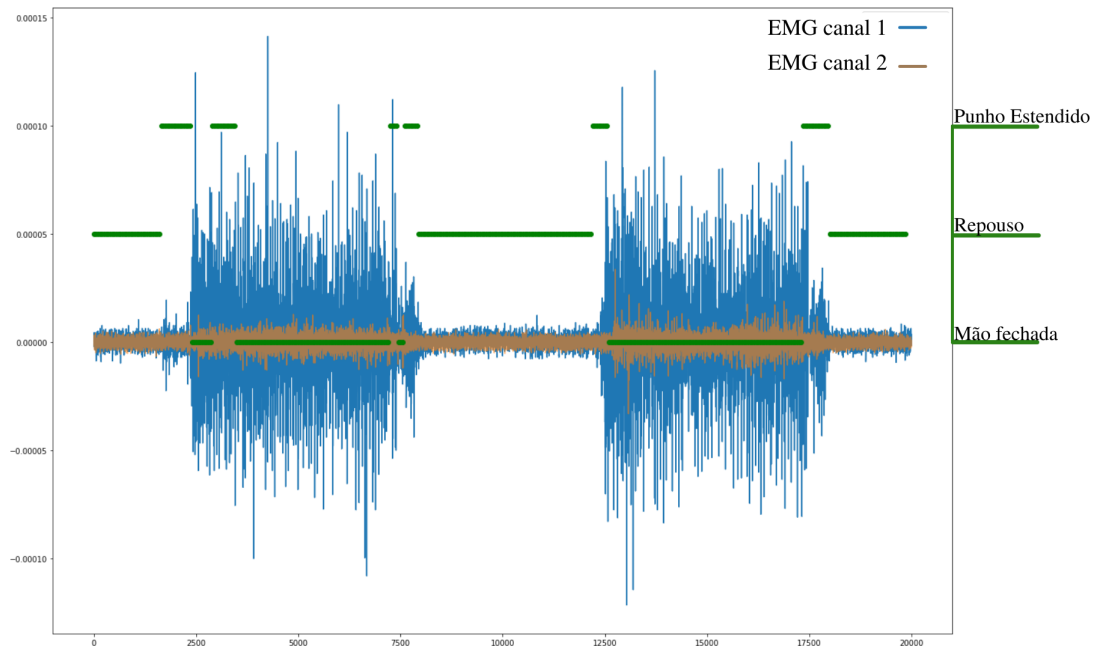


Figura 4.7: Classificação do gesto de fechar a mão

Observando ambas as figuras, é possível ver que a classificação de cada um dos gestos foi dentro do esperado, com alguns enganos principalmente nas áreas de transição entre o gesto e o repouso. Como esses momentos de transição são praticamente instantâneos em aplicações reais, é provável que esses erros não teriam tanto impacto., mas mesmo assim deve ser tratado. Apesar dos poucos gestos classificados, o resultado dessa classificação poderia ser explorado em tempo real. Uma prótese poderia utilizar o sinal da mão fechada para segurar objetos (principalmente objetos), de estender o punho para se estender a palma de uma mão robótica, para se acenar, por exemplo, e o repouso pode ser utilizado para se desligar alguns componentes com o objetivo de poupar bateria.

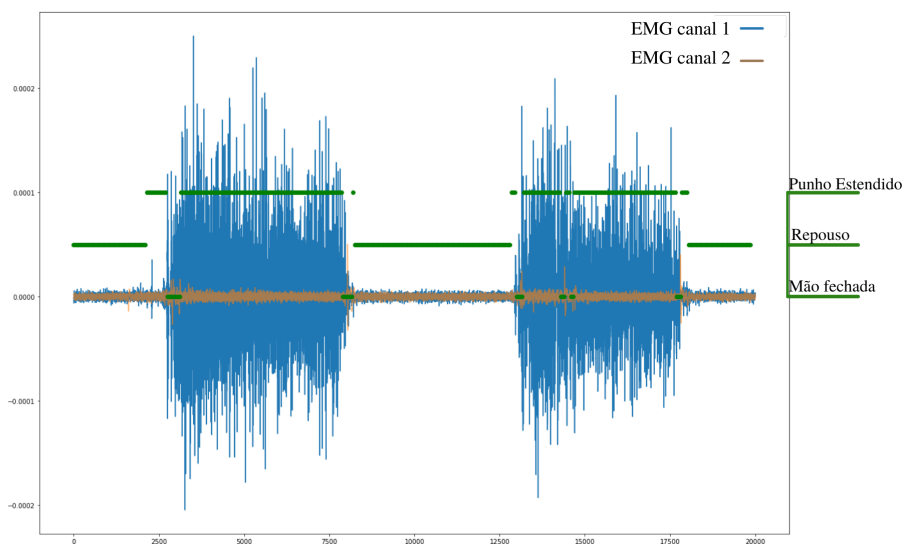


Figura 4.8: Classificação do gesto de estender o punho

A aplicação desse modelo em tempo real não é imediata e possui seus próprios desafios.

