

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

MARIANO MAJOLO

**MÁQUINA DE VETORES DE SUPORTE EMBARCADA EM
HARDWARE PARA CLASSIFICAÇÃO DE SINAIS
MIOELÉTRICOS**

Porto Alegre

2019

MARIANO MAJOLO

**MÁQUINA DE VETORES DE SUPORTE EMBARCADA EM
HARDWARE PARA CLASSIFICAÇÃO DE SINAIS
MIOELÉTRICOS**

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, da Universidade Federal do Rio Grande do Sul, como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica.

Área de concentração: Controle e Automação – Instrumentação.

ORIENTADOR: Prof. Dr. Alexandre Balbinot

Porto Alegre

2019

MARIANO MAJOLO

**MÁQUINA DE VETORES DE SUPORTE EMBARCADA EM
HARDWARE PARA CLASSIFICAÇÃO DE SINAIS
MIOELÉTRICOS**

Esta dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia Elétrica e aprovada em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: _____

Prof. Dr. Alexandre Balbinot, UFRGS

Doutor pela UFRGS – Porto Alegre, Brasil

Banca Examinadora:

Prof. Dr. Francisco José Fraga da Silva, PPGINF-UFABC

Doutor pelo Instituto Tecnológico de Aeronáutica – São José dos Campos, Brasil

Prof. Dr. Raphael Martins Brum, DELET-UFRGS

Doutor pela Université Montpellier 2 – Montpellier, França

Prof. Dr. Altamiro Amadeu Susin, PPGEE-UFRGS

Doutor pelo Institut National Polytechnique de Grenoble – Grenoble, França

Coordenador do PPGEE: _____

Prof. Dr. João Manoel Gomes da Silva Jr.

Porto Alegre, julho de 2019.

DEDICATÓRIA

Dedico este trabalho a todas as pessoas que acreditam na inclusão e buscam fazer a diferença para um mundo melhor.

AGRADECIMENTOS

Ao Programa de Pós-Graduação em Engenharia Elétrica e seu corpo docente, pelo ensino de qualidade e pela oportunidade de aprender.

Ao professor Dr. Alexandre Balbinot, que acreditou no projeto e orientou esta dissertação.

Aos colegas do Laboratório de Instrumentação Eletro-Eletrônica, IEE, pelo auxílio nas tarefas desenvolvidas durante o curso e apoio no desenvolvimento deste trabalho.

À família, esposa e amigos pelo incentivo ao estudo e compreensão nos momentos de ausência.

À BP&M Representações, em especial a João Dullius, que forneceu suporte e ferramentas essenciais para o desenvolvimento do projeto.

RESUMO

Sinais mioelétricos são foco de muitos estudos que visam melhorar a capacidade motora de pessoas amputadas ao proporem próteses eletromecânicas cada vez mais inteligentes. Técnicas de aprendizado de máquina são aplicadas a estes sinais e sistemas que preveem movimentos com taxa de acerto acima de 90% para alguns sujeitos são encontrados na literatura. Para atingir estas altas taxas de acerto, porém, o algoritmo de aprendizado deve treinar por minutos ou horas, tempo que acompanha toda e qualquer calibração e que prejudica a popularização de sistemas realimentados nesta área. Com o objetivo de acelerar o processo de treinamento, este trabalho propõe embarcar o sistema em uma FPGA pois estudos mostram que tem capacidade de reduzir o tempo necessário para aprendizado, e sistemas em *hardware* são pouco explorados pela literatura especializada. O sistema proposto consiste no desenvolvimento de módulos em VHDL para o filtro digital, extração de características, treinamento de Máquinas de Vetores de Suporte, validação e classificação embarcados na FPGA. Ensaios com a base de dados Ninapro indicam taxas de acerto acima de 70% no aprendizado de 10 movimentos executados por alguns sujeitos não amputados após o curto período de tempo de 10 s de treinamento, e acima de 50% quando os movimentos são executados por alguns sujeitos amputados. Estas taxas são inferiores às obtidas em estudos feitos em computadores, onde os recursos de processamento e memória são abundantes, porém, possui as vantagens de portabilidade, possibilitar calibração rápida com treinamento em menos de 10 s, e estar preparado para sistemas realimentados. A velocidade de aprendizado do sistema é aproximadamente 1,76 épocas/ms.

Palavras-chave: sEMG, SVM, Inteligência Computacional Embarcada, FPGA.

ABSTRACT

Myoelectric signals are the focus of many studies that aim to improve the motor capacity of amputees by proposing increasingly intelligent electromechanical prostheses. Machine learning techniques are applied to these signals and systems that predict movements with accuracy above 90% for some subjects are proposed. In order to achieve these high classification rates, however, the learning algorithm must train for minutes or hours, time that follows any and all calibration and makes feedback systems unfeasible. In order to accelerate the training process, this work proposes to embed the system in a FPGA since studies show that it has the capacity to reduce the time required for learning, and hardware systems are little explored in the literature. The proposed system consists of digital filtering, feature extraction, support vector machine training, validation and classification embedded in the FPGA. Experiments with the Ninapro database indicate accuracy rates above 70% in the classification of 10 movements performed by some non-amputated subjects after 10 s of training, and over 50% when performed by some amputees. These rates are lower than those obtained in computer studies, where processing and memory resources abound, but have the advantages of portability, enabling quick calibration with training in less than 10 s, and being prepared to feedback systems. The system learning speed is approximately 1.76 epochs/ms.

