

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Nibele Rodrigues

**ANÁLISE DE MOVIMENTOS DO
SEGMENTO MÃO-BRAÇO UTILIZANDO
LÓGICA *FUZZY***

Porto Alegre

2017

Nibele Rodrigues

ANÁLISE DE MOVIMENTOS DO SEGMENTO MÃO-BRAÇO UTILIZANDO LÓGICA *FUZZY*

Projeto de Diplomação apresentado ao Departamento de Engenharia Elétrica da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para Graduação em Engenharia Elétrica.

Orientador: Prof. Dr. Alexandre Balbinot

Porto Alegre

2017

Nibele Rodrigues

ANÁLISE DE MOVIMENTOS DO SEGMENTO MÃO-BRAÇO UTILIZANDO LÓGICA *FUZZY*

Projeto de Diplomação apresentado ao Departamento de Engenharia Elétrica da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para Graduação em Engenharia Elétrica.

Prof. Dr. Eng. Alexandre Balbinot
UFRGS

Prof^a. Dr^a. Eng^a. Leia Bagesteiro

Trabalho aprovado em: ____/____/____

Banca Examinadora:

Me^a. Eng^a. Karina Moura - U.F.R.G.S. _____

Me. Eng. Vinicius Cene - U.F.R.G.S. _____

Prof^a. Dr^a. Eng^a. Leia Bagesteiro - U.F.R.G.S. _____

Porto Alegre

2017

À minha mãe que não tomou café da tarde na mesa.

Agradecimentos

Agradeço principalmente à minha família, minha mãe por ter me dado as condições de chegar até aqui, minha irmã pelo suporte incondicional e aos meus sobrinhos pelo amor e carinho.

Agradeço também aos meus amigos, em especial a Franciele Prietsch, Wagner Cuty, Gabriela Zanin, Heider Carlos, Isnei Junior e Priscila Ben por estarem sempre ao meu lado, indicando o melhor caminho.

Aos meus colegas de curso e também amigos Braian, Fábio, Felipe Saraiva, Felipe Marques, Thamys e Derek . Por sofrerem e lutarem junto comigo nas adversidades que este curso pode proporcionar, sou muito grata por ter tido vocês como colegas. Um agradecimento também aos meus colegas de trabalho, pelos ouvidos atentos e braços abertos.

Aos grandes funcionários e professores deste departamento, que embora existam em pequeno número, se sobresaem aos demais.

Ao meu orientador, um agradecimento especial pela paciência, apoio e busca pela qualidade .

Resumo

Eletromiografia de Superfície (sEMG) é a captação dos sinais elétricos originários dos músculos ao serem comandados pelo cérebro, sendo assim, estes dados podem ser utilizados para identificar qual foi a intenção de um determinado movimento realizado por um sujeito. Desta forma, o objetivo deste trabalho é utilizar sinais de sEMG para analisar movimentos do segmento mão-braço utilizando lógica de decisão *Fuzzy*. Os dados utilizados são da base de dados NINAPRO que possui informações de 12 canais de eletrodos posicionados no segmento mão-braço, foram avaliados 12 sujeitos (6 amputados e 6 intactos) e 5 movimentos diferentes. A segmentação do sinal é feita através do método de detecção de limiar, a característica avaliada pelo algoritmo é a tensão elétrica RMS dos segmentos ativos e o agrupamento dos dados foi realizado através do *Fuzzy C – means Clustering*. O processamento dos dados é realizado pelo Sistema de Inferência *Fuzzy* com entradas *Fuzzyficadas* por três funções de pertinência do tipo Gaussiana e uma saída correspondente ao movimento realizado. Foram implementadas 1322 regras para o sistema e a taxa de acerto médio global resultante para todos os movimentos da base, sob a melhor configuração do Sistema de Inferência, foi de 77%. Para os 5 movimentos de interesse obteve-se taxa de acerto médio de 84% a 97%, ao passo que para os movimentos que não são de interesse deste trabalho, ou seja, cujas regras *Fuzzy* não foram especificadas para identificar, resultaram em uma taxa média de 72%.

Palavras-chave: Sinais Mioelétricos de Superfície (sEMG). Movimentos do Segmento Mão-Braço. Classificação de Movimentos por Lógica *Fuzzy*. Agrupamento *Fuzzy*.

Abstract

Surface Electromyography (SEMG) is the measurement of electrical signals originated from the muscle when they are commanded by the brain, as a result, those data could be used to identify which movement was intended by a subject. Therefore, the objective of this work is to use the SEMG signals to analyze hand-wrist movements using *Fuzzy* decision logic. The used data were from the database NINAPRO, that has the information from 12 electrode channels positioned in the hand-wrist region, 12 subjects were evaluated (6 amputees and 6 intact) and 5 different movements. The signal segmentation is made through threshold detection, the characteristic evaluated by the algorithm is RMS voltage from each active segment and the cluster data were made through *Fuzzy C – Means Clustering*. The data processing is made by the Fuzzy Inference System with *Fuzzyfied* inputs by three Gaussian type membership functions and one output corresponding to the performed movement. There were implemented 1322 rules to the system and the mean global success rate for all the sub-database movements, using the best *Fuzzy Inference System* configuration, was 77%. For the 5 movements of interest the mean success rate was from 84% until 97%, in contrast with the no-interest movements in this work, in other words, the ones that the *Fuzzy* rules were not specified to identify, result in a success rate of 72%. Par

Keywords: Superficial Myoelectric Signals (sEMG). Hand-wrist Movements. Movement classification with *Fuzzy* Logic. Fuzzy Clustering.

Lista de Figuras

Figura 1 – Constituição Neurônio Motor.	18
Figura 2 – Etapas de ocorrência do potencial de ação.	20
Figura 3 – Placa Motora.	21
Figura 4 – Músculo Esquelético: fascículos e fibras musculares.	22
Figura 5 – Retículo Sarcoplasmático.	23
Figura 6 – Composição de um sarcômetro.	24
Figura 7 – Exemplo de sEMG no domínio do tempo.	26
Figura 8 – Exemplo de sEMG no domínio da frequência.	26
Figura 9 – Função Pertinência Triangular ($x;1,5,9$).	29
Figura 10 – Função Pertinência Trapezoidal ($x;3,4,6,7$).	30
Figura 11 – Função Pertinência Gaussiana ($x;2,5$).	30
Figura 12 – Função Pertinência Bell ($x;2,5,2,5$).	31
Figura 13 – Exemplo da Implicação das saídas pelo método de mínimo (ou <i>min</i>).	34
Figura 14 – Exemplo da Implicação das saídas pelo método de produto (ou <i>prod</i>).	34
Figura 15 – Exemplo de métodos de Agregação: a) <i>max</i> ; b) <i>sum</i> e c) <i>probor</i>	35
Figura 16 – Lógica de decisão <i>Fuzzy</i> com método Mandani.	36
Figura 17 – Lógica de decisão <i>Fuzzy</i> com método Sugeno.	36
Figura 18 – Exemplificação gráfica dos métodos de <i>Defuzzyficação</i> : <i>centroid</i> , <i>bisector</i> , <i>mom</i> , <i>lom</i> e <i>som</i>	38
Figura 19 – Diagrama da Etapa de Pré-Processamento.	40
Figura 20 – Sinal dividido em janelas de 500ms.	43
Figura 21 – Fluxograma da Segmentação por detecção de limiar do sinal.	45
Figura 22 – Exemplo do procedimento de segmentação por detecção de limiar.	47
Figura 23 – Movimentos da Base de dados NINAPRO.	48
Figura 24 – Diagrama do processamento do Sinal.	50
Figura 25 – Fluxograma para determinar o valor RMS por segmento.	51
Figura 26 – Fluxograma do algoritmo de <i>Clustering</i> FCM.	53
Figura 27 – Diagrama de criação de um sistema de Inferência <i>Fuzzy</i>	56
Figura 28 – Lógica de ordenamento das Funções de Pertinência.	59
Figura 29 – Vetor de Adição de Regras.	62
Figura 30 – Lógica de Composição das Regras do Sistema de Inferência <i>Fuzzy</i>	64
Figura 31 – Lógica para caracterização do Movimento e Determinação do Acerto.	67
Figura 32 – Sinal de sEMG bruto no domínio do tempo.	69
Figura 33 – FFT do Sinal bruto no domínio da frequência.	70
Figura 34 – Resposta Simulada em Frequência do filtro Passa-Faixa.	71
Figura 35 – Sinal Filtrado no Tempo.	71

Figura 36 – FFT do Sinal Filtrado.	72
Figura 37 – Sinal Retificado no Tempo.	72
Figura 38 – FFT do Sinal Retificado (1 até 1000Hz).	73
Figura 39 – Sinal Normalizado no Tempo.	73
Figura 40 – FFT do Sinal Normalizado.	74
Figura 41 – Exemplo de Sinal Segmentado no Tempo.	75
Figura 42 – Vetor Segmentação por detecção de limiar x Sinal normalizado.	76
Figura 43 – Valor RMS dos 12 canais para os segmentos 20 até 25.	77
Figura 44 – Distribuição de amplitudes de todos os canais para voluntário 3 da sub-base II.	81
Figura 45 – Distribuição de amplitudes de todos os canais para voluntário 2 da sub-base III.	82
Figura 46 – Curva de graus de pertinência por <i>cluster</i> para canal 3 do voluntário 3 da sub-base II.	83
Figura 47 – Curva de graus de pertinência por <i>cluster</i> para canal 5 do voluntário 2 da sub-base III.	84
Figura 48 – Curva de graus de pertinência por <i>cluster</i> para o voluntário 3 da sub-base II.	86
Figura 49 – Curva de graus de pertinência por <i>cluster</i> para o voluntário 2 da sub-base III.	87
Figura 50 – Sistema de Inferência <i>Fuzzy</i> 12 entradas e 1 saída.	88
Figura 51 – Entrada 3 para voluntário 3 da sub-base II.	88
Figura 52 – Saída para todos os voluntários e sub-bases.	89
Figura 53 – Saída gerada para um conjunto de 26 regras para uma entrada de VRMS.	90
Figura 54 – Taxa de Acerto por Movimento por Sujeito Intacto.	96
Figura 55 – Taxa de Acerto por Movimento por Sujeito Amputado.	96
Figura 56 – Distribuição de amplitudes de todos os canais para voluntário 11 da sub-base II.	117
Figura 57 – Distribuição de amplitudes de todos os canais para voluntário 19 da sub-base II.	118
Figura 58 – Distribuição de amplitudes de todos os canais para voluntário 4 da sub-base III.	118
Figura 59 – Distribuição de amplitudes de todos os canais para voluntário 10 da sub-base III.	119
Figura 60 – Curva de graus de pertinência por <i>cluster</i> para o vol. 11 da sub-base II.	120
Figura 61 – Curva de graus de pertinência por <i>cluster</i> para o vol. 19 da sub-base II.	121
Figura 62 – Curva de graus de pertinência por <i>cluster</i> para o vol. 4 da sub-base III.	122
Figura 63 – Curva de graus de pertinência por <i>cluster</i> para o vol. 10 da sub-base III.	123

Figura 64 – Saída gerada para um conjunto de 26 regras para uma entrada de VRMS
- exemplo 2. 126

Lista de Tabelas

Tabela 1 – Variáveis que compõem os arquivos de dados da base NINAPRO. . . .	39
Tabela 2 – Detalhes do comando de <i>design()</i> de Filtros do <i>MatlabR2016a</i>	41
Tabela 3 – Detalhes do comando de <i>fdesign()</i> de Filtros do <i>MatlabR2016a</i>	41
Tabela 4 – Movimentos utilizados neste trabalho do Exercício 1 da sub-bases II e III da base de dados NINAPRO.	48
Tabela 5 – Identificação dos sujeitos da base de dados NINAPRO utilizados neste trabalho.	49
Tabela 6 – Detalhes do comando do FCM do <i>MatlabR2016a</i>	54
Tabela 7 – Detalhes do comando de <i>fit()</i> do <i>MatlabR2016a</i>	55
Tabela 8 – Detalhes do vetor <i>fioptions</i> do <i>MatlabR2016a</i>	55
Tabela 9 – Detalhes do vetor <i>prepareCurveData</i> do <i>MatlabR2016a</i>	55
Tabela 10 – Combinações dos métodos de Implicação, Agregação e Defuzzificação para Avaliação dos Resultados do Sistema de Inferência <i>Fuzzy</i>	66
Tabela 11 – Parâmetros utilizados no projeto do Filtro utilizando comando <i>design()</i> do <i>MatlabR2016a</i>	70
Tabela 12 – Combinações dos métodos de Implicação, Agregação e Defuzzificação para Avaliação dos Resultados do Sistema de Inferência <i>Fuzzy</i>	78
Tabela 13 – Taxa de acerto por voluntário por combinação do Sistema de Inferência <i>Fuzzy</i>	92
Tabela 14 – Taxa de Acerto por movimento.	93
Tabela 15 – Taxa de Acerto por sujeito por movimento.	94

Lista de Abreviaturas e Siglas

ACI	<i>Acetilcolina</i>
EMG	<i>Eletromiografia</i>
FCM	<i>Fuzzy C-means Clustering</i>
FP	<i>Função de Pertencência</i>
LG	<i>Lógica Fuzzy</i>
MUAP	<i>MotorUnitActionPotential</i>
MUAPs	<i>MotorUnitActionPotentialTrain</i>
RMS	<i>Root Mean Square</i>
SBE	<i>Sinal Bioelétrico</i>
sEMG	<i>Eletromiografia de Superfície</i>
SME	<i>Sinal Mioelétrico</i>
TA	<i>Tecnologia Assistiva</i>

Sumário

1	INTRODUÇÃO	15
2	REVISÃO BIBLIOGRÁFICA	17
2.1	Princípios Básicos sobre Sinais Mioelétricos	17
2.1.1	Neurônio Motor	17
2.1.2	Potencial de Ação Neural	18
2.1.3	Transmissão do Impulso	20
2.1.4	Músculo Esquelético	21
2.1.5	Potencial de Ação da Unidade Motora	24
2.2	Princípios Básicos sobre Eletromiografia	25
2.3	Princípios básicos sobre Lógica <i>Fuzzy</i>	27
2.3.1	<i>Fuzzy</i> ificação e Funções de Pertinência	27
2.3.2	Operadores <i>Fuzzy</i>	31
2.3.3	Sistema de Inferência <i>Fuzzy</i>	32
2.3.3.1	Sistema de Inferência <i>Fuzzy</i> : regras	33
2.3.3.2	Sistema de Inferência <i>Fuzzy</i> : Implcação	33
2.3.3.3	Sistema de Inferência <i>Fuzzy</i> : Agregação	34
2.3.3.4	Sistema de Inferência <i>Fuzzy</i> : Mandani e Sugeno	35
2.3.4	<i>Defuzzy</i> ificação	37
3	METODOLOGIA EXPERIMENTAL	39
3.1	Descrição da Base NINAPRO	39
3.2	Etapa de Pré-Processamento	39
3.2.1	Filtragem do Sinal	40
3.2.2	Retificação e Normalização do Sinal	41
3.2.3	Segmentação por Detecção de Limiar	42
3.3	Descrição dos Movimentos Avaliados e dos Voluntários	47
3.4	Etapa de Processamento do sinal	49
3.4.1	Obtenção dos parâmetros dos movimentos	50
3.4.2	<i>Clustering</i> dos parâmetros	52
3.4.3	Determinação da Função de Pertinência	54
3.4.4	Criação do Sistema de Inferência <i>Fuzzy</i>	55
3.4.5	Composição das Entradas do Sistema de Inferência <i>Fuzzy</i>	57
3.4.6	Composição das Saídas do Sistema de Inferência <i>Fuzzy</i>	60
3.4.7	Composição das Regras do Sistema de Inferência <i>Fuzzy</i>	61
3.4.8	Geração dos Resultados do Sistema de Inferência <i>Fuzzy</i>	65

4	RESULTADOS E DISCUSSÕES	69
4.1	Etapa de Pré-Processamento	69
4.1.1	Resultados da Filtragem do Sinal	70
4.1.2	Retificação e Normalização do sinal	72
4.1.3	Resultados da Segmentação do Sinal	74
4.2	Resultados da Etapa de Processamento do sinal	76
4.2.1	Resultados da Obtenção dos parâmetros dos movimentos	77
4.2.2	Resultados do <i>Clustering</i> dos parâmetros	81
4.2.3	Resultados da Determinação da Função de Pertinência	84
4.2.4	Resultados da Composição das Entradas e Saídas do Sistema de Inferência <i>Fuzzy</i>	88
4.2.5	Resultados da Composição das Regras do Sistema de Inferência <i>Fuzzy</i>	89
4.2.6	Taxa de acerto do Sistema de Inferência <i>Fuzzy</i>	91
5	CONCLUSÕES	97
6	PROPOSTA DE TRABALHOS FUTUROS	99
	REFERÊNCIAS	100
	ANEXOS	103
	ANEXO A – CÓDIGOS MATLAB	104
A.1	Código central	104
A.2	Função de Retificação, Normalização e Filtragem	104
A.3	Função de Segmentação	106
A.4	Função de Obtenção das Propriedades	108
A.5	Função de <i>Clustering</i> e Obtenção das Funções de Pertinência	109
A.6	Criação do Sistema de Inferência <i>Fuzzy</i>	111
A.7	Adição das Entradas	111
A.8	Função de avaliação do método de Implicação	113
A.9	Função de avaliação do método de Agregação	113
A.10	Função de avaliação do método de <i>Defuzzyficação</i>	114
A.11	Função de avaliação do Sistema <i>Fuzzy</i>	115
A.12	Função de geração da saída	116
	ANEXO B – DISTRIBUIÇÃO DE AMPLITUDES VRMS	117
B.1	Distribuição de Amplitudes VRMS voluntário 11 da sub-base II	117
B.2	Distribuição de Amplitudes VRMS voluntário 19 da sub-base II	118
B.3	Distribuição de Amplitudes VRMS voluntário 4 da sub-base III	118

B.4	Distribuição de Amplitudes VRMS voluntário 10 da sub-base III . . .	119
	ANEXO C – CURVA DE GRAUS DE PERTINÊNCIA	120
C.1	Curva de graus de pertinência para vol. 11 da sub-base II	120
C.2	Curva de graus de pertinência para vol. 19 da sub-base III	121
C.3	Curva de graus de pertinência para vol. 4 da sub-base III	122
C.4	Curva de graus de pertinência para vol. 10 da sub-base III	123
	ANEXO D – EXEMPLOS DE CRIAÇÃO DE REGRAS <i>Fuzzy</i> . . .	124
	ANEXO E – EXEMPLOS DE UTILIZAÇÃO DE REGRAS <i>Fuzzy</i> .	126

1 Introdução

De acordo com o Instituto Brasileiro de Geografia e Estatística (IBGE, 2010), estima-se que 7% da população brasileira apresenta algum tipo de deficiência motora. Em outras palavras, mais de 13 milhões de brasileiros sofrem todos os dias com algum tipo de restrição física. Portanto, em benefício desta parcela da população, nos últimos anos, diversas pesquisas estão em desenvolvimento para propor novas soluções ou melhorias em dispositivos de Tecnologia Assistiva (TA). Sendo assim, Tecnologia Assistiva é a área que promove soluções para pessoas portadoras de deficiência e para idosos, buscando melhorar a qualidade de vida desta parcela da população (GARCIA & FILHO, 2012).

Existem técnicas de desenvolvimento experimental para corrigir ou minimizar diversas situações diárias dos portadores de deficiência física. Dentre elas, destaca-se a Eletromiografia de Superfície (sEMG), que consiste na captação de sinais elétricos advindos de músculos do corpo ao serem comandados pelo cérebro. É uma das técnicas mais utilizadas e relevantes na área de instrumentação biomédica, pois acessa com facilidade (quando comparada a outros biosinais) os fenômenos fisiológicos que ocasionam movimento (DE LUCA, 1979).

Nas últimas décadas houve uma intensificação do estudo e desenvolvimento de aplicações com sinais biológicos, porém ainda são necessários novos estudos e o aprimoramento das técnicas e métodos de controle destes sinais para tornar as TA mais compatíveis com o usuário final (ATZORI, 2015).

Em função das colocações iniciais, este trabalho justifica-se pela aplicação de uma metodologia para análise de sinais de sEMG em uma área em constante desenvolvimento, assim como, de forte impacto social e futuramente econômico relacionado aos dispositivos de TA. Além disso, justifica-se também o desenvolvimento deste trabalho pela importância da inserção de alunos de graduação em áreas da bioengenharia onde o aprendizado e, portanto, a compreensão dos fenômenos biológicos e sua interface com a área das ciências exatas são essenciais para a resolução dos problemas existentes. Como por exemplo, neste trabalho, a realização da identificação de movimentos pelo uso das variáveis linguísticas.

Portanto, o objetivo principal deste trabalho é desenvolver um algoritmo capaz de identificar diferentes movimentos do segmento mão-braço (Flexão dos dedos em formato de punho, Supinação do punho, Flexão do punho, Extensão do punho e Desvio ulnar do punho) a partir de sinais musculares de 12 voluntários. Para a classificação dos movimentos utilizou-se a lógica de decisão *Fuzzy* através da implementação de um Sistema de Inferência *Fuzzy*, que possui regras linguísticas para interpretar a intenção do movimento.

Os objetivos secundários são:

-
- Comparar a taxa de acerto de Sistemas de Inferência *Fuzzy* para diferentes configurações de métodos de Implicação, Agregação e *Defuzzyficação*;
 - Avaliar o desempenho do Sistema criado em função dos voluntários;
 - Avaliar o desempenho do Sistema criado em função dos movimentos.

2 REVISÃO BIBLIOGRÁFICA

2.1 Princípios Básicos sobre Sinais Mioelétricos

Sinais bioelétricos (SBE) são gerados por células nervosas ou musculares, que sob certas condições podem ser excitadas para gerar um potencial de ação. Por definição, as informações elétricas geradas pela ação de diversas células constituem o sinal bioelétrico, sendo o sinal biomédico aquele que é gerado por cada célula e medido individualmente. Considerando-se que os mais importantes sistemas biológicos utilizam células excitáveis, é possível utilizar biosinais para estudar e monitorar as principais funções destes sistemas. Em função disto, biosinais são amplamente utilizados na maioria dos campos da biomedicina (GUYTON, 2006). Sinal mioelétrico (SME) é o sinal elétrico originado das células musculares resultantes da contração muscular (CRAM, 2011).

Em 1773, Walsh demonstrou que o músculo consegue emitir eletricidade, mas apenas em 1791 Galvani provou a relação direta entre eletricidade e contração muscular. Galvani publicou um estudo, através de experimentos com sapos, onde utilizava de placas e varas metálicas para causar movimentos dos membros dos animais. Em 1938, Matteucci utilizou o galvanômetro para descobrir que, na contração de um músculo esquelético intacto, ocorria um evento bioelétrico transiente. Assim foi descoberto o que mais tarde seria denominado de potencial de ação (BRONZINO & ELDERLE, 2012). O século XX foi marcado pelo estudo da forma dos sinais mioelétricos, dando origem a Eletromiografia (EMG) que utiliza de representações visuais do sinal mioelétrico. Os pioneiros no estudo de SME utilizando eletromiografia foram Hebert Gasser e Joseph Erlanger, ganhadores do prêmio Nobel em 1944 (CRAM, 2011).

2.1.1 Neurônio Motor

Neurônios são as células nervosas responsáveis por diversas funções do sistema nervoso, incluindo o controle da atividade muscular. O neurônio motor conduz potenciais de ação entre o Sistema Nervoso Central (SNC), o Sistema Nervoso Periférico (SNP), músculos e glândulas, através de nervos cranianos e espinhais (TORTORA, 2000).

Usualmente os neurônios são constituídos de três partes, como mostrado na Figura 1, de acordo com TORTORA(2000):

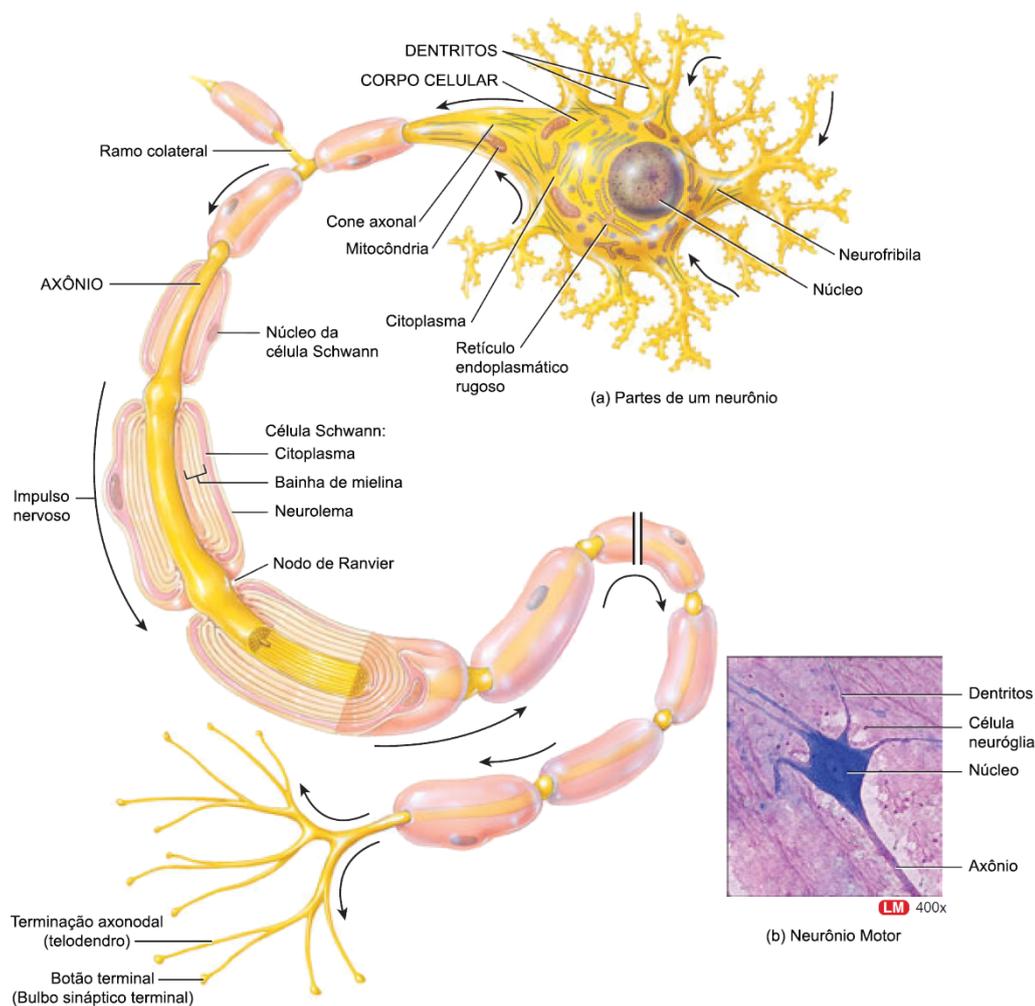
- 1) Corpo Celular: possui um núcleo circundado por citoplasma granular, é onde a maior parte das moléculas necessárias para o funcionamento do neurônio são sintetizadas;
- 2) Dendritos: possuem formato de “árvore” e emergem do corpo do neurônio, são

as partes de entrada ou recebimento do neurônio;

3) Axônio: são as ramificações de saída do neurônio, conduzem impulsos nervosos para os outros neurônios, células musculares ou glândulas.

O ponto de junção de um neurônio com o outro é o botão sináptico, que armazenam neurotransmissores, e também são responsáveis por determinar a direção na qual o sinal nervoso viaja (GUYTON, 2006).

Figura 1 – Constituição Neurônio Motor.



Fonte: Adaptado livremente de TORTORA, 2000.

2.1.2 Potencial de Ação Neural

O método usual do sinal nervoso circular pelo axônio é através da variação de potencial elétrico da membrana, sendo que, o conjunto destas variações é chamado de potencial de ação. As mudanças de tensão elétrica ocorrem, pois os neurônios são eletricamente e quimicamente excitáveis e a célula da membrana dos neurônios possui proteínas

específicas que facilitam o fluxo de íons. Com este fluxo, as cargas são redistribuídas e correntes elétricas são criadas alterando a diferença de potencial da membrana e por consequência causando uma despolarização (KANDEL *etal.*,2013).

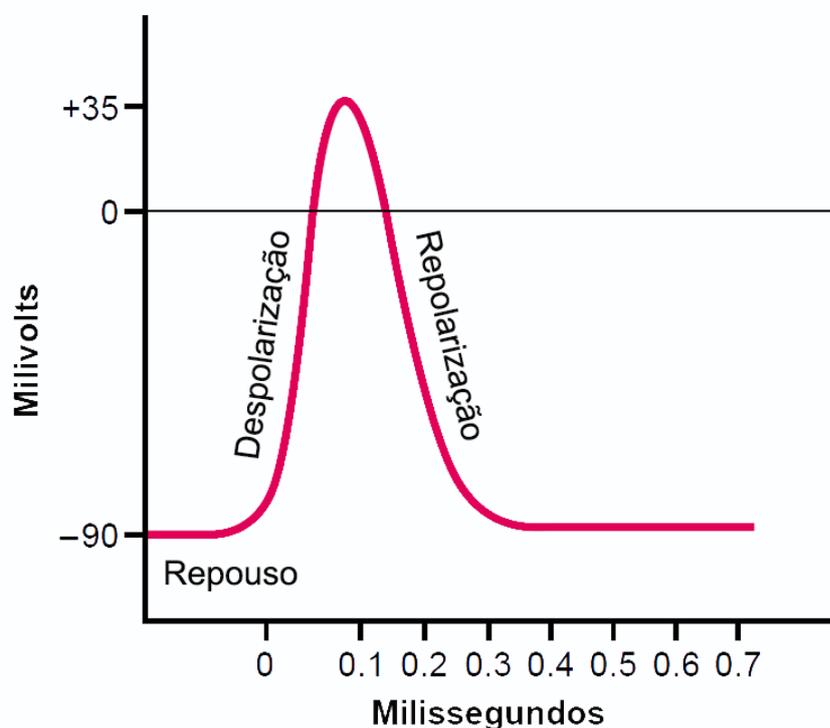
De acordo com GUYTON (2006), todo potencial de ação tem início a partir de uma tensão elétrica de repouso negativa, então ocorre uma mudança abrupta para um potencial positivo e, logo após, retorna para a situação de repouso. Conforme mostrado na Figura 2, é possível dividir a ocorrência do potencial de ação em três etapas distintas, são elas de acordo com GUYTON (2006):

1) Repouso: antes da ocorrência do potencial de ação a membrana é dita polarizada, pois possui potencial total em torno de -90mV . Isto ocorre, pois a membrana está impermeável para íons de sódio (Na^+), mas facilita a permeação dos íons de potássio (K^+). Sendo assim, há uma alta concentração de K^+ , o que provoca difusão para fora da membrana. Dada a carga positiva do potássio, a parte externa da membrana fica positivamente carregada. Dentro da fibra permanecem moléculas de proteínas, que são carregadas negativamente, a carga resultante da membrana fica negativa.

2) Despolarização: com o estímulo a membrana torna-se facilmente permeável por íons de Na^+ que, ao fluírem para dentro da fibra, tornam o potencial da membrana positivo.

3) Repolarização: a membrana torna-se novamente impermeável para os íons de Na^+ ; devido a polaridade positiva de seu interior, permanecendo impermeável para os íons de K^+ . Porém, devido à alta concentração dos íons de K^+ no interior, ocorre novamente a difusão destes para o exterior da membrana. Assim o potencial da membrana se torna negativo outra vez e a membrana está apta a transmitir um novo impulso nervoso.

Figura 2 – Etapas de ocorrência do potencial de ação.



Fonte: Adaptado livremente de TORTORA, 2000.

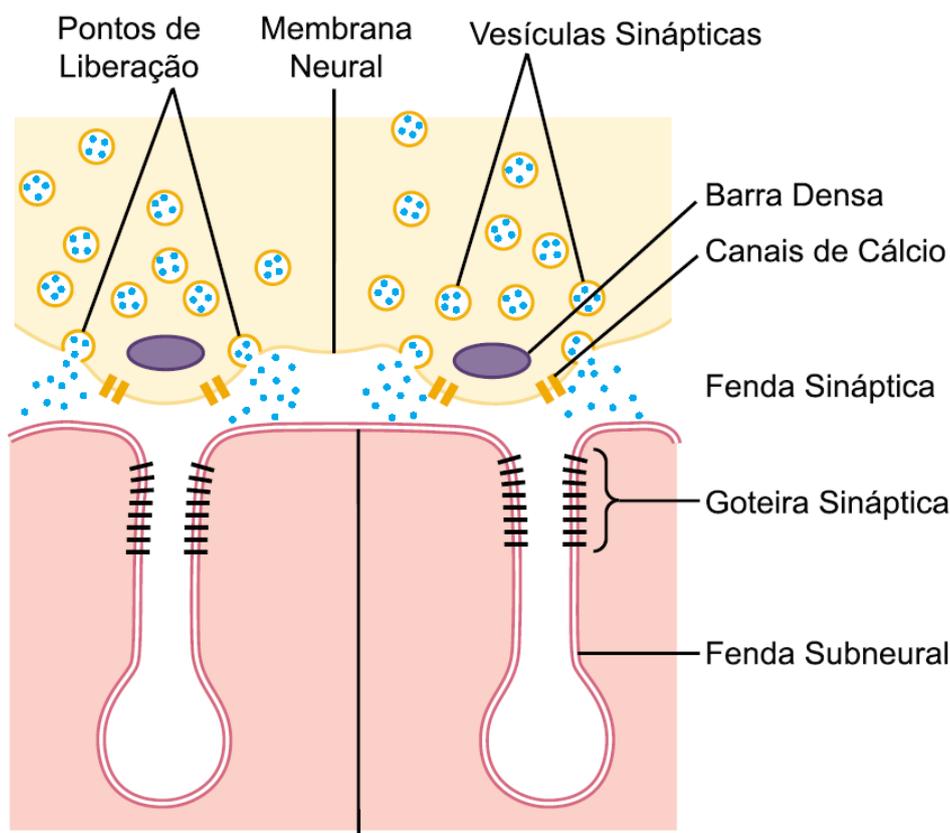
2.1.3 Transmissão do Impulso

A interface com que os neurônios se conectam com os músculos é chamada junção neuromuscular, que é a sinapse formada entre o fim do axônio e a fibra muscular. A fibra nervosa forma um complexo emaranhado de terminais nervosos que penetram na superfície da fibra muscular, esta estrutura é chamada de placa motora (GUYTON, 2006).