Aquisição de dados inerciais

Como dito anteriormente, ao fim do experimento foram coletados alguns dados inerciais do sensor Trigno Avanti. Com ele é possível selecionar se deseja coletar os dados dos sensores diretamente (acelerômetro e giroscópio), ou se os dados já devem vir em formato de orientação, seja ele em ângulos de Euler ou Quaternions. Não é possível obter diretamente a medida do magnetômetro da mesma forma que as medidas do acelerômetro ou giroscópio, porém, esse dado é utilizado pelo sensor para o cálculo da orientação a bordo.

Nas figuras 4.9, 4.10, se observa a medida do giroscópio e do acelerômetro coletados pelo sensor. Em nenhuma das medidas para se obter dados inerciais, se seguiu um protocolo específico de movimentos. Apenas se moveu o braço com o objetivo observar a variação nos eixos. Além disso, para cada tipo de medida (medidas dos sensores, orientação em Euler e Orientação em Quaternions), foi feita uma aquisição diferente de dados, de forma que um não reflete o outro.

Esses dois tipos de dados também podem ter suas características extraídas para serem englobadas pelo HMM. Por exemplo, é possível calcular o valor médio (MV) do sinal dentro da janela de processamento, e esse valor poderá ser computado no vetor de características.

Já nas figuras 4.11 e 4.12 são vistos, respectivamente, dados de orientação em Euler e em Quaternions. Um dado interessante fornecido pelo sensor é a acurácia da orientação em Quaternions, vista na imagem 4.13. É possível configurar o sensor para ler um dos dois tipos de orientação, mas como mostrado pelas equações 2.2 e 2.3, é possível fazer a conversão entre um e outro.

Apesar de não terem sido usadas medidas dos IMUs embutidos no sensor, essas medições ainda podem ser utilizadas para compor o conjunto de dados usados para treinar um modelo de reconhecimento de padrões, principalmente quando se trata de movimentos mais complexos da