Keywords: sEMG, SVM, Embedded Computational Intelligence, FPGA.

LISTA DE FIGURAS

Figura 1 - Potencial de ativação gerado a partir de um estímulo.	17
Figura 2 - Composição de um MUAP: somatório de potenciais de ação individuais de fibras musculares adquirido por um eletrodo.	17
Figura 3 - Representação visual da distância entre hiperplanos.	21
Figura 4 - Diagrama de blocos do sistema.	30
Figura 5 - Resposta em frequência do Filtro FIR implementado.	31
Figura 6 - Diagrama do filtro FIR implementado.	31
Figura 7 - Diagramas de blocos das Extrações das Características: (a) Média Móvel, (b) RMS, (c) Potência e (d) Comprimento de Onda.	34
Figura 8 - Exemplo da característica RMS do sinal de sEMG do canal 1 representando os 17 movimentos da base avaliada.	35
Figura 9 - Bloco em VHDL responsável pelo treinamento SVM.	37
Figura 10 - Operação em pipeline de SVM iteration respeitando escalonamento de SVM COUNT.	38
Figura 11 - Representação da operação multiplicação escalar para arquitetura em pipeline. ...	39
Figura 12 - Representação gráfica da geração de endereços randômicos para escolha das amostras.	41
Figura 13 - Bloco em VHDL responsável pela validação dos parâmetros treinados pela SVM.	43
Figura 14 - Bloco em VHDL responsável pela classificação do sinal de sEMG.	45
Figura 15 - Lógica de registro dos contadores para eleição do movimento previsto.	46
Figura 16 - Movimentos de dedos e pulsos da base Ninapro.	48
Figura 17 - Simulação do filtro digital com entrada composta pelas senoides de 1 Hz e 9 Hz.	53
Figura 18 - Simulação do filtro digital com entrada composta pelas senoides de 300 Hz e 400 Hz.	54
Figura 19 - Resposta em frequência do filtro FIR: (a) senoides de 1 Hz e 9 Hz na entrada do filtro, (b) saída do filtro para esta entrada, (c) senoides de 300 Hz e 400 Hz na entrada do filtro, (d) saída do filtro para esta entrada.	55
Figura 20 - Resposta do filtro digital com os sinais de sEMG do primeiro sujeito não amputado da base de dados Ninapro executando o movimento 13.	55
Figura 21 - Resposta em frequência do filtro FIR com os sinais de sEMG do primeiro sujeito não amputado da base de dados Ninapro executando o movimento 13.	56
Figura 22 - Simulação da extração de características de onda quadrada.	57
Figura 23 - Simulação da extração de características de dente de serra.	57
Figura 24 - Extração de características utilizando as funções do Matlab (Versão 2016a).	58
Figura 25 - Respostas da Extração de Características com os sinais de sEMG do primeiro sujeito não amputado da base de dados Ninapro executando o movimento 13. ...	59
Figura 26 - Evolução do número de acertos das 20 primeiras SVMs para o primeiro sujeito não-amputado da base de dados Ninapro.	64
Figura 27 - Evolução do EQM das 136 SVMs para o primeiro sujeito não-amputado da base de dados Ninapro.	64
Figura 28 - Classificação dos movimentos utilizados na simulação na ordem da apresentação das amostras para o primeiro sujeito não-amputado da base de dados Ninapro. ...	67
Figura 29 - Matriz de confusão resultante da simulação da etapa de classificação para o primeiro sujeito não-amputado da base de dados Ninapro.	67