A fenda sináptica separa a fibra nervosa da placa motora com um espaçamento de 20 a 30nm. A penetração da membrana ocorre na goteira sináptica, onde no fundo possui numerosas pregas que formam fendas subneurais. As fendas subneurais possuem agregados da enzima colinesterase, que é capaz de quebrar o neurotransmissor acetilcolina (ACh) em ácido acético e colina. A Figura 3 ilustra a placa motora e suas partes. A chegada do impulso nervoso acarreta na liberação do neurotransmissor acetilcolina que é difundida do neurônio motor até a placa motora. As ligações de ACh abrem canais iônicos da placa motora e permite a passagem de cátions pequenos, principalmente Na⁺. Assim ocorre a despolarização da membrana muscular e, por consequência, o potencial de ação atua sobre as fibras musculares. O efeito da acetilcolina é breve, pois é quebrada rapidamente por enzimas nas pregas subneurais. Desta forma, ocorre a repolarização da membrana

(TORTORA,2000).

Figura 3 – Placa Motora.

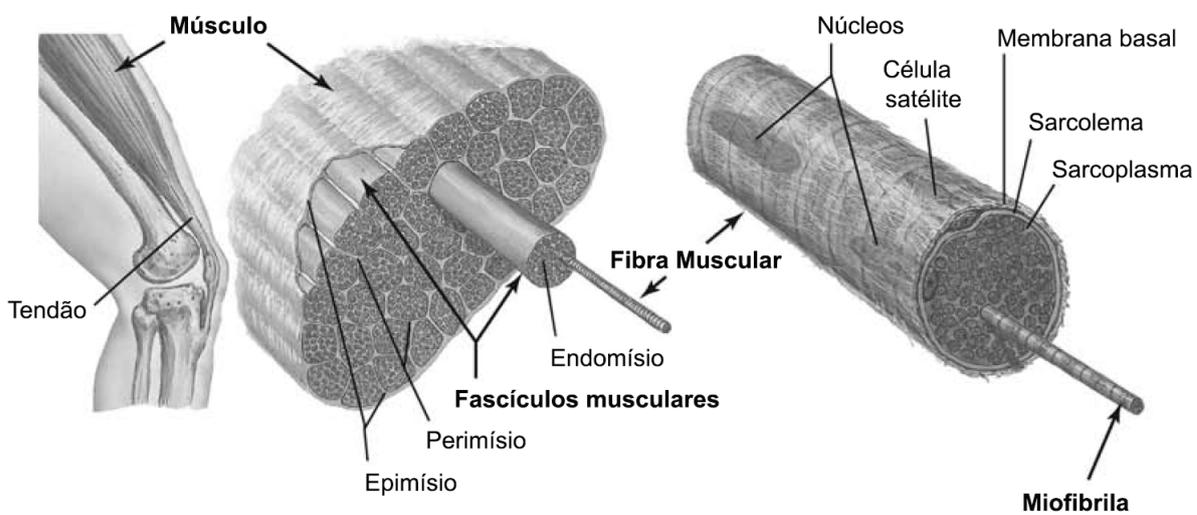


Fonte: Adaptado livremente de TORTORA, 2000.

2.1.4 Músculo Esquelético

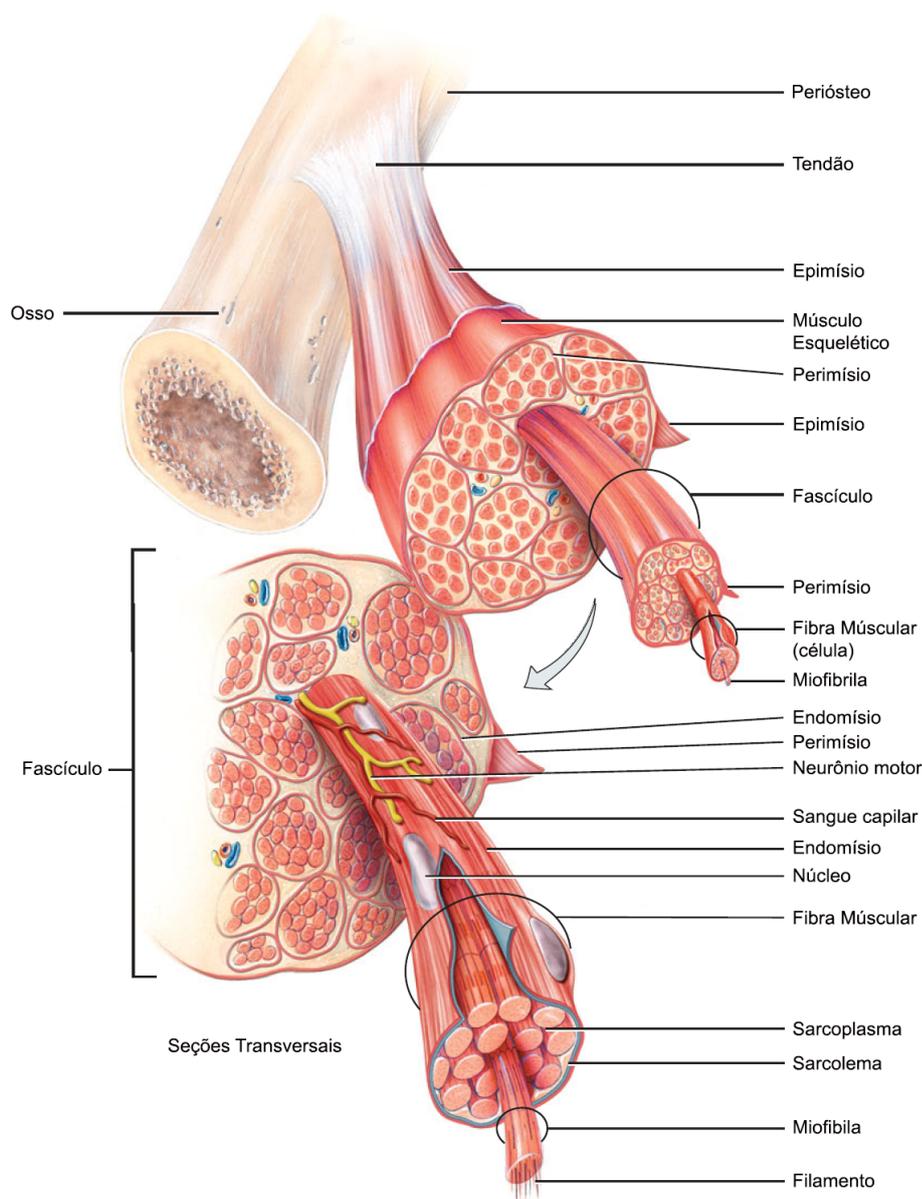
Músculos esqueléticos são ligados aos ossos e podem ser relaxados e contraídos voluntariamente. São compostos por centenas de milhares de células chamadas de fibras musculares, um aglomerado de fibras musculares caracteriza um fascículo e, por sua vez, um aglomerado de fascículos caracteriza um músculo esquelético, como demonstrado na Figura 4. Na Figura 4 também é possível observar que cada uma das fibras é formada por unidades cada vez menores, chamados de: miofibrilas, núcleo celular e retículo sarcoplasmático, envoltos por uma membrana chamada de sarcolema (TORTORA, 2000).

Figura 4 – Músculo Esquelético: fascículos e fibras musculares.



Fonte: Adaptado livremente de TORTORA, 2000.

Figura 5 – Retículo Sarcoplasmático.



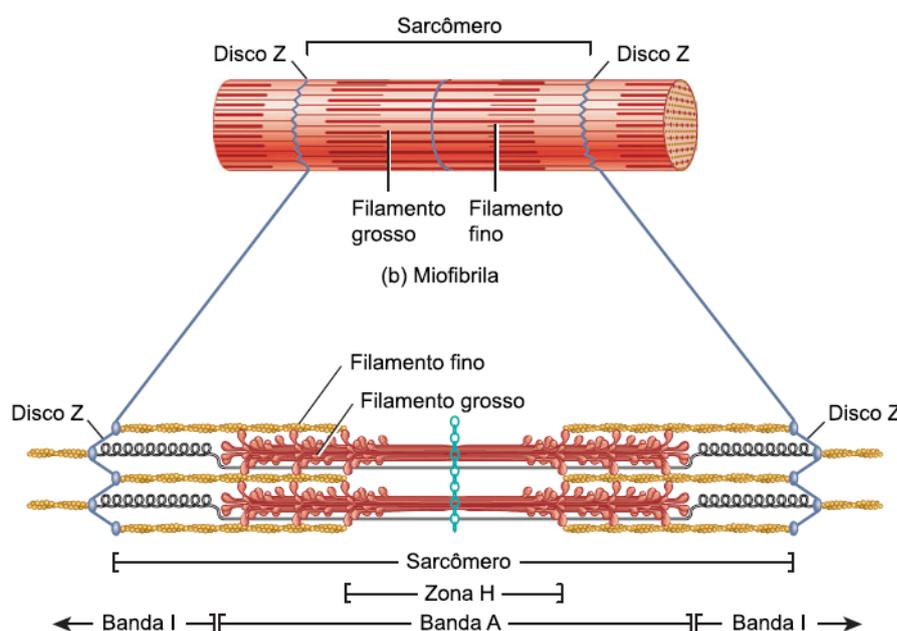
Fonte: Adaptado livremente de TORTORA, 2000.

O Retículo Sarcoplasmático é um retículo extensível e é extremamente importante para controlar a contração muscular, pois possui grande concentração de cálcio (Ca^{++}) que é liberado depois de um potencial de ação. As miofibras são compostas por filamentos chamados de miofilamentos que, por sua vez, são formados por filamentos de proteína miosina (miofilamento espesso) e actina (miofilamento delgado), conforme indica Figura 5 (GUYTON, 2006).

A unidade funcional básica dos músculos estriados são os sarcômeros, e são

separados uns dos outros por zonas de alta densidade de proteína chamadas de disco Z. O sarcômetro é composto pela alternância de bandas escuras A e bandas claras I, que dão a aparência estriada da fibra muscular. A banda A se estende por todo o comprimento do filamento espesso, mas apenas a zona H está livre de filamentos delgados. A banda I é formada apenas por filamentos delgados. A Figura 6 identifica a composição de um sarcômetro (TORTORA,2000).

Figura 6 – Composição de um sarcômetro.



Fonte: Adaptado livremente de TORTORA, 2000.

Resumidamente o potencial de ação viaja através dos neurônios motores para as fibras musculares através dos neurotransmissores, este potencial viaja pela membrana da fibra muscular e realiza um processo de despolarização. A eletricidade no centro do músculo causa uma liberação de íons de cálcio pelo retículo sarcoplasmático. Estes íons ativam forças de atração entre os filamentos delgados e espessos, causando o deslizamento dos mesmos, o que provoca a contração das miofibrilas e, portanto, a contração muscular. Ao final do potencial de ação os íons de cálcio retornam ao retículo sarcoplasmático e permanecem ali armazenados até novo potencial de ação, o que ocasiona o fim da contração das miofibrilas (GUYTON, 2006).

2.1.5 Potencial de Ação da Unidade Motora

A unidade básica organizacional do sistema nervoso é a unidade motora, que consiste de um neurônio motor e as fibras musculares que inerva. Todas as fibras musculares de uma

unidade motora se contraem ao mesmo tempo, ao passo que músculos que pertencem a diferentes unidades motoras podem não se contraírem ao mesmo tempo. Pode haver ligeira diferença de tempo entre as contrações de uma mesma unidade motora dada a diferença de tempo de propagação entre as ramificações ou a natureza estocástica do processo de despolarização. As fibras musculares em cada unidade motora não estão todas juntas, mas se sobrepõem às outras unidades motoras em micro feixes de 3 até 5 fibras. A interdigitação permite às unidades motoras separadas contraírem de forma a dar suporte as demais. O potencial de ação da unidade motora (MUAP) é a soma algébrica dos potenciais de ação de cada uma das fibras que compõem a unidade (BRONZINO & ELDERLE, 2012)

2.2 Princípios Básicos sobre Eletromiografia

A maioria das contrações musculares envolvem a ativação de várias unidades motoras, a soma das correntes elétricas originárias da atividade muscular, produzem sinais que podem ser detectados por instrumentos chamados de Eletromiógrafos. Sendo assim, a Eletromiografia pode ser definida como a captação e o estudo das variações elétricas que ocorrem no tecido muscular (TORTORA, 2000).

Eletromiografia de superfície (sEMG), como o nome indica, é um método de aquisição de dados de forma não invasiva, utilizando eletrodos posicionados na superfície da pele. Através da sEMG é possível observar os padrões do sinal da atividade muscular, do repouso até todo o curso do movimento (CRAM, 2011). A amplitude do sinal gerado por um músculo é de magnitude bastante reduzida, geralmente na casa de microvolts, e é associada à impedância do tecido corporal. São necessários instrumentos de alta sensibilidade para captar a diferença de potencial gerada pela atividade muscular (CRAM, 2011). O sinal da eletromiografia de superfície é de natureza estocástica, pois varia conforme o número de unidades motoras ativas, força da contração, interações mecânicas entre as fibras musculares, amplitude, duração e morfologia dos MUAPs (DE LUCA, 1979).

O sinal elétrico gerado pelo movimento de um músculo esquelético encontra-se usualmente na faixa de frequência de 2-500Hz e assume amplitude de 50 μ V até 5mV (BRONZINO, 2006). Estes são os valores típicos e podem variar dependendo do tipo de músculo, tipo de movimento e posicionamento dos eletrodos. A Figura 7 ilustra um típico sinal de sEMG para o movimento de extensão de braço com a mão fechada, medições do primeiro voluntário da base de dados NinaPro. A Figura 8 apresenta a resposta em frequência do mesmo movimento e é claramente observável, neste exemplo, que a maior parte da potência está concentrada na faixa de 30-300Hz.

Figura 7 – Exemplo de sEMG no domínio do tempo.

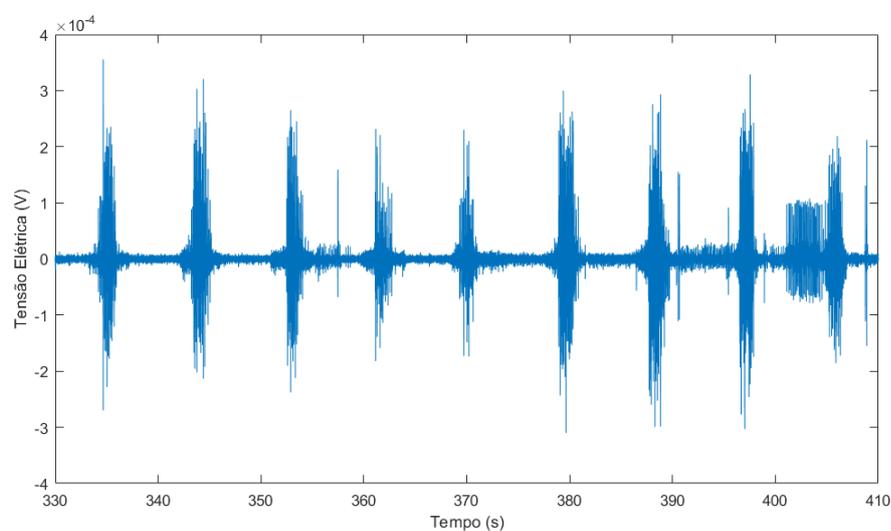
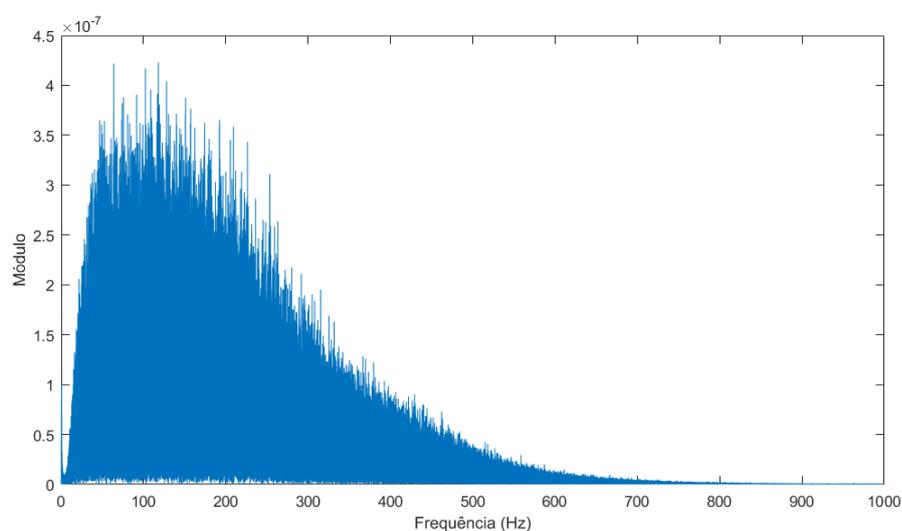


Figura 8 – Exemplo de sEMG no domínio da frequência.



Este capítulo apresentou uma sucinta revisão sobre a fisiologia e os sinais mioelétricos envolvidos neste trabalho. Para informações mais aprofundadas sobre esses temas é recomendada a leitura dos seguintes livros:

- BRONZINO, J.; ENDERLE, J. **Introduction to Biomedical Engineering**. 3rd. ed. Burlington: Elsevier, 2012. cap 1 e 3.
- GUYTON, A. C. **Fisiologia Humana**. 11. ed. Rio de Janeiro: Guanabara Koogan, 2006. cap 4-8.

- KANDEL, E. *et al.* **Principles of Neural Science**. 5. ed. New York: McGraw-Hill, 2013. cap 6, 33 e 34.
- TORTORA, GERARD J. **Corpo humano: fundamentos de anatomia e fisiologia**. Porto Alegre: Artmed, 2000. cap 1 e 8.

2.3 Princípios básicos sobre Lógica *Fuzzy*

Problemas complexos do dia a dia necessitam de sistemas que possuam conhecimento e sejam capazes de tomar decisões que se aproximam à lógica humana (JANG *et al.*, 1997). Na teoria clássica dos conjuntos, onde um indivíduo apenas pode ser classificado como pertencente ou não pertencente a um conjunto, normalmente não é inteiramente capaz de descrever diversas aplicações do dia a dia. A clara distinção entre os dois estados (chamado geralmente de lógica *Booleana*) não abrange problemas que envolvem elementos com pertinência parcial (CHEN&PHAM, 2001).

É importante salientar que o cérebro humano é capaz de interpretar informações incompletas e parciais e a teoria de conjuntos *Fuzzy* proporciona um sistema de cálculo que possibilita lidar linguisticamente com estas informações (JANG *et al.*, 1997). Conjuntos de definição imprecisa têm um papel importante no pensamento humano, particularmente no domínio de reconhecimento de padrões, comunicação de informações e abstração. A teoria de conjuntos *Fuzzy* inclui também aqueles elementos que não podem ser descritos pela teoria clássica dos conjuntos (ZADEH, 1965).

Como (ZADEH, 1965) definiu, conjuntos *Fuzzy* são uma classe de objetos que possuem níveis de pertinência contínuos. Estes níveis são determinados por funções de pertinência (FP) e podem assumir valores de 0 (não pertence) até 1 (pertence totalmente). O nome *Fuzzy* advém da palavra em inglês "*fuzziness*" que significa "vago", o que remete a generalidade e ambiguidade da associação de valores aos conjuntos, visto que na teoria de conjuntos *Fuzzy* não existem fronteiras binariamente definidas (DUBOIS & PRADE, 1980).

2.3.1 *Fuzzy*ificação e Funções de Pertinência

É possível aproximar a definição de um conjunto *Fuzzy* como sendo uma simples extensão de um conjunto clássico onde o grau de associação de um objeto ao conjunto pode assumir valores entre 0 e 1 (JANG *et al.*, 1997). Funções de pertinência, ou funções características, são comumente representadas pela letra grega μ e determinam os níveis de pertinência de um conjunto de valores pertencentes a um universo de valores. A Equação (1) apresenta a clássica representação de um conjunto *Fuzzy* A :

$$A = \{(x, \mu_A(x)), x \in X\} \quad (1)$$

onde:

A é o conjunto *Fuzzy*;

μ_A é a função de pertinência do conjunto *Fuzzy*;

X é o universo de valores;

x é elemento genérico do universo de valores.

Sendo assim, as funções de pertinência delimitam as fronteiras que determinam a pertinência de um indivíduo ao conjunto *Fuzzy*, e o processo de atribuir um grau de pertinência a um valor numérico é chamado de *Fuzzyficação* (JANG *et al.*, 1997). Os tipos mais comuns de FP são::

1) Triangular: é determinada por três parâmetros a , b e c . A Equação (2) indica a expressão matemática em função destes parâmetros e do ponto x e a Figura 9 ilustra a forma gráfica desta FP (JANG *et al.*, 1997).

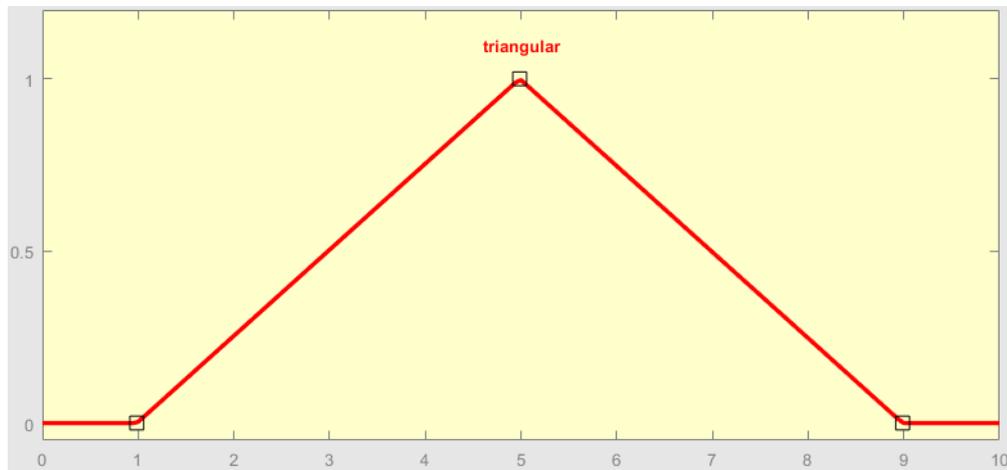
$$triangular(x; a, b, c) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & c \leq x \end{cases} \quad (2)$$

onde:

triangular é a função de pertinência triangular;

x é variável independente;

a, b, c são parâmetros da função de pertinência.

Figura 9 – Função Pertinência Triangular $(x;1,5,9)$.

2) Trapezoidal: é determinada por quatro parâmetros a , b , c e d . A Equação (3) indica a expressão matemática em função destes parâmetros e do ponto x e a Figura 10 ilustra a forma gráfica desta FP (JANG *et al.*, 1997).

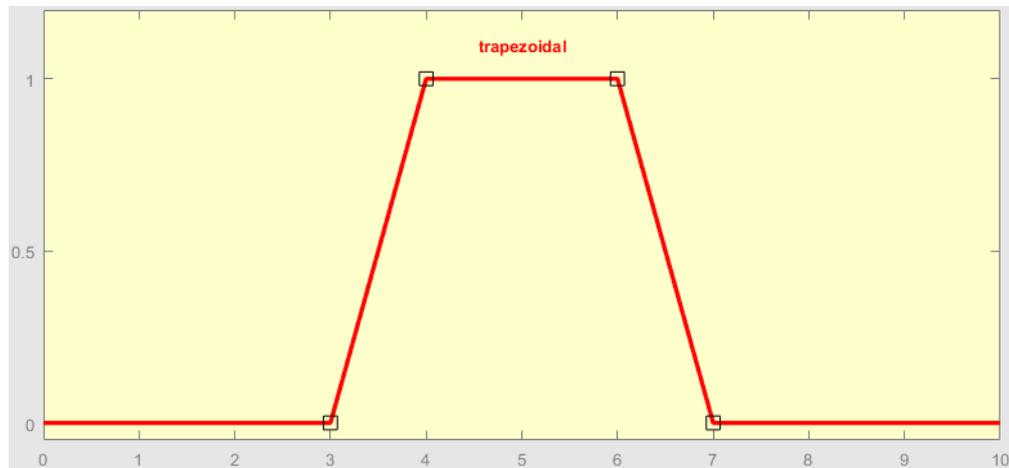
$$\text{trapezoidal}(x; a, b, c, d) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & d \leq x \end{cases} \quad (3)$$

onde:

trapezoidal é a função de pertinência trapezoidal;

x é variável independente;

a, b, c, d são parâmetros da função de pertinência.

Figura 10 – Função Pertinência Trapezoidal ($x;3,4,6,7$).

3) Gaussiana: é determinada por dois parâmetros c e σ . A Equação (4) indica a expressão matemática em função destes parâmetros e do ponto x e a Figura 11 ilustra a forma gráfica desta FP (JANG *et al.*, 1997).

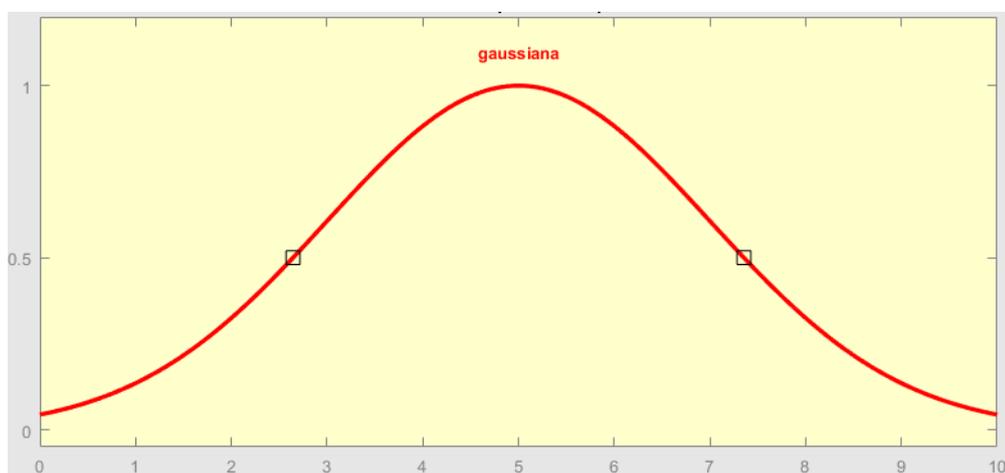
$$\text{Gaussiana}(x; c, \sigma) = e^{-\frac{1}{2} \frac{x-c}{\sigma}} \quad (4)$$

onde:

Gaussiana é a função de pertinência Gaussiana;

x é variável independente;

c e σ são parâmetros da função de pertinência.

Figura 11 – Função Pertinência Gaussiana ($x;2,5$).

4) *Bell*: é determinada por três parâmetros a , b e c . A Equação (5) indica a expressão matemática em função destes parâmetros e do ponto x e a Figura 12 ilustra a forma gráfica desta FP (JANG *et al.*, 1997).

$$Bell(x; a, b, c) = \frac{1}{1 + \left| \frac{x-c}{a} \right|^{2b}} \quad (5)$$

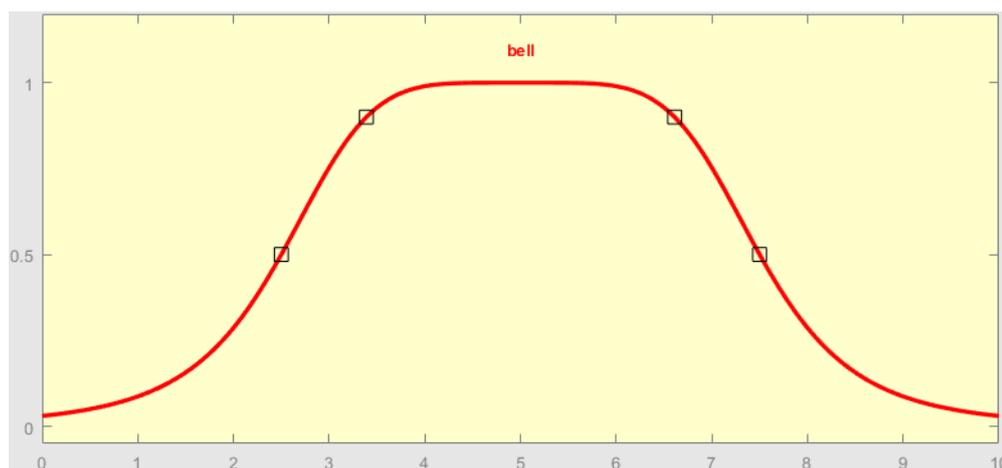
onde:

$Bell$ é a função de pertinência Bell;

x é variável independente;

a, b, c são parâmetros da função de pertinência.

Figura 12 – Função Pertinência Bell ($x;2,5,2,5$).



Embora haja uma notória semelhança entre a função característica de um conjunto *Fuzzy* e uma função probabilística, ambos são conceitos essencialmente diferentes. O grau de pertinência não informa a probabilidade que um objeto possui de pertencer a um conjunto, mas sim o nível que este objeto de fato pertence ao conjunto (ZADEH, 1965).

2.3.2 Operadores *Fuzzy*

A teoria de conjuntos *Fuzzy* deriva da teoria ordinária de conjuntos, como consequência as operações entre conjuntos podem ser extrapoladas e é possível obter o racional matemático para realizar estas operações também com conjuntos *Fuzzy*. Toda operação para conjuntos *Fuzzy*, quando aplicada a uma função de pertinência binária, retorna àquela aplicada em conjuntos clássicos (DUBOIS & PRADE, 1980). Foi no ano de 1965 que (ZADEH, 1965) criou as seguintes definições:

• *Pertinência* ou subconjunto: indica que um conjunto A está contido em um conjunto B e é representado pela Equação (6):

$$\mu_A \leq \mu_B, \text{ para todo } x \quad (6)$$

• *União (OR)*: é o conjunto junção entre os conjuntos A e B . Pode ser descrito como o menor conjunto contendo A e B . Está representado pela Equação (7):

$$\mu_C(x) = \max(\mu_A(x), \mu_B(x)) = \mu_A(x) \vee \mu_B(x) \quad (7)$$

• *Intersecção (AND)*: é o conjunto comum entre os conjuntos A e B . Pode ser descrito como maior conjunto contendo A e B . Está representado pela Equação (8):

$$\mu_C(x) = \min(\mu_A(x), \mu_B(x)) = \mu_A(x) \wedge \mu_B(x) \quad (8)$$

• *Complemento (NOT)*: é o conjunto complementar do conjunto A . É representado pela Equação (9):

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad (9)$$

onde:

μ_A , μ_B e μ_C são funções de pertinência dos conjuntos *Fuzzy*;

A, B e C são conjuntos *Fuzzy*;

x é a variável independente.

2.3.3 Sistema de Inferência *Fuzzy*

É importante observar que várias premissas são parcialmente verdadeiras, bem como parcialmente falsas. Para analisar tais valores vagamente assertivos, de uma forma que abrange a já utilizada lógica binária, as lógicas de multivalores foram criadas. A Lógica de decisão *Fuzzy*, ou simplesmente Lógica *Fuzzy*, tem como principal objetivo fornecer as bases para o raciocínio aproximado usando premissas imprecisas baseadas na teoria de conjuntos *Fuzzy* (CHEN & PHAM, 2001). Valores linguísticos correspondem a um valor aproximado que tem a funcionalidade de resumir e expressar valores quantitativos. São expressões muito comuns na forma de raciocínio humano e assumem valores como “alto”,

“bastante”, “irrelevante”, etc. São necessários pois, conforme a complexidade do sistema aumenta, nossa habilidade de criar premissas precisas e ainda significativas a respeito do seu comportamento diminuí até se tornarem mutuamente exclusivas. Sendo assim, os Sistemas de Inferência *Fuzzy* baseiam-se na Lógica de decisão *Fuzzy* para determinar relações de entrada e saída e são compostas por três principais etapas: Regras, Implicação e Agregação (JANG *et al.*, 1997).

2.3.3.1 Sistema de Inferência *Fuzzy*: regras

As regras *Fuzzy* são responsáveis por atribuir relação de causa e efeito para níveis de pertinência das entradas e resposta do sistema. Também conhecidas como “*if – then Fuzzy rule*” (regra *Fuzzy* se-então), as regras apresentam uma abordagem dos valores linguísticos. Como exemplo, a Equação (10) representa matematicamente esta regra, sendo “*x is A*” é a premissa, a regra em si é a implicação e “*y is B*” é a conclusão (ZADEH, 1965):

$$\text{If } x \text{ is } A, \text{ then } y \text{ is } B. \quad (10)$$

onde:

x e y variáveis numéricas;

A e B valores linguísticos definidos pelos conjuntos *Fuzzy*.

2.3.3.2 Sistema de Inferência *Fuzzy*: Implicação

Cada regra individualmente pode ser composta por diversas condições interligadas pelos conectores lógicos "E" ou "OU" e possui um conjunto de saída resultante determinados pelos métodos de Implicação. A Equação (11) mostra uma adaptação da “*if – then Fuzzy rule*” para três entradas, três condições e interligadas pelo operador "E":

$$\text{If } w \text{ is } A \text{ "E"} \text{ If } x \text{ is } B \text{ "E"} \text{ If } y \text{ is } C, \text{ then } z \text{ is } D. \quad (11)$$

onde:

w , x , y e z variáveis numéricas;

A , B , C e D valores linguísticos definidos pelos conjuntos *Fuzzy*.

Cada uma das condições que constituem uma regra possui um conjunto resultado igual a área abaixo da curva de pertinência no ponto correspondente à variável de entrada, a forma de união destes conjuntos determina o método de Implicação. Os métodos de

Implicação mais comuns são mínimo (ou *min*) e produto (ou *prod*). A Figura 13 ilustra o método de mínimo, as áreas em amarelo representam a pertinência individual de cada "if" da regra para os valores indicados pela linha vertical vermelha e a área em azul o conjunto resultante, é possível acompanhar a linha em verde de forma que o conjunto solução é determinado pelo valor mínimo entre os conjuntos, ou seja, apenas o menor nível de pertinência determina a saída. A Figura 14 ilustra o método de produto, é notável que o menor nível de pertinência entre os conjuntos intermediários determina a saída, pois em uma multiplicação o menor valor tem influência significativa no resultado final (JANG *et al.*, 1997).

Figura 13 – Exemplo da Implicação das saídas pelo método de mínimo (ou *min*).