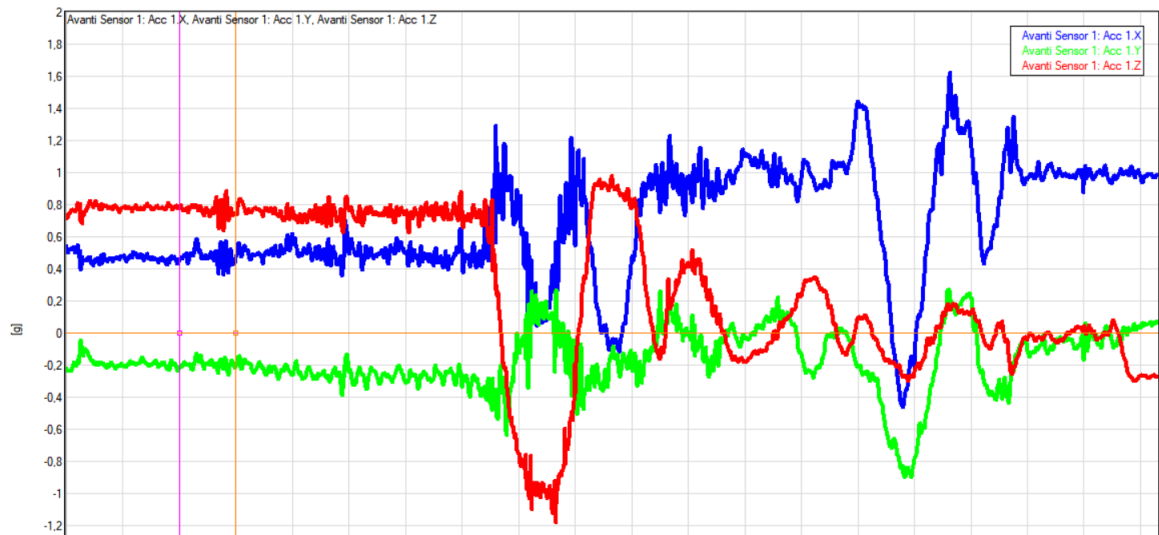


Figura 4.9: Aquisição de dados de acelerômetro utilizando Trigno Avanti

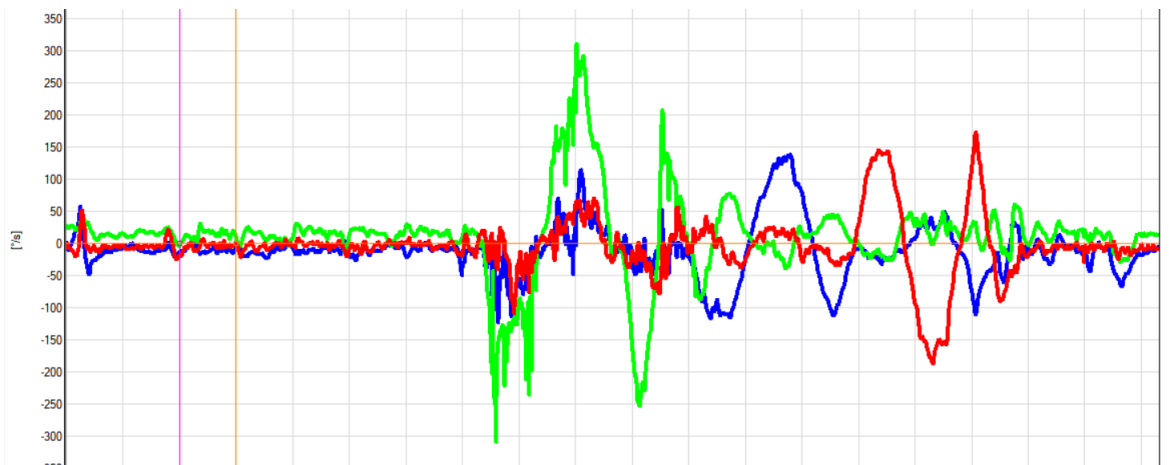


Figura 4.10: Aquisição de dados de giroscópio utilizando Trigno Avanti

mão e do braço.