Figura 30 - Evolução do EQM da classificação de 17 movimentos para o primeiro sujeito não-amputado da base de dados Ninapro.	68
Figura 31 - Histograma dos resíduos do modelo de experimento.	70
Figura 32 - Distribuição dos resíduos no tempo conforme a ordem de observação.....	71
Figura 33 - Gráficos dos resíduos versus a média dos níveis das variáveis controladas do experimento (sujeito, tempo e movimento) e média dos blocos (nº de movimentos).	71
Figura 34 - Sobreposição das matrizes de confusão geradas para sujeitos não amputados após 75 ms de treinamento.	72
Figura 35 - Sobreposição das matrizes de confusão geradas para sujeitos não amputados após 10 s de treinamento.....	73
Figura 36 - Sobreposição das matrizes de confusão geradas para sujeitos amputados após 75 ms de treinamento.....	74
Figura 37 - Sobreposição das matrizes de confusão geradas para sujeitos amputados após 10 s de treinamento.	75
Figura 38 - Valores preditivos positivos para cada movimento executado por sujeito não amputado após 75 ms e 10 s de treinamento.	76
Figura 39 - Sensitivity para cada movimento executado por sujeito não amputado após 75 ms e 10 s de treinamento.	76
Figura 40 - Valores preditivos positivos para cada movimento executado por sujeito amputado após 75 ms e 10 s de treinamento.	76
Figura 41 - Sensitivity para cada movimento executado por sujeito amputado após 75 ms e 10 s de treinamento.	77
Figura 42 - Taxa de acerto de cada uma das 136 SVMs no ensaio com 40 sujeitos não-amputados após 10 s de treinamento.	77
Figura 43 - Taxa de acerto de cada uma das 136 SVMs no ensaio com 11 sujeitos amputados após 10 s de treinamento.....	78
Figura 44 - Taxa de acerto média de cada uma das 136 SVMs no ensaio com 40 sujeitos não amputados e 11 sujeitos amputados após 10 s de treinamento.	79
Figura 45 - Taxa de acerto das SVMs obtida para cada um dos 51 sujeitos após 10 s de treinamento.	79
Figura 46 - Taxa de acerto da classificação para cada um dos 51 sujeitos após 10 s de treinamento.	80
Figura 47 - Sobreposição das matrizes de confusão geradas para sujeitos não amputados após 75 ms de treinamento do conjunto reduzido de movimentos.	81
Figura 48 - Sobreposição das matrizes de confusão geradas para sujeitos amputados após 75 ms de treinamento do conjunto reduzido de movimentos.	81
Figura 49 - Sobreposição das matrizes de confusão geradas para sujeitos não amputados após 10 s de treinamento do conjunto reduzido de movimentos.	82
Figura 50 - Sobreposição das matrizes de confusão geradas para sujeitos amputados após 10 s de treinamento do conjunto reduzido de movimentos.....	82
Figura 51 - Taxa de acerto da classificação por sujeito.....	83
Figura 52 - Valores preditivos positivos para cada movimento executado por sujeito não amputado após 75 ms e 10 s de treinamento do conjunto reduzido de movimentos.	84
Figura 53 - Sensitivity para cada movimento executado por sujeito não amputado após 75 ms e 10 s de treinamento do conjunto reduzido de movimentos.....	84
Figura 54 - Valores preditivos positivos para cada movimento executado por sujeito amputado após 75 ms e 10 s de treinamento do conjunto reduzido de movimentos.....	84

Figura 55 - Sensitivity para cada movimento executado por sujeito amputado após 75 ms e 10 s de treinamento do conjunto reduzido de movimentos.	85
Figura 56 - Matrizes de confusão da classificação do sujeito 33 da base de dados de não amputados com treinamento de 75 ms e com treinamento de 10 s.	86
Figura 57 - Classificação dos movimentos na ordem da apresentação das amostras para o sujeito 33 da base de dados de não-amputados.	86
Figura 58 - Matrizes de confusão da classificação do sujeito 1 da base de dados de não amputados com treinamento de 75 ms e com treinamento de 10 s.	87
Figura 59 - Classificação dos movimentos na ordem da apresentação das amostras para o sujeito 1 da base de dados de não-amputados.	87
Figura 60 - Matrizes de confusão da classificação do sujeito 24 da base de dados de não amputados com treinamento de 75 ms e com treinamento de 10 s.	88
Figura 61 - Classificação dos movimentos na ordem da apresentação das amostras para o sujeito 24 da base de dados de não-amputados.	88
Figura 62 - EQM observado na classificação de 10 movimentos para o sujeito 33 da base de dados de não-amputados.	89
Figura 63 - EQM observado na classificação de 10 movimentos para o sujeito 1 da base de dados de não-amputados.	89
Figura 64 - EQM observado na classificação de 10 movimentos para o sujeito 24 da base de dados de não-amputados.	89

LISTA DE TABELAS

Tabela 1 - Exemplos de Kernel para SVM.....	24
Tabela 2 - Cálculo do ganho.....	36
Tabela 3 - Limites para o contador do gerador de endereços.....	42
Tabela 4 - Descrição dos sujeitos amputados da base Ninapro.....	48
Tabela 5 - Movimentos de dedos e pulsos da base Ninapro.....	49
Tabela 6 - Ensaios realizados.....	50
Tabela 7 - Projeto do experimento.....	51
Tabela 8 - Ganhos utilizados para gerar Figura 25.....	59
Tabela 9 - Ganhos calculados para os 17 movimentos executados pelo primeiro sujeito não-amputado da base de dados Ninapro.....	60
Tabela 10 - Ganhos calculados para as 136 SVMs binárias para o primeiro sujeito não-amputado da base de dados Ninapro.....	61
Tabela 11 - Teste do algoritmo de treinamento com padrões gerados em simulação.....	63
Tabela 12 - Acertos calculados para as 136 SVMs para o primeiro sujeito não-amputado da base de dados Ninapro.....	65
Tabela 13 - Percentual de ocupação lógica dos blocos implementados na correspondente FPGA.....	69
Tabela 14 - Análise de variância.....	70
Tabela 15 - Conjunto reduzido de movimentos de dedos e pulsos utilizados nos ensaios com 10 movimentos.....	80