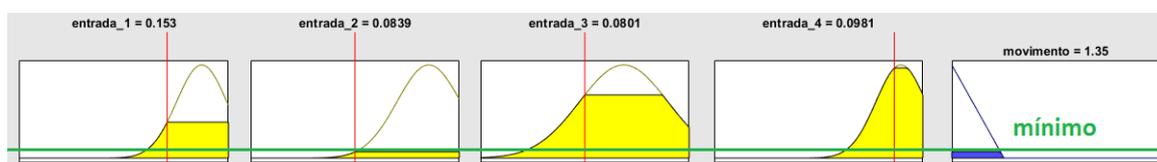
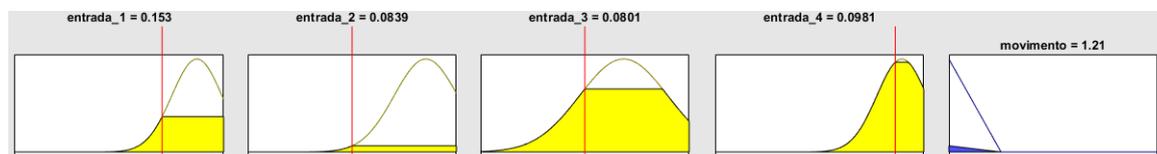
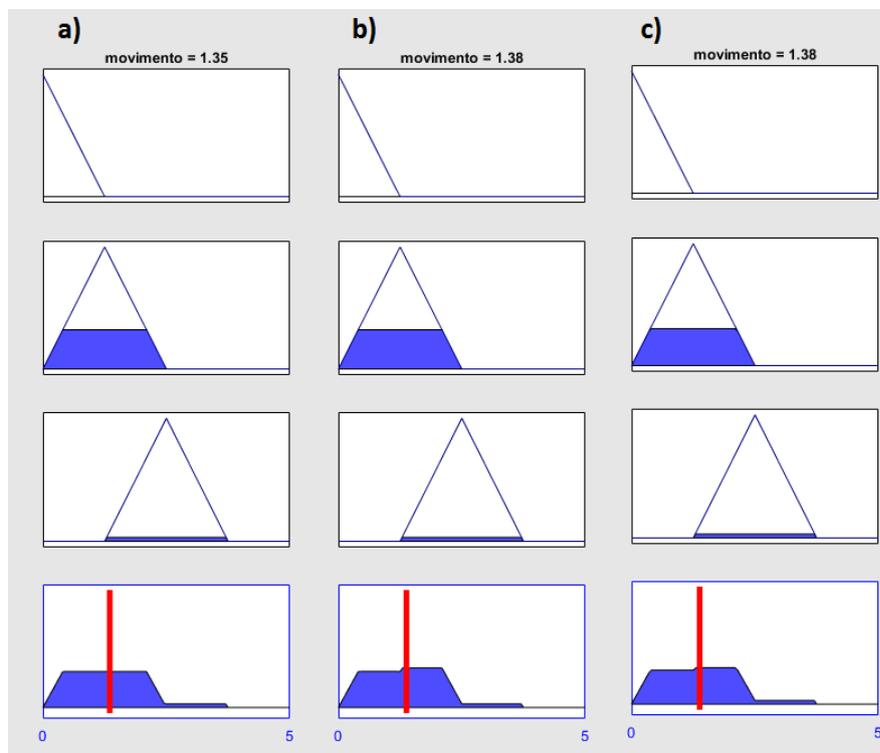


Figura 14 – Exemplo da Implicação das saídas pelo método de produto (ou *prod*).



2.3.3.3 Sistema de Inferência *Fuzzy*: Agregação

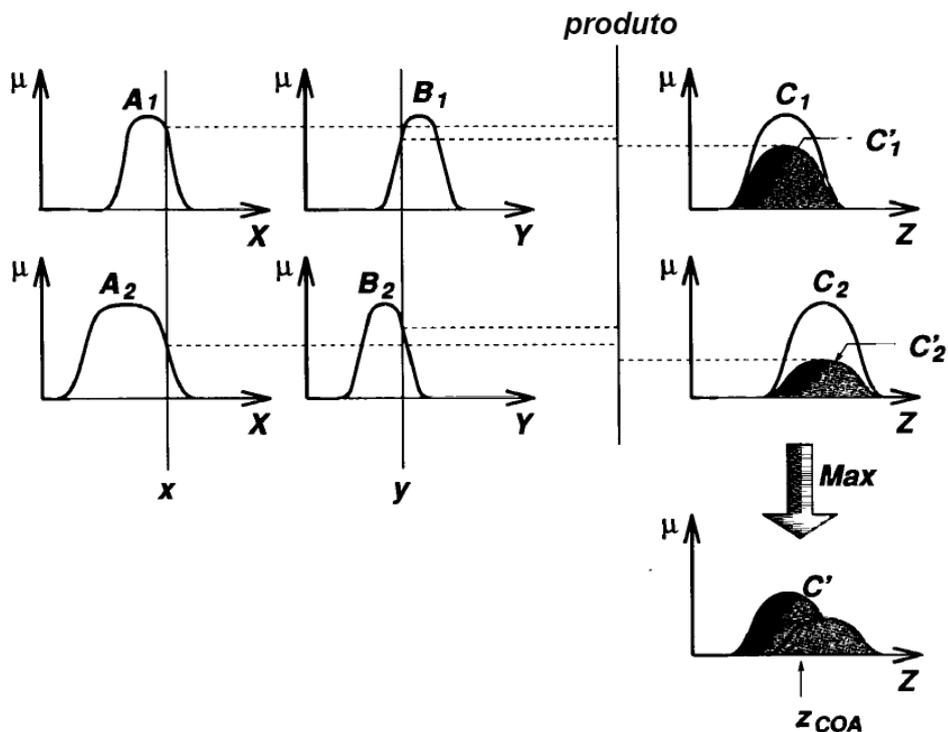
A região resultante de um Sistema *Fuzzy* é dada por uma ou mais regras “*if-then*” e uma vez determinada a saída resultante de cada uma delas é necessário uma etapa de composição, também chamada de Agregação, que determina uma saída resultante através de operadores considerando também a relevância de cada uma das regras. Os principais métodos de Agregação são máximo (ou *max*), soma (ou *sum*) e *probor* (DUBOIS & PRADE, 1980). A Figura 15 apresenta os três diferentes métodos de Agregação para conjuntos resultantes de três regras, as três primeiras linhas indicam a saída de uma regra diferente e cada coluna representa um método de Agregação, a quarta linha representa o conjunto resultante. A Figura 15 a) ilustra o método de máximo, ou seja, o resultado é o máximo valor entre os conjuntos, a Figura 15 b) ilustra o método de soma, ou seja, apresenta o valor da soma dos conjuntos resultantes e a Figura 15 c) ilustra o método de *probor*, que apresenta o maior conjunto resultante.

Figura 15 – Exemplo de métodos de Agregação: a) *max*; b) *sum* e c) *probor*.

2.3.3.4 Sistema de Inferência *Fuzzy*: Mandani e Sugeno

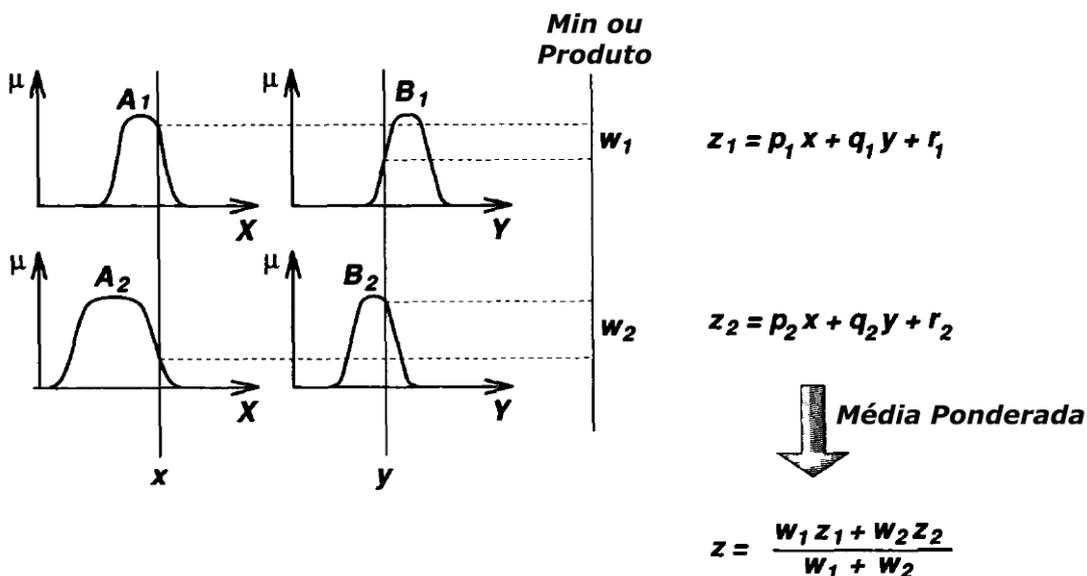
Formatos muito utilizados para Sistemas de Inferência *Fuzzy* são o Mamdani e Sugeno. O primeiro utiliza operadores de máximo e mínimo para agregação e composição e pode ser exemplificado pela Figura 16. O segundo método obtém a composição de saída através de uma média ponderada, o que otimiza o tempo de processamento e obtém as saídas individuais através dos operadores de mínimo ou produto. É possível observar o método Sugeno na Figura 17 (JANG *et al.*, 1997).

Figura 16 – Lógica de decisão *Fuzzy* com método Mandani.



Fonte: Adaptado livremente de JANG *et al.*, 1997

Figura 17 – Lógica de decisão *Fuzzy* com método Sugeno.



Fonte: Adaptado livremente de JANG *et al.*, 1997

2.3.4 Defuzzificação

Uma vez determinado o conjunto de saída a partir da composição dos resultados das regras definidas é necessária uma etapa que converta todos os termos *Fuzzy* resultantes em variáveis numéricas dentro do universo de números com fronteiras booleanas. Esta etapa resulta em um sinal de controle que assume os valores físicos necessários para o processo, como por exemplo, posição, tensão elétrica, força, etc (CHEN & PHAM, 2001).

Os métodos principais de *defuzzificação*, são: Centróide ou *centroid* (Equação (12)), Biseção ou *bisector* (Equação (13)), Máximo Intermediário ou *mom*, Maior Máximo ou *lom*, Menor Máximo ou *som* (CHEN & PHAM, 2001).

$$u(n+1) = \frac{\sum_{i=1}^N \mu_{A_i}(u(n)) \cdot u(n)}{\sum_{i=1}^N \mu_{A_i}(u(n))} \quad (12)$$

$$u(n+1) = \frac{\sum_{i=1}^N u(n) \sum_{j=1}^N \mu_{A_j}(u(n))}{\sum_{j=1}^N \mu_{A_j}(u(n))} \quad (13)$$

onde:

$u(n)$ é a função que se deseja *Defuzzificar*;

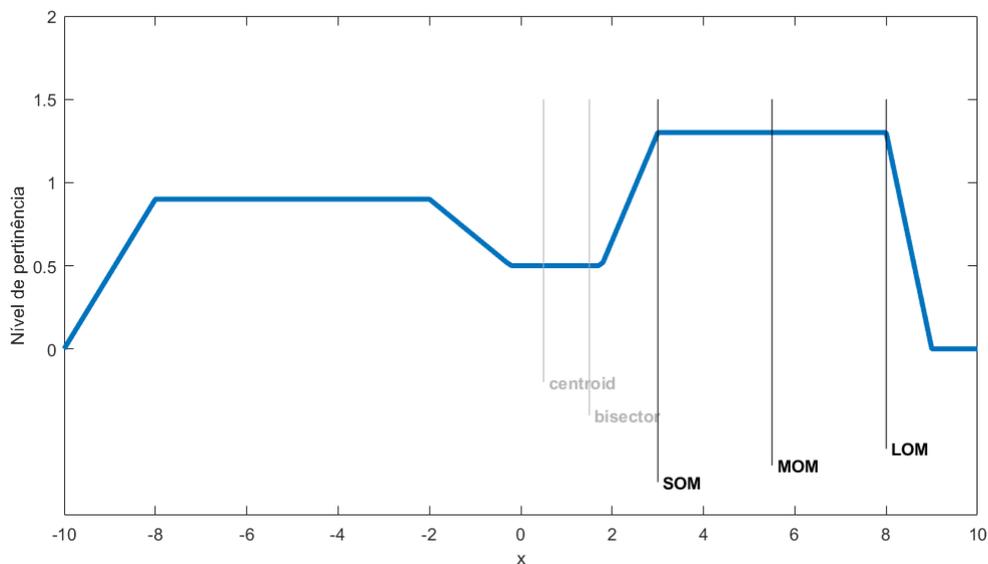
$u(n+1)$ é o resultado da *Defuzzificação*;

μ_A é a função de pertinência de $u(n)$;

N é o número de pontos da função $u(n)$.

A Figura 18 mostra um exemplo de conjunto resultante a partir da Agregação de regras e as retas verticais apresentam os valores resultantes dos 5 métodos de *Defuzzificação* cujas equações foram apresentadas, desta forma é possível observar a diferença entre os resultados numéricos obtidos para cada um deles. Como o conjunto apresentado apresenta certa simetria o método de Centróide, que também é conhecido como método do centro de massa, está centralizado na Figura 18, mesmo que este ponto coincida com um dos pontos mais baixos da curva. O resultado do método da Biseção se aproxima muito do Centróide, pois indica qual o ponto da curva que apresenta igual área para ambos os lados. Por fim SOM, MOM e LOM indicam, respectivamente, o menor ponto cuja curva atinge seu valor máximo, o ponto médio entre os picos da curva e o maior ponto cuja curva atinge seu valor máximo.

Figura 18 – Exemplificação gráfica dos métodos de *Defuzzyficação*: *centroid*, *bisector*, *mom*, *lom* e *som*.



Este capítulo apresentou uma sucinta revisão sobre a Lógica *Fuzzy*. É importante salientar que esta área está em constante desenvolvimento. Portanto, para informações básicas sobre os princípios da Lógica Fuzzy é recomendada a leitura dos seguintes livros:

- CHEN, G. PHAM, T. **Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems**. Boca Raton: CRC Press LLC, 2001.
- DUBOIS, D. PRADE, H. **Fuzzy set and systems: theory and applications**. New York: Academic Press, 1980.
- JANG, J. R.; SUN, C.; MIZUTANI, E. **Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence**. New York: Prentice Hall, 1997.
- SUGENO, M.; KANG, G. T. Structure identification of fuzzy model. **Fuzzy Sets and Systems**, North Holland, v.28, n.3, p. 15-33, May 1988.

3 METODOLOGIA EXPERIMENTAL

3.1 Descrição da Base NINAPRO

A base de dados NINAPRO foi desenvolvida com o objetivo de estimular e facilitar estudos na área de biosinais e é composta por três sub-bases de dados distintas. Serão utilizadas, neste trabalho, apenas as sub-bases II e III, pois foram geradas através do mesmo procedimento científico, com diferença apenas na composição dos voluntários. A sub-base II foi obtida com voluntários não amputados e a sub-base III com voluntários amputados. Para a captação dos sinais mioelétricos, as sub-bases citadas anteriormente utilizaram 12 eletrodos de superfície sem fio. Cada um dos eletrodos é equipado com uma bateria recarregável e podem operar a uma distância de 40m. Os sinais foram adquiridos à uma frequência de 2kHz com um ruído menor que 750nV (ATZORI, 2015).

A base utilizada contém um total de 49 movimentos separados em três classes de exercícios, da mesma forma as sub-bases são divididas em três arquivos, um para cada classe de exercício. Os movimentos são repetidos seis vezes em sequência e são identificados por um número. Cada arquivo de dados é composto por 12 variáveis, das quais 7 serão utilizadas no presente trabalho. A Tabela 1 contém a nomenclatura das variáveis, a dimensão do arranjo de dados e uma breve descrição do conteúdo.

Tabela 1 – Variáveis que compõem os arquivos de dados da base NINAPRO.

Nome	Dimensão	Descrição
<i>Subject</i>	1 x 1	Identificação do Voluntário
<i>Exercise</i>	1 x 1	Classe de exercícios
<i>emg</i>	12 x amostras	Doze canais de sinais mioelétricos
<i>stimulus</i>	1 x amostras	Identificação do Movimento
<i>restimulus</i>	1 x amostras	Identificação refinada do Movimento
<i>repetition</i>	1 x amostras	Identificação da Repetição
<i>rerepetition</i>	1 x amostras	Identificação refinada da Repetição

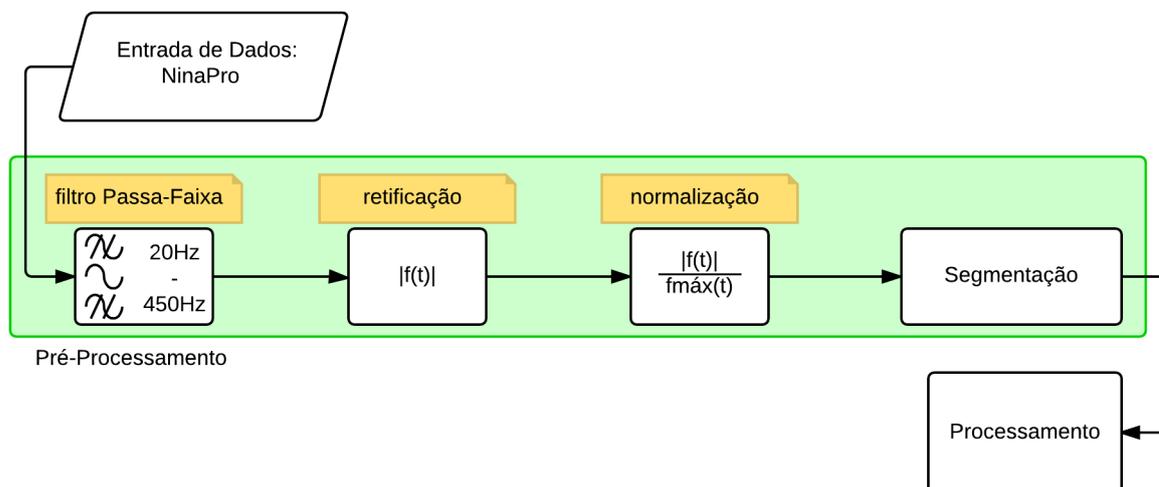
Fonte: Adaptado livremente de ATZORI, 2015.

3.2 Etapa de Pré-Processamento

O primeiro passo para processar o sinal e poder analisar os movimentos através de dados coletados é realizar um pré-processamento. Trata-se de ajustes no sinal que envolvem eliminação de imperfeições e que facilitam a manipulação matemática. A Figura 19 apresenta o diagrama do pré-processamento, que filtra, retifica, normaliza e segmenta os

dados advindos da Base NinaPro. Os próximos capítulos desta seção abordam em detalhes a implementação das diferentes etapas do pré-processamento.

Figura 19 – Diagrama da Etapa de Pré-Processamento.



3.2.1 Filtragem do Sinal

Conforme visto anteriormente os valores típicos de frequência de um sinal mioelétrico encontram-se na faixa de 2Hz até 500Hz (BRONZINO, 2006), entretanto pode-se estipular que a distribuição espectral que contém maior parte da informação encontra-se de 15-450Hz (JANG, 2014). Desta forma, se torna necessário a filtragem do sinal através de um filtro Passa-Faixa que atenua os valores contidos fora da faixa de passagem determinada. O *MatlabR2016a* possui uma função específica para projeto de filtros chamada pelo comando *design()*, que está descrito na Tabela 2. Para utilização desta função é necessário criar um vetor configuração do filtro Passa-Faixa através do comando *fdesign.bandpass()*, que está descrito na Tabela 3, sendo assim, com este vetor é possível selecionar as características do filtro desejado. O código completo utilizado para Filtragem do Sinal pode ser observado no Anexo A.2.

Tabela 2 – Detalhes do comando de *design()* de Filtros do *MatlabR2016a*.

Nome	Descrição
Sintaxe	$h = \text{design.bandpass}(F_{\text{stop1}}, F_{\text{pass1}}, F_{\text{pass2}}, F_{\text{stop2}}, A_{\text{stop1}}, A_{\text{pass}}, A_{\text{stop2}}, F_s);$
F_{stop1}	Frequência de parada inferior;
F_{pass1}	Frequência de passagem inferior;
F_{pass2}	Frequência de passagem superior;
F_{stop2}	Frequência de parada superior;
A_{stop1}	Atenuação de parada inferior;
A_{pass}	Atenuação de passagem;
A_{stop2}	Atenuação de parada superior;
F_s	Frequência de amostragem;

Tabela 3 – Detalhes do comando de *fdesign()* de Filtros do *MatlabR2016a*.

Nome	Descrição
Sintaxe	$Hd = \text{design}(h, \text{'butter'}, \text{'MatchExactly'}, \text{match});$
h	Vetor que armazena as configurações;
Hd	Função transferência do filtro;
<i>'butter'</i>	Indica que o tipo de filtro é <i>Butterworth</i> ;
<i>'MatchExactly'</i>	Indica que o filtro deve corresponder exatamente as especificações de corte e atenuação;
<i>match</i>	Indica que a instrução anterior referia-se ao parâmetro de correspondência;

3.2.2 Retificação e Normalização do Sinal

Todos os canais sEMG passam por um processo de retificação e normalização. Para retificar o sinal, basta aplicar a operação de módulo representada pela Equação (14). A etapa de normalização ocorre quando todos os elementos do canal são divididos pelo maior elemento, como mostrado na Equação (15). O código completo para Retificação e Normalização do Sinal pode ser observado no Anexo A.2.

$$f_{\text{retificada}}(t) = |f_{\text{filtrada}}(t)| \quad (14)$$

$$f_{\text{normalizada}}(t) = \frac{f_{\text{retificada}}(t)}{f_{\text{retificada}}^{\text{máx}}} \quad (15)$$

onde:

$f_{\text{retificada}}(t)$ é a função retificada no tempo;

$f_{\text{filtrada}}(t)$ é a função filtrada no tempo;

$f_{\text{normalizada}}(t)$ é a função normalizada no tempo;

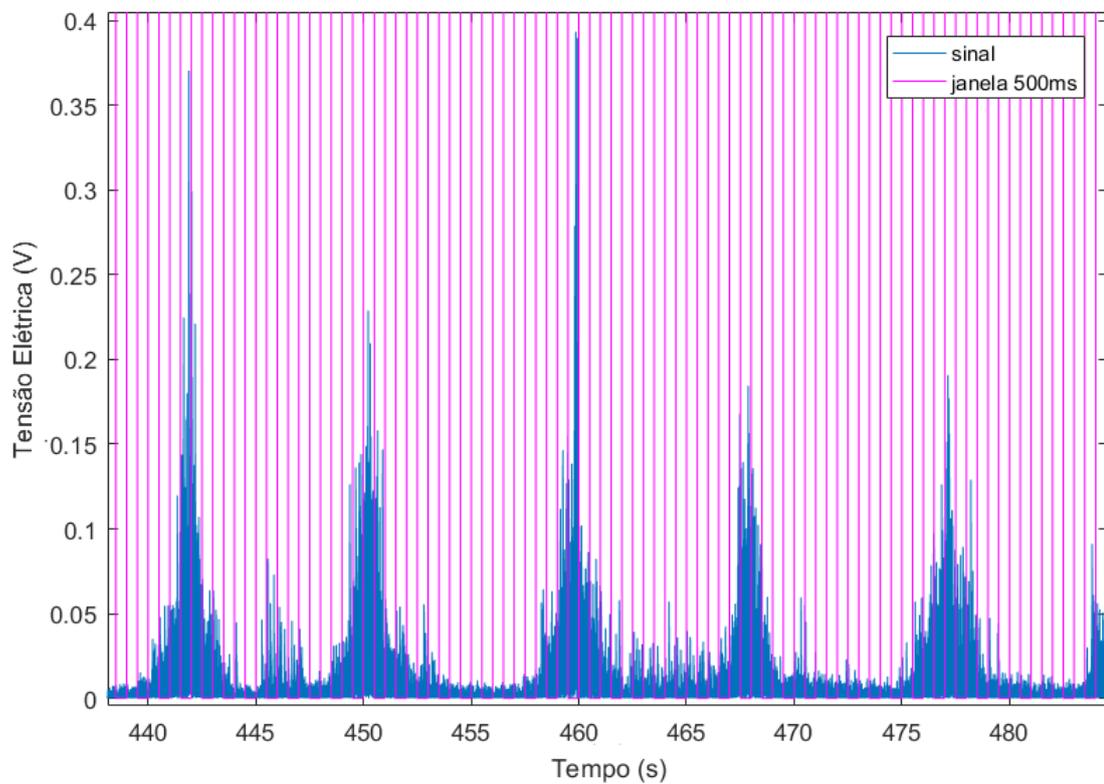
$f_{\text{retificada}}^{\text{máx}}$ é o ponto máximo da função retificada para todo o tempo.

3.2.3 Segmentação por Detecção de Limiar

A segmentação do sinal consiste em identificar os intervalos do sinal em que há sinal de sEMG caracterizando movimento ativo, desta forma, é possível analisar separadamente o respectivo conjunto de dados. A segmentação será realizada através do método de segmentação por detecção de limiar. O presente capítulo aborda em detalhes a implementação da segmentação. Para facilitar o entendimento deste trabalho e representar de forma mais resumida a informação de interesse será utilizado "há movimento ativo", para quando existe sinal de sEMG que representa de forma significativa a ativação muscular de um dado movimento.

O procedimento de segmentação por detecção de limiar, adotado neste trabalho, divide o sinal em janelas de mesmo tamanho e para cada uma destas janelas é identificada ou não a presença do sinal característico do movimento através da porcentagem de elementos que ultrapassam um limiar determinado. Caso o movimento seja detectado dentro da janela esta é destacada e o sinal é recortado unindo todas as janelas adjacentes. O tamanho fixo da janela de tempo é de 500ms, ou seja, considerando-se a taxa de amostragem, é composta de 1000 amostras do sinal. A Figura 20 apresenta um exemplo de um canal dividido em janelas de 500ms. Como pode ser observado são necessárias várias janelas para abranger um movimento, é importante salientar que nenhuma janela foi descartada e a Figura 20 apenas ilustra a segmentação. Analisando a base de dados observou-se que, em média, um movimento tem o tamanho entre 5 até 6 janelas, em outras palavras, 2500ms até 3000ms.

Figura 20 – Sinal dividido em janelas de 500ms.



A metodologia para determinar se uma janela contém ou não um sinal de sEMG que caracteriza a ativação de um dado movimento é baseada no valor RMS do sinal, ou seja, se várias amostras ultrapassarem o valor RMS a janela pode conter um movimento considerado ativo. Isto porque não é acertivo sinalizar que existe movimento dentro da janela indicada se apenas uma amostra, ou uma pequena quantidade de amostras, ultrapassarem o valor RMS do sinal. O sinal sEMG é caracterizado pela natureza estocástica e a presença de ruídos, estes elementos que ultrapassam o limiar de teste podem fornecer um falso positivo. Ao observar o sinal sEMG da Figura 20 é possível identificar visualmente que existe movimento quando uma extensa quantidade de elementos se destacam do valor de repouso, existe uma densidade maior de amostras com valores acima do limiar. Sendo assim foi atribuído o seguinte multiplicador empírico à lógica de decisão: se $x\%$ das amostras dentro da janela ultrapassam o valor RMS do sinal então esta janela contém um movimento. Através de diversos testes empíricos com a base de dados foi observado que o multiplicador ótimo para este algoritmo foi 80%. A Equação (16) indica a expressão utilizada para determinar o valor RMS do sinal:

$$X_{RMS} = \sqrt{\frac{1}{N} \sum_{n=1}^N |X_n|^2} \quad (16)$$

onde:

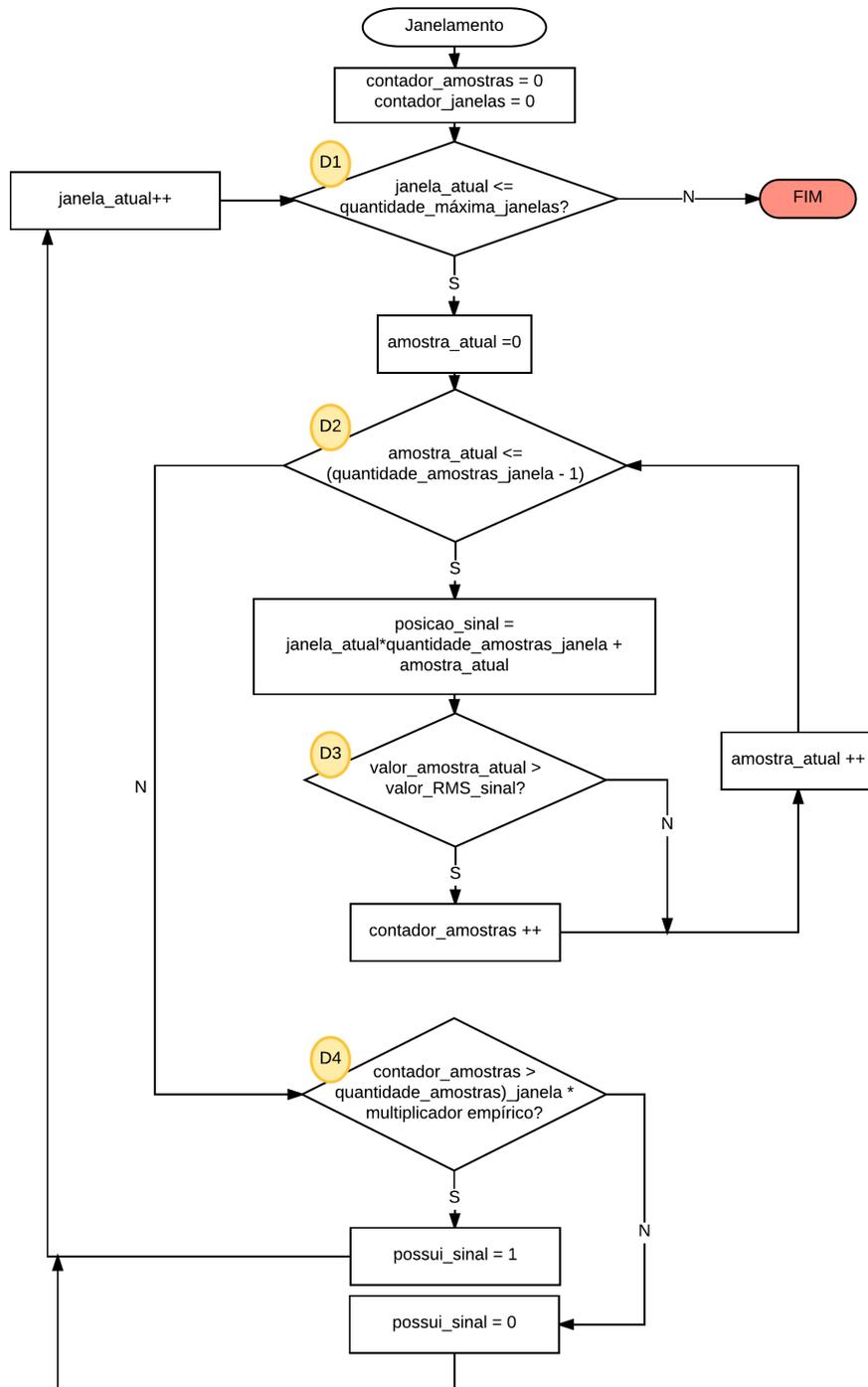
X_{RMS} é o valor RMS do sinal X ;

N é o número de amostras do sinal;

X_n é o valor da amostra da posição n do sinal X .

O algoritmo de segmentação precisa testar todas as janelas de 500ms que o sinal está dividido, a Figura 21 ilustra a lógica por trás da segmentação e está dividido em quatro lógicas de decisão diferentes. A primeira decisão do fluxograma da Figura 21, chamada de “D1”, finaliza o programa assim que a contagem de janelas chegar ao valor máximo, caso contrário, reinicia o laço que repete o resto do programa para todas as janelas do sinal. A segunda lógica de decisão (“D2”) retorna o programa para a próxima janela caso o valor máximo de amostras contidas em uma janela seja atingido, caso contrário, reinicia o laço que repete o resto do programa para todas as amostras que compõe a janela. A terceira decisão (“D3”) testa se o valor da amostra atual é maior do que o valor RMS do sinal previamente calculado e incrementa um contador caso verdadeiro. Além disto, independente do resultado, a decisão “D3” reinicia o *loop* que repete este mesmo teste para todas as amostras da janela. A quarta e última decisão (“D4”) identifica se a quantidade de amostras que ultrapassam o limiar dentro da janela atual é maior do que o valor empírico determinado, caso verdadeiro o programa aciona a flag “*possuisinal*” colocando-a em valor alto, caso falso a *flag* é colocada em valor baixo. A Figura 22, como exemplo, apresenta visualmente o processo de segmentação por detecção de limiar. As janelas destacadas em rosa representam a janela que o programa estaria verificando e as janelas destacadas em verde são as janelas que após verificação foram sinalizadas como contendo movimento. O código completo de Segmentação do Sinal pode ser observado no Anexo A.3.

Figura 21 – Fluxograma da Segmentação por detecção de limiar do sinal.



onde:

contador_amostras conta a quantidade de amostras por janela;

contador_janelas conta a quantidade de janelas do sinal;

janela_atual indica qual janela está sendo testada;

quantidade_maxima_janelas indica qual a quantidade de janelas de 500ms o sinal possui ao máximo;

amostra_atual indica qual amostra está sendo testada;

posicao_sinal calcula qual o elemento do vetor está sendo testado;

valor_amostra_atual é o valor da tensão elétrica da amostra que está sendo testada;

valor_RMS_sinal é o valor RMS do sinal;

contador_amostras indica a quantidade de amostras que ultrapassam o valor RMS do sinal;

multiplicador_empirico indica qual a porcentagem das amostras deve estar acima do limiar para ser considerado movimento;

possui_sinal_flag que quando 1 indica que há movimento na janela atual e quando 0 indica que não há movimento.

Figura 22 – Exemplo do procedimento de segmentação por detecção de limiar.



3.3 Descrição dos Movimentos Avaliados e dos Voluntários

No presente trabalho foram analisados os movimentos do Exercício 1 das sub-bases II e III da base de dados NINAPRO, que são ilustrados pela Figura 23. Entretanto, apenas 5 dos 17 movimentos foram analisados. A Tabela 4 mostra quais são estes movimentos e suas

respectivas descrições, estes movimentos foram escolhidos com base no comportamento, observou-se que estes apresentavam maior diferença entre o comportamento dos canais.

Tabela 4 – Movimentos utilizados neste trabalho do Exercício 1 da sub-bases II e III da base de dados NINAPRO.

Movimento	Descrição
6	Flexão dos dedos em formato de punho
9	Supinação do punho
13	Flexão do punho
14	Extensão do punho
16	Desvio ulnar do punho

Figura 23 – Movimentos da Base de dados NINAPRO.