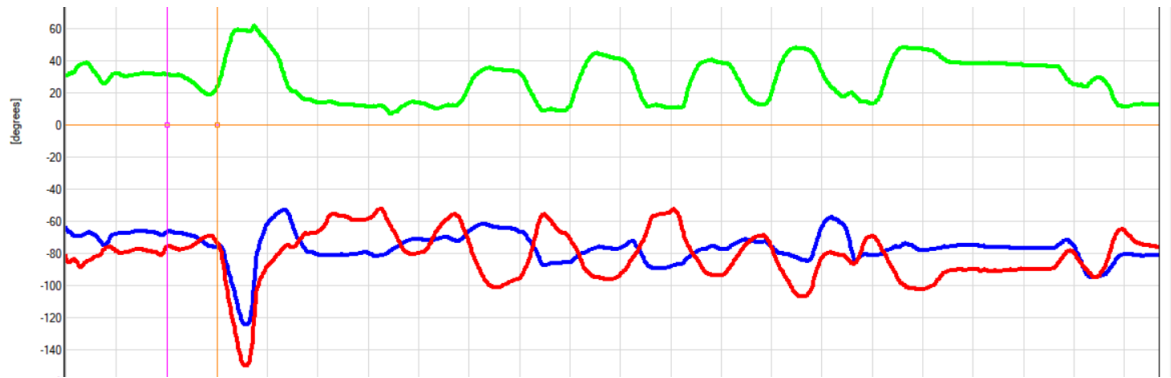


Figura 4.11: Aquisição de dados de acelerômetro utilizando Trigno Avanti

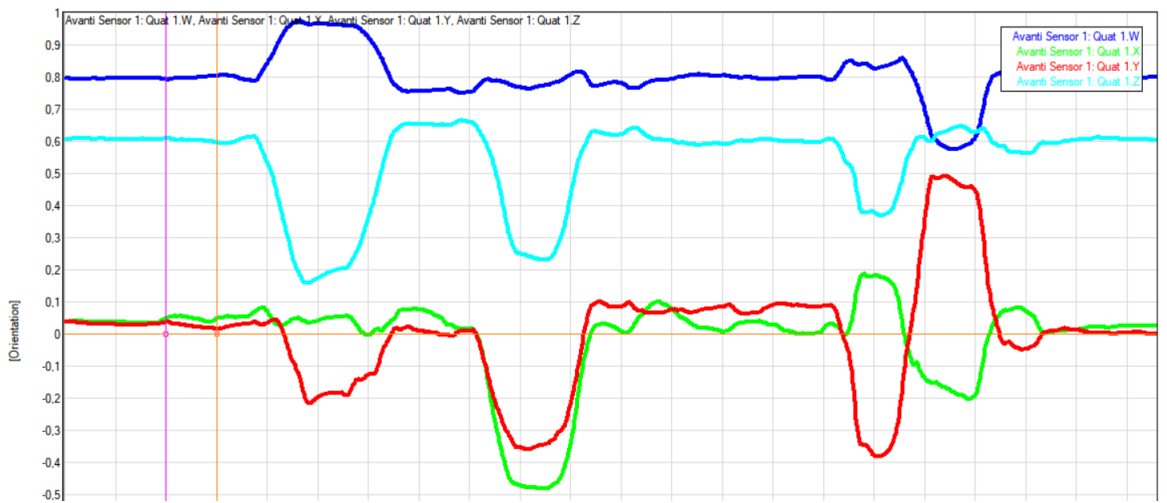


Figura 4.12: Aquisição de dados de giroscópio utilizando Trigno Avanti

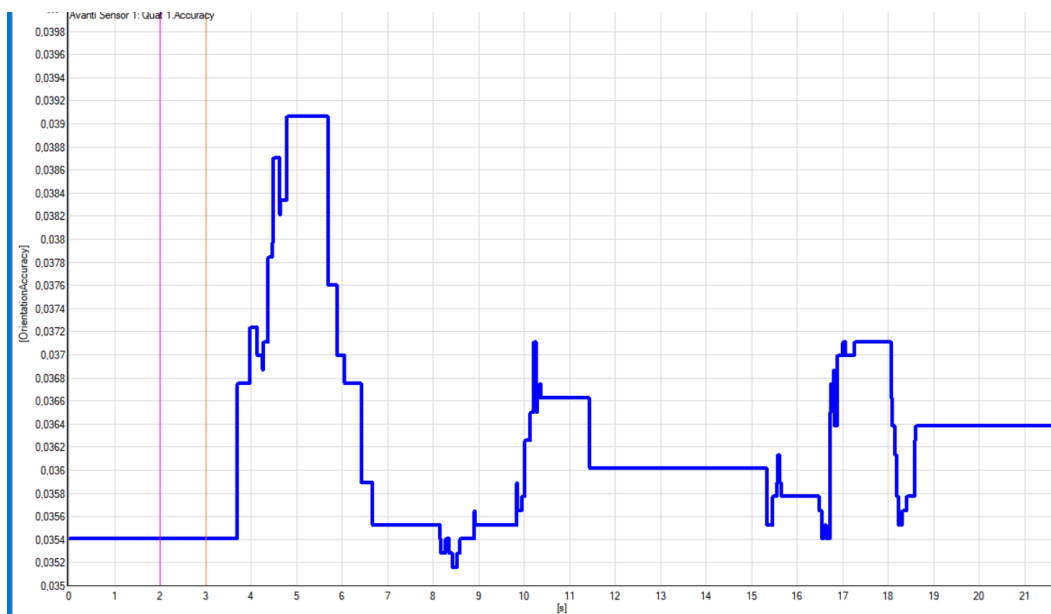


Figura 4.13: Aquisição de dados de acelerômetro utilizando Trigno Avanti

Capítulo 5

Conclusão

5.1 Conclusões e Trabalhos Futuros

Quando se trata de classificação de movimentos de mão existem diversas abordagens possíveis. A escolhida neste trabalho é através do reconhecimento de padrões. Para que esse reconhecimento seja feito, existem uma série de etapas que necessitam ser cumpridas. O objetivo deste trabalho era fechar o ciclo do reconhecimento de padrões passando por todas as suas etapas. Começando pela aquisição de dados, são coletados sinais mio-elétricos gerados pela atividade muscular. Foi necessário definir os tipos de movimentos que seriam classificados. Sendo este um primeiro trabalho de reconhecimento de padrões, apenas movimentos mais simples foram realizados. Após adquirido e condicionado o sinal, o passo seguinte é separar os dados em janelas de análise e de cada janela são extraídas características do sinal EMG. As características extraídas são o valor RMS e os coeficientes do Modelo Auto-Regressivo. Essas características são organizadas em um vetor de características e servem como as observações do modelo HMM. Em posse dessas observações, é possível estimar o modelo HMM e, com o modelo estimado, é possível classificar um gesto de mão dado o sinal que foi gerado por ele. Para o conjunto de testes realizados, foi avaliada a acurácia do modelo que resultou em valores acima de 90% para os três gestos sendo classificados.

Apesar do classificador ter um desempenho com acurácia de 90% para cima, vale ressaltar que o modelo é relativamente simples, classificando apenas 3 gestos estáticos e utilizando 2 canais do sinal EMG. O uso dos Modelos Ocultos de Markov é interessante pela possibilidade de se decodificar uma sequência de gestos dado o sinal coletado e das suas características, coisa que soluções mais simples não fariam com tanta facilidade. Além disso, a análise é feita a partir das características do sinal, que podem ser influenciadas por diversos fatores como amplitude, autocorrelação, variação no tempo e na frequência, entre outros. Existem muitos parâmetros que podem ser ajustados e adicionados ao modelo, aumentando a sua complexidade. Embora seja um modelo com poucos gestos, poderia ser adaptado a um sistema de tempo real, como uma prótese, capaz de replicar os gestos classificados.

Abrindo as portas para muitos trabalhos que virão a seguir, pode-se considerar este trabalho como um ponta-pé inicial, sendo este um primeiro trabalho no tema. Foi possível fechar o ciclo e