LISTA DE ABREVIATURAS

BT-ED	<i>Evenly Divided Binary Tree</i>
BT-KM	<i>K-Means clustering Binary Tree</i>
DSP	<i>Digital Signal Processor</i>
DAG	<i>Directed Acyclic Graph</i>
EMG	Eletromiografia
EQM	Erro Quadrático Médio
FIR	<i>Finite Impulse Response</i>
FPGA	<i>Field Programmable Gate Array</i>
IMC	Índice de Massa Corpórea
LDA	<i>Linear Discriminant Analysis</i>
MAV	<i>Moving Average</i>
MLP	<i>Multi-Layer Perceptron</i>
MUAP	<i>Motor Unit Action Potential</i>
MUAPT	<i>Motor Unit Action Potential Train</i>
Ninapro	<i>Non Invasive Adaptive Prosthetics</i>
NLR	<i>Nonlinear Logistic Regression</i>
OVA	<i>One Versus All</i>
OVO	<i>One Versus One</i>
PWR	<i>Power</i>
RAM	<i>Random Access Memory</i>
RBF	<i>Radial Basis Function</i>
sEMG	Eletromiografia de superfície
SME	Sinal Mioelétrico
SMO	<i>Sequential Minimal Optimization</i>
SoC	<i>System on Chip</i>
SSC	<i>Slope Signal Change</i>
SVM	<i>Support Vector Machines</i>
UART	<i>Universal Asynchronous Receiver-Transmitter</i>
VHDL	<i>Very High Speed Integrated Circuit Hardware Description Language</i>
WL	<i>Waveform Length</i>
ZC	<i>Zero Crossing</i>

SUMÁRIO

1	INTRODUÇÃO	13
2	REVISÃO BIBLIOGRÁFICA	16
2.1	PRINCÍPIOS SOBRE ANATOMIA, FISIOLOGIA E ELETROMIOGRAFIA	16
2.2	CLASSIFICAÇÃO E MÁQUINAS DE VETORES DE SUPORTE	19
2.3	TRABALHOS RELACIONADOS À CLASSIFICAÇÃO DE MOVIMENTOS UTILIZANDO SEMG	26
3	METODOLOGIA EXPERIMENTAL	30
3.1	DESCRIÇÃO DO SISTEMA EMBARCADO	30
3.1.1	Implementação do filtro FIR em VHDL	30
3.1.2	Implementação do bloco de extração de características em VHDL	32
3.1.3	Implementação em VHDL do bloco de ganho do sistema	35
3.1.4	Implementação em VHDL da etapa de treinamento da SVM	37
3.1.5	Implementação em VHDL do bloco de validação da SVM	43
3.1.6	Implementação em VHDL da etapa de classificação da SVM	44
3.1.7	Implementação do bloco de controle em VHDL	47
3.2	DESCRIÇÃO DO PROTOCOLO DO EXPERIMENTO	47
3.2.1	Base de dados utilizada	47
3.2.2	Procedimentos executados para avaliar o desempenho do sistema	49
3.2.3	Método utilizado para validação estatística	51
4	RESULTADOS E DISCUSSÕES	53
4.1	VALIDAÇÃO DO SISTEMA EMBARCADO	53
4.1.1	Validação do filtro FIR	53
4.1.2	Validação do bloco de extração de características	57
4.1.3	Validação do bloco de ganho do sistema	60
4.1.4	Validação da etapa de treinamento da SVM	62
4.1.5	Validação do bloco de validação da SVM	63
4.1.6	Validação da etapa de classificação da SVM	66
4.1.7	Avaliação da ocupação lógica do sistema	68
4.2	AVALIAÇÃO DO DESEMPENHO DO SISTEMA COMPLETO	70
4.2.1	Ensaio com aprendizado de 17 movimentos	72
4.2.2	Ensaio com aprendizado de 10 movimentos	80
4.2.3	Análise dos sujeitos com maior taxa de acerto	85
4.2.4	Comparação dos resultados com trabalhos relacionados da área	90
5	CONCLUSÕES	92
	REFERÊNCIAS BIBLIOGRÁFICAS	94

1 INTRODUÇÃO

A Engenharia Biomédica é a aplicação dos conhecimentos das ciências exatas e da engenharia nas áreas biológicas e médicas para resolver problemas de saúde e bem-estar. Com o aumento da expectativa de vida da população, ela vem ganhando importância desde a sua criação na década de 50 (BUREAU OF LABOR STATISTICS, 2015; JO *et al.*, 2015; NEBEKER, 2003). Um dos desafios da Engenharia Biomédica, mais especificamente da Engenharia de Reabilitação, é a busca pelo aumento da qualidade de vida de pessoas com mobilidade comprometida devido à idade avançada, doenças, nascença sem algum membro ou perda de um membro devido a acidente, entre outros motivos. A reabilitação com uso de próteses ou exoesqueletos pode possibilitar que essas pessoas tenham as mesmas capacidades de um indivíduo saudável. Portanto, com este foco, esforços são empregados para que exoesqueletos e próteses de pernas e braços tenham grande número de graus de liberdade e executem movimentos similares aos naturais (BURCK; BIGELOW; HARSHBARGER, 2011).