Fonte: Adaptado livremente de ATZORI, 2015

Foram selecionados 6 voluntários selecionados aleatoriamente da sub-base II da base de dados NINAPRO, composta apenas por pessoas sem amputação, e 6 voluntários selecionados aleatoriamente da sub-base III da base de dados NINAPRO, composta apenas por amputados. Totalizando 12 sujeitos, a Tabela 5 indica quais as identificações dos sujeitos nas respectivas bases e o nome adotado para eles ao longo deste trabalho. A Tabela 5 também apresenta idade, altura e peso para todos os voluntários, para os sujeitos amputados indica também a causa da deficiência.

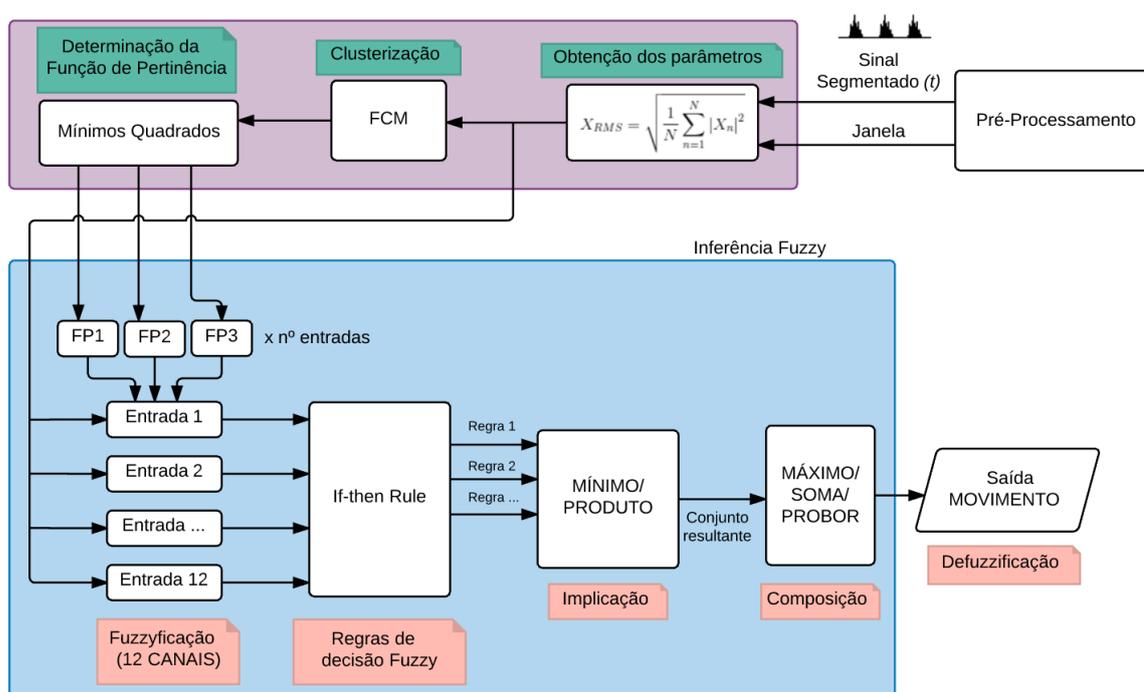
Tabela 5 – Identificação dos sujeitos da base de dados NINAPRO utilizados neste trabalho.

nome	sub-base	identificação	idade	altura	peso	causa
S2	II	S2	29	183	75	
S3	II	S3	31	174	69	
S11	II	S11	32	150	54	
S19	II	S19	31	158	52	
S25	II	S25	31	168	70	
S39	II	S39	31	183	96	
aS1	III	S1	32	172	86	Acidente
aS2	III	S2	35	183	81	Acidente
aS4	III	S4	34	166	68	Acidente
aS5	III	S5	67	175	75	Acidente
aS8	III	S8	33	175	80	Acidente
aS10	III	S10	59	177	86	Acidente

3.4 Etapa de Processamento do sinal

O pré-processamento fornece um sinal no tempo segmentado e a posição de início e fim das janelas de movimento, desta forma, é possível calcular os parâmetros desejados para a aplicação da lógica *Fuzzy*. A Figura 24 apresenta o diagrama do processamento, identificando que o sinal passa por uma etapa de obtenção de parâmetros, onde será calculado o valor RMS de cada segmento considerado ativo, ou seja, o segmento cujo movimento foi detectado. A etapa seguinte realiza um processo de *Clustering*, que agrupa os dados em três *clusters* diferentes e uma etapa de determinação da Função de pertinência, que utiliza os níveis de pertinência obtidos no *Clustering* para encontrar os parâmetros das funções Gaussianas que mais se aproximam do comportamento do sinal. Após esta preparação das informações é possível determinar a resposta do Sistema de Inferência *Fuzzy*, como também apresenta a Figura 24. Esta etapa inicia pelo processo de *Fuzzyficação* que transforma as variáveis de tensão elétrica RMS em níveis de pertinência, através das Funções de Pertinência determinadas. Após, são aplicadas as regras de decisão *Fuzzy*, onde os níveis de pertinência de cada entrada determinam um conjunto de resposta de saída. As respostas individuais de cada regra são determinadas pelo método de implicação e o agrupamento de todas as saídas em uma única se dá pelo método de composição (ou agregação). Por fim, o conjunto resultante final é submetido à um método de *Defuzzyficação* e fornece um resultado numérico que identifica qual movimento foi realizado. O presente capítulo descreve as etapas até a determinação do movimento final.

Figura 24 – Diagrama do processamento do Sinal.



3.4.1 Obtenção dos parâmetros dos movimentos

A segmentação do sinal fornece duas informações diferentes: um vetor para cada canal contendo todas as amostras do sinal segmentado e um vetor contendo as posições de início e fim de cada conjunto de elementos em que houve movimento identificado. A partir disto, é possível calcular as características de cada segmento do sinal. No presente trabalho, será utilizada uma métrica no domínio do tempo, que é o valor RMS da tensão elétrica do correspondente sinal de sEMG de um dado segmento ativo.

O valor médio quadrático da amplitude do sinal é dada pela Equação (17) e a lógica desta função consiste em um laço *for* que percorre todos os elementos do vetor do sinal segmentado para cada movimento, conforme mostrado na Figura 25. O diagrama da Figura 25 contém um *loop* que começa na posição de início do segmento e termina na posição final. Para cada amostra da janela de movimento é calculada a soma quadrática do valor RMS do sinal e ao fim do segmento divide-se esta soma pela quantidade de amostras da janela e retira-se a raiz quadrada. O código completo para obtenção dos parâmetros dos movimentos pode ser observado no Anexo A.4.

$$X_{RMS} = \sqrt{\frac{1}{N} \sum_{n=1}^N |X_n|^2} \quad (17)$$

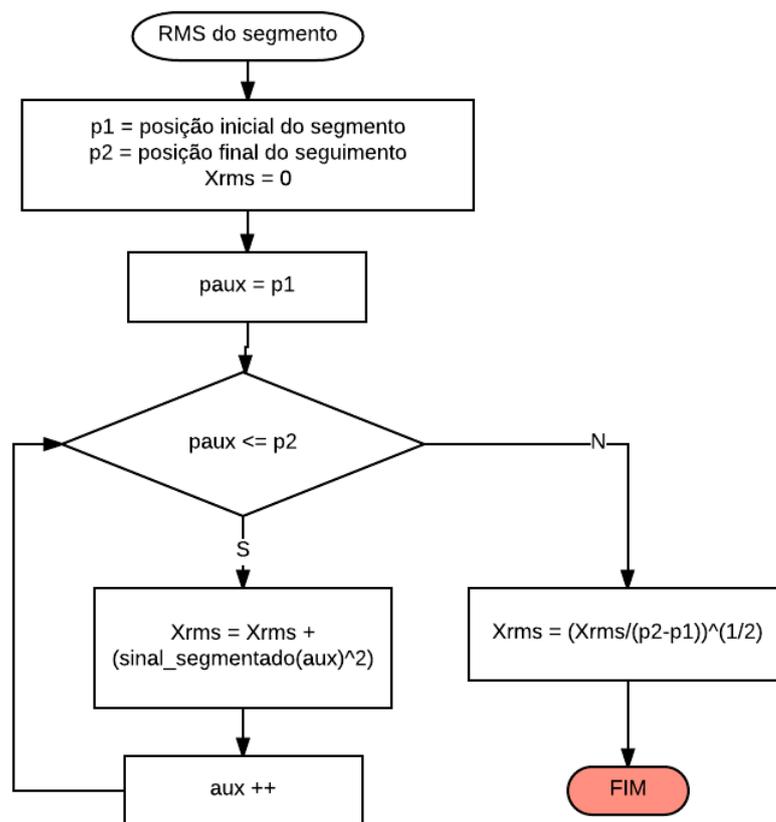
onde:

X_{RMS} é o valor RMS do sinal $X[n]$;

N é a quantidade de amostras que compõe o segmento;

X_n é o valor do sinal $X[n]$ de posição n .

Figura 25 – Fluxograma para determinar o valor RMS por segmento.



onde:

$p1$ é a posição inicial do segmento analisado;

$p2$ é a posição final do segmento analisado;

$paux$ é a posição auxiliar;

$sinal_segmentado$ é o sinal cujo valor RMS está sendo calculado;

aux é o index do elemento do $sinal_segmentado$;

$Xrms$ é o valor RMS calculado do segmento.

3.4.2 *Clustering* dos parâmetros

Com a finalidade de determinar a melhor função de pertinência para o processo de *fuzzyficação* das características representativas do sinal segmentado será utilizado uma metodologia de agrupamento de dados (chamado de *Clustering*). A partir do histórico de medidas de cada uma das janelas de movimento foram determinados três grupos de valores classificados como baixo, médio e alto, sendo assim, determina-se também o grau de pertinência de todos os elementos para os respectivos grupos. A metodologia empregada foi o algoritmo de *Fuzzy Clustering*, chamado também de *Fuzzy c – means* ou ainda pela sigla FCM. Como o nome indica é um método baseado no erro mínimo quadrático. Com base na quantidade de agrupamentos indicados, o método encontra os centróides dos agrupamentos de dados através da Equação (18).

O algoritmo é iterativo, ou seja, em cada iteração busca minimizar o erro quadrático através da realocação dos níveis de pertinência (inicialmente assumem valores aleatórios) utilizando a Equação (19). O critério de parada ocorre quando a Função Objetivo descrita pela Equação (20) assume valores inferiores ao valor mínimo estipulado ou quando o número máximo de iterações é atingido. A Figura 26 ilustra o algoritmo de *Clustering* descrito por (BEZDEK,1981), onde calcula-se o centro do agrupamento de dados a partir de dados aleatórios iniciais, após determina os níveis de pertinência pelo método dos mínimos quadrados com os novos valores de centro. Por fim o algoritmo testa se a Função Objetivo atingiu os critérios de parada, caso contrário fecha o *loop* para recalcular os centros e os níveis de pertinência.

$$c_j = \frac{\sum_{i=1}^D \mu_{ij}^m x_i}{\sum_{i=1}^D \mu_{ij}^m} \quad (18)$$

$$\mu_{ij} = \frac{1}{\sum_{k=1}^N \left(\frac{\|x_i - c_j\|}{\|x_i - c_k\|} \right)^{\frac{2}{m-1}}} \quad (19)$$

$$J_m = \sum_{i=1}^D \sum_{j=1}^N \mu_{ij}^m \|x_i - c_j\|^2 \quad (20)$$

onde:

D é o número de elementos;

N é o número de *clusters* determinado;

m é a divisão *Fuzzy* da matriz expoente que controla o nível de sobreposição;

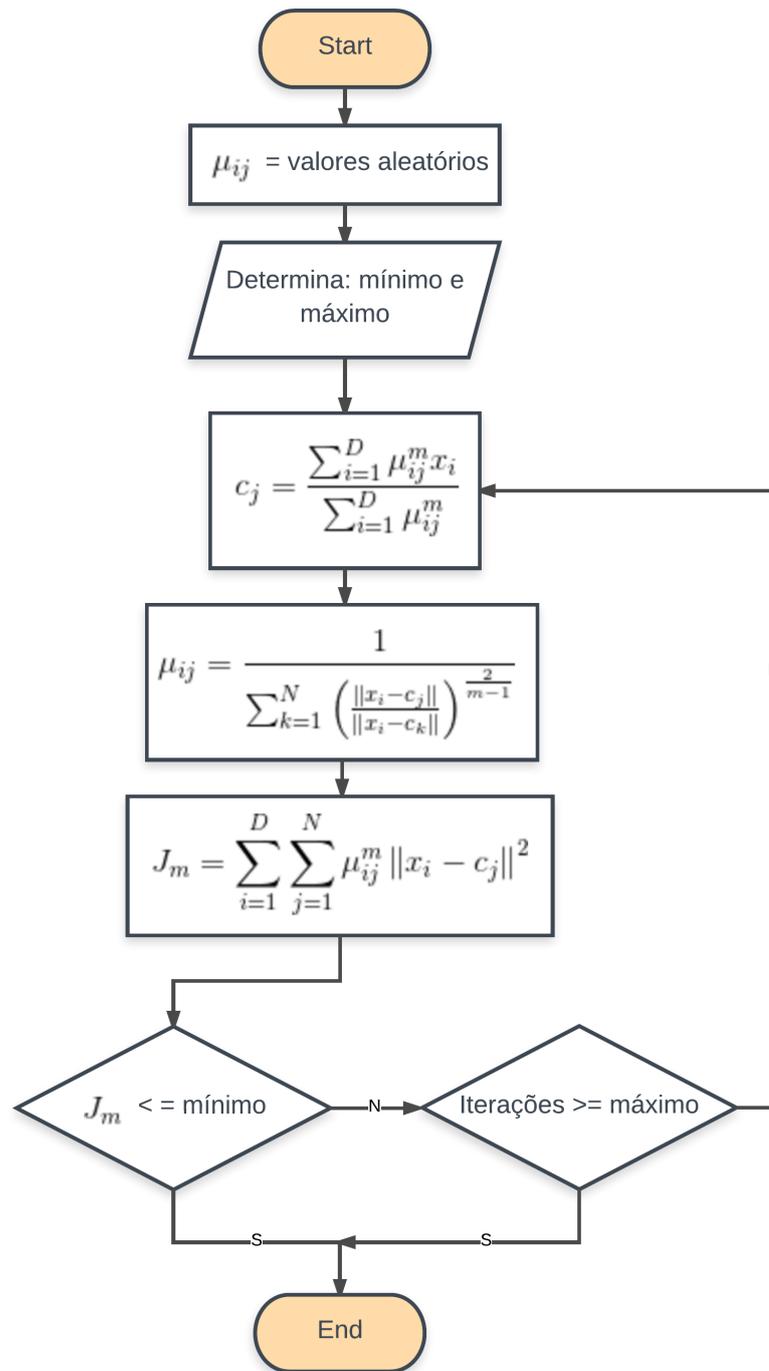
x_i é o ponto de posição i ;

c_j é o centro do *cluster* de número j ;

μ_{ij} é o grau de pertinência de x_i no *cluster* j .

J_m é a Função Objetivo que é comparada ao critério de parada.

Figura 26 – Fluxograma do algoritmo de *Clustering* FCM.



onde:

minimo é valor de parada da função;

maximo é maior número de iterações para o algoritmo.

O *MatlabR2016a* possui uma função que realiza o algoritmo descrito pela Figura 26, executado pelo comando “*fcm()*”, cuja sintaxe encontra-se na Tabela 6. É interessante salientar que para qualquer valor das características analisadas será atribuído um grau de pertinência para cada *cluster*. O código completo para o agrupamento de dados pode ser observado no Anexo A.5.

Tabela 6 – Detalhes do comando do FCM do *MatlabR2016a*.

Nome	Descrição
Sintaxe	$[centers,U] = fcm(data, Nc, options)$
<i>data</i>	conjunto de dados para <i>Clustering</i> ;
<i>Nc</i>	quantidade de <i>clusters</i> em que os dados serão agrupados;
<i>options</i>	vetor que informa respectivamente a ordem da matriz expoente, máximo número de iterações, valor mínimo do critério de parada e a flag que, quando colocada em 0, não mostra os resultados das iterações individuais;
<i>centers</i>	vetor que retorna os centros de cada um dos <i>Nc clusters</i> ;
<i>U</i>	matriz que retorna para cada um dos <i>Nc clusters</i> o nível de pertinência para todos os elementos do vetor <i>data</i> ;

3.4.3 Determinação da Função de Pertinência

No presente trabalho serão utilizadas três funções de pertinência por canal, para cada voluntário diferente, por isto o *Clustering* foi realizado com três *clusters*. Isto porque dependendo da pessoa ou do músculo que será avaliado se tem diferentes noções do que é um sinal “alto”, “baixo” ou “médio”. A partir do vetor *U*, indicado na Tabela 6, determina-se uma série de elementos que representam o grau de pertinência de cada nível de entrada das características originárias do sinal. Afim de utilizar estes valores para realizar a Inferência *Fuzzy* é necessário transformá-los em uma função de transferência que relaciona a variável de entrada não *fuzzyficada* com um determinado grau de pertinência. Sendo assim, será utilizada a função própria do *MatlabR2016a* que determina os parâmetros de uma função indicada que melhor se adapta aos elementos desejados. O comando “*fit()*” realiza esta relação através da sintaxe e configurações descritas na Tabela 7, um vetor de opções e um vetor de dados previamente preparados são necessários para utilização do comando e são indicados respectivamente pela Tabela 8 e Tabela 9. Os dados necessitam de preparação, pois os tipos de dados e formato do vetor necessitam estar adequados para o processamento do comando “*fit()*”. O código completo para Determinação da Função de Pertinência pode ser encontrado no Anexo A.5.

Tabela 7 – Detalhes do comando de *fit()* do *MatlabR2016a*.

Nome	Descrição
Sintaxe	[fitresult, gof] = fit(xData, yData, ft, opts);
<i>xData</i>	conjunto de dados do eixo independente <i>x</i> ;
<i>yData</i>	conjunto de dados do eixo dependente <i>y</i>
<i>ft</i>	tipo de função que será originada;
<i>opts</i>	vetor de opções que determina as características do comando

Tabela 8 – Detalhes do vetor *fitoptions* do *MatlabR2016a*.

Nome	Descrição
Sintaxe	opts = fitoptions('Method', 'NonlinearLeastSquares');
<i>opts</i>	vetor de opções que determina as características do comando
'Method'	indica que o parâmetro método será modificado;
'NonlinearLeastSquares'	atribuí como método mínimos quadrados;

Tabela 9 – Detalhes do vetor *prepareCurveData* do *MatlabR2016a*.

Nome	Descrição
Sintaxe	[xData, yData] = prepareCurveData(vetorx, vetory);
<i>xData</i>	conjunto de dados preparados do eixo independente <i>x</i> ;
<i>yData</i>	conjunto de dados preparados do eixo dependente <i>y</i>
<i>vetorx</i>	conjunto de dados original do eixo independente <i>x</i> ;
<i>vetory</i>	conjunto de dados original do eixo dependente <i>y</i>

3.4.4 Criação do Sistema de Inferência *Fuzzy*

O sistema de Inferência *Fuzzy* é composto por entradas, saídas, conjunto de regras e as características do mesmo, como ilustra a Figura 27 que indica as etapas para criação de um sistema *Fuzzy*. Com as informações e parâmetros encontrados até o presente capítulo é possível dar início a criação do sistema *Fuzzy* que possibilitará a determinação dos movimentos. As características selecionáveis dentro do sistema são: operação lógica "E", operação lógica "OU", método de Implicação, método de Agregação e método de *Defuzzyficação*. Afim de compor por completo um sistema *Fuzzy* é necessário criá-lo dentro da área de trabalho do *MatlabR2016a*, desta forma é possível tratar o sistema como um objeto que pode ser modificado. O *MatlabR2016a* apresenta uma função própria para criação de um sistema *Fuzzy*, comando *newfis()*, que além de criar o objeto *sistema*, também atribuí as características deste objeto. O código completo para criação do Sistema de Inferência *Fuzzy* pode ser observado no Anexo A.6, as linhas de comando utilizadas para criar o sistema foram:

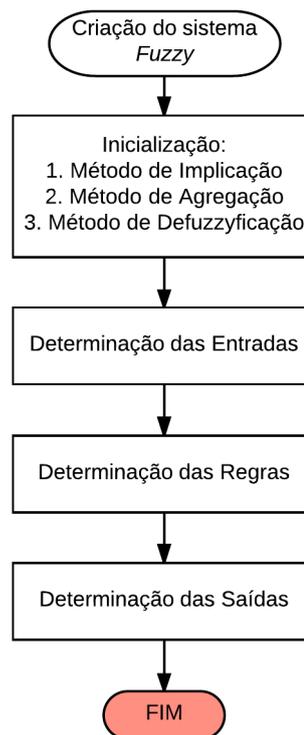
```

AndMethod='min';
OrMethod='max';
ImpMethod='prod';
AggMethod='max';
DefuzzMethod='centroid';

sistemaFIS = newfis('sistemaFIS','mamdani', AndMethod, OrMethod, ImpMethod, AggMethod, DefuzzMethod);

```

Figura 27 – Diagrama de criação de um sistema de Inferência *Fuzzy*.



onde:

$AndMethod = 'min'$ indica que a operação lógica "E" é caracterizada pelo operador de *Fuzzy* mínimo;

$ORMethod = 'max'$ indica que a operação lógica "OU" é caracterizada pelo operador *Fuzzy* de máximo;

$ImpMethod = 'prod'$ indica que o método de Implicação é caracterizado pelo operador *Fuzzy* de produto;

$AggMethod = 'max'$ indica que o método de Agregação é caracterizado pelo operador *Fuzzy* de máximo;

$DefuzzMethod = 'centroid'$ indica que o método de Defuzzyficação é o centróide;

$sistemaFIS$ é o nome dado ao objeto Sistema de Inferência *Fuzzy*;

'*mandani*' indica que o tipo de Sistema de Inferência *Fuzzy* criado é *Mandani*.

3.4.5 Composição das Entradas do Sistema de Inferência *Fuzzy*

Uma vez determinadas as funções de pertinência é possível configurar as entradas do sistema de inferência *Fuzzy* que irão realizar a *fuzzyficação* do sinal RMS obtido na parametrização dos sinais. Serão determinadas doze entradas, uma para cada canal do EMG, com três funções de pertinência já dimensionadas anteriormente.

As características principais para determinar uma entrada são: o tipo de função de pertinência, quantidade, parâmetros das funções de pertinência e a faixa de valores possíveis para entrada. Cada canal e cada indivíduo possui suas funções de pertinência própria e uma faixa de valores própria, o máximo valor RMS do sinal pode ser muito diferente entre os voluntários e entre os canais.

O *MatlabR2016a* apresenta uma função própria para adição de variáveis ao sistema *Fuzzy*, comando *addvar()*, a utilização deste comando pode ser observado nas seguintes linhas de código:

```
sistemaFIS = addvar(sistemaFIS,'input', 'nome',[0 max_v_rms]);
sistemaFIS = addmf(sistemaFIS,'input',CANAL,'alto','gaussmf',[sigma(MAIOR) c(MAIOR)]);
```

onde:

sistemaFIS é o objeto Sistema de Inferência *Fuzzy*;

'*input*' indica que a variável que está sendo adicionada é do tipo entrada ou que a FP adicionada pertence a uma entrada;

'*nome*' indica o nome dado a variável que está sendo adicionada;

[0 *max_v_rms*] indica os limites dos valores que as entradas podem assumir, no exemplo os limites vão de 0 até o valor RMS máximo para o sujeito em questão;

CANAL indica qual das variáveis de entrada a FP está sendo adicionada, *CANAL* é um número que pode assumir valores de 1 até o número de variáveis

'*alto*' indica o nome desta Função de Pertinência;

'*gaussmf*' indica o que o tipo de FP que está sendo adicionada é do tipo gaussiana;

[*sigma(MAIOR) c(MAIOR)*] indica os parâmetros da FP gaussiana, no exemplo está sendo adicionada uma FP com nome de '*alto*', por isto os parâmetros estão indexados como *MAIOR*.

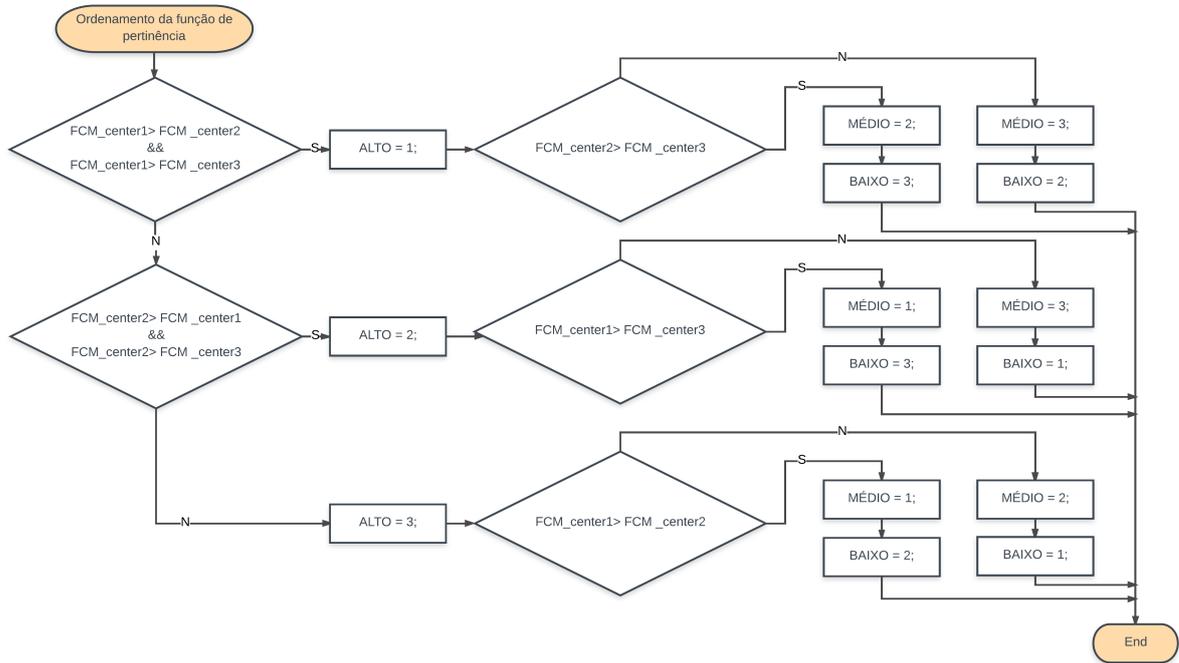
Desta forma, "*SistemaFIS*" será igual ao próprio sistema anteriormente criado mais uma variável de entrada. Também é necessário determinar os limites máximos e

mínimos que a entrada pode apresentar, para o programa desenvolvido são os valores assumidos pelos valores de tensão elétrica RMS extraídos das janelas de movimento, ou seja, de 0 até o valor máximo do vetor v_rms .

Após inserir uma variável é necessário determinar quais as funções de pertinência que compõe a *fuzzyficação* dos valores de entrada, estas funções são aquelas previamente obtidas através do *Clustering* e da aproximação por mínimos quadrados. O *MatlabR2016a* apresenta uma função própria para adição de funções de pertinência à uma variável, comando *addmf()*, a utilização deste comando pode ser observado nas linhas de código acima. Desta forma, “*sistemaFIS*” será igual ao próprio sistema anteriormente criado mais uma função de pertinência atribuída à uma entrada. As demais características determinadas para execução do comando *addmf()* são: o índice da variável à qual função de pertinência será associada; o nome dado à função, que para este exemplo é '*alto*', mas também tem-se para cada entrada mais duas FP's denominadas de '*médio*' e '*baixo*'; a morfologia da função de pertinência, que conforme indicado nos capítulos anteriores é do tipo gaussiana e os parâmetros da função gaussiana *sigma* e *c*.

O *Clustering* realizado não fornece os parâmetros de forma padronizada e se faz necessário realizar um ordenamento das funções encontradas da maior para a menor. Sendo assim, três variáveis são utilizadas para indexar os parâmetros das gaussianas, são elas *MAIOR*, *MEDIO* e *MENOR*, como pode ser observado nas linhas de código apresentadas anteriormente. A lógica de ordenamento ocorre através de um agrupamento de laços “se” que podem ser observados na Figura 28, estes testam qual é o maior dos centros dos *clusters* e a partir disso atribuem a posição da gaussiana. Em outras palavras, precisa-se identificar, a partir dos três *clusters* gerados anteriormente, qual deles refere-se as gaussianas das funções de pertinência "baixo", "médio" e "alto". O código completo para criação das entradas *Fuzzy* pode ser observado no Anexo A.7.

Figura 28 – Lógica de ordenamento das Funções de Pertinência.



onde:

$FCM_center1$ é o centro do *cluster* de número 1;

$FCM_center2$ é o centro do *cluster* de número 2;

$FCM_center3$ é o centro do *cluster* de número 3;

ALTO indexa qual *cluster* é o alto;

MÉDIO indexa qual *cluster* é o médio;

BAIXO indexa qual *cluster* é o baixo.

As Funções de pertinência são adicionadas utilizando os parâmetros previamente determinados pelo comando *fit()*, cujos parâmetros são fornecidos conforme a Equação (21), ao passo que a entrada das gaussianas do Sistema de Inferência *Fuzzy* possuem a morfologia indicada pela Equação (22), portanto se faz necessário uma adaptação dos valores afim de obter os mesmos resultados. A Equação (23) apresenta a relação entre os parâmetros σ e c_i e a Equação (24) apresenta a relação entre c e b_i .

$$FP_{encontrada}(x; b_i, c_i) = e^{-\frac{x-b_i}{c_i}} \quad (21)$$

$$FP_{FIS}(x; c, \sigma) = e^{-\frac{1}{2} \frac{x-c}{\sigma}} \quad (22)$$

$$\sigma = \sqrt{\frac{c_i^2}{2}} \quad (23)$$

$$c = b_i \quad (24)$$

onde:

$FP_{encontrada}$ a função gaussiana encontrada pelo comando $fit()$ a partir dos dados de *Clustering*;

FP_{FIS} a função gaussiana de entrada do Sistema de Inferência *Fuzzy*;

x é a variável independente das funções;

b_i é o parâmetro do numerador do expoente da função gaussiana encontrada pelo comando $fit()$ a partir dos dados de *Clustering*;

c_i é o parâmetro do denominador do expoente da função gaussiana encontrada pelo comando $fit()$ a partir dos dados de *Clustering*;

c é o parâmetro do numerador do expoente da função gaussiana de entrada do Sistema de Inferência *Fuzzy*;

σ é o parâmetro do denominador do expoente da função gaussiana de entrada do Sistema de Inferência *Fuzzy*.

3.4.6 Composição das Saídas do Sistema de Inferência *Fuzzy*

Este trabalho possui apenas uma variável de saída do Sistema de Inferência *Fuzzy*, pois o interesse está em determinar o movimento resultante do processo de análise dos resultados. Desta forma, também não há necessidade de uma complexidade de valores de saída, pois trabalha-se apenas com cinco movimentos e quanto mais identificáveis estes movimentos forem, melhor será a taxa de acerto. A saída *Fuzzy* para o Sistema de Inferência projetado é composta de 6 Funções de Pertinência do formato *Bell*, uma para cada movimento que deseja-se identificar e uma para o repouso ou movimento não identificado. As FP's não precisam se interrelacionarem e por isto são projetadas de forma a não se cruzarem, pode-se dizer que a saída do sistema tem comportamento booleano. O *MatlabR2016a* apresenta uma função própria para adição de variáveis ao sistema *Fuzzy*, comando $addvar()$. A utilização deste comando pode ser observado nas seguintes linhas:

```
sistemaFIS = addvar(sistemaFIS, 'output', 'movimento', [0 20]);
sistemaFIS = addmf(sistemaFIS, 'output', 1, 'movimento1', 'gbellmf', [0.3125 5 1]);
```

onde:

sistemaFIS é o objeto Sistema de Inferência *Fuzzy*;

'*output*' indica que a variável que está sendo adicionada é do tipo saída ou que a FP adicionada pertence a uma saída;

'*movimento*' indica o nome dado a variável que está sendo adicionada;

[0 20] indica os limites dos valores que as entradas podem assumir, no exemplo os limites vão de 0 até 20;

1 indica qual das variáveis de saída a FP está sendo adicionada, pode assumir valores de 1 até o número de variáveis de saída, para presente projeto este número é sempre 1;

'*movimento1*' indica o nome desta Função de Pertinência;

'*gbellmf*' indica o que o tipo de FP que está sendo adicionada é do tipo Bell;

[0, 3125 5 1] indica os parâmetros da FP Bell.

3.4.7 Composição das Regras do Sistema de Inferência *Fuzzy*

A composição das regras do Sistema de Inferência *Fuzzy* depende das premissas e determinações criadas pelo especialista. O especialista é a inteligência que possui conhecimento do que está sendo analisado e é capaz de determinar padrões e assim gerar as regras necessárias para obter as saídas desejadas. No presente trabalho o especialista é a própria autora da presente monografia e que elaborou as seguintes premissas:

1. na maioria dos sinais uma das funções de pertinência possui grau de pertinência maior do que 70%;
2. no sinal que não possui tal característica, pelo menos uma das funções de pertinência possui um valor próximo de 0% de pertinência;
3. os sinais de mesmo movimento ou mesmo sujeito não apresentam padrão de forma a permitir a criação de regras que se apliquem as repetições do movimento;

Para criação das regras foram aplicadas as premissas descritas anteriormente e, considerando-se a sintaxe disponível no *MatlabR2016a*, são necessárias as determinações:

1. para cada uma das entradas, será testada a FP alto, baixo, médio ou nenhuma?
2. serão testados os graus de pertinência altos ou baixos da FP selecionada?
3. se esta regra for verdadeira, qual a FP de saída ela resulta?
4. qual o nível de importância desta regra (de 0 até 1)?
5. qual o tipo de conector lógico entre as regras (1 para "E" e 2 para "OU")?

O *MatlabR2016a* apresenta uma função própria para adição de regras ao sistema *Fuzzy*, comando *addrule()*, a utilização deste comando pode ser observado nas linhas de código a seguir:

```

sistemaFIS = addrule(sistemaFIS, regras(i_regra,:));
    
```

onde:

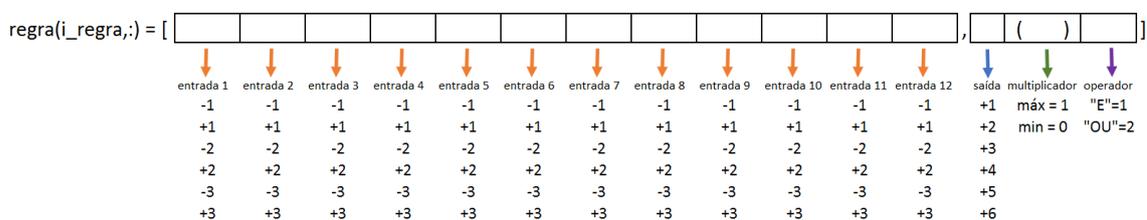
sistemaFIS é o objeto Sistema de Inferência *Fuzzy*;

regras(i_regras,:) adiciona o vetor que contém a regra que está sendo adicionada.