reproduzir todos os passos para realizar o reconhecimento de padrões para classificar alguns gestos de mão. Este trabalho é relevante pois seus resultados poderão ser explorados e utilizados em diversas aplicações, desde próteses robóticas de mão quanto outras formas de interação humano-máquina utilizando reconhecimento de gestos de mão, como por exemplo a reprodução dos gestos em ambiente virtual. Além disso, existe uma grande variedade de mudanças que podem ser feitas com o objetivo de melhorar a classificação de movimentos utilizando sinais eletromiográficos. É esperado que em trabalhos futuros mais gestos possam ser classificados, ao ponto que contrações dinâmicas também possam ser classificadas utilizando também os sinais inerciais do movimento. Outros pontos a serem explorados são outras regiões no braço, onde os sensores podem ser posicionados, outras características podem ser extraídas do sinal com o objetivo de aumentar a qualidade da classificação. Além de melhorar o modelo, deve-se adaptá-lo para testes em tempo real, para seguir com o desenvolvimento de próteses robóticas e para outras aplicações. Existem muitas formas de melhorar e evoluir o sistema em questão, e este trabalho é apenas o início.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] OPENBIONICS. Nossa história. 2020. Disponível em: <<https://openbionics.com/about/>>.
- [2] OPENHAND. dextrus. Disponível em: <<http://www.openhandproject.org/dextrus.php>>.
- [3] BENNETT, D. A.; GOLDFARB, M. IMU-Based Wrist Rotation Control of a Transradial Myoelectric Prosthesis. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, IEEE, v. 26, n. 2, p. 419–427, 2018. ISSN 15344320.
- [4] LUCA, G. D. *Fundamental Concepts in EMG Signal Acquisition Table of Contents*. 2003.
- [5] WANG, N.; LAO, K.; ZHANG, X. Design and Myoelectric Control of an Anthropomorphic Prosthetic Hand. *Journal of Bionic Engineering*, Jilin University, v. 14, n. 1, p. 47–59, 2017. ISSN 16726529. Disponível em: <[http://dx.doi.org/10.1016/S1672-6529\(16\)60377-3](http://dx.doi.org/10.1016/S1672-6529(16)60377-3)>.
- [6] RABINER, L.; JUANG, B. An introduction to hidden markov models. *IEEE ASSP Magazine*, v. 3, n. 1, p. 4–16, 1986.
- [7] DELSYS INCORPORATED. *Trigno® Wireless Biofeedback System User's Guide*. [S.l.], feb 2020.
- [8] SAÚDE, M. da. *Informações de Saúde*. Disponível em: <<http://www2.datasus.gov.br/DATASUS/index.php?area=0202>>.
- [9] PEIXOTO, A. M. et al. Prevalência de amputações de membros superiores e inferiores no estado de Alagoas atendidos pelo SUS entre 2008 e 2015. *Fisioterapia e Pesquisa*, v. 24, n. 4, p. 378–384, 2017. ISSN 1809-2950.
- [10] Ministério da Saúde. *Diretrizes de atenção às pessoas submetidas a amputação Ministério da Saúde*. [s.n.], 2013. 38 p. ISBN 9788533419810. Disponível em: <www.saude.gov.br>.
- [11] THURSTON, A. J. Paré and prosthetics: The early history of artificial limbs. *ANZ Journal of Surgery*, v. 77, n. 12, p. 1114–1119, 2007. ISSN 14451433.
- [12] KRASOULIS, A. et al. Improved prosthetic hand control with concurrent use of myoelectric and inertial measurements. *Journal of NeuroEngineering and Rehabilitation*, Journal of NeuroEngineering and Rehabilitation, v. 14, n. 1, p. 1–14, 2017. ISSN 17430003.
- [13] PROJETO Mão 3D. Disponível em: <<https://www.mao3d.com.br/>>.

- [14] HACKBERRY. exiii-hackberry. Disponível em: <exiii-hackberry>.
- [15] GEETHANJALI, P. Myoelectric control of prosthetic hands: State-of-the-art review. *Medical Devices: Evidence and Research*, v. 9, p. 247–255, 2016. ISSN 11791470.
- [16] DALLEY, S. A.; VAROL, H. A.; GOLDFARB, M. A method for the control of multigrasp myoelectric prosthetic hands. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, v. 20, n. 1, p. 58–67, 2012. ISSN 15344320.
- [17] NINAPRO Protocol. 2021. Disponível em: <<https://www.idiap.ch/webarchives/sites/www.idiap.ch/project/ninapro/>>.
- [18] ATZORI, M. et al. Building the Ninapro database: A resource for the biorobotics community. *Proceedings of the IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics*, n. December, p. 1258–1265, 2012. ISSN 21551774.
- [19] YANG, J.; PAN, J.; LI, J. SEMG-based continuous hand gesture recognition using GMM-HMM and threshold model. *2017 IEEE International Conference on Robotics and Biomimetics, ROBIO 2017*, v. 2018-Janua, p. 1509–1514, 2018.
- [20] CHAN, A. D. C.; ENGLEHART, K. B. Continuous myoelectric control for powered prostheses using hidden Markov models. v. 52, n. 1, p. 121–124, 2005.
- [21] MUKUNDAN, R. Quaternions : From Classical Mechanics to Computer Graphics , and Beyond. *Proceedings of the 7th Asian Technology Conference in Mathematics*, p. 97–106, 2002.
- [22] DAY, S. Important factors in surface emg measurement. *Bortec Biomedical Ltd Publishers*, p. 1–17, 01 2002.
- [23] TRONTELJ, J. V.; JABRE, J.; MIHELIN, M. Needle and Wire Detection Techniques. *Electromyography*, p. 27–46, 2005.
- [24] Asghari Oskoei, M.; HU, H. Myoelectric control systems—a survey. *Biomedical Signal Processing and Control*, v. 2, n. 4, p. 275–294, 2007. ISSN 1746-8094. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1746809407000547>>.
- [25] MCGILL, K. C. Surface electromyogram signal modelling. v. 42, 2004.
- [26] JARAMILLO-YÁNEZ, A.; BENALCÁZAR, M. E.; MENA-MALDONADO, E. Real-time hand gesture recognition using surface electromyography and machine learning: A systematic literature review. *Sensors (Switzerland)*, v. 20, n. 9, p. 1–36, 2020. ISSN 14248220.
- [27] LUCA, C. J. D. *Surface Electromyography: Detection and Recording*. 2002. 1–10 p.
- [28] LIANG, S. et al. Identification of Gesture Based on Combination of Raw sEMG and sEMG Envelope Using Supervised Learning and Univariate Feature Selection. *Journal of Bionic Engineering*, v. 16, n. 4, p. 647–662, 2019. ISSN 25432141.
- [29] ZSCHORLICH, V. Digital filtering of EMG-signals. n. April 1989, 2018.