Um tipo de sinal bioelétrico bastante explorado na área de reabilitação é o sinal mioelétrico, que contém a informação da intenção de movimento de um sujeito. Esses sinais circulam pelo sistema nervoso a fim de ativar os músculos motores e podem ser detectados através de eletrodos de superfície (ASGHARI OSKOEI; HU, 2007). É na classificação destes sinais com técnicas de Eletromiografia (EMG), Inteligência Computacional e Instrumentação Inteligente que se buscam soluções para o controle e atuação de próteses com elevado número de graus de liberdade. Tipicamente os sinais mioelétricos são obtidos através de eletrodos não invasivos posicionados na região ventral dos músculos, manipulados para extração de informação útil e apresentados a um algoritmo classificador. Por fim, o classificador apresenta como resposta o movimento executado ou a intenção de movimento de acordo com os sinais recebidos (ATZORI *et al.*, 2014; ISON; ARTEMIADIS, 2014). Além da aplicação em próteses e exoesqueletos, a classificação de movimentos é uma tarefa explorada no controle de dispositivos remotos por dispensar o uso das mãos (ISON *et al.*, 2015; PETERNEL *et al.*, 2016; SMITH; KUIKEN; HARGROVE, 2016).

Dois desafios verificados nos estudos que envolvem classificação de sinais mioelétricos são atingir altas taxas de acerto e/ou diminuir o tempo necessário para treinamento do classificador. Altas taxas de acerto são apresentadas em vários estudos (PEERDEMAN *et al.*, 2011; SUBASI *et al.*, 2018; TOSIN *et al.*, 2017), e alguns artigos propõem uso de novos classificadores (CENE; SANTOS; BALBINOT, 2018; SUBASI *et al.*, 2018), melhoria dos algoritmos (CRAMMER; SINGER, 2001; DRINEAS; MAHONEY, 2010; JOACHIMS;

DORTMUND; JOACHIMSCSUNI-DORTMUNDDE, 1999; LEE; LIN; WAHBA, 2004; PLATT, 1999), uso de técnicas de pré-processamento (TOSIN *et al.*, 2017) e pós-processamento (HUANG *et al.*, 2016), ou emprego de *hardware* dedicado para redução do tempo necessário para treinar o classificador (BOSCHMANN *et al.*, 2016; PAPADONIKOLAKIS; BOUGANIS, 2010).

Novos algoritmos estão surgindo a cada ano e os antigos sendo melhorados constantemente, mas a natureza sequencial das implementações em *software* dificilmente permitirá atingir os baixos tempos de processamento e latência possíveis com implementações com *hardware* dedicado. As implementações em *hardware* possuem o potencial de atingir os requisitos de tempo de processamento, consumo de energia e portabilidade que se deseja em um sistema de classificação de sinais de eletromiografia e mesmo assim ainda são pouco estudadas na respectiva área abordada neste trabalho (SALIM; SALIM; GUAN, 2011).

Inserido neste cenário, a presente dissertação explora o algoritmo da técnica Máquina de Vetores de Suporte (SVM) para propor um sistema embarcado dedicado à tarefa de classificação de sinais de EMG de Superfície (sEMG) utilizando FPGA. Pretende-se embarcar desde a etapa de pré-processamento (filtragem do sinal e extração de características), treinamento do classificador através de algoritmo de aprendizagem supervisionada, até o pós-processamento (validação dos parâmetros para o classificador) para reduzir o tempo necessário para treinar o classificador. Almeja-se treinar as Máquinas de Vetores de Suporte em menos de 300 ms para distinguir os movimentos de dedos e mão disponíveis na base de dados Ninapro (*Non Invasive Adaptive Prosthetics*) descrita por ATZORI; MULLER (2015), uma base internacional aceita e utilizada pela literatura especializada.

A escolha da SVM se deve a um trabalho executado por BELLINGEGNI *et al.* (2017) que compara os classificadores considerados estado da arte no reconhecimento de padrões: Regressão Logística Não Linear (NLR), Perceptron de Múltiplas Camadas (MLP), Análise Discriminante Linear (LDA) e SVM. O artigo conclui que a SVM apresenta a melhor performance quando utilizada para classificar cinco movimentos de mão executados por 30 indivíduos com amputação transradial, mas apesar das altas taxas de classificação, o tempo necessário para treinar este classificador é muito alto para utilização em aplicações de tempo real. Além disso, o Laboratório IEE possui uma ampla experiência em estudos sobre classificação de sinais de sEMG e vários dos seus trabalhos demonstram a vantagem da SVM sobre outros modelos básicos de inteligência computacional.

Uma forte candidata para melhoria do tempo de resposta do classificador é a FPGA, um *hardware* bastante utilizado no desenvolvimento de sistemas dedicados. Uma arquitetura em

FPGA que utilize técnicas de paralelização e *pipeline* permite melhorar a latência e tempo de treinamento de tarefas com baixa dependência temporal quando comparados a microprocessadores que executam instruções sequenciais.

Este trabalho apresenta no Capítulo 2 uma breve revisão bibliográfica sobre anatomia, fisiologia e eletromiografia, que são conhecimentos essenciais para o entendimento da dinâmica dos sinais de entrada do sistema de classificação; revisão sobre classificação focado nas Máquinas de Vetores de Suporte; e trabalhos relacionados a este encontrados na literatura. No Capítulo 3 é descrita a metodologia experimental com a descrição detalhada do sistema embarcado implementado e descrição do protocolo do experimento, com informações da base de dados utilizada, condução dos ensaios e procedimento utilizado para a validação estatística. O quarto capítulo apresenta os resultados e discussões gerados a partir dos testes relacionados ao sistema embarcado e dos ensaios executados com a base de dados Ninapro. Por fim, o Capítulo 5 contém as conclusões e propostas de trabalhos futuros com o sistema criado.