O Formato do vetor de adição da regra pode ser observado na Figura 29, as posições marcadas como entrada podem assumir valores inteiros de -1 até +3, onde o módulo do valor indica qual das FP's a regra irá testar (0- "nenhuma", 1- "alto", 2- "médio" e 3- "baixo"), o sinal positivo testa níveis de pertinência altos e o negativo níveis de pertinência baixos. A posição de saída pode assumir valores inteiros de 1 até 6 (1- movimento 1, 2- movimento 2, 3- movimento 3, 4- movimento 4, 5- movimento 5 e 6- repouso ou movimento não identificado) e indica qual das FP's de saída esta regra se refere. A posição multiplicador indica a importância desta regra perante as demais e pode assumir valores reais de 0 até 1. Por fim, a posição operador indica se a regra é do tipo "E" ou do tipo "OU".

Exemplificando, para a regra linguística “Se (entrada 1 não é alta) E (entrada 2 é alta) E (entrada 3 não é baixa) E (entrada 4 não é baixa) E (entrada 5 é média) E (entrada 6 não é alta) E (entrada 7 é baixa) E (entrada 8 é baixa) E (entrada 9 é media) E (entrada 10 é baixa) E (entrada 11 é média) E (entrada 12 é baixa) ENTÃO (movimento é o 1), esta regra tem peso 1”, o vetor correspondente seria "regra(1,:) = [-1 1 -3 -3 2 -1 3 3 2 3 2 3,1 (1) 1];".

Figura 29 – Vetor de Adição de Regras.

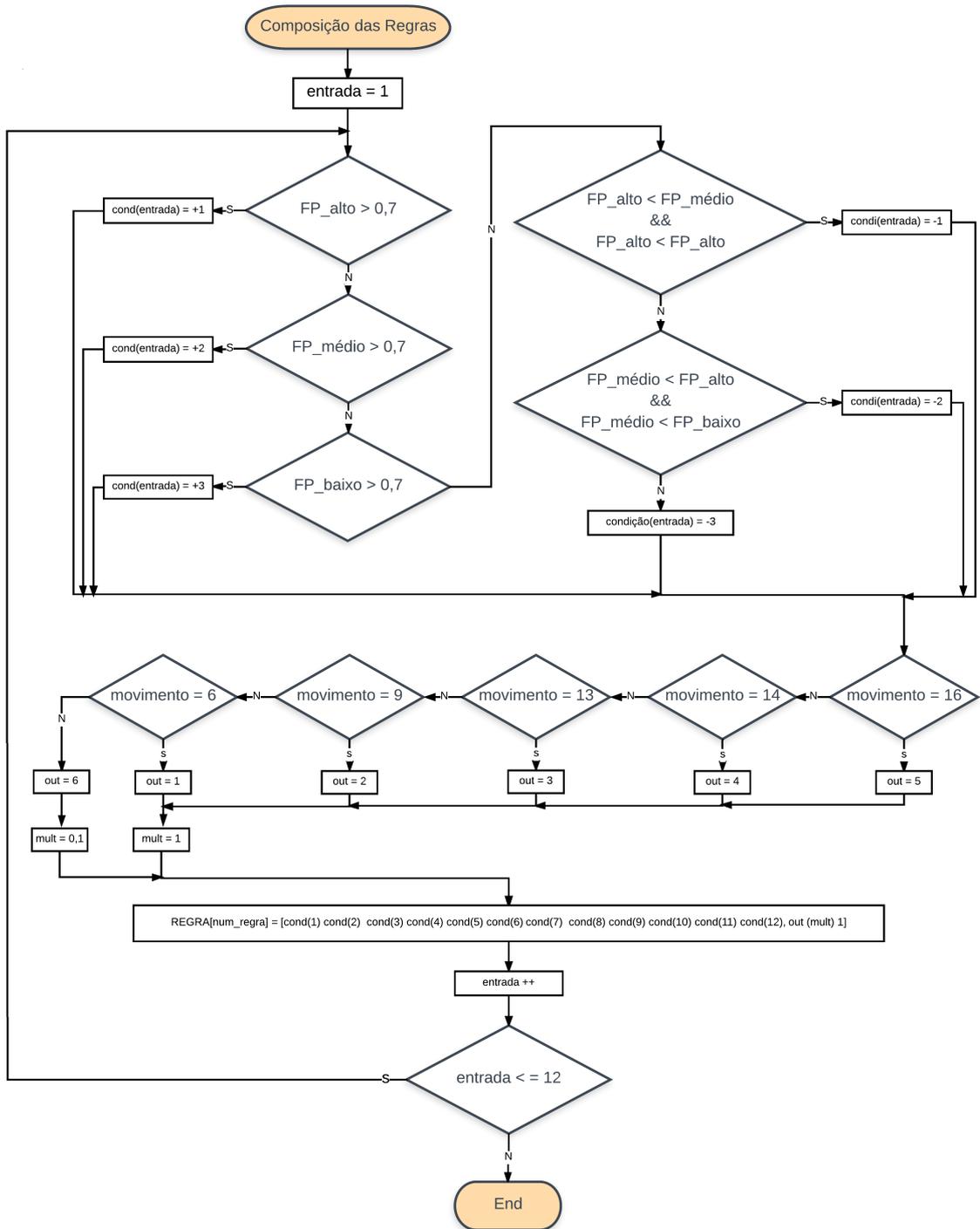


Dadas as observações do sinal e a possibilidade de construção das regras foi possível determinar um algoritmo de geração de regras que segue o raciocínio apresentado na Figura 30, onde entrada representa as 12 entradas advindas dos 12 canais de medição do sEMG. *FP_alto*, *FP_médio* e *FP_baixo* são os valores de pertinência, para uma

dada entrada de tensão elétrica RMS, das funções de pertinência alto, médio e baixo, respectivamente. Sempre que um nível de pertinência apresentar valor maior do que 70% a regra será criada com o número da função de transferência positivo, significando que irá testar se existe pertinência para com esta função. Caso nenhuma das três FP's apresente um valor acima do estipulado, então testa-se qual a menor pertinência das três funções restantes e a regra é criada com o número desta FP negativo, significando que irá testar se a pertinência desta função é baixa.

Para compor a regra ainda restam dois importantes parâmetros, o primeiro deles seria o movimento. Toda regra criada gera um movimento de saída, ou seja, se esta regra for verdadeira, então a saída deve ser determinado movimento. Isto é o princípio da "if-then rule" descrita anteriormente. Para a geração das regras são utilizados dados conhecidos, ou seja, tem-se a entrada e sabe-se qual é a saída desejada. A composição de regras cria um valor de saída que vai de 1 até 5 e cada um dos valores representa um movimento avaliado diferente. Quando a saída equivale a 6 indica que esta regra não representa nenhum dos movimentos avaliados, em outras palavras, é uma regra que caracteriza o repouso ou os movimentos que não se enquadram no presente trabalho. Por último, estipula-se um fator multiplicador que potencializa ou não o efeito de uma regra, ou seja, se mais de uma regra for verdadeira aquela que possuir maior fator multiplicador será a mais representativa na saída. Como está sendo avaliado um número pequeno de movimentos e tem-se uma variedade muito maior de janelas que não abrangem nenhum movimento avaliado, todas as regras que tem como saída um dos movimentos analisados por este trabalho ganham fator multiplicador 1, enquanto os demais ganham fator multiplicador 0,1.

Figura 30 – Lógica de Composição das Regras do Sistema de Inferência *Fuzzy*.



onde:

entrada indica qual das 12 entradas está sendo avaliada;

FP_alto é o valor de pertinência da FP alto para a entrada avaliada;

FP_médio é o valor de pertinência da FP médio para a entrada avaliada;

FP_baixo é o valor de pertinência da FP baixo para a entrada avaliada;

cond é a condição que compõe a regra, indexado conforme a entrada;

movimento é o movimento que compõe a regra;

out é a saída que compõe a regra;

mult é o fator multiplicador que compõe a regra.

3.4.8 Geração dos Resultados do Sistema de Inferência *Fuzzy*

Após a construção de todo o Sistema de Inferência *Fuzzy*, adicionando as entradas, saídas e regras é possível realizar a Inferência *Fuzzy*. O presente trabalho propõe-se a avaliar os movimentos usando todos os métodos disponíveis pelo *MatlabR2016a* de Implicação, Agregação e *Defuzzyficação*, sendo assim, as saídas serão geradas a partir de uma combinação de resultados para possibilitar a comparação entre as diversas configurações do Sistema *Fuzzy*. A Tabela 10 apresenta todas as combinações que serão avaliadas, dentre elas, dois métodos de Implicação (*mínimo* e *produto*), três métodos de Agregação (*máximo*, *soma* e *probor*) e cinco métodos de *Defuzzyficação* (*centroid*, *bisector*, *mom*, *som* e *lom*), totalizando 30 diferentes análises *Fuzzy* para cada uma das janelas de movimento. O código completo para geração dos Resultados do Sistema de Inferência *Fuzzy* podem ser observados nos Anexos [A.8](#) até [A.12](#).

Tabela 10 – Combinações dos métodos de Implicação, Agregação e Defuzzificação para Avaliação dos Resultados do Sistema de Inferência *Fuzzy*.

combinação	IMPLICAÇÃO	AGREGAÇÃO	DEFUZZIFICAÇÃO
1	<i>min</i>	<i>max</i>	<i>centroid</i>
2	<i>min</i>	<i>max</i>	<i>bisector</i>
3	<i>min</i>	<i>max</i>	<i>mom</i>
4	<i>min</i>	<i>max</i>	<i>lom</i>
5	<i>min</i>	<i>max</i>	<i>som</i>
6	<i>min</i>	<i>sum</i>	<i>centroid</i>
7	<i>min</i>	<i>sum</i>	<i>bisector</i>
8	<i>min</i>	<i>sum</i>	<i>mom</i>
9	<i>min</i>	<i>sum</i>	<i>lom</i>
10	<i>min</i>	<i>sum</i>	<i>som</i>
11	<i>min</i>	<i>probor</i>	<i>centroid</i>
12	<i>min</i>	<i>probor</i>	<i>bisector</i>
13	<i>min</i>	<i>probor</i>	<i>mom</i>
14	<i>min</i>	<i>probor</i>	<i>lom</i>
15	<i>min</i>	<i>probor</i>	<i>som</i>
16	<i>prod</i>	<i>max</i>	<i>centroid</i>
17	<i>prod</i>	<i>max</i>	<i>bisector</i>
18	<i>prod</i>	<i>max</i>	<i>mom</i>
19	<i>prod</i>	<i>max</i>	<i>lom</i>
20	<i>prod</i>	<i>max</i>	<i>som</i>
21	<i>prod</i>	<i>sum</i>	<i>centroid</i>
22	<i>prod</i>	<i>sum</i>	<i>bisector</i>
23	<i>prod</i>	<i>sum</i>	<i>mom</i>
24	<i>prod</i>	<i>sum</i>	<i>lom</i>
25	<i>prod</i>	<i>sum</i>	<i>som</i>
26	<i>prod</i>	<i>probor</i>	<i>centroid</i>
27	<i>prod</i>	<i>probor</i>	<i>bisector</i>
28	<i>prod</i>	<i>probor</i>	<i>mom</i>
29	<i>prod</i>	<i>probor</i>	<i>lom</i>
30	<i>prod</i>	<i>probor</i>	<i>som</i>

O *MatlabR2016a* apresenta uma função própria para reprodução do sistema *Fuzzy*, comando *evalfis()*, a utilização deste comando pode ser observado como:

```
out(movimento,out_avaliacao) = evalfis(v_rms(:),sistemaFIS);
```

onde:

out é a matriz de saída que indica o movimento resposta da Inferência *Fuzzy*;

movimento é o segmento cuja Inferência *Fuzzy* está sendo aplicada para verificação do movimento;

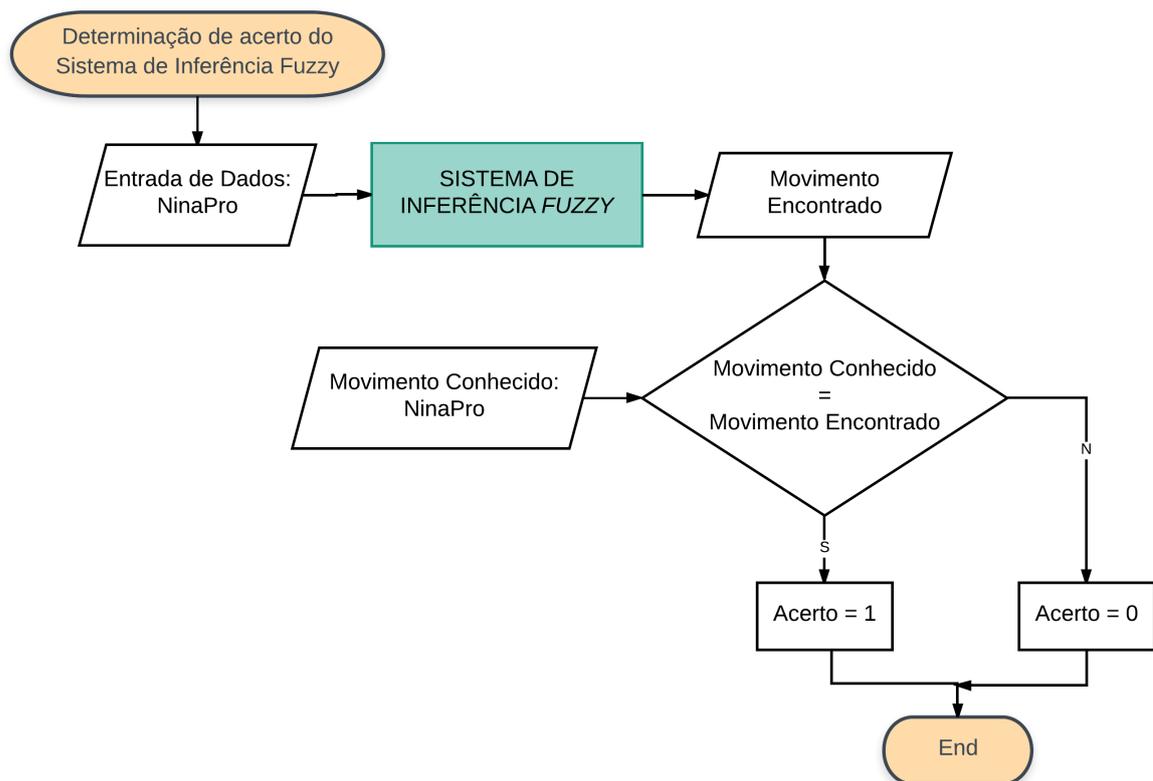
$out_avaliacao$ indica qual combinação da Tabela 10 está sendo avaliada, assume valores de 1 até 30;

v_{rms} é o valor RMS do segmento do sinal que está sendo avaliado;

$sistemaFIS$ é o objeto Sistema de Inferência *Fuzzy*.

Por fim, obtém-se a taxa de acerto da análise *Fuzzy* a partir do sistema determinado, como para cada um dos elementos da base de dados NinaPro tem-se o conhecimento do movimento realizado, pode-se comparar este valor ao obtido pela Inferência *Fuzzy* realizada. A Figura 31 indica o algoritmo de determinação da taxa de acerto do Sistema de Inferência *Fuzzy*, se o movimento encontrado foi igual ao movimento conhecido pela base de dados atribuí-se o valor 1 a *flag* de acerto, caso contrário, atribuí-se 0. A taxa de acerto será a quantidade de movimentos acertados dividido pela quantidade total de movimentos avaliados, como mostrado na Equação (25).

Figura 31 – Lógica para caracterização do Movimento e Determinação do Acerto.



$$taxa_{acerto} = \frac{movimentos_{acertados}}{movimentos_{avaliados}} \quad (25)$$

onde:

$taxa_{acerto}$ é taxa de acerto do Sistema de Inferência *Fuzzy*;

$movimentos_{acertados}$ é a quantidade de movimentos acertados pela Inferência *Fuzzy*;

$movimentos_{avaliados}$ é o total de movimentos avaliados pela Inferência *Fuzzy*.

4 RESULTADOS E DISCUSSÕES

4.1 Etapa de Pré-Processamento

Com intuito de avaliar o processo de pré-processamento, todas as etapas descritas pela Figura 19 serão exemplificadas utilizando o sinal do canal 1, do voluntário 1 da sub-base II no período de 330s até 410s. O sinal original, ou seja, sem nenhum processamento prévio, é indicado na Figura 32 no domínio do tempo e pela Figura 33 no domínio da frequência. Como é possível observar na Figura 33 a região que apresenta maior concentração de energia do sinal encontra-se de 20Hz até 400Hz. Na Figura 32 tem-se uma janela de tempo que abrange nove movimentos e torna-se claro a existência de ruído entre estes, observa-se picos de amplitude entre os movimentos 4 e 5, por exemplo.

Figura 32 – Sinal de sEMG bruto no domínio do tempo.

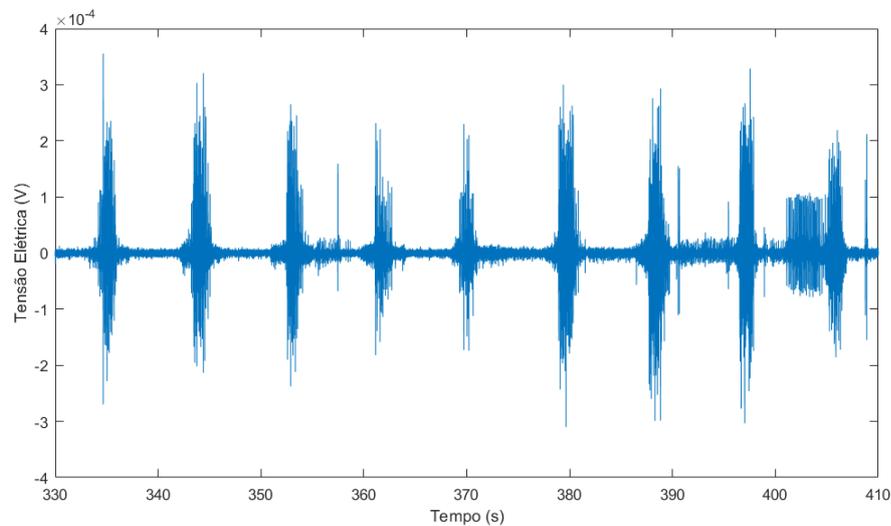
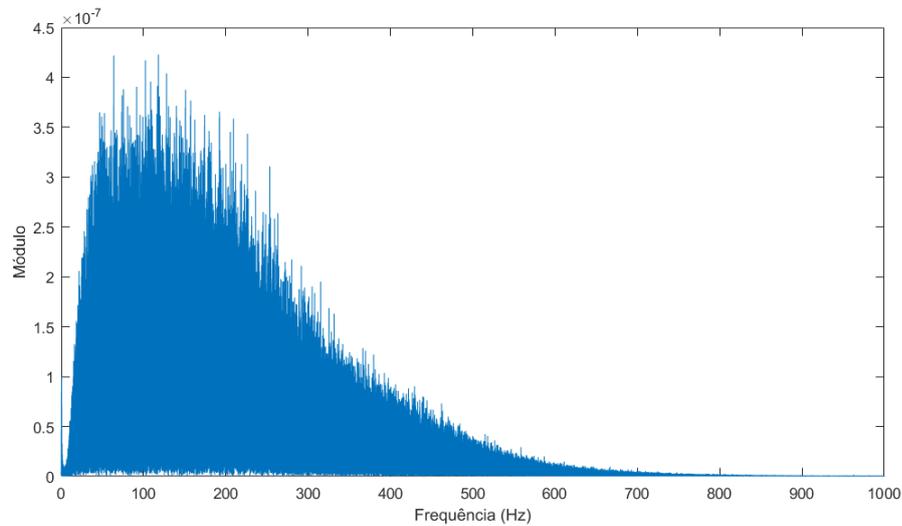


Figura 33 – FFT do Sinal bruto no domínio da frequência.



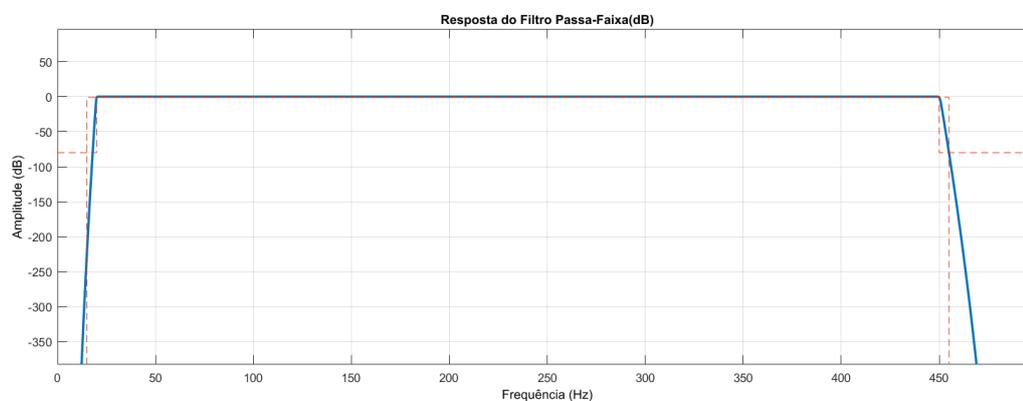
4.1.1 Resultados da Filtragem do Sinal

Dadas as configurações do vetor h conforme indicado na Tabela 3, o filtro projetado foi do tipo *Butterworth*, com frequência de passagem de 20Hz, frequência de corte de 450Hz e atenuação de 80dB. A Tabela 11 indica os parâmetros utilizados para o projeto do filtro Passa-Faixa do presente trabalho, baseado na descrição detalhada da sintaxe na Tabela 3. A Figura 34 apresenta a resposta em frequência simulada do filtro proposto.

Tabela 11 – Parâmetros utilizados no projeto do Filtro utilizando comando *design()* do *MatlabR2016a*.

Nome	Descrição
F_{stop1}	15Hz
F_{pass1}	20Hz
F_{pass2}	450Hz
F_{stop2}	455Hz
A_{stop1}	-80dB
A_{pass}	0dB
A_{stop2}	-80dB
F_s	2kHz

Figura 34 – Resposta Simulada em Frequência do filtro Passa-Faixa.



Considerando o Filtro Passa-Faixa descrito no presente capítulo, o sinal de sEMG da base de dados NINAPRO assumiu a morfologia indicada pela Figura 35 no domínio do tempo e pela Figura 36. Embora não seja tão perceptível a diferença no domínio do tempo é claramente identificável que há uma significativa atenuação das frequências acima de 450Hz (ver Figuras 32 e 33).

Figura 35 – Sinal Filtrado no Tempo.

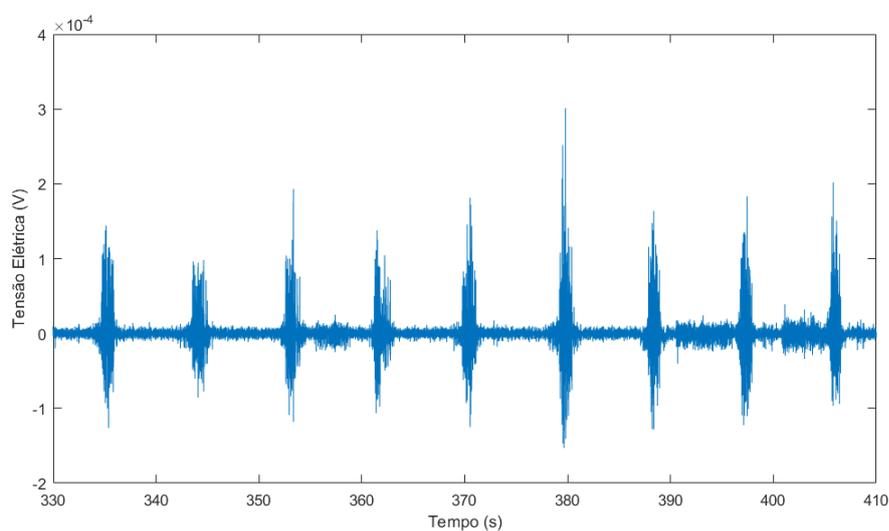
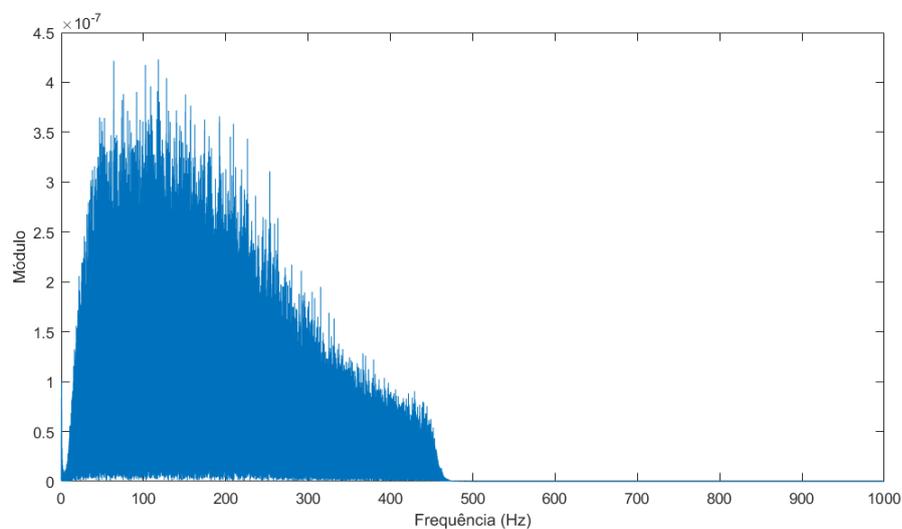


Figura 36 – FFT do Sinal Filtrado.



4.1.2 Retificação e Normalização do sinal

Após passar pela etapa de filtragem o sinal é normalizado utilizando a função módulo indicada pela Equação (14). Os efeitos sobre o sinal no domínio do tempo podem ser observados na Figura 37 e no domínio da frequência na Figura 38.

Figura 37 – Sinal Retificado no Tempo.

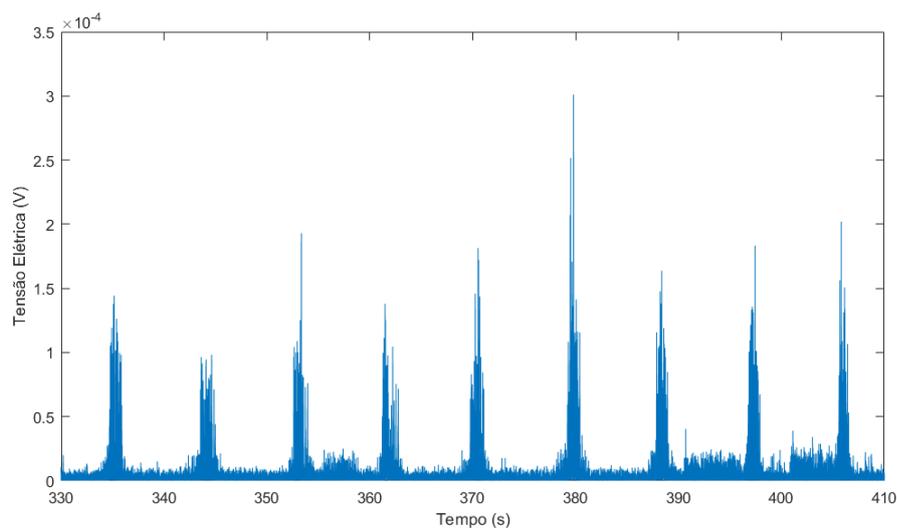
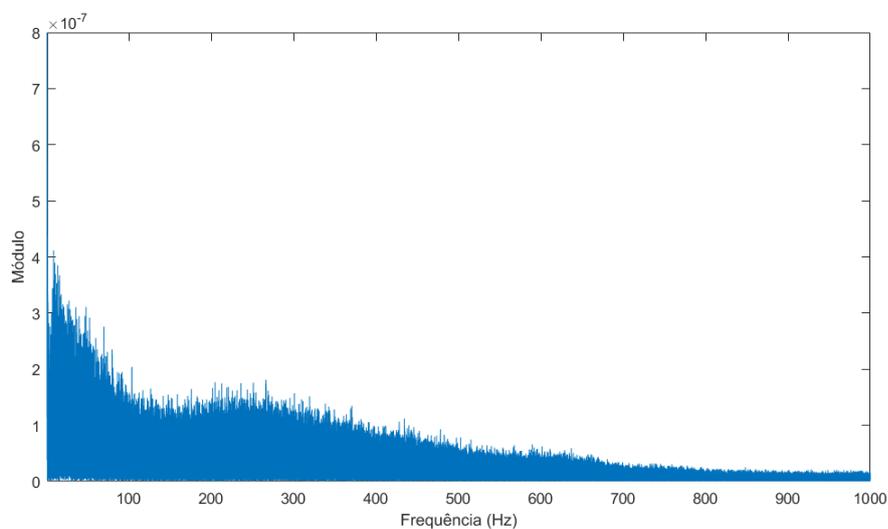


Figura 38 – FFT do Sinal Retificado (1 até 1000Hz).



A normalização do sinal ocorre após a etapa de retificação e divide todos os elementos do vetor do sinal retificado pelo ponto de maior amplitude, assim o vetor normalizado assume valores intermediários entre zero e um. Os efeitos sobre o sinal no domínio do tempo podem ser observados na Figura 39 e no domínio da frequência na Figura 40.

Figura 39 – Sinal Normalizado no Tempo.

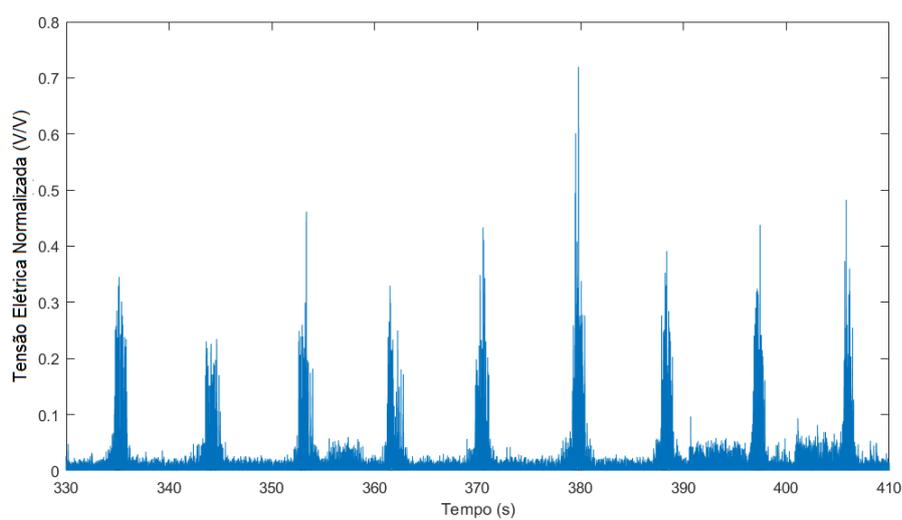
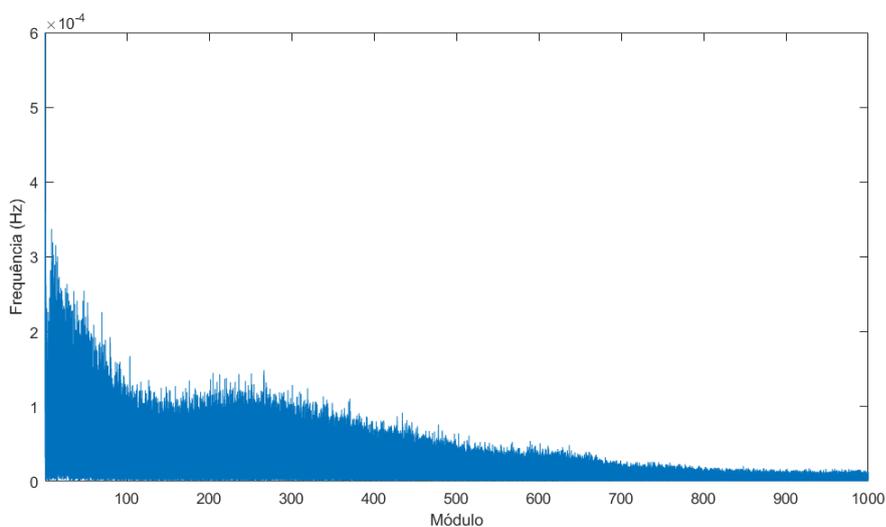


Figura 40 – FFT do Sinal Normalizado.

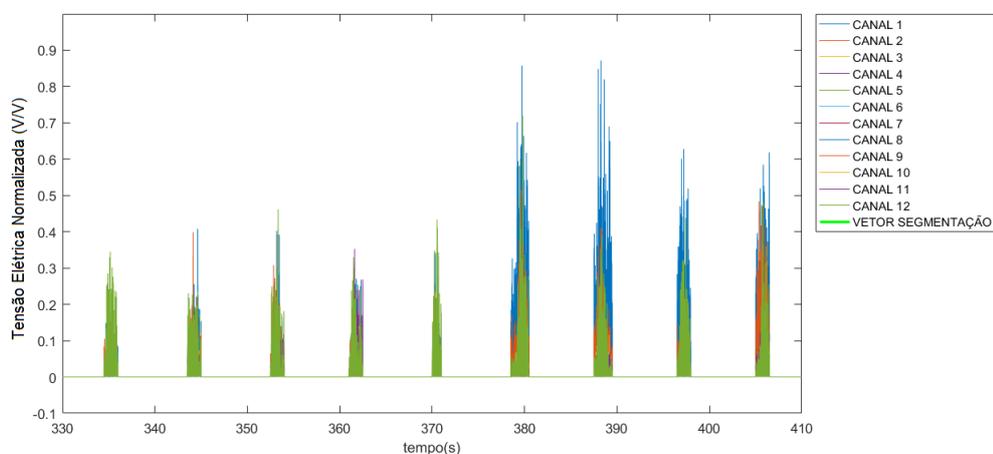


É importante, ressaltar, que todas as etapas anteriormente descritas e exemplificadas com apenas um sinal de um voluntário foram executadas para todos os sinais selecionados de sEMG da base analisada.