- [30] SCHEME, E.; ENGLEHART, K. Electromyogram pattern recognition for control of powered upper-limb prostheses: State of the art and challenges for clinical use. *The Journal of Rehabilitation Research and Development*, v. 48, n. 6, p. 643, 2011.
- [31] ENGLEHART, K.; HUDGINS, B.; MEMBER, S. A Robust , Real-Time Control Scheme for Multifunction Myoelectric Control. v. 50, n. 7, p. 848–854, 2003.
- [32] SMITH, L. H. et al. Determining the Optimal Window Length for Pattern Recognition-Based Myoelectric Control : Balancing the Competing Effects of Classification Error and Controller Delay. *IEEE*, v. 19, n. 2, p. 186–192, 2011.
- [33] LI, H.; CHEN, X.; LI, P. Human-computer interaction system design based on surface EMG signals. *Proceedings of 2014 International Conference on Modelling, Identification and Control, ICMIC 2014*, Swinburne University of Technology, Australia, p. 94–98, 2015.
- [34] RABINER, L. R. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. v. 77, n. 2, 1989.

ANEXOS

I. CÓDIGOS E PROGRAMAS UTILIZADOS

```
Splitting Data between Train and Test

1 train_rest = []
2 test_rest = []
3
4 train_close = []
5 test_close = []
6
7 train_wrist = []
8 test_wrist = []
9
10 for i in range(int(n_features/2)):
11     #Create Training Set
12     train_rest.append( rest_features[i] )
13     train_close.append( close_features[i] )
14     train_wrist.append( wrist_features[i] )
15
16     #Create Test Set
17     test_rest.append ( rest_features[ int(n_features/2) + i])
18     test_close.append( close_features[ int(n_features/2) + i])
19     test_wrist.append ( wrist_features[ int(n_features/2) + i])
20
21
22 lengths = [len(train_rest), len(train_close), len(train_wrist)]
23 X_train = np.concatenate([train_rest, train_close, train_wrist])
24 X_rest = test_rest
25 X_close = test_close
26 X_wrist = test_wrist
27 X_test = np.concatenate([test_rest, test_close, test_wrist])
```

Figura I.1: Código para separar dados em conjunto de treinamento, e conjuntos de teste rotulado e não rotulado

```

1 from statsmodels.tsa.arima.model import ARIMA
2
3 def rms(a, wdw_size=150):
4     return np.sqrt(sum([a[wdw_size-i-1:len(a)-i]**2 for i in range(wdw_size-1)])/wdw_size)
5
6 rest_features = []
7 close_features = []
8 wrist_features = []
9
10 rest_features2 = []
11 close_features2 = []
12 wrist_features2 = []
13
14 for i in range(int(n_windows)*gesture_samples):
15     #extracting for rest gesture
16     rest_model = ARIMA(rest_hand_window[i], order = (4,0,0))
17     rest_fit = rest_model.fit(method='yule_walker')
18     rest_features.append(rest_fit.polynomial_ar)
19     rest_features[i] = np.delete(rest_features[i], 0)
20     rest_features[i] = -rest_features[i]
21
22     rest_model = ARIMA(rest_hand_window2[i], order = (4,0,0))
23     rest_fit = rest_model.fit(method='yule_walker')
24     rest_features2 = rest_fit.polynomial_ar
25     rest_features2 = np.delete(rest_features2, 0)
26     rest_features[i] = np.append(rest_features[i], -rest_features2)
27
28     rest_features[i] = np.append(rest_features[i], rms(rest_hand_window[i].to_numpy().flatten()), len(rest_hand_window[i]))
29     rest_features[i] = np.append(rest_features[i], rms(rest_hand_window2[i].to_numpy().flatten()), len(rest_hand_window2[i]))
30
31     #extracting for close gesture
32     close_model = ARIMA(close_hand_window[i], order = (4,0,0))
33     close_fit = close_model.fit(method='yule_walker')
34     close_features.append(close_fit.polynomial_ar)
35     close_features[i] = np.delete(close_features[i], 0)
36     close_features[i] = -close_features[i]
37
38     close_model = ARIMA(close_hand_window2[i], order = (4,0,0))
39     close_fit = close_model.fit(method='yule_walker')
40     close_features2 = close_fit.polynomial_ar
41     close_features2 = np.delete(close_features2, 0)
42     close_features[i] = np.append(close_features[i], -close_features2)
43
44     close_features[i] = np.append(close_features[i], rms(close_hand_window[i].to_numpy().flatten()), len(close_hand_window[i]))
45     close_features[i] = np.append(close_features[i], rms(close_hand_window2[i].to_numpy().flatten()), len(close_hand_window2[i]))
46