2 REVISÃO BIBLIOGRÁFICA

Conceitos de anatomia, fisiologia e eletromiografia são essenciais para o correto posicionamento de eletrodos, aquisição e filtragem do sinal. O entendimento das técnicas de classificação de sinais, pré-processamento, pós-processamento e classificação permitem a construção do sistema de classificação. Neste capítulo são descritas as informações buscadas na literatura pertinentes ao tema proposto nesta dissertação.

2.1 PRINCÍPIOS SOBRE ANATOMIA, FISIOLOGIA E ELETROMIOGRAFIA

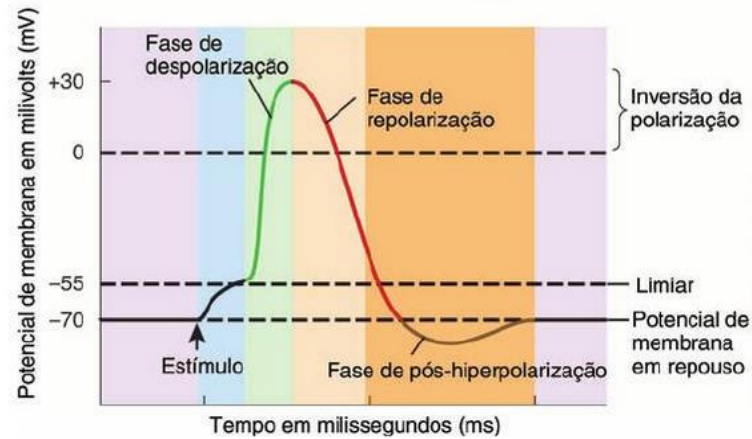
Movimentos como flexão e extensão de braços são voluntários e envolvem várias partes do corpo: ossos, articulações, tecido muscular esquelético e sistema nervoso. Neste contexto, os ossos são os tecidos que fornecem suporte e proteção ao corpo, e as articulações são formadas por tecidos conjuntivos flexíveis que mantêm os ossos unidos de forma a permitir o movimento e a flexibilidade. Os músculos consistem em células alongadas especializadas em gerar força, produzem movimentos através da alternância entre contração e relaxamento; e os neurônios são células do sistema nervoso que convertem estímulos nervosos em potenciais de ação e os conduzem às fibras musculares (TORTORA; DERRICKSON, 2010). Assim sendo, um movimento voluntário é iniciado por um estímulo nervoso que é conduzido aos músculos através de potenciais de ação, gerando a contração deste.

Os potenciais de ação ocorrem tanto nos neurônios quanto nas fibras musculares, são reações eletroquímicas geradas quando um estímulo leva o potencial elétrico da membrana celular a um certo nível denominado limiar (ver Figura 1). De acordo com TORTORA; DERRICKSON (2010), algumas características do potencial são:

- o típico potencial de membrana de um neurônio é de -70 mV, e nas fibras musculares esqueléticas está mais próximo de -90 mV;
- a duração do impulso nervoso é de 0,5 a 2,0 ms, enquanto o potencial de ação muscular dura aproximadamente 1,0 a 5,0 ms, nas fibras musculares esqueléticas;
- a velocidade de propagação dos potenciais de ação ao longo dos axônios mielinizados de maior diâmetro é aproximadamente 18 vezes maior do que a velocidade de propagação ao longo do sarcolema das fibras musculares esqueléticas.

- caso o estímulo seja superior ao limiar, um ou mais potenciais de ativação são gerados, todos com mesma amplitude, estando o estímulo relacionado com a frequência dos potenciais de ativação e não com a amplitude deles.

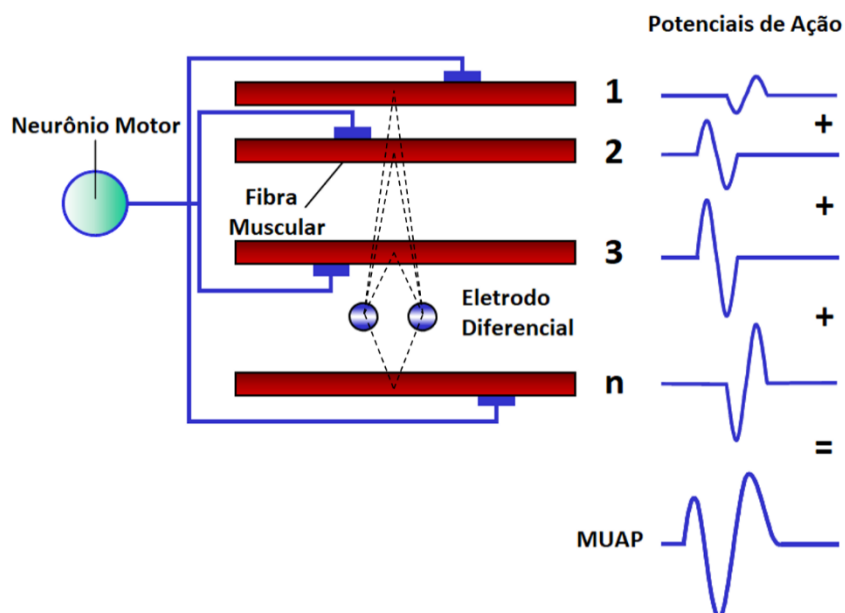
Figura 1 - Potencial de ativação gerado a partir de um estímulo.