4.1.3 Resultados da Segmentação do Sinal

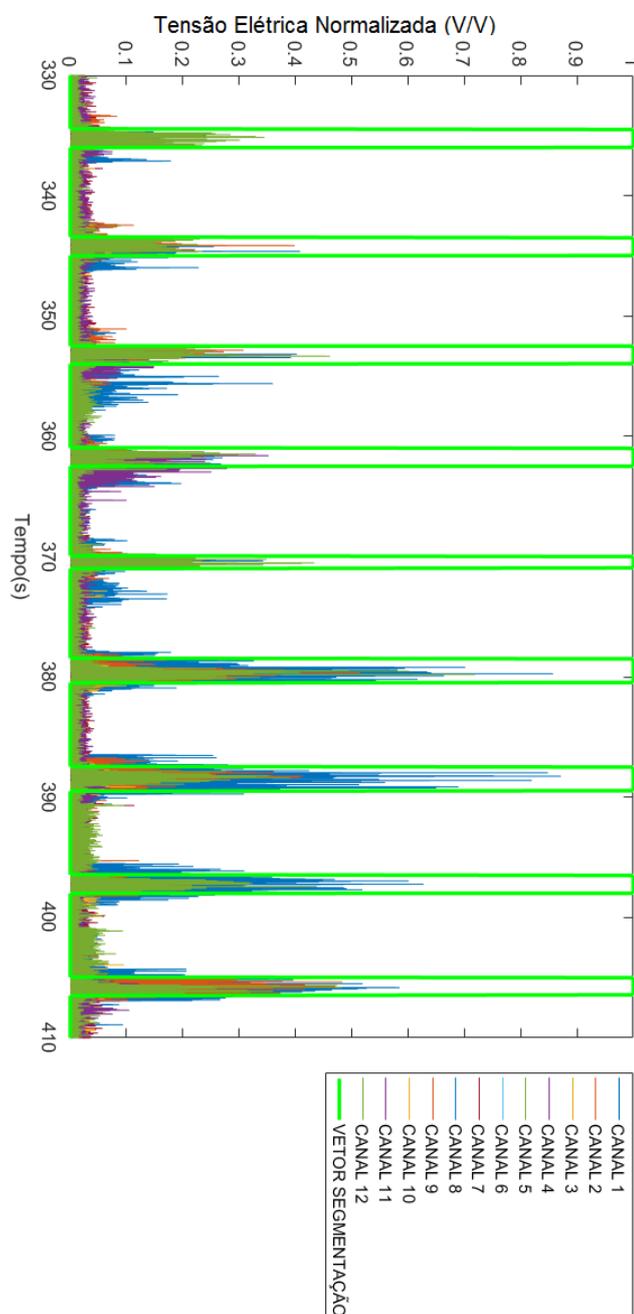
O vetor sinal segmentado nada mais é do que o vetor sinal, após a filtragem, retificação e normalização, multiplicado ponto a ponto pelo vetor que identifica o início e o fim de cada movimento. A Figura 41 mostra um exemplo com sinal segmentado. Analisando-se esta Figura 41, é possível observar que os intervalos entre movimentos agora são nítidos e iguais a zero. Portanto, com esta metodologia é possível analisar somente os trechos de sinal onde há movimento.

Figura 41 – Exemplo de Sinal Segmentado no Tempo.



Utilizando o algoritmo descrito pela Figura 21 é obtido um vetor de mesmo tamanho do vetor do sinal normalizado que tem valor com nível alto (valor um) quando algum movimento foi identificado e valor com nível baixo (valor zero) caso contrário. A Figura 42 apresenta graficamente, no domínio do tempo, o conteúdo deste vetor segmentação por detecção de limiar contrastando com o sinal normalizado de todos os canais. Este método é capaz de realizar a identificação de movimentos, porém apresenta janelas estreitas que não abrangem as extremidades da atividade muscular.

Figura 42 – Vetor Segmentação por detecção de limiar x Sinal normalizado.



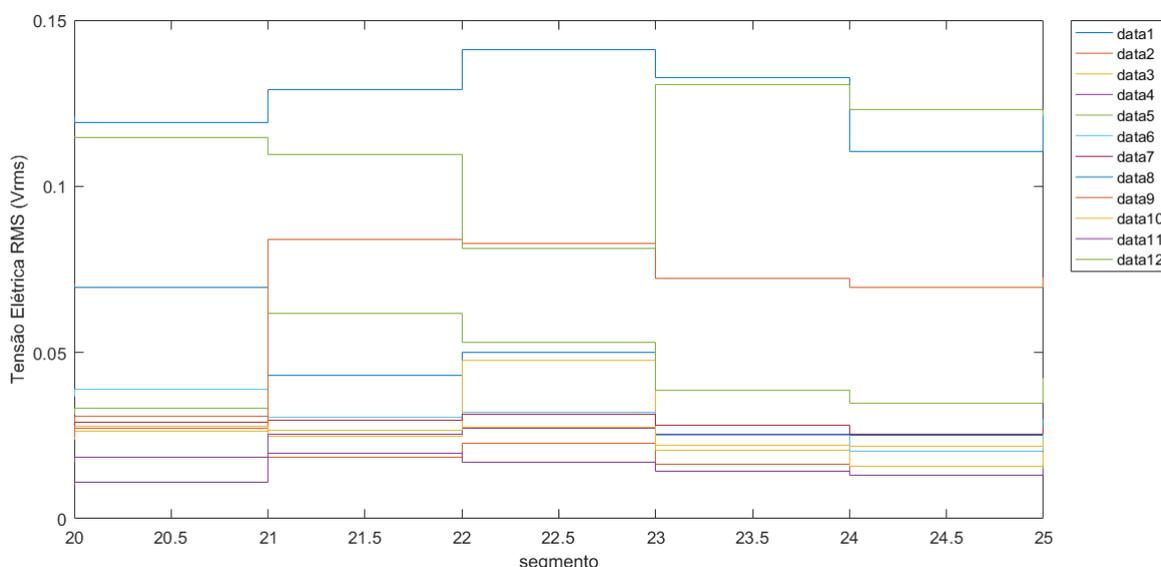
4.2 Resultados da Etapa de Processamento do sinal

Com objetivo de avaliar a etapa de processamento, todas as etapas descritas pela Figura 24 serão exemplificadas utilizando o sinal do canal 1, do voluntário 1 da sub-base II no período de 330s até 410s. Os dois sinais bases utilizados nesta etapa serão o sinal segmentado e o vetor segmentação.

4.2.1 Resultados da Obtenção dos parâmetros dos movimentos

Para cada um dos segmentos de todos os 12 canais de sEMG foi extraída a tensão elétrica RMS do sinal, sendo assim, foi formada uma matriz de dimensões (segmentos x canal x característica). Como exemplo, a Figura 43 mostra para os segmentos de 20 até 25 os níveis RMS de tensão elétrica para os doze canais, desta forma, é possível compreender a relação entre a amplitude do sinal dos canais e movimento.

Figura 43 – Valor RMS dos 12 canais para os segmentos 20 até 25.



As informações obtidas neste capítulo serão utilizadas como entrada do Sistema de Inferência *Fuzzy* projetado, a partir destas informações que toda a lógica *Fuzzy* será dimensionada. A grande importância desta etapa é compreender que as tensões RMS dos segmentos apresentam comportamento distinto entre os canais e, principalmente, entre os movimentos. Os próximos capítulos visam compreender e determinar o comportamento da VRMS através de agrupamentos e determinação de padrões.

Não serão apresentados neste capítulo todos os dados obtidos, mas sim os limites máximos e mínimos, como mostram as Tabelas 12, ?? e ?. Note que os limites dos valores variam entre sujeitos e canais, ou seja, não há uma padronização para valores e se faz necessário caracterizar individualmente o comportamento o parâmetro de tensão elétrica RMS.

Tabela 12 – Combinações dos métodos de Implicação, Agregação e Defuzzyficação para Avaliação dos Resultados do Sistema de Inferência *Fuzzy*.

Voluntário	CANAL	VRMS máximo	VRMS mínimo
aS1	1	0.201330693	0
aS1	2	0.105340075	0
aS1	3	0.154060653	0
aS1	4	0.174505018	0
aS1	5	0.14562004	0
aS1	6	0.161379177	0
aS1	7	0.242823277	0
aS1	8	0.21099488	0
aS1	9	0.165515897	0
aS1	10	0.170150831	0
aS1	11	0.071386228	0
aS1	12	0.050591503	0
aS10	1	0.143419057	0
aS10	2	0.149670651	0
aS10	3	0.115041954	0
aS10	4	0.108348791	0
aS10	5	0.108705309	0
aS10	6	0.110458879	0
aS10	7	0.138177526	0
aS10	8	0.164086895	0
aS10	9	0.106600986	0
aS10	10	0.211171537	0
aS10	11	0.068926096	0
aS10	12	0.064409135	0
aS2	1	0.150395433	0
aS2	2	0.195917648	0
aS2	3	0.115739588	0
aS2	4	0.152001612	0
aS2	5	0.231966094	0
aS2	6	0.188908754	0
aS2	7	0.192913883	0
aS2	8	0.1857712	0
aS2	9	0.190642148	0
aS2	10	0.165427379	0
aS2	11	0.151649545	0
aS2	12	0.176532882	0
aS4	1	0.171827548	0
aS4	2	0.150698283	0
aS4	3	0.166841931	0
aS4	4	0.096898943	0
aS4	5	0.090005047	0
aS4	6	0.184133768	0
aS4	7	0.195208881	0
aS4	8	0.19646726	0
aS4	9	0.109092434	0
aS4	10	0.148213295	0
aS4	11	0.127097754	0
aS4	12	0.137848037	0

Voluntário	CANAL	VRMS máximo	VRMS mínimo
aS5	1	0.056213193	0
aS5	2	0.111390815	0
aS5	3	0.014394828	0
aS5	4	0.063037214	0
aS5	5	0.132736316	0
aS5	6	0.026661762	0
aS5	7	0.027705854	0
aS5	8	0.047599314	0
aS5	9	0.027142555	0
aS5	10	0.15881851	0
aS5	11	0.044037171	0
aS5	12	0.063297215	0
aS8	1	0.203679516	0
aS8	2	0.136012112	0
aS8	3	0.133210661	0
aS8	4	0.168730091	0
aS8	5	0.144220445	0
aS8	6	0.187829478	0
aS8	7	0.242336335	0
aS8	8	0.222803953	0
aS8	9	0.173823265	0
aS8	10	0.245347744	0
aS8	11	0.143419285	0
aS8	12	0.121547786	0
S2	1	0.141521089	0
S2	2	0.146504288	0
S2	3	0.112837974	0
S2	4	0.190070276	0
S2	5	0.098961959	0
S2	6	0.121813689	0
S2	7	0.129989304	0
S2	8	0.121538338	0
S2	9	0.155001775	0
S2	10	0.138917133	0
S2	11	0.132249086	0
S2	12	0.143034577	0
S3	1	0.077915372	0
S3	2	0.08894298	0
S3	3	0.137713728	0
S3	4	0.130837552	0
S3	5	0.107742354	0
S3	6	0.145458746	0
S3	7	0.165172898	0
S3	8	0.184690907	0
S3	9	0.166466815	0
S3	10	0.180474873	0
S3	11	0.126896561	0
S3	12	0.142093417	0

Voluntário	CANAL	VRMS máximo	VRMS mínimo
S11	1	0.119710442	0
S11	2	0.113150954	0
S11	3	0.184178802	0
S11	4	0.148265178	0
S11	5	0.104986697	0
S11	6	0.12663584	0
S11	7	0.151400059	0
S11	8	0.158281677	0
S11	9	0.15116289	0
S11	10	0.131156743	0
S11	11	0.134467025	0
S11	12	0.144341138	0
S19	1	0.172257756	0
S19	2	0.165153534	0
S19	3	0.159234965	0
S19	4	0.117690288	0
S19	5	0.140285039	0
S19	6	0.09132138	0
S19	7	0.110736005	0
S19	8	0.156875601	0
S19	9	0.139045269	0
S19	10	0.1389036	0
S19	11	0.13213201	0
S19	12	0.140217121	0
S25	1	0.132116038	0
S25	2	0.120447084	0
S25	3	0.176642435	0
S25	4	0.156197655	0
S25	5	0.114005752	0
S25	6	0.177049886	0
S25	7	0.128027453	0
S25	8	0.220416748	0
S25	9	0.141173148	0
S25	10	0.180728489	0
S25	11	0.150101121	0
S25	12	0.158770445	0
S39	1	0.093113771	0
S39	2	0.124212632	0
S39	3	0.104003284	0
S39	4	0.107948791	0
S39	5	0.05260104	0
S39	6	0.046217282	0
S39	7	0.142863692	0
S39	8	0.158434668	0
S39	9	0.102608517	0
S39	10	0.128922634	0
S39	11	0.067636282	0
S39	12	0.042870516	0

4.2.2 Resultados do *Clustering* dos parâmetros

A partir da matriz de dados obtida na etapa anterior é possível agrupar os dados em *clusters*. No presente trabalho serão três *clusters* por voluntário e por canal, gerando um total de 432 *clusters*. A Figura 59 mostra, como exemplo, a distribuição das amplitudes para todos os canais do voluntário 3 da sub-base II. É possível concluir que, embora existam algumas concentrações de elementos, como por exemplo, o canal 11 entre 0.04Vrms e 0.06Vrms, os valores se espalharam ao longo do intervalo compreendido entre 0Vrms e 0.2Vrms. A Figura 45 mostra como estão distribuídas as amplitudes para todos os canais do voluntário amputado 2 da sub-base III, afim de comparar com o comportamento de um voluntário intacto. Da mesma forma, embora existam algumas concentrações de elementos, como por exemplo, o canal 7 entre 0.05Vrms e 0.1Vrms, os valores se espalharam ao longo do intervalo compreendido entre 0Vrms e 0.25Vrms. Mais exemplos de distribuição para outros sujeitos deste trabalho encontram-se no Anexo B. Lembrando que os valores variaram entre os canais, devido aos diferentes grupos musculares que estão sendo analisados e também variaram por indivíduo, pois as características fisiológicas alteraram os sinais de sEMG.

Figura 44 – Distribuição de amplitudes de todos os canais para voluntário 3 da sub-base II.

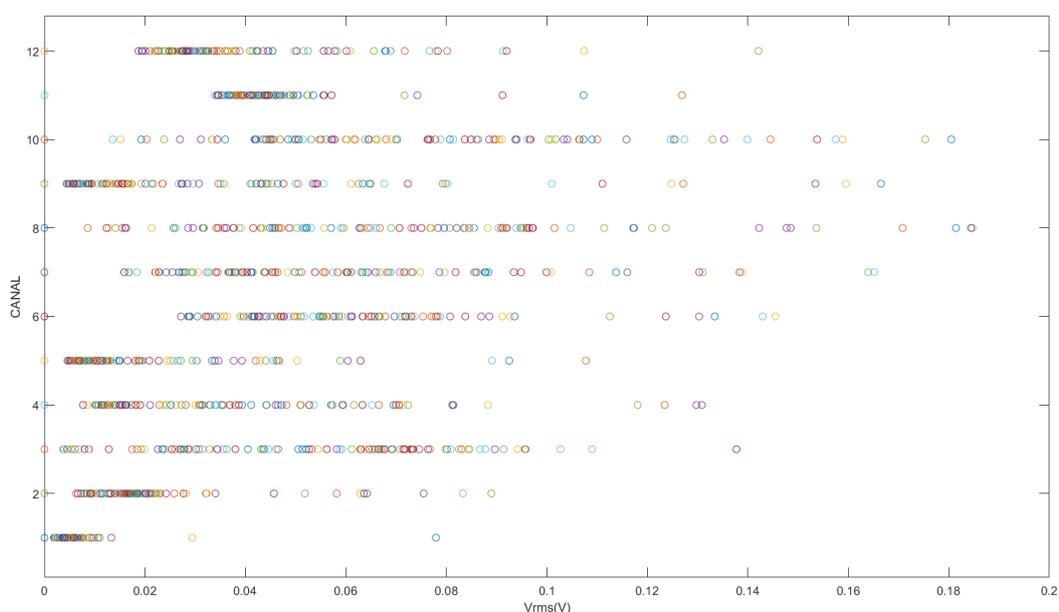
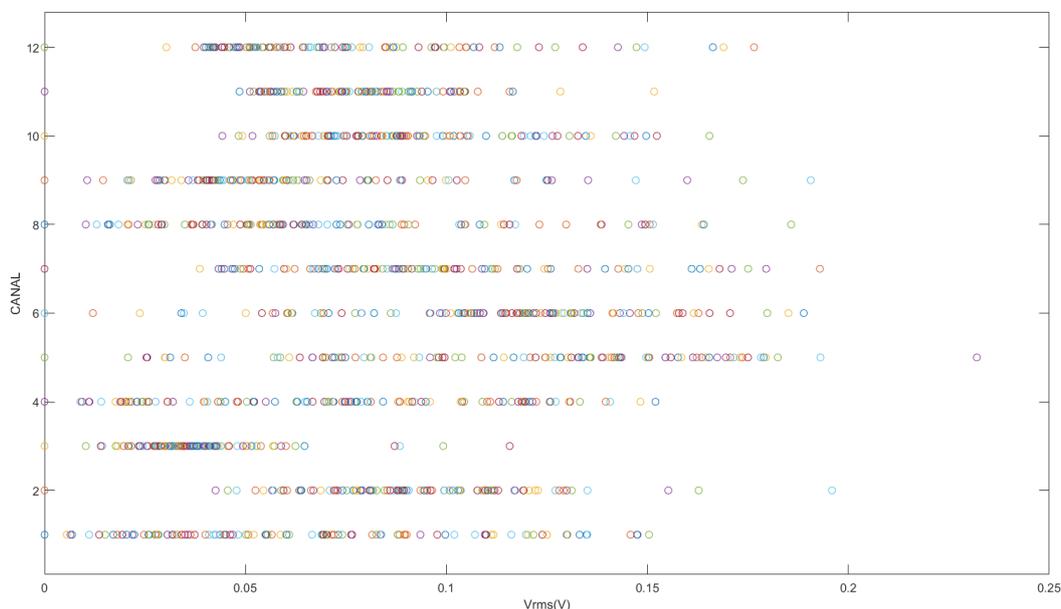


Figura 45 – Distribuição de amplitudes de todos os canais para voluntário 2 da sub-base III.



Utilizando a função do *MatlabR2016a* descrita pela Figura 26 determinou-se o grau de pertinência de cada elemento do vetor que contém as características do sinal para cada um dos três *clusters*. As opções configuradas pelo vetor *options*, determinado pela Tabela 6, são:

- Ordem matriz expoente = 2, pois quanto menor a ordem mais distantes estão os centros dos *clusters* e mais distintas são os graus de pertinência de cada ponto;
- Máximo número de iterações = 1000;
- Valor mínimo do critério de parada = 1e-15;
- *flag* = 0, portanto é apresentado o resultado das iterações individualmente;

Tem-se como resultado da utilização do comando *fcm()* uma matriz *U* de dimensão (número de *clusters* x número de segmentos) para cada um dos canais. Ao traçar a curva formada pelos graus de pertinência para cada *cluster*, como mostrado na Figura 46, é possível observar o comportamento Gaussiano e a clara distinção entre *clusters* para o canal 3 do voluntário 3 da sub-base II. Da mesma forma, na Figura 47, é possível observar o comportamento Gaussiano e a clara distinção entre *clusters* para o canal 5 do voluntário amputado 2 da sub-base III. Nota-se que a função em laranja possui altos valores de pertinência para valores de tensão elétrica RMS mais próximos de zero, ao passo que a função ao amarelo apresenta altos valores de pertinência para tensões elétricas RMS

intermediárias e a função em azul apresenta altos valores de pertinência para tensões elétricas mais próximas do valor máximo. Com isto é possível concluir que as curvas em laranja representam o *cluster* que pode ser associado a variável linguística "baixo", em amarelo o *cluster* que pode ser associado a variável linguística "médio" e em azul o *cluster* que pode ser associado a variável linguística "alto". Outros exemplos de distribuições de pertinência para todos os canais de sujeitos diferentes encontram-se no Anexo C.

Figura 46 – Curva de graus de pertinência por *cluster* para canal 3 do voluntário 3 da sub-base II.

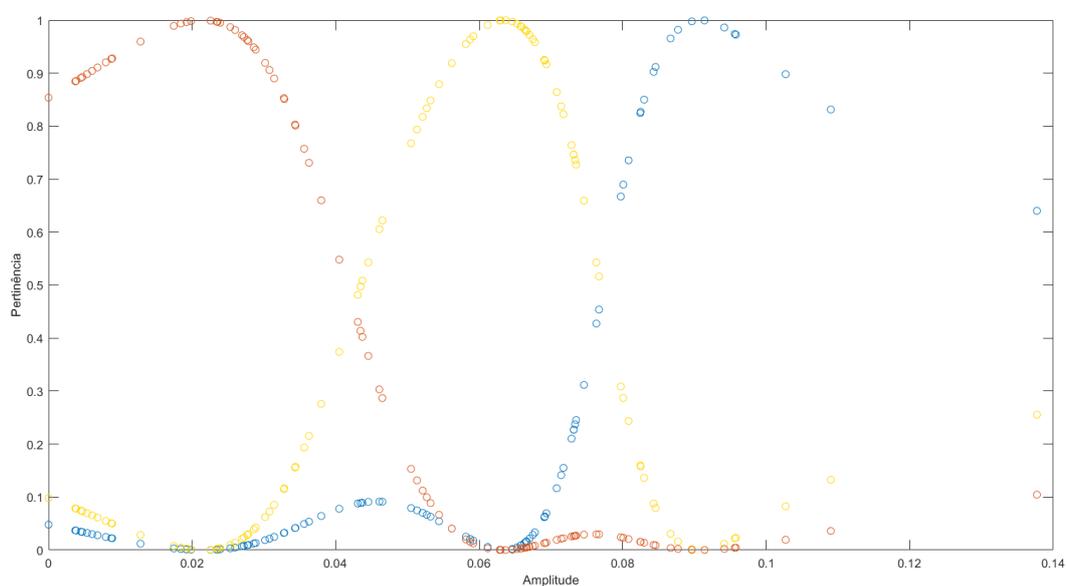
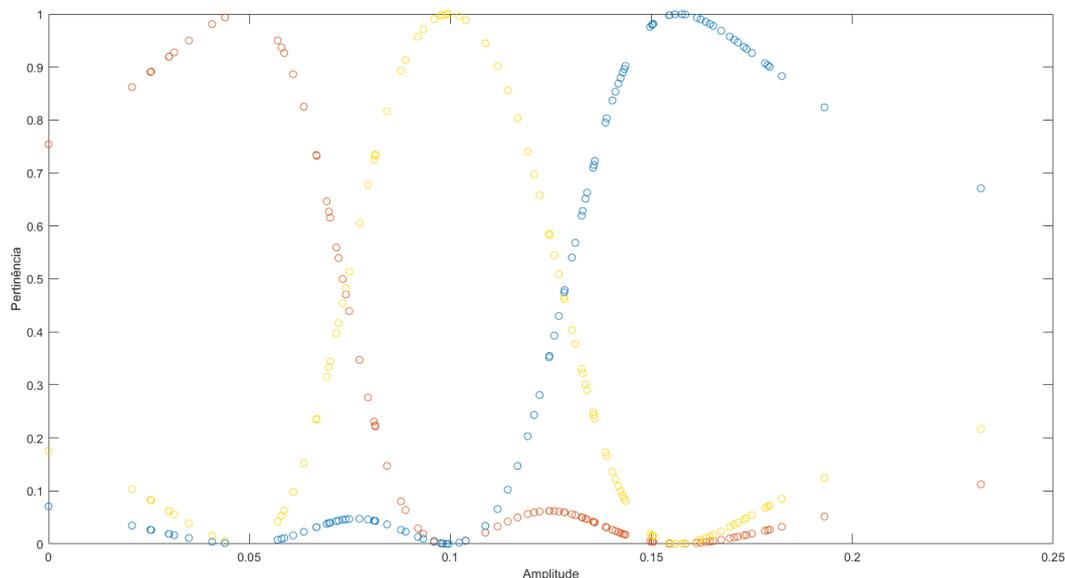


Figura 47 – Curva de graus de pertinência por *cluster* para canal 5 do voluntário 2 da sub-base III.



4.2.3 Resultados da Determinação da Função de Pertinência

Na etapa de *Clustering* notou-se a característica gaussiana dos níveis de pertinência das características dos movimentos. Portanto, afim de determinar as funções de pertinência para o sistema de inferência *Fuzzy*, o comando descrito pela Tabela 7 foi configurado para adaptar os elementos a uma função gaussiana. Para isto foi necessário atribuir 'gauss1' para a entrada *ft* do comando *fit()*. A Figura 48 mostra um exemplo de uma curva aproximada utilizando o comando *fit()* através do método dos mínimos quadrados para todos os canais do voluntário 3 da sub-base II. Da mesma forma, a Figura 49 mostra um exemplo de uma curva aproximada utilizando o comando *fit()* através do método dos mínimos quadrados para todos os canais do voluntário amputado 2 da sub-base III. Outros exemplos de distribuições de pertinência para todos os canais e sujeitos diferentes encontram-se no Anexo C.

As Figuras 48 e 49 apresentam em linhas contínuas as funções de pertinência determinadas a partir dos elementos do vetor gerado pelo procedimento de *Clustering*, indicados por pontos circulados. As funções de pertinência estão nomeadas com "baixo", "médio" e "alto" de forma crescente, mostrando o funcionamento do algoritmo de ordenamento apresentado pela Figura 28. É possível observar que na grande maioria dos casos as o comportamento das funções de pertinência se aproximou muito dos elementos agrupados. Mesmo os casos em que a coincidência dos pontos apresentou maior distanciamento, tem-se claramente a diferenciação de três *clusters* diferentes o que possibilitou a *fuzzyficação*

da entrada dos sinais.

Um problema agregado pela imperfeição do ajuste das curvas parte da premissa que a soma dos níveis de pertinência dos três agrupamentos para um determinado ponto sempre é igual a 1. Visto que as funções apresentaram resultados aproximados, tem-se *fuzzyficações* onde esta soma é menor do que 1. Em um caso extremo é possível um valor de entrada apresentar pertinência igual a 0 para os três *clusters*.

Figura 48 – Curva de graus de pertinência por *cluster* para o voluntário 3 da sub-base II.

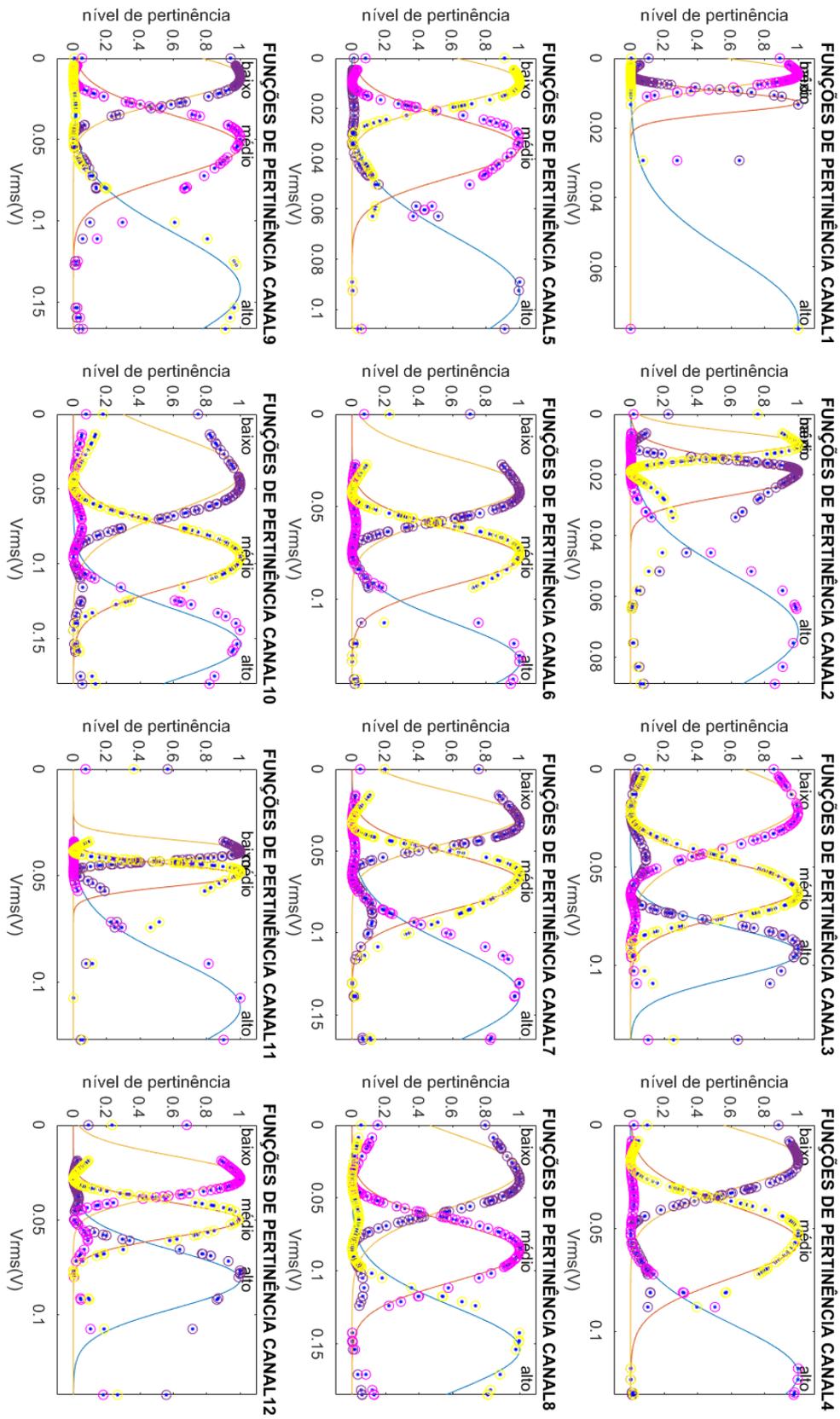
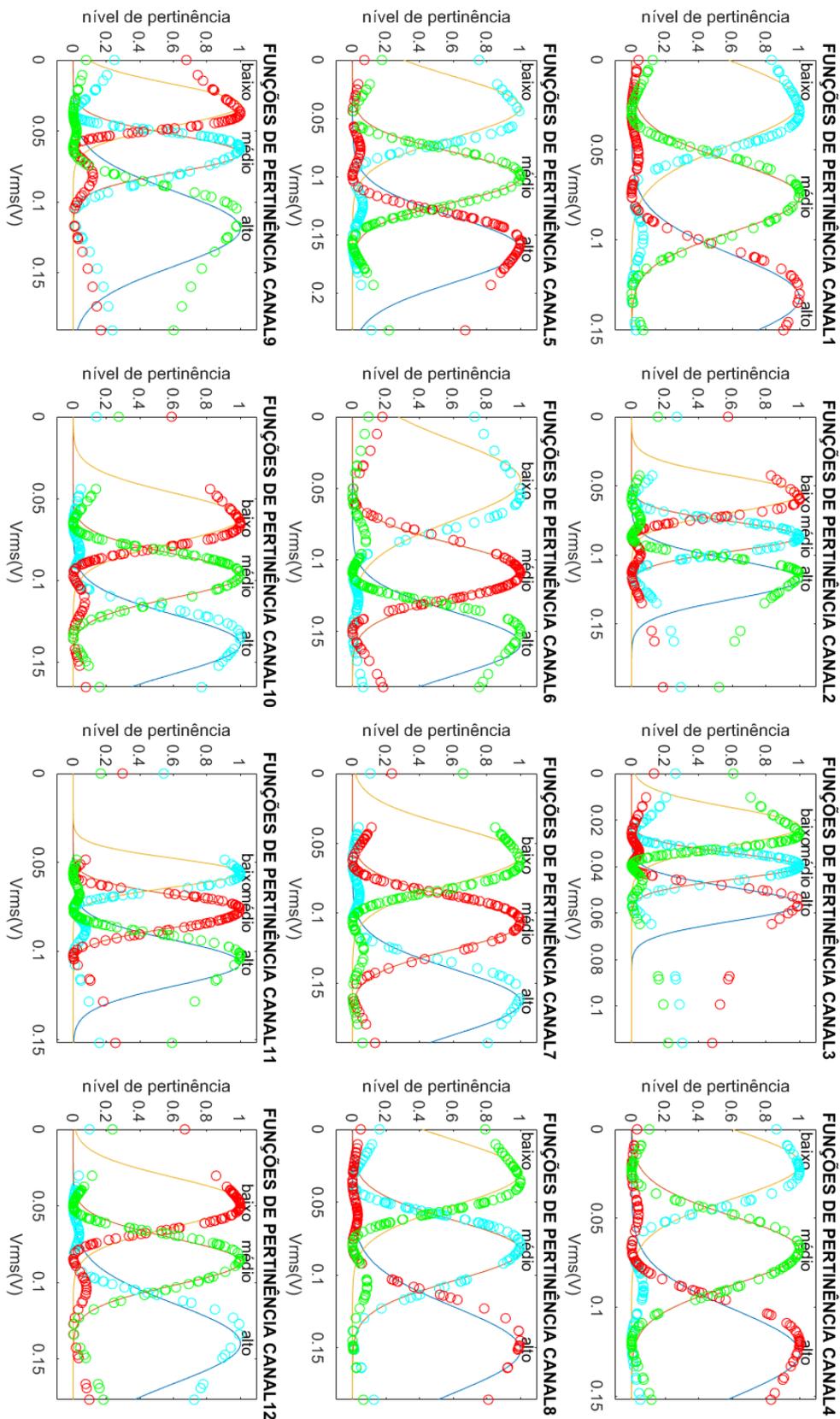


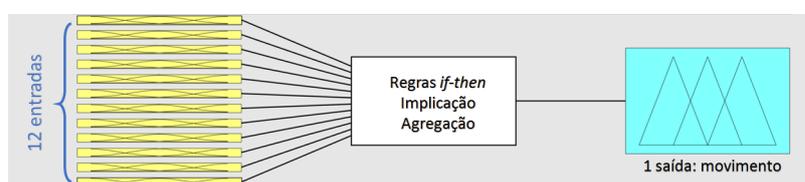
Figura 49 – Curva de graus de pertinência por *cluster* para o voluntário 2 da sub-base III.