```

Figura I.2: Código para a extrair características das janelas para cada gesto para cada canal. No pedaço de código visto não inclui o terceiro gesto.


```

1 from hmmlearn import hmm
2 import matplotlib.pyplot as plt
3
4 def fit_hmm(n_components=3, n_mix=32):
5     # try:
6     # startprob = np.zeros(n_components)
7     # startprob[0] = 1
8     # print(startprob)
9     hmm_model = hmm.GMMHMM(n_components=n_components,
10                            n_mix=n_mix,
11                            covariance_type="tied",
12                            algorithm='viterbi',
13                            init_params="stmcw", params="stmcw")
14
15     # hmm_model.startprob_ = np.array(startprob)
16     hmm_model.fit(X_train, lengths=lengths)
17
18     train_scores = []
19     rest_scores = []
20     close_scores = []
21     wrist_scores = []
22
23     for i in range(len(X_train)):
24         train_score = hmm_model.score(np.reshape(X_train[i],[len(X_train[i]),1]))
25         train_scores.append(train_score)
26
27     for j in range(len(X_rest)):
28         rest_score = hmm_model.score(np.reshape(X_rest[j],[len(X_rest[j]),1]))
29         rest_scores.append(rest_score)
30
31     for j in range(len(X_close)):
32         close_score = hmm_model.score(np.reshape(X_close[j],[len(X_close[j]),1]))
33         close_scores.append(close_score)
34
35     for j in range(len(X_wrist)):
36         wrist_score = hmm_model.score(np.reshape(X_wrist[j],[len(X_wrist[j]),1]))
37         wrist_scores.append(wrist_score)
38
39     :
40     :
41     :
42     :
43     :
44
45
46
47
48
49
50
51
52
53     return hmm_model

```

```

1 classifier = fit_hmm(n_components=3, n_mix=16)
2 A = np.array(classifier.transmat_)
3 print(A)
4 B = np.array(classifier.weights_)
5 print(B)
6 pi = np.array(classifier.startprob_)
7 print(pi)

```

Figura I.3: Código treinamento do modelo HMM

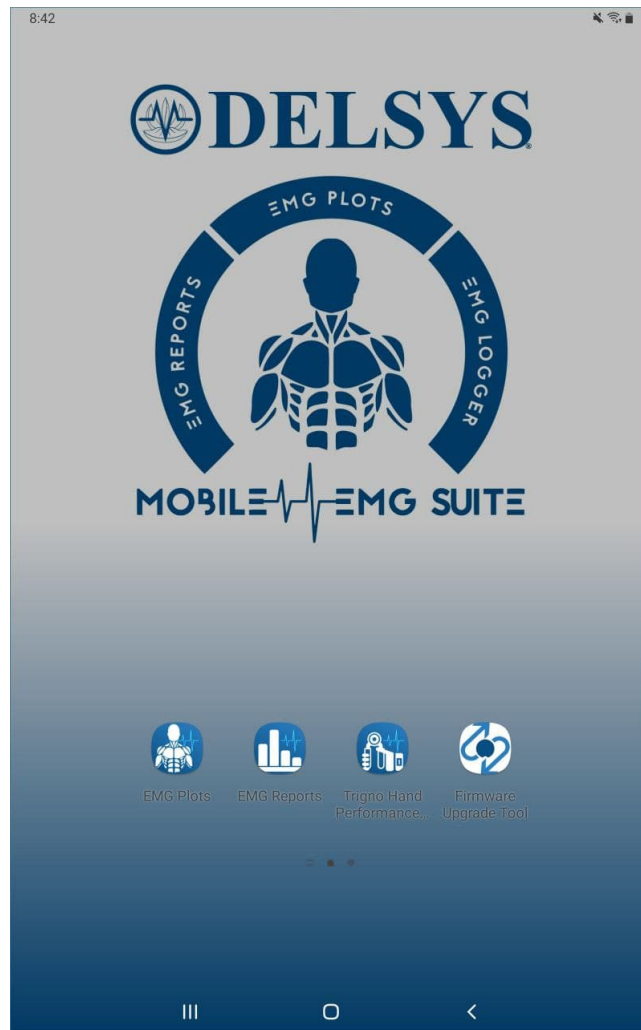


Figura I.4: Tela inicial do *tablet* que contém as aplicações móveis da Delsys Inc.