Fonte: TORTORA; DERRICKSON, 2010

Cada motoneurônio (neurônio capaz de fazer um músculo entrar em atividade) que emerge da medula espinhal inerva várias fibras musculares, sendo o conjunto de todas as fibras musculares inervadas por uma fibra nervosa chamada unidade motora, e o somatório do potencial de ação das fibras individuais denominado MUAP (*Motor Unit Action Potential*) (Figura 2). Quanto menos fibras musculares compõem uma unidade motora, maior a graduação fina da força total do músculo (HALL, 2017).

Figura 2 - Composição de um MUAP: somatório de potenciais de ação individuais de fibras musculares adquirido por um eletrodo.



Fonte: adaptado de (KONRAD, 2006)

As fibras musculares de uma unidade ficam dispersas por todo o músculo em microfeixes de 3 a 15 fibras, e esses microfeixes ocorrem intercalados com outros microfeixes de diversas unidades motoras. Isso permite que unidades motoras distintas se contraíam em apoio umas às outras: no início da contração muscular poucas unidades motoras são recrutadas e na medida em que o músculo é mais solicitado mais unidades motoras são ativadas. Portanto a força que um músculo executa depende de dois fatores: número de unidades motoras recrutadas e frequência de estimulação das unidades.

Com o aumento da frequência de estimulação, há um momento em que cada nova contração ocorre antes do término da anterior, ocorrendo fusão das contrações em contrações mais fortes. Essa sequência de MUAPs gerada pela unidade motora ao ser ativada repetidamente é chamado de MUAPT (*Motor Unit Action Potential Train*) (HALL, 2017).

Os MUAPs formam um campo eletromagnético em suas proximidades ao percorrerem as fibras musculares e um eletrodo-agulha ou eletrodo de superfície próximo ao músculo é capaz de detectar o Sinal Mioelétrico (SME) resultante desses campos eletromagnéticos. Geralmente o SME é formado por vários MUAPTs, sendo que a ativação de apenas uma unidade motora só é possível em contrações musculares muito fracas. Portanto, de forma resumida, a Eletromiografia (EMG) é uma técnica para estudo dos músculos através destes sinais.

O uso de eletrodos de superfície para obter o sinal de EMG apresenta vantagens e desvantagens com relação ao uso de eletrodos-agulha: a forma invasiva de eletrodo captura os sinais diretamente do músculo no qual se deseja verificar a ativação, porém as agulhas são desconfortáveis ao usuário principalmente em uso por longos períodos de tempo. A forma não invasiva, também chamada de Eletromiografia de superfície (sEMG), apresenta ruído, interferência e sinais mais fracos, porém possibilita a obtenção do sinal de forma mais confortável e mais fácil (BASMAJIAN; DE LUCA, 1985). Segundo KONRAD (2006), a atenuação e o ruído do sinal de sEMG se deve a alguns fatores:

- características dos tecidos: a condutividade varia de acordo com o tipo de tecido, espessura, mudanças fisiológicas e temperatura. Ao usar eletrodo de superfície deve-se fazer a correta preparação da pele, eliminando-se pelos e tecido morto; mesmo assim, o tecido epitelial atenua o sinal entre músculo e eletrodo;
- *cross talk* fisiológico: músculos vizinhos podem interferir no sinal de EMG ao produzir um sinal forte o suficiente para ser capturado pelo eletrodo;

- desalinhamento de eletrodo com relação ao centro do músculo: durante um movimento, o centro do músculo se desloca e, como o eletrodo está fixo na pele, pode haver um desalinhamento entre eletrodo e músculo;
- ruído externo, eletrodos e amplificadores: interferência de equipamentos presentes no ambiente, qualidade dos eletrodos e amplificadores impactam na qualidade do sinal adquirido.

Apesar das desvantagens do sEMG, ele contém informação suficiente da intenção de movimento do sujeito e é largamente utilizado na literatura (ATZORI; MULLER, 2015). Tipicamente o sinal de sEMG pode variar entre $\pm 5000 \mu\text{V}$, estando concentrado no espectro de frequência de 6 a 500Hz com a maior parte da potência entre 20 e 150 Hz. Este biosinal é de natureza estocástica devido às unidades motoras recrutadas estarem em constante mudança e à forma que os potenciais de ativação se sobrepõem. As características temporais, de frequência e espectro-temporais são complexas no sEMG e atributos (também denominadas de métricas ou características) como RMS (*Root Mean Square*) e MAV (*Moving Average*) são frequentemente utilizados na análise e na classificação deste sinal (KONRAD, 2006).