4.2.4 Resultados da Composição das Entradas e Saídas do Sistema de Inferência *Fuzzy*

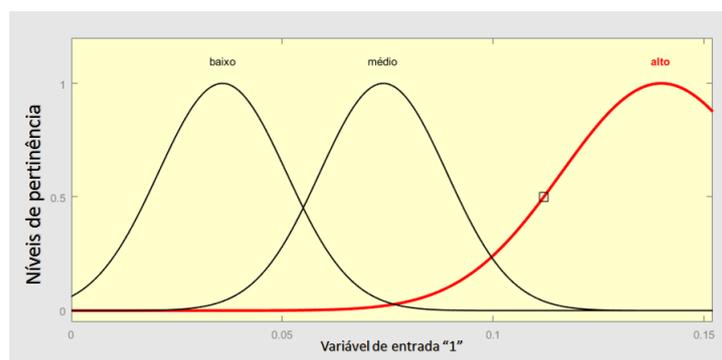
A visão geral do resultado obtido da composição das entradas e saídas do Sistema de Inferência *Fuzzy* pode ser observado na Figura 50. Portanto, o sistema final está configurado como possuindo 12 entradas distintas e 1 saída. O programa *MatlabR2016a* conta com a ferramenta "*FuzzyLogicDesign*" que permite visualizar e alterar Sistemas *Fuzzy*, todas as Figuras deste capítulo são retiradas desta ferramenta e mostram as configurações e características do Sistema de Inferência *Fuzzy* configurado pelo programa.

Figura 50 – Sistema de Inferência *Fuzzy* 12 entradas e 1 saída.



Visualizando o comportamento individual das variáveis de entrada através da ferramenta "*Fuzzy Logic Design*" é possível constatar que as funções de transferência adicionadas ao sistema são idênticas àquelas geradas pelo ajuste de curvas realizado anteriormente. A Figura 51 apresenta o formato das Funções de Pertinência para a entrada de número 3 para o voluntário 3 da sub-base II.

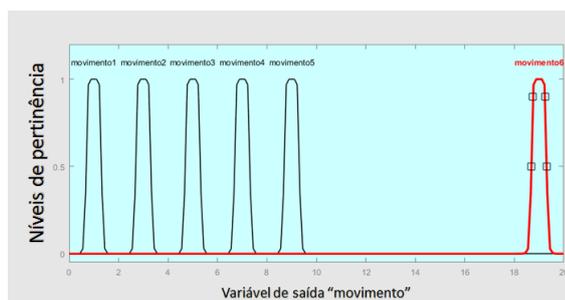
Figura 51 – Entrada 3 para voluntário 3 da sub-base II.



Visualizando o comportamento individual da variável de saída, que é idêntica para todos os sujeitos, através da ferramenta "*Fuzzy Logic Design*" é possível constatar que as funções de transferência adicionadas ao sistema são não-dependentes entre si, como mostrado na Figura 52. Nota-se também que apresentam comportamento no formato "Bell" muito aproximado de uma função retangular, o movimento 1 está centralizado na

saída de valor 1, o movimento 2 no valor 3, o movimento 3 no valor 5, o movimento 4 no valor 7, o movimento 5 no valor 9 e os demais movimentos no valor 19. Com base nos testes preliminares tem-se que o maior erro advém dos movimentos não identificados, por este motivo tem-se como centro mais distante dos demais.

Figura 52 – Saída para todos os voluntários e sub-bases.



4.2.5 Resultados da Composição das Regras do Sistema de Inferência *Fuzzy*

Foram geradas mais de 1300 regras para compor o sistema *Fuzzy* deste projeto, uma regra para cada uma das janelas de movimento, dando preferência à identificação das FP's com altos níveis de pertinência para cada canal. A declaração da matriz contendo outros exemplos de regras criadas neste trabalho encontra-se no Anexo D.

O *MatlabR2016a* permite a visualização gráfica das regras criadas para o Sistema de Inferência, identificando qual o nível de pertinência da FP contida na regra, baseado no valor de entrada. A Figura 53 apresenta uma destas análises para um sinal VRMS de [0.007979; 0.02197; 0.06886; 0.06542; 0.05387; 0.07273; 0.1224; 0.1324; 0.02306; 0.09024; 0.04816; 0.02482], a coluna da direita apresenta o movimento de saída, quando a área abaixo da função de pertinência está azul significa que aquela regra identificou a saída como pertinente. As regras de 1-12 e 24-25 resultam em movimentos não avaliados, as regras de 13-18 resultam no movimento 9 e as regras de 19-23 resultam no movimento 13. A regra resulta em uma pertinência elevada quando todas as colunas apresentam alta pertinência para as entradas especificadas, em outras palavras, se faz necessário que todas as 12 entradas, representadas pelas 12 colunas da esquerda, apresentem a área abaixo da função de pertinência preenchida em amarelo. É possível observar que para estes valores de entrada a regra 19 possui maior influência na saída, resultando no movimento 13. Em função do número elevado de regras é impossível mostrar de forma completa o resultado gráfico individual de cada regra para a entrada apresentada, entretanto observa-se que para regra 22, que foi gerada para o mesmo movimento e mesmo sujeito que a 19, também apresenta saída relevante, embora muito menor. Mais resultados da composição das Regras do Sistema de Inferência *Fuzzy* podem ser observados no Anexo E.

Figura 53 – Saída gerada para um conjunto de 26 regras para uma entrada de VRMS.



4.2.6 Taxa de acerto do Sistema de Inferência *Fuzzy*

A taxa de acerto foi determinada conforme indicado na Figura 31 para cada uma das combinações apresentadas pela Tabela 10, os resultados percentuais de acerto encontram-se na Tabela 13. Como esperado as taxas de acerto variaram consideravelmente entre os sistemas com diferentes combinações de parâmetros. Para os métodos de Implcação não houve diferença entre os resultados obtidos com o operador *min* ou com o *prod*. Para as combinações contendo opções diferentes para Agregação as taxas de acerto se mostraram na mesma ordem de grandeza para os métodos de *sum* e *probor*, mas apresentou resultados inferiores para o método *max*.

Os métodos de *Defuzzyficação* foram os que apresentaram maior diferença entre as saídas das combinações. Para *Defuzzyficação* utilizando o método de *lom*, o acerto foi 0%, com este resultado é possível assumir que o método não é indicado para a aplicação no sistema projetado. Isto se deve ao fato de que as regras são elaboradas de forma não relacionada, não apresentando picos intermediário, portanto o "menor ponto de máximo" é um ponto de 0. Os métodos que apresentaram melhor taxa de acerto foram *mom* e *som*, possuindo resultados idênticos e até 71% maior do que o segundo pior método. Já esperava-se que os métodos de *Defuzzyficação* que utilizam os valores máximos apresentassem as melhores taxas de acerto, pois as regras foram geradas de forma que a saída que possuir maior valor indica o movimento correto. Dada a grande quantidade de movimentos não avaliados, mas ainda assim submetidos ao algoritmo de decisão, tem-se um número proporcionalmente elevado de falsos positivos, que acaba resultado em erros na taxa de acerto se o resultado da saída leva em consideração todas as saídas que apresentam resultados diferentes de zero.

Tabela 13 – Taxa de acerto por voluntário por combinação do Sistema de Inferência *Fuzzy*.

				NÃO AMPUTADOS						AMPUTADOS					
IMPLICAÇÃO	AGREGAÇÃO	DEFUZZY		S2	S3	S11	S19	S25	S39	aS1	aS2	aS4	aS5	aS8	aS10
1	min	max	centroid	37%	8%	11%	30%	20%	14%	34%	16%	50%	41%	10%	9%
2	min	max	bisector	46%	22%	21%	26%	29%	22%	20%	28%	24%	48%	23%	16%
3	min	max	mom	50%	47%	47%	48%	38%	31%	49%	34%	26%	64%	26%	18%
4	min	max	lom	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
5	min	max	som	50%	47%	47%	48%	38%	32%	49%	34%	26%	64%	26%	18%
6	min	sum	centroid	68%	63%	58%	69%	60%	57%	71%	52%	60%	77%	55%	38%
7	min	sum	bisector	61%	55%	31%	50%	53%	38%	56%	48%	24%	64%	19%	18%
8	min	sum	mom	86%	79%	75%	83%	76%	78%	84%	72%	71%	93%	74%	51%
9	min	sum	lom	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
10	min	sum	som	86%	79%	75%	83%	76%	78%	84%	72%	71%	93%	74%	51%
11	min	probor	centroid	53%	40%	33%	46%	21%	43%	46%	41%	67%	67%	27%	32%
12	min	probor	bisector	49%	38%	15%	34%	26%	18%	41%	23%	14%	52%	15%	18%
13	min	probor	mom	86%	72%	72%	81%	75%	60%	81%	72%	74%	92%	51%	49%
14	min	probor	lom	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
15	min	probor	som	86%	72%	72%	81%	75%	60%	81%	72%	74%	92%	51%	49%
16	prod	max	centroid	44%	12%	11%	34%	20%	16%	35%	17%	50%	43%	13%	9%
17	prod	max	bisector	46%	23%	21%	27%	29%	23%	21%	29%	24%	49%	23%	17%
18	prod	max	mom	50%	47%	47%	48%	38%	31%	49%	34%	26%	64%	26%	18%
19	prod	max	lom	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
20	prod	max	som	50%	47%	47%	48%	38%	32%	49%	34%	26%	64%	26%	18%
21	prod	sum	centroid	65%	56%	51%	64%	60%	56%	65%	48%	60%	70%	48%	34%
22	prod	sum	bisector	59%	45%	27%	43%	39%	31%	50%	31%	24%	62%	21%	19%
23	prod	sum	mom	86%	79%	75%	83%	76%	78%	84%	72%	71%	93%	74%	51%
24	prod	sum	lom	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
25	prod	sum	som	86%	79%	75%	83%	76%	78%	84%	72%	71%	93%	74%	51%
26	prod	probor	centroid	49%	36%	33%	40%	21%	38%	37%	40%	57%	61%	26%	32%
27	prod	probor	bisector	51%	41%	18%	35%	27%	19%	41%	26%	21%	51%	17%	18%
28	prod	probor	mom	86%	72%	72%	81%	75%	60%	81%	72%	74%	92%	51%	49%
29	prod	probor	lom	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
30	prod	probor	som	86%	72%	72%	81%	75%	60%	81%	72%	74%	92%	51%	49%

Portanto, dada a qualidade da taxa de acerto de todas as combinações, uma das que apresentou melhor assertividade foi aquela gerada a partir do Sistema de Inferência *Fuzzy* com método de Implicação *min*, Agregação *sum* e *Defuzzificação mom*. Sendo assim, as demais análises realizadas para este trabalho levam em consideração a combinação de número 8, cuja taxa de acerto levando em conta todos os movimentos para todos os sujeitos para o Sistema de Inferência *Fuzzy* projetado é de 77%.

As Taxas de acerto para cada movimento avaliado encontram-se na Tabela 14, onde "outro" representa todos os outros 12 movimentos que não foram contemplados pelas regras. Como é possível observar, os movimentos avaliados tem alta taxa de acerto, o que compromete a qualidade do resultado são os movimentos classificados como "outros". Isto acontece, pois tem-se uma regra para cada movimento e, muitas vezes, dada a característica estocástica dos sinais, os movimentos não avaliados possuem valores de tensão elétrica RMS muito parecidos com os avaliados. Tentou-se minimizar estes erros colocando as regras que testam os movimentos especificados apresentando maior importância para o sistema e que estas regras sejam as mais singulares possíveis, ou seja, por isso as regras geradas pelo algoritmo da Figura 30 valorizam a FP com maior nível de pertinência. Dentre os movimentos, aquele que apresentou a melhor taxa de acerto foi o de número 16 e o pior foi o de número 13.

Tabela 14 – Taxa de Acerto por movimento.

Movimento	Taxa de acerto
6	93%
9	90%
13	84%
14	95%
16	97%
outro	72%

As Tabelas 15 e ?? apresentam as taxas de acerto por movimento por sujeito, estas informações estão representadas graficamente para os amputados na Figura 55 e para os não-amputados na Figura 54, onde é possível observar que o movimento com menor taxa de acerto é o “outros”. Os demais movimentos possuem taxas de acerto de até 100%, tanto para os voluntários com amputações, quanto para os com membro intacto. Estes resultados se devem a produção das regras ser individualizada por movimento e por sujeito, o *Clustering* dos dados também é importante, pois consegue adaptar os limites do sistema ao sujeito. Entretanto, um sistema de identificação de movimento deveria apresentar resultados independentes do sujeito, embora a etapa de *Clustering* vise diminuir esta influência.

Tabela 15 – Taxa de Acerto por sujeito por movimento.

Voluntário	Movimento	Taxa de acerto
aS1	6	100%
aS1	9	100%
aS1	13	100%
aS1	14	100%
aS1	16	100%
aS1	outro	80%
aS1 Total		84%
aS10	6	67%
aS10	9	40%
aS10	13	67%
aS10	14	100%
aS10	16	67%
aS10	outro	45%
aS10 Total		51%
aS2	6	100%
aS2	9	100%
aS2	13	100%
aS2	14	67%
aS2	16	100%
aS2	outro	65%
aS2 Total		72%
aS4	6	80%
aS4	9	100%
aS4	13	66%
aS4	14	80%
aS4	16	80%
aS4	outro	70%
aS4 Total		71%
aS5	6	100%
aS5	9	100%
aS5	13	100%
aS5	14	100%
aS5	16	100%
aS5	outro	92%
aS5 Total		93%
aS8	6	100%
aS8	9	67%
aS8	13	83%
aS8	14	100%
aS8	16	100%
aS8	outro	68%
aS8 Total		74%

Voluntário	Movimento	Taxa de acerto
S11	6	71%
S11	9	86%
S11	13	100%
S11	14	83%
S11	16	100%
S11	outro	71%
S11 Total		75%
S19	6	100%
S19	9	100%
S19	13	83%
S19	14	100%
S19	16	100%
S19	outro	79%
S19 Total		83%
S2	6	100%
S2	9	100%
S2	13	100%
S2	14	100%
S2	16	100%
S2	outro	83%
S2 Total		87%
S25	6	100%
S25	9	100%
S25	13	33%
S25	14	100%
S25	16	100%
S25	outro	73%
S25 Total		76%
S3	6	100%
S3	9	83%
S3	13	83%
S3	14	100%
S3	16	100%
S3	outro	75%
S3 Total		79%
S39	6	83%
S39	9	100%
S39	13	83%
S39	14	100%
S39	16	100%
S39	outro	73%
S39 Total		78%

Figura 54 – Taxa de Acerto por Movimento por Sujeito Intacto.

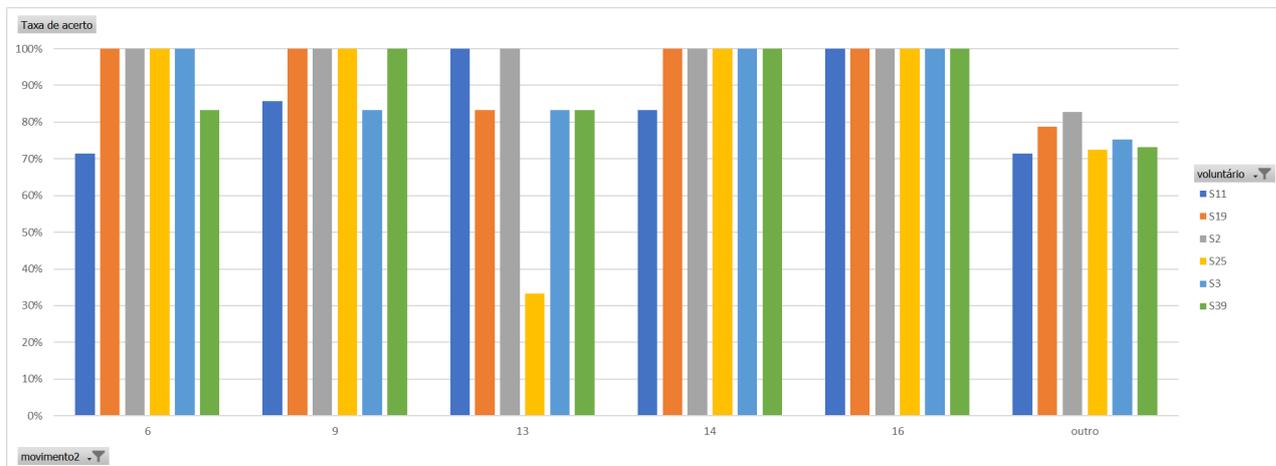
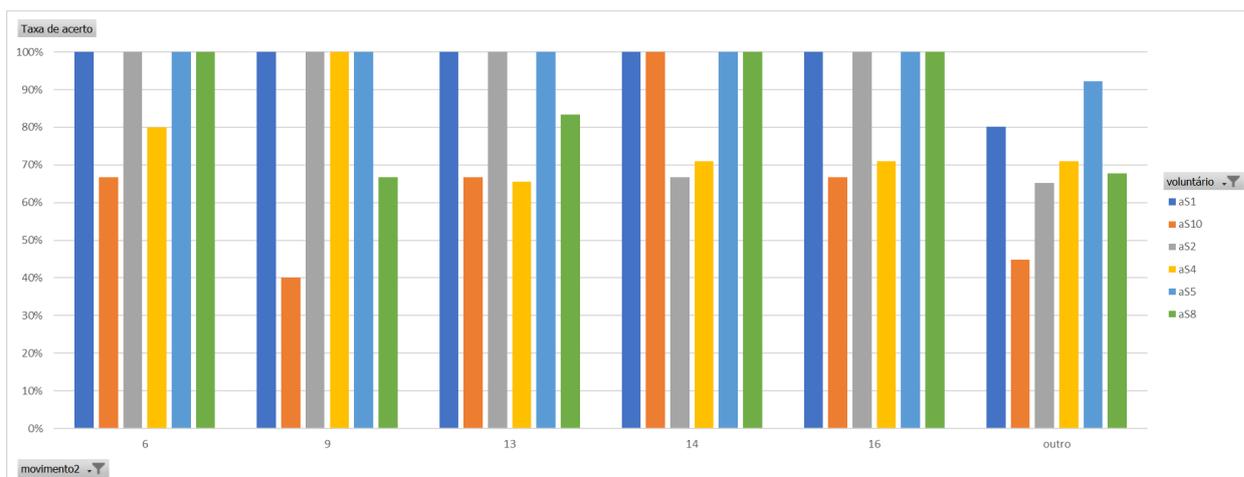


Figura 55 – Taxa de Acerto por Movimento por Sujeito Amputado.



5 CONCLUSÕES

O presente trabalho tinha como objetivo utilizar os dados da base de dados NINAPRO para criar um algoritmo de decisão utilizando lógica *Fuzzy* capaz de identificar movimentos do segmento mão-braço. Foram avaliados 5 movimentos diferentes para 12 voluntários, 6 destes amputados.

Com a segmentação do sinal por detecção de limiar por janela fixa foi possível seccionar o sinal nas áreas com e sem movimento. Desta forma, obteve-se os parâmetros de tensão elétrica RMS para cada um dos segmentos ativos, que se apresentaram bem distribuídos ao longo de uma faixa de valores - com valores máximos variando de 0,2453Vrms até 0,0144Vrms. Esta distribuição permitiu o agrupamento em clusters através da técnica de *Fuzzy Clustering*, os resultados obtidos foram padronizados, apresentando para todos os canais e sujeitos três comportamentos gaussianos com centros espaçados.

O Sistema de Inferência *Fuzzy* foi projetado contendo 12 entradas, uma para cada canal, cujas Funções de Pertinência foram calculadas através do ajuste de curvas feitas a partir do *Clustering*. Foi possível determinar para cada sujeito um Sistema *Fuzzy* próprio, cujos parâmetros são determinados a partir dos dados passados. A saída do sistema e as regras são comuns para todos os movimentos e sujeitos.

A lógica de criação de regras determinada pelo especialista, permitiu elaborar uma regra individual para cada segmento ativo de todos os sujeitos. A grande quantidade de regras se fez necessário para minimizar o ruído devido aos movimentos não avaliados, mas ainda assim considerados na taxa de acerto. Foi de suma importância adequar o fator multiplicador das regras de forma a aumentar a relevância das regras que indicavam algum dos movimentos avaliados.

A taxa de acerto se mostrou diferente para as combinações de características do Sistema *Fuzzy*, apresentando resultado inferior para o método de Agregação *max* e para o método de *Defuzzyficação lom*. Os níveis de acerto não variaram conforme o método de implicação, e apresentaram melhores resultado para os métodos de Agregação *sum* e *probor* e para os métodos de *Defuzzyficação mom* e *som*. O sistema de configuração número 8 (Tabela 10) apresentou uma taxa de acerto geral de 77% e em função disto essa configuração foi utilizada no restante do trabalho.

Ao avaliar individualmente os movimentos constatou-se que a melhor taxa de acerto correspondeu ao movimento de número 16 (Desvio ulnar do punho), apontando 97% de acerto. Os movimentos avaliados apresentaram taxa de acerto até 12% maior do que a identificação do movimento *outros*, que representa todos os outros movimentos não analisados neste trabalho. Como na base NINAPRO, Exercício 1 da sub-bases II e III,

apresentou um total de 17 movimentos, tem-se 12 movimentos não avaliados contra 5 avaliados. Desta forma, se a taxa de acerto total considera todos os movimentos e apresenta pior desempenho para os 12 não avaliados, estes puxam a média total para baixo. Por esta razão a taxa média encontra-se em 77%, ao passo que as taxas individuais dos movimentos assumem valores de 84% até 97%.

Analisando os dados dos movimentos por indivíduo tem-se o mesmo comportamento do geral, apenas dois sujeitos manifestam taxa de acerto inferior para um dos movimentos analisados em relação ao movimento *outros*. Para diversos movimentos apresentou-se taxa de acerto igual a 100%, por exemplo, para o sujeito aS8 o movimento 14 foi identificado como tal em todas as repetições.

Por fim, constatou-se que é possível elaborar um sistema de Inferência *Fuzzy* que identifique movimentos a partir de dados sEMG no tempo. O sistema criado apresentou resultados acima de 84% para os movimentos analisados e uma taxa de acerto total de 77%.

6 Proposta de Trabalhos Futuros

Como proposta de trabalhos futuros se sugere:

- aplicação em um sistema online do algoritmo proposto;
- aperfeiçoar a base de dados com a criação de uma base com repetições e análise de maior número de sujeitos e voluntários;
- realizar um projeto de experimentos para analisar os resultados e sua validação do ponto de vista estatístico;
- aperfeiçoar o Sistema de Inferência *Fuzzy* de forma que a taxa de acerto não dependa do sujeito;
- elaborar *hardware* e realizar as medições dos voluntários.

Referências

ATZORI, M. *et al.* **Building the Ninapro database: a resource for the biorobotics community.** In: INTERNATIONAL CONFERENCE ON BIOMEDICAL-ROBOTICS AND BIOMECHATRONICS (BioRob), 4., 2012, Rome. Proceedings...[S.l.]: IEEE, 2012. p.1258-1265.

ATZORI, M. *etal.* **Electromyography data for non-invasive naturally-controlled robotic hand prostheses.** NatureScientific Data. [S.l.], n. 1, v. 1, p. 1-13, 2014. Disponível em: <<http://www.nature.com/articles/sdata201453>>. Acesso em: 13 fev. 2015. DOI: 10.1038/sdata.2014.53.

BASMAJIAN, J.V.; DE LUCA, C.J. **Muscle alive:** their functions revealed by electromyography. Baltimore: Wlliams & Wilkins, 1985.

BEZDEK, J. **Patter Recognition with Fuzzy Objective Function Algorithms.** New York: Plenum Press, 1981.

BRONZINO, J.; ENDERLE, J. **Introduction to Biomedical Engineering.** 3rd. ed. Burlington: Elsevier, 2012. 1272 p.

BRONZINO, J. (edical Devices and Systems. 3rd. Ed. Boca Roton: CRC Press, 2006.

BÚRIGO, A. **Classificação de Movimentos da Mão Utilizando Eletromiografia de Superfície, Regressão Logística, Redes Neurais, Máquina de Vetores de Suporte e Base de Dados NinaPro.** 2014. Monografia (Bacharel em Engenharia Elétrica) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2014.

CENE, V. **Avaliação da Influência da Variação de Características e Canais Para a Caracterização de Sinais Mielétricos Através do uso de Regressão Logística.** 2015. 132p. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal do Rio Grande do Sul, Porto Algre, 2015.

CHEN, G. PHAM, T. **Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems.** Boca Raton: CRC Press LLC, 2001.

CRAM, J.R. The History of Surface Electromyography. **Applied Psychophysiology**

and Biofeedback. New York, v. 28, n. 2, jun. 2003.

CRAM, J.R. **Cram's Introduction to Surface Electromyography**. 2da. ed. Sudbury: Jones and Bartlett Publishers, 2011.

DAS, S. Pattern Recognition using the Fuzzy c-means Technique. **International Journal of Energy, Information and Communications**, v.4, n.1, fev. 2013.

DE LUCA, C.J. Physiology and Mathematics of Myoelectric Signals. **IEEE Transactions on Biomedical Engineering**, Gainesville, v.26, n.6, p. 313-325, June 1979.

DE LUCA, C.J. The Use of Surface Electromyography in Biomechanics. **Journal Of Applied Biomechanics**, v.13, p. 135-163, 1997.

DUBOIS, D. PRADE, H. **Fuzzy set and systems: theory and applications**. New York: Academic Press, 1980.

FAVIEIRO G. W. **Desenvolvimento de um sistema neuro-fuzzy para análise de sinais mioelétricos do segmento mão-braço**. 2012. 142p. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2012.

GALVÃO FILHO, T. A., GARCIA, J. C. D. **Pesquisa Nacional de Tecnologia Assistiva**. São Paulo: Instituto de Tecnologia Social - ITS BRASIL e Ministério da Ciência, Tecnologia e Inovação - MCTI/SECIS, 2012.

GUYTON, A. C. **Fisiologia Humana**. 11. ed. Rio de Janeiro: Guanabara Koogan, 2006.

HATA, Y. *et al.* A Fuzzy Biosignal Detection Algorithm and its Application to Health Monitoring. **Applied Computing Math**, v.10, n.1, p. 133-145, 2011.

IBGE. **Censo Demográfico 2010: Características Gerais da População, Religião e Pessoas com Deficiência**. Censo demogr. Rio de Janeiro, p.1-215, 2010. Disponível em:

JANG, J. R.; SUN, C.; MIZUTANI, E. **Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence**. New York: Prentice Hall, 1997.

KANDEL, E. *et al.* **Principles of Neural Science**. 5. ed. New York: McGraw-Hill, 2013.

LOPES, I. **Caracterização dos sinais mioelétricos dos movimentos do segmento mão braço através de regressão logística**. 2014. Monografia (Bacharel em Engenharia de Computação) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2014.

SUGENO, M.; KANG, G. T. Structure identification of fuzzy model. **Fuzzy Sets and Systems**, North Holland, v.28, n.3, p. 15-33, May 1988.

TAKAGI, T.; SUGENO, M. Fuzzy identification of systems and its applications to modeling and control. **IEEE Transactions on Systems, Man, and Cybernetics**, New York, v.15, n.1, p. 116-132, Jan. 1985.

TORTORA, GERARD J. **Corpo humano: fundamentos de anatomia e fisiologia**. Porto Alegre: Artmed, 2000.

ZADEH, L. A. Fuzzy sets. **Information and control**, North Holland, v.8, n.3, p. 338-353, 1965.

Anexos


```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% SINAL PRÉ-PROCESSADO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
sinal_pre_processado(:,retCANAL) = retsinal_norm;
rms_sinal_pre_processado(retCANAL) = rms(sinal_pre_processado(:,retCANAL));
end

```

A.3 Função de Segmentação

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% janelamento
% *cria apenas uma janela de sinal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
jan_Fs = 2000;           % Sampling frequency
jan_T = 1/jan_Fs;       % Sampling period
jan_L = length(emg);    % Length of signal
jan_t = (0:jan_L-1)*jan_T; % Time vector
jan_f = jan_Fs*(0:(jan_L/2))/jan_L; % Frequency vector

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% JANELAMENTO
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%caracteristicas
jan_t_janela = 0.5;     %500mili segundos o tamanho da janela
jan_amostras_janela = round(jan_t_janela / jan_T);
jan_amostras_porcentagem = round(jan_amostras_janela*0.83);
jan_quantidade_maxima_janelas = round((length(emg)/jan_amostras_janela) - 1);
rms_sinal_pre_processado = rms_sinal_pre_processado .*0.8;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% "COMPARANDO VALORES DENTRO DAS JANELAS"
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                jan_cont_amostras = 0;
                jan_cont_janelas = 2;
                jan_aux = 0;

                for jan_x = 1:1:(jan_quantidade_maxima_janelas-1)
                    %AQUI ESTOU DENTRO DA JANELA

                for jan_y = 0:1:(jan_amostras_janela-1)

```

```

        jan__z = jan__x*jan__amostras_janela + jan__y; %INDICE DA JANELA

        jan__canal_aux = 0;
        for jan__CANAL1=1:1:12
            if sinal_pre_processado(jan__z,jan__CANAL1) >...
... rms_sinal_pre_processado(jan__CANAL1);
                jan__canal_aux = 1;
            end
        end
        if jan__canal_aux == 1
            jan__cont_amostras = jan__cont_amostras + 1;
        end
    end
    %verifico se minha janela contém sinal
    if jan__cont_amostras > jan__amostras_porcentagem

        if jan__aux == 0
            jan__aux = 1;
            janela(jan__cont_janelas,1)= jan__x;
            janela(jan__cont_janelas,2)= (jan__x+1);
        else
            if janela(jan__cont_janelas,2) == jan__x
                janela(jan__cont_janelas,2) = jan__x+1;
            else
                jan__cont_janelas = jan__cont_janelas+1;
                janela(jan__cont_janelas,1) = jan__x;
                janela(jan__cont_janelas,2) = (jan__x+1);
            end
        end
    end

    jan__possui_sinal = 0.1;

else
    jan__possui_sinal = 0;
end
jan__cont_amostras = 0;
%construo a vetor seguimentação
for jan__y = 0:1:(jan__amostras_janela-1)
    jan__z = jan__x*jan__amostras_janela + jan__y;%INDICE DA JANELA
    seguimentacao(jan__z) = jan__possui_sinal;
end

end

% ultima janela/que é incompleta
for jan__x = ((jan__quantidade_maxima_janelas-1)*jan__amostras_janela):1:length(emg)
    seguimentacao(jan__x) = 0;

```



```

% CÁLCULO DAS PROPRIEDADES
% - calcula os valores rms e frequencia media de cada janela do sinal
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
prop_Fs = 2000;           % Sampling frequency
prop_T = 1/prop_Fs;      % Sampling period
prop_L = length(emg);    % Length of signal
prop_t = (0:prop_L-1)*prop_T; % Time vector
prop_f = prop_Fs*(0:(prop_L/2))/prop_L; % Frequency vector
prop_t_janela = 0.5;     %500mili segundos o tamanho da janela
prop_amostras_janela = round(prop_t_janela / prop_T);
prop_amostras_porcentagem = round(prop_amostras_janela*0.8);
prop_quantidade_maxima_janelas = round((length(emg)/prop_amostras_janela) - 1);
rms_sinal_pre_processado = rms_sinal_pre_processado;

for prop_CANAL = 1:1:12

for prop_x = 1:1:length(janela)
    prop_aux_vetor = 0;
    aux_vetor_stimulus = 0;
    prop_vetor_comeco = janela(prop_x,1)*prop_amostras_janela;
    prop_vetor_fim = (janela(prop_x,2)*prop_amostras_janela)-1;

    for prop_y = (prop_vetor_comeco):1:(prop_vetor_fim)
        prop_z = prop_y + 1 - (prop_vetor_comeco);
        prop_aux_vetor(prop_z,1) = sinal_seguintado(prop_y,prop_CANAL);
        aux_vetor_stimulus(prop_z,1) = stimulus(prop_y);
    end
    v_rms(prop_x,prop_CANAL) = rms(prop_aux_vetor);
    movimento_stimulus(prop_x) = max(aux_vetor_stimulus);
end

end
end

```

A.5 Função de *Clustering* e Obtenção das Funções de Pertinência

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ESTA FUNÇÃO CALCULA AS FUNÇÕES DE PERTINÊNCIA
% - a partir dos valores típicos de cada um dos canais para cada um dos
% voluntário, cria o FCM e retorna o vetor de pertencencia
% - crio a função gaussiana baseada na FCM, retorna os coeficientes
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
fcm_options = [2 1000 1e-15 0]; %configura FCM

```

```
for fcm__CANAL = 1:1:12 %gera o vetor para todos os canais

[FCMcenters(:,fcm__CANAL),FCMU(:, :,fcm__CANAL)] = fcm(v_rms(:,fcm__CANAL),3,fcm__options);

end

for CANAL = 1:1:12

FP__options = [2 1000 1e-15 0];
[FCMcenters(:,CANAL),FCMU(:, :,CANAL),FCMobjFun] = fcm(v_rms(:,CANAL),3,FP__options);

[FP__xData, FP__yData] = prepareCurveData( v_rms(:,CANAL), FCMU(1,:,CANAL) );
FP__ft = fittype( 'gauss1' );
FP__opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
FP__opts.Display = 'Off';
FP__opts.Lower = [1 -Inf -Inf];
FP__opts.Upper = [1 Inf Inf];
[FP__fitresult, FP__gof] = fit( FP__xData, FP__yData, FP__ft, FP__opts );
gaussiana_a1 (CANAL,1) = FP__fitresult.a1;
gaussiana_b1 (CANAL,1) = FP__fitresult.b1;
gaussiana_c1 (CANAL,1) = FP__fitresult.c1;

[FP__xData, FP__yData] = prepareCurveData( v_rms(:,CANAL), FCMU(2,:,CANAL) );
FP__ft = fittype( 'gauss1' );
FP__opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
FP__opts.Display = 'Off';
FP__opts.Lower = [1 -Inf -Inf];
FP__opts.Upper = [1 Inf Inf];
[FP__fitresult, FP__gof] = fit( FP__xData, FP__yData, FP__ft, FP__opts );
gaussiana_a1 (CANAL,2) = FP__fitresult.a1;
gaussiana_b1 (CANAL,2) = FP__fitresult.b1;
gaussiana_c1 (CANAL,2) = FP__fitresult.c1;

[FP__xData, FP__yData] = prepareCurveData( v_rms(:,CANAL), FCMU(3,:,CANAL) );
FP__ft = fittype( 'gauss1' );
FP__opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
FP__opts.Display = 'Off';
FP__opts.Lower = [1 -Inf -Inf];
FP__opts.Upper = [1 Inf Inf];
[FP__fitresult, FP__gof] = fit( FP__xData, FP__yData, FP__ft, FP__opts );
gaussiana_a1 (CANAL,3) = FP__fitresult.a1;
gaussiana_b1 (CANAL,3) = FP__fitresult.b1;
gaussiana_c1 (CANAL,3) = FP__fitresult.c1;
```

```
end
```