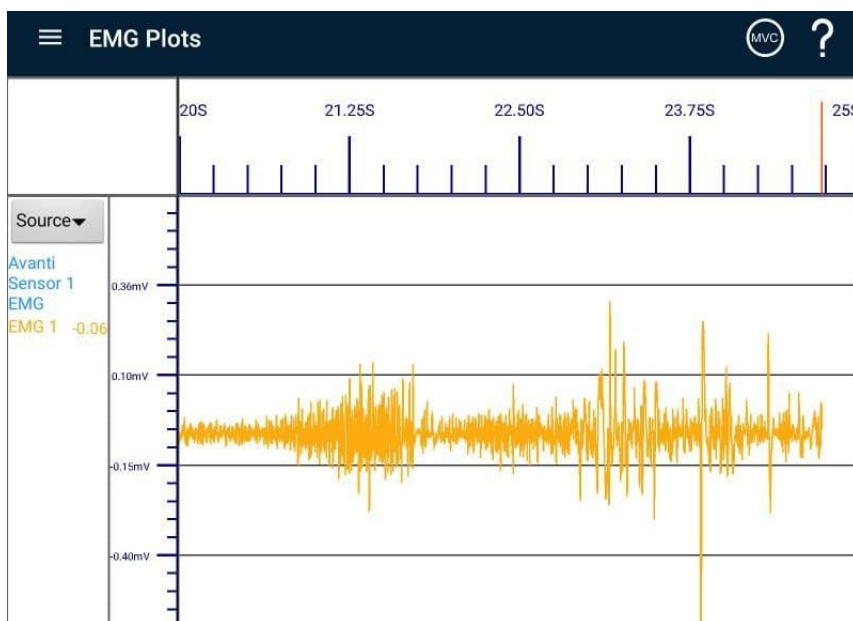


Figura I.5: Aplicação *EMG Plots* disponível no *tablet* durante a captura de sinais EMG

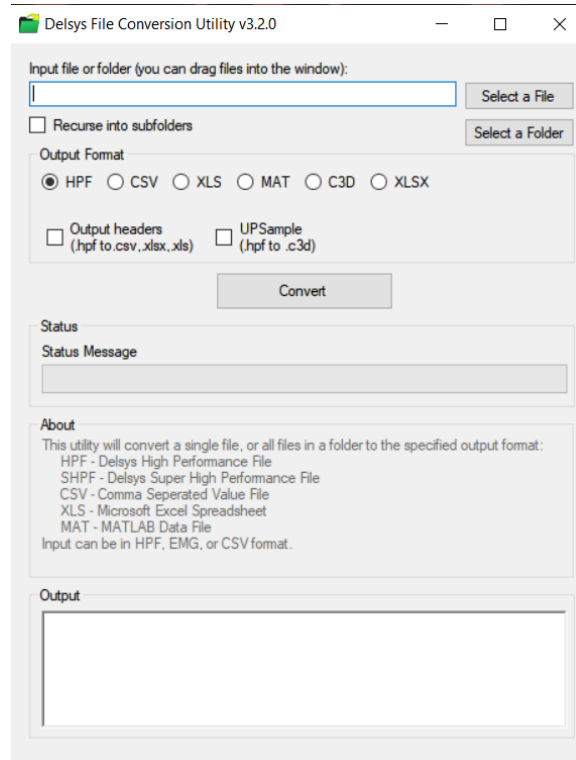


Figura I.6: Programa de conversão de dados

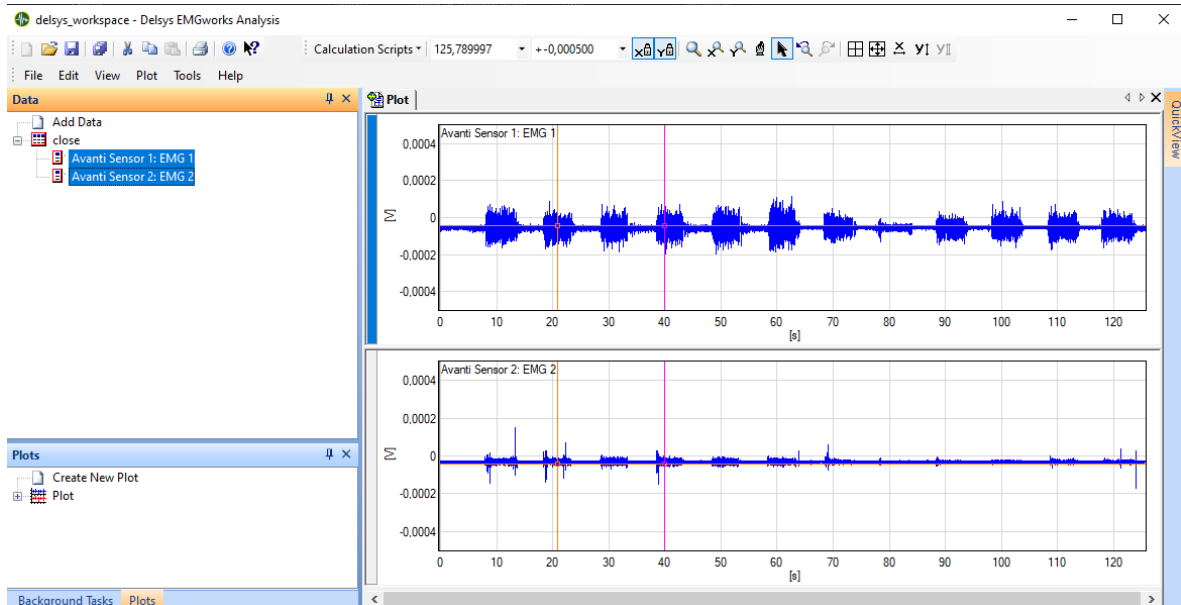


Figura I.7: Programa de visualização e análise de dados