A.6 Criação do Sistema de Inferência *Fuzzy*

```
clear -regexp ^sys__
clear -regexp ^mFIS__

sys__AndMethod='min';
sys__OrMethod='max';
sys__ImpMethod='prod';
sys__AggMethod='max';
sys__DefuzzMethod='centroid';

% cria o fis
my_first_fis = newfis('my_first_fis_inside','mamdani',sys__AndMethod,sys__OrMethod,...
...sys__ImpMethod,sys__AggMethod,sys__DefuzzMethod);
montagemFIS      %adiciona os inputs

% adiciona os outputs
my_first_fis = addvar(my_first_fis,'output','movimento',[0 20]);
my_first_fis = addmf(my_first_fis,'output',1,'movimento1','gbellmf',[0.3125 5 1]);
my_first_fis = addmf(my_first_fis,'output',1,'movimento2','gbellmf',[0.313 5 3]);
my_first_fis = addmf(my_first_fis,'output',1,'movimento3','gbellmf',[0.313 5 5]);
my_first_fis = addmf(my_first_fis,'output',1,'movimento4','gbellmf',[0.313 5 7]);
my_first_fis = addmf(my_first_fis,'output',1,'movimento5','gbellmf',[0.313 5 9]);
my_first_fis = addmf(my_first_fis,'output',1,'movimento6','gbellmf',[0.313 5 19]);

my_first_fis.rule = []; %limpa todas as regras
regras              %carrega as regras atuais
% escreve as regras no FIS
for sys__RULE = 1:1:length(sys__rule)
    my_first_fis = addrule(my_first_fis,sys__rule(sys__RULE,:));
end

avalia_imp          %chama função que calcula o movimento final
```

A.7 Adição das Entradas

```
%////////////////////////////////////
%////////////////////////////////////
% Criação das entradas FIS
% *carrego os valores gerados pelo ajuste de curvas
% *ordeno os clusters
%////////////////////////////////////
%////////////////////////////////////
```

```

for CANAL = 1:1:12
    mFIS__B = max(v_rms(:,CANAL));
    my_first_fis = addvar(my_first_fis,'input',num2str(CANAL),[0 mFIS__B]);

    %ORDENAMENTO DOS CLUSTERS
    if gaussiana_b1(CANAL,1)> gaussiana_b1(CANAL,2) &&...
...gaussiana_b1(CANAL,1)> gaussiana_b1(CANAL,3)
        mFIS__MAIOR = 1;
        if gaussiana_b1(CANAL,2)> gaussiana_b1(CANAL,3)
            mFIS__INTERMEDIARIO = 2;
            mFIS__MENOR = 3;
        else
            mFIS__INTERMEDIARIO = 3;
            mFIS__MENOR = 2;
        end
    else
        if gaussiana_b1(CANAL,2)> gaussiana_b1(CANAL,3)
            mFIS__MAIOR = 2;
            if gaussiana_b1(CANAL,1)> gaussiana_b1(CANAL,3)
                mFIS__INTERMEDIARIO = 1;
                mFIS__MENOR = 3;
            else
                mFIS__INTERMEDIARIO = 3;
                mFIS__MENOR = 1;
            end
        else
            mFIS__MAIOR = 3;
            if gaussiana_b1(CANAL,1)> gaussiana_b1(CANAL,2)
                mFIS__INTERMEDIARIO = 1;
                mFIS__MENOR = 2;
            else
                mFIS__INTERMEDIARIO = 2;
                mFIS__MENOR = 1;
            end
        end
    end

    end

    %adiciono em ordem as FP's
    my_first_fis = addmf(my_first_fis,'input',CANAL,'alto','gaussmf',...
        ...[(sqrt((gaussiana_c1(CANAL,mFIS__MAIOR)^2)/2))...
        ...gaussiana_b1(CANAL,mFIS__MAIOR)]);
    my_first_fis = addmf(my_first_fis,'input',CANAL,'médio','gaussmf',...
        ...[(sqrt((gaussiana_c1(CANAL,mFIS__INTERMEDIARIO)^2)/2))...
        ...gaussiana_b1(CANAL,mFIS__INTERMEDIARIO)]);

```

```

my_first_fis = addmf(my_first_fis,'input',CANAL,'baixo','gaussmf',...
    ...[(sqrt((gaussiana_c1(CANAL,mFIS__MENOR)^2)/2))...
    ....gaussiana_b1(CANAL,mFIS__MENOR)]);
subplot(3,4,CANAL)
legend('off')
plotmf(my_first_fis,'input',CANAL)
hold on;

end

```

A.8 Função de avaliação do método de Implicação

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% define as características do sistema FIS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
my_first_fis = setfis(my_first_fis,'ImpMethod', 'min');
getfis(my_first_fis, 'ImpMethod')
% writefis(my_first_fis,'myfirstfis')
sys__z_out = 0;
avalia_agr

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% define as características do sistema FIS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
my_first_fis = setfis(my_first_fis,'ImpMethod', 'prod');
getfis(my_first_fis, 'ImpMethod')
% writefis(my_first_fis,'myfirstfis')
sys__z_out = 15;
avalia_agr

```

A.9 Função de avaliação do método de Agregação

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% define as características do sistema FIS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
my_first_fis = setfis(my_first_fis,'AggMethod', 'max');
getfis(my_first_fis, 'AggMethod')
% writefis(my_first_fis,'myfirstfis')
sys__j_out = 0;
avalia_defuzz

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% define as características do sistema FIS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
my_first_fis = setfis(my_first_fis,'AggMethod', 'sum');
getfis(my_first_fis, 'AggMethod')

```

```

% writefis(my_first_fis,'myfirstfis')
sys__j_out = 5;
avalia_defuzz

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% define as características do sistema FIS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
my_first_fis = setfis(my_first_fis,'AggMethod', 'probor');
getfis(my_first_fis, 'AggMethod')
% writefis(my_first_fis,'myfirstfis')
sys__j_out = 10;
avalia_defuzz

```

A.10 Função de avaliação do método de *Defuzzyficação*

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% define as características do sistema FIS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
my_first_fis = setfis(my_first_fis,'DefuzzMethod', 'centroid');
getfis(my_first_fis, 'DefuzzMethod')
% writefis(my_first_fis,'myfirstfis')
sys__i_out = 1+ sys__j_out + sys__z_out;
generate_out

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% define as características do sistema FIS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
my_first_fis = setfis(my_first_fis,'DefuzzMethod', 'bisector');
getfis(my_first_fis, 'DefuzzMethod')
% writefis(my_first_fis,'myfirstfis')
sys__i_out = 2+ sys__j_out + sys__z_out;
generate_out

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% define as características do sistema FIS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
my_first_fis = setfis(my_first_fis,'DefuzzMethod', 'mom');
getfis(my_first_fis, 'DefuzzMethod')
% writefis(my_first_fis,'myfirstfis')
sys__i_out = 3 + sys__j_out + sys__z_out;
generate_out

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% define as características do sistema FIS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
my_first_fis = setfis(my_first_fis,'DefuzzMethod', 'lom');

```

```

getfis(my_first_fis, 'DefuzzMethod')
% writefis(my_first_fis,'myfirstfis')
sys__i_out = 4+ sys__j_out + sys__z_out;
generate_out

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% define as características do sistema FIS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
my_first_fis = setfis(my_first_fis,'DefuzzMethod', 'som');
getfis(my_first_fis, 'DefuzzMethod')
% writefis(my_first_fis,'myfirstfis')
sys__i_out = 5+ sys__j_out + sys__z_out;
generate_out

```

A.11 Função de avaliação do Sistema *Fuzzy*

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% define as características do sistema FIS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
aaa = setfis(aaa,'DefuzzMethod', 'centroid');
getfis(aaa, 'DefuzzMethod')
i_out = 1;
generate_out

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% define as características do sistema FIS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
aaa = setfis(aaa,'DefuzzMethod', 'bisector');
getfis(aaa, 'DefuzzMethod')

i_out = 2;
generate_out

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% define as características do sistema FIS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
aaa = setfis(aaa,'DefuzzMethod', 'mom');
getfis(aaa, 'DefuzzMethod')

i_out = 3;
generate_out

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% define as características do sistema FIS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
aaa = setfis(aaa,'DefuzzMethod', 'lom');

```

```
getfis(aaa, 'DefuzzMethod')

i_out = 4;
generate_out
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% define as características do sistema FIS
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
aaa = setfis(aaa,'DefuzzMethod', 'som');
getfis(aaa, 'DefuzzMethod')

FUZZY_START

i_out = 5;
generate_out
```

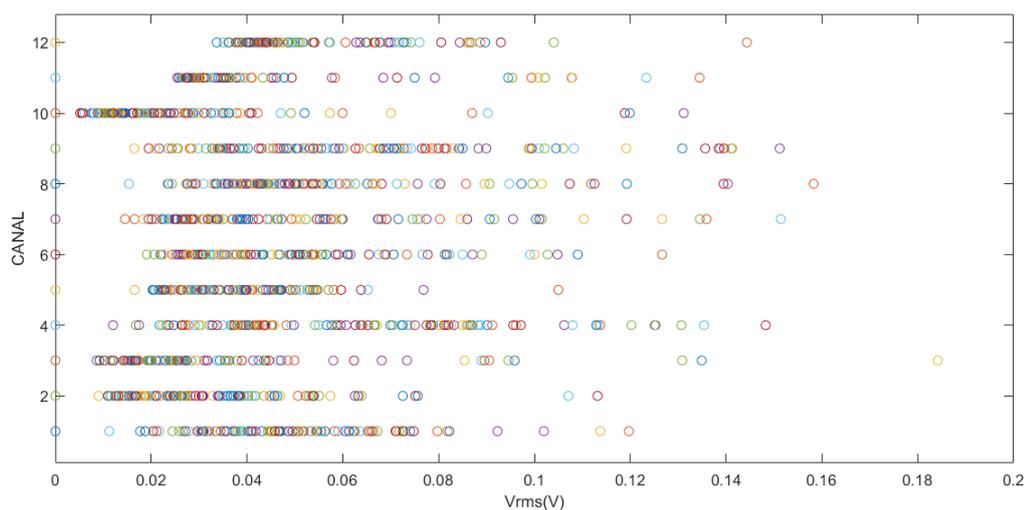
A.12 Função de geração da saída

```
for sys__m=1:1:length(v_rms)
    for CANAL = 1:1:12
        sys__v_rms_teste(CANAL) = v_rms(sys__m,CANAL);
    end
    out(sys__m,sys__i_out) = evalfis(sys__v_rms_teste(:),my_first_fis);
end
```

ANEXO B – Distribuição de Amplitudes VRMS

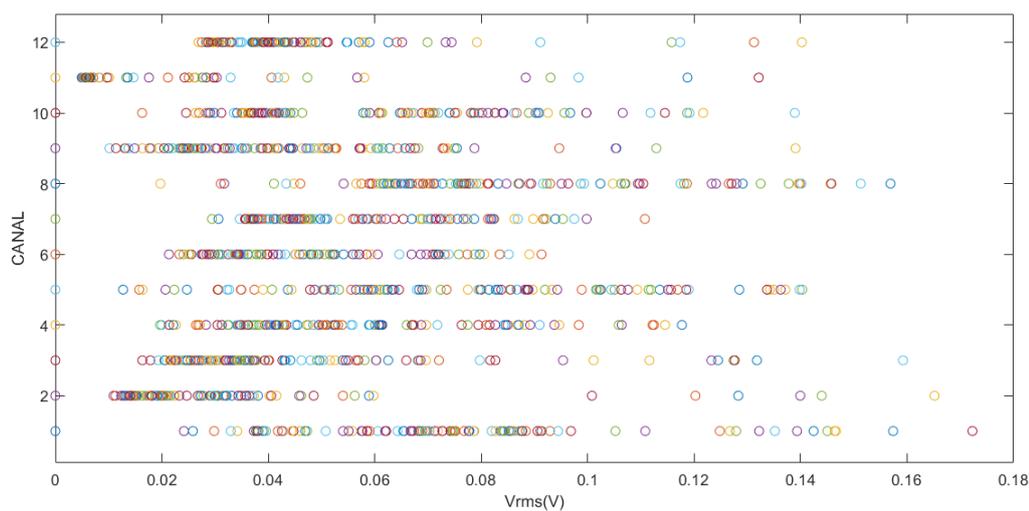
B.1 Distribuição de Amplitudes VRMS voluntário 11 da sub-base II

Figura 56 – Distribuição de amplitudes de todos os canais para voluntário 11 da sub-base II.



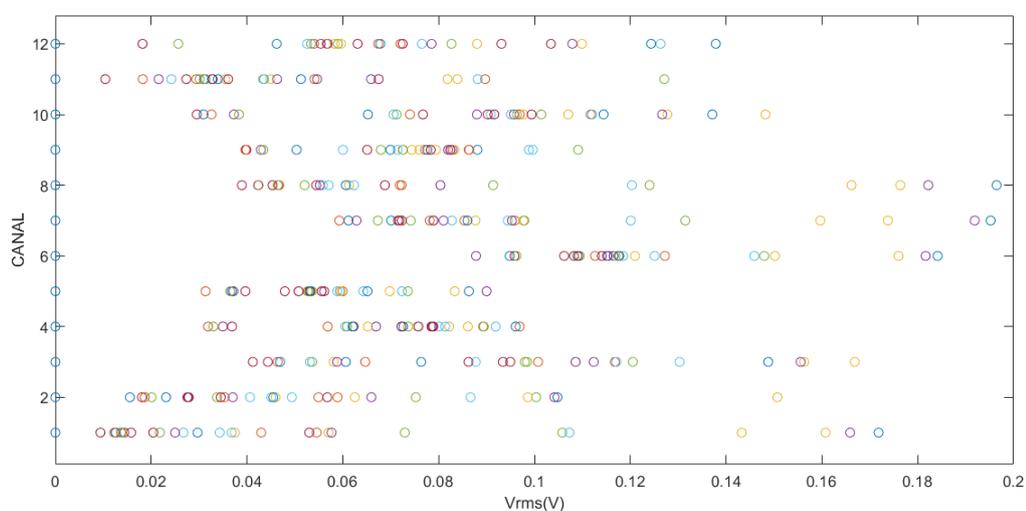
B.2 Distribuição de Amplitudes VRMS voluntário 19 da sub-base II

Figura 57 – Distribuição de amplitudes de todos os canais para voluntário 19 da sub-base II.



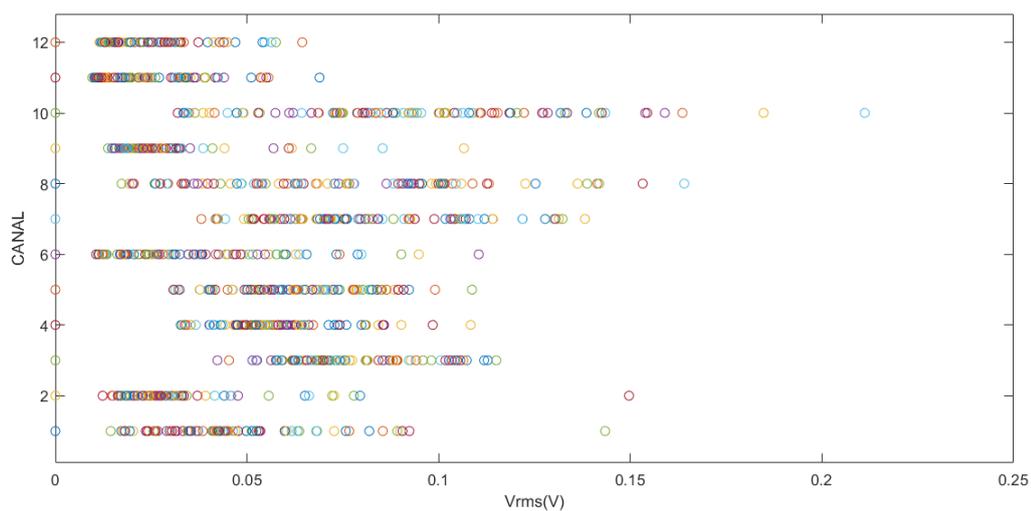
B.3 Distribuição de Amplitudes VRMS voluntário 4 da sub-base III

Figura 58 – Distribuição de amplitudes de todos os canais para voluntário 4 da sub-base III.



B.4 Distribuição de Amplitudes VRMS voluntário 10 da sub-base III

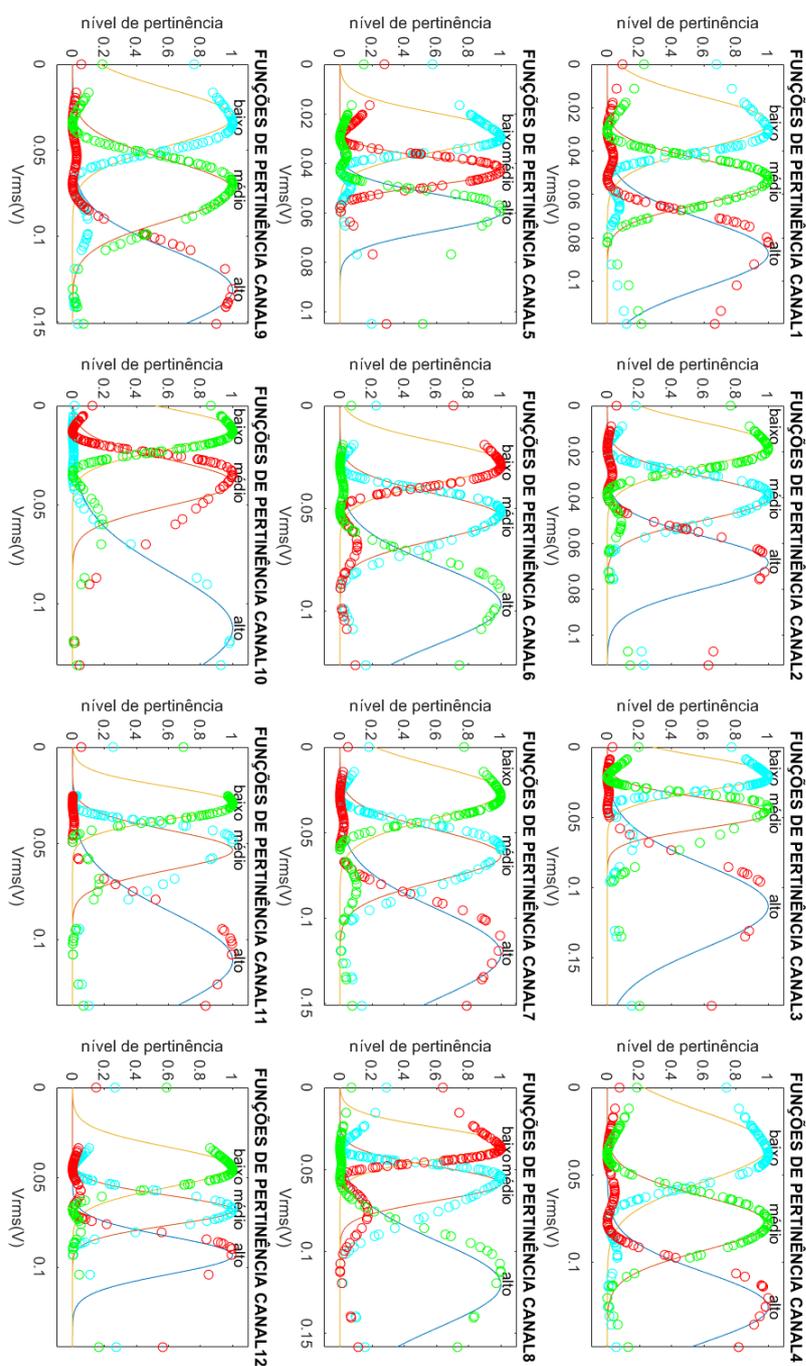
Figura 59 – Distribuição de amplitudes de todos os canais para voluntário 10 da sub-base III.



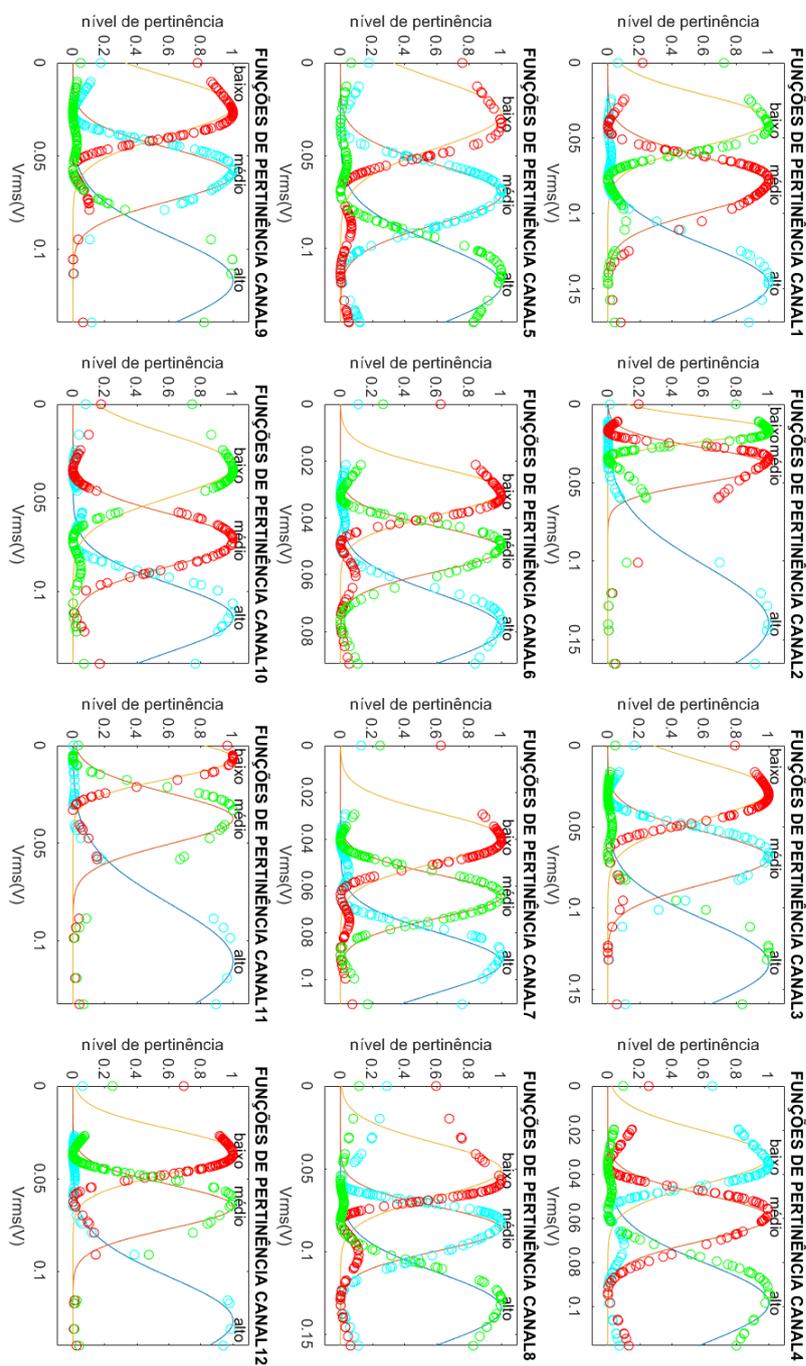
ANEXO C – Curva de graus de pertinência

C.1 Curva de graus de pertinência para vol. 11 da sub-base II

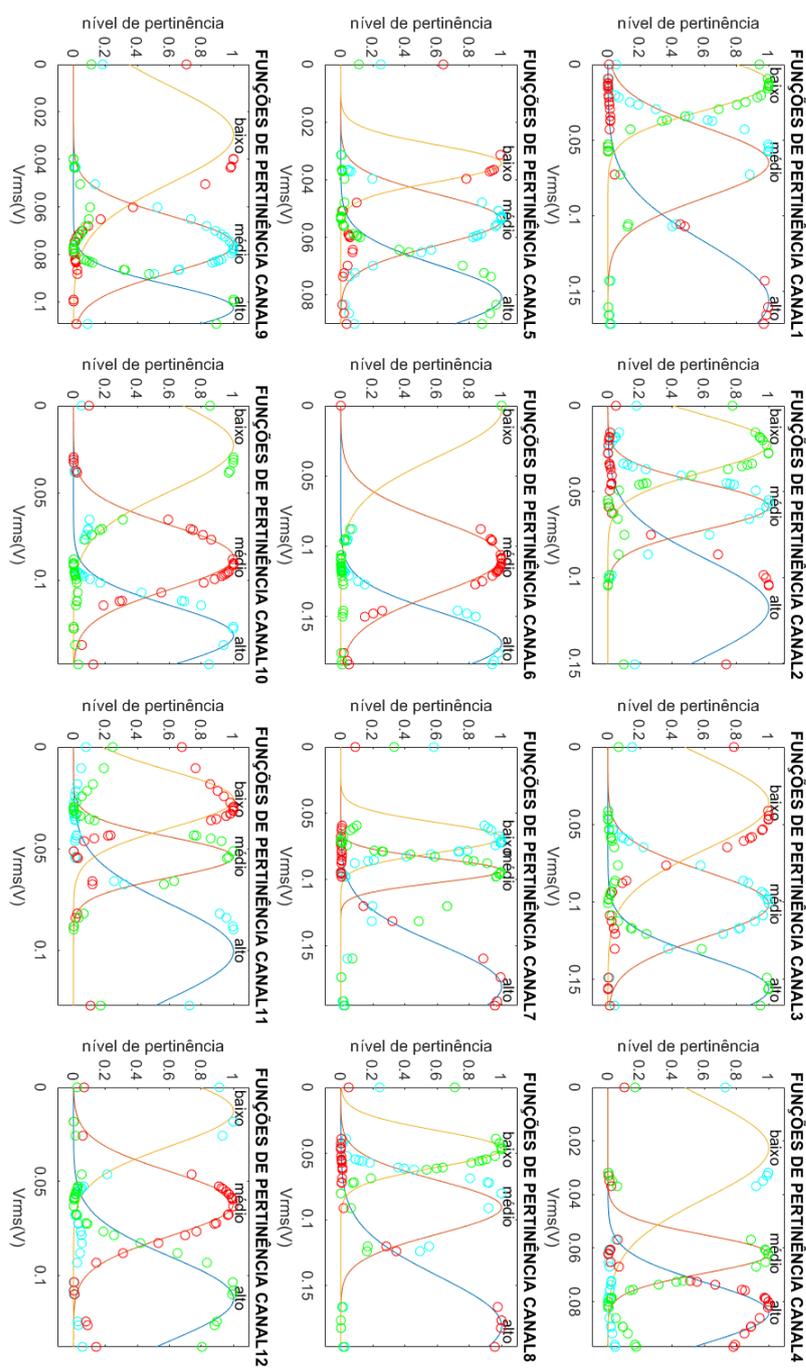
Figura 60 – Curva de graus de pertinência por *cluster* para o vol. 11 da sub-base II.



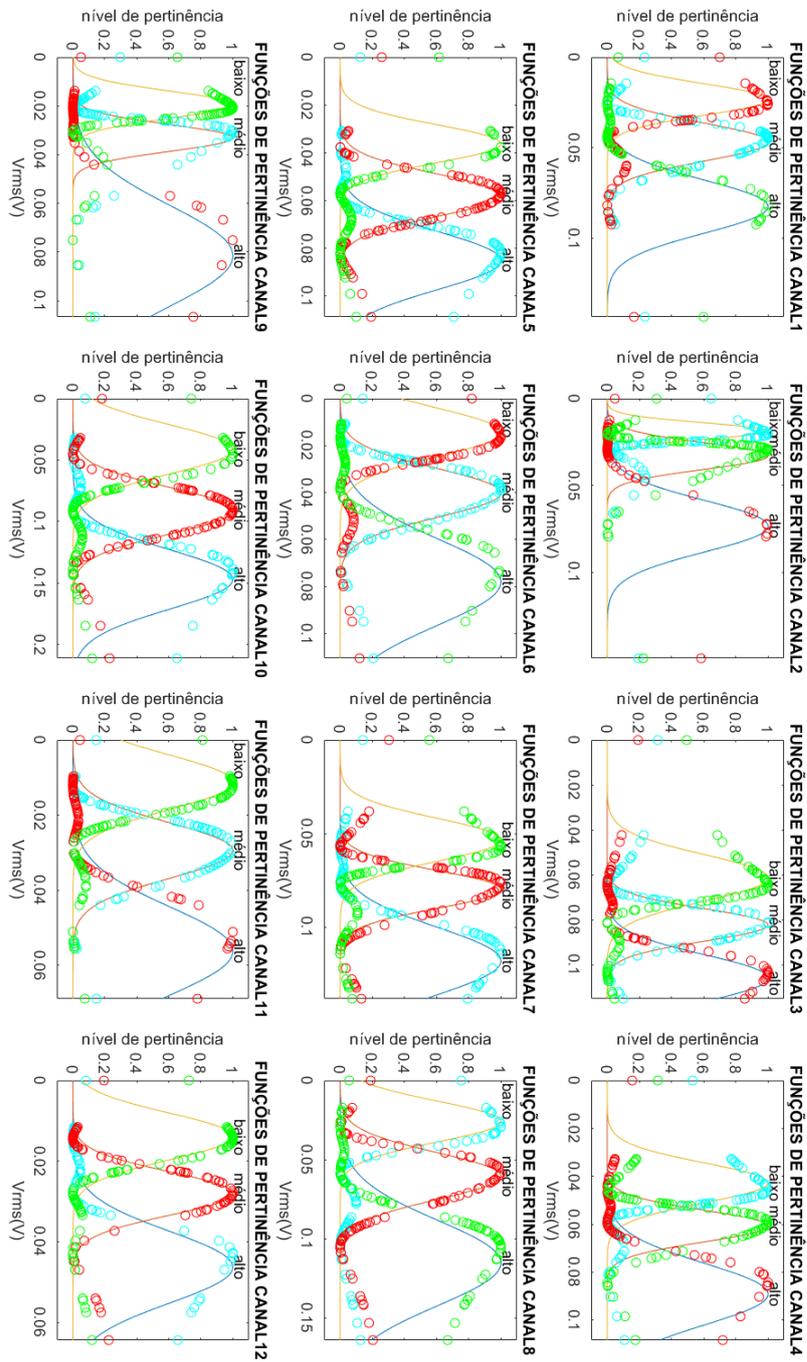
C.2 Curva de graus de pertinência para vol. 19 da sub-base III

Figura 61 – Curva de graus de pertinência por *cluster* para o vol. 19 da sub-base II.

C.3 Curva de graus de pertinência para vol. 4 da sub-base III

Figura 62 – Curva de graus de pertinência por *cluster* para o vol. 4 da sub-base III.

C.4 Curva de graus de pertinência para vol. 10 da sub-base III

Figura 63 – Curva de graus de pertinência por *cluster* para o vol. 10 da sub-base III.

ANEXO D – Exemplos de criação de Regras

Fuzzy

```

sys__rule(98,:) = [-3 1 2 2 -1 1 1 1 3 2 3 3,6 (0.1) 1];
sys__rule(99,:) = [-3 3 3 -3 2 2 3 2 -1 3 3 3,5 (1) 1];
sys__rule(100,:) = [-3 3 3 -3 2 2 3 2 2 3 3 3,5 (1) 1];
sys__rule(101,:) = [-3 3 3 -3 2 2 -1 2 -1 3 3 3,5 (1) 1];
sys__rule(102,:) = [-3 -1 3 -3 2 2 -1 2 -1 3 3 3,5 (1) 1];
sys__rule(103,:) = [1 3 3 1 -3 -3 -1 2 2 3 3 2,5 (1) 1];
sys__rule(104,:) = [2 -1 3 1 -3 -3 -1 -1 2 3 3 2,5 (1) 1];
sys__rule(105,:) = [2 2 -1 2 3 3 3 1 -1 3 3 -1,6 (0.1) 1];
sys__rule(106,:) = [1 2 -1 2 3 3 3 1 -1 3 3 2,6 (0.1) 1];
sys__rule(107,:) = [2 2 3 2 3 3 3 -3 3 3 3 2,6 (0.1) 1];
sys__rule(108,:) = [1 2 3 2 -1 3 3 1 3 3 3 2,6 (0.1) 1];
sys__rule(109,:) = [1 2 3 2 3 3 3 1 3 3 3 2,6 (0.1) 1];
sys__rule(110,:) = [1 2 3 2 3 3 3 1 3 3 3 2,6 (0.1) 1];
sys__rule(111,:) = [1 2 3 2 3 3 3 1 3 3 3 2,6 (0.1) 1];
sys__rule(112,:) = [1 2 3 2 3 3 3 1 3 3 3 2,6 (0.1) 1];
sys__rule(113,:) = [1 2 3 2 3 3 3 1 3 3 3 2,6 (0.1) 1];
sys__rule(114,:) = [1 2 3 2 3 3 3 1 3 3 3 2,6 (0.1) 1];
sys__rule(115,:) = [1 2 3 2 3 3 3 1 3 3 3 2,6 (0.1) 1];
sys__rule(116,:) = [1 2 3 2 3 3 3 1 3 3 3 2,6 (0.1) 1];
sys__rule(117,:) = [1 2 3 2 3 3 3 1 3 3 3 2,6 (0.1) 1];
sys__rule(118,:) = [1 2 3 2 3 3 3 1 3 3 3 2,6 (0.1) 1];
sys__rule(119,:) = [1 2 3 2 3 3 3 1 3 3 3 2,6 (0.1) 1];
sys__rule(120,:) = [1 2 3 2 3 3 3 1 3 3 3 2,6 (0.1) 1];
sys__rule(121,:) = [1 2 3 2 3 3 3 1 3 3 3 2,6 (0.1) 1];
sys__rule(122,:) = [1 2 3 2 3 3 3 1 3 3 3 2,6 (0.1) 1];
sys__rule(123,:) = [1 2 3 2 3 3 3 1 3 3 3 2,6 (0.1) 1];
sys__rule(124,:) = [1 2 3 2 3 3 3 1 3 3 3 2,6 (0.1) 1];
sys__rule(125,:) = [1 2 3 2 3 3 3 1 3 3 3 2,6 (0.1) 1];
sys__rule(126,:) = [1 2 3 2 3 3 3 1 3 3 3 2,6 (0.1) 1];
sys__rule(127,:) = [1 2 3 2 3 3 3 1 3 3 3 2,6 (0.1) 1];
sys__rule(128,:) = [1 2 3 2 3 3 3 1 3 3 3 2,6 (0.1) 1];
sys__rule(129,:) = [1 2 3 2 3 3 3 1 3 3 3 2,6 (0.1) 1];
sys__rule(130,:) = [1 2 3 2 3 3 3 1 3 3 3 2,6 (0.1) 1];

```


ANEXO E – Exemplos de utilização de Regras Fuzzy

Figura 64 – Saída gerada para um conjunto de 26 regras para uma entrada de VRMS - exemplo 2.