2.2 CLASSIFICAÇÃO E MÁQUINAS DE VETORES DE SUPORTE

Classificação é uma tarefa amplamente estudada e apresenta diversas aplicações, como por exemplo, na taxonomia (FISHER, 1936), no diagnóstico de doenças (WOLBERG; MANGASARIAN, 1990), na área de telecomunicações (ANGUITA; BONI; RIDELLA, 2003; GROLEAT; ARZEL; VATON, 2012) e nas pesquisas e no desenvolvimento de interfaces que utilizem sinais biológicos (CHEN *et al.*, 2016; PIÑA-RAMÍREZ; VALDÉS-CRISTERNA; YÁÑEZ-SUÁREZ, 2006; SALIM; SALIM; GUAN, 2011; SHIMA; TSUJI, 2007; TOSIN *et al.*, 2017; WANG *et al.*, 2005) entre outras.

A área de aprendizagem de máquina é um campo de estudo cujo principal interesse é o de resolver problemas de categorização através de algoritmos. As técnicas de aprendizado de máquina podem ser divididas em dois tipos principais: supervisionado – quando existe um conjunto de entradas e rótulos (também chamados de alvo ou saída desejada) correspondentes utilizado como exemplo e ponto de partida do classificador no seu treinamento – e não-supervisionado – quando o classificador deve agrupar as entradas em rótulos com base em uma métrica de qualidade sem ter o alvo desejado, como por exemplo, na maioria dos métodos de agrupamento de dados (*clustering*) (HAYKIN, 2009).

O aprendizado supervisionado necessita que o algoritmo primeiramente aprenda os padrões que deve detectar através de uma fase de treinamento para posteriormente classificar. Portanto, divide-se a sua operação em:

- fase de aprendizado ou treinamento: momento em que o classificador aprende o problema que deve resolver. Um conjunto de atributos (\mathbf{x}_i) com respectivos rótulos (y_i) é usado para treinar o classificador em uma tarefa de otimização. Busca-se uma função $f(\mathbf{x})$ que consiga classificar os sinais de entrada apresentados através de alguma técnica de aprendizado de máquina; e, além disso, que generalize o comportamento do classificador para classificar entradas desconhecidas;
- fase de detecção ou classificação: o classificador treinado recebe dados de entrada e prevê um rótulo ou uma classe para essas entradas de acordo com o que aprendeu na fase anterior. A função $f(\mathbf{x})$ é utilizada nesta fase para gerar os rótulos (y_i) para um novo conjunto de atributos (\mathbf{x}_i) cujos rótulos são desconhecidos pelo sistema.

De acordo com AL OMARI *et al.* (2014); ATZORI *et al.* (2014); MOURA; FAVIEIRO; BALBINOT (2016), uma técnica de aprendizado de máquina com bons resultados na classificação de SME é a técnica Máquinas de Vetores de Suporte representada pela Equação (1):

$$f(\mathbf{x}) = \boldsymbol{\omega} \cdot \boldsymbol{\varphi}(\mathbf{x}) + b \quad (1)$$

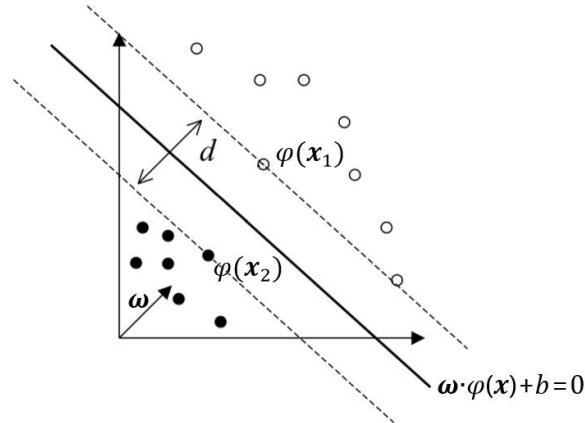
onde \mathbf{x} é o vetor de entrada, $\boldsymbol{\varphi}$ é uma função que permite classificação não-linear através do uso de diferentes tipos de *kernel*, $\boldsymbol{\omega}$ é o vetor de pesos do classificador, b é o *bias* e $f(\mathbf{x})$ é a saída predita.

SVM foi inicialmente desenvolvida para solução de problemas binários e posteriormente estendido para problemas multiclasse. Seu embasamento está na teoria de aprendizado estatístico desenvolvida por (VAPNIK, 1999) para obtenção de classificadores com boa generalização. No problema de classificação de movimentos em função do processamento dos sinais de EMG, bons resultados são obtidos tanto com função *kernel* linear, função de base radial ou polinomial (AL OMARI *et al.*, 2014; ATZORI *et al.*, 2014; MOURA; FAVIEIRO; BALBINOT, 2016).

Na fase de treinamento das SVMs, o objetivo é encontrar os pesos para a função que maximizam a distância entre classes. Essa distância é representada na Figura 3 por d , e pode ser calculada através da projeção de $\boldsymbol{\varphi}(\mathbf{x}_1) - \boldsymbol{\varphi}(\mathbf{x}_2)$ na direção de $\boldsymbol{\omega}$ indicada na Equação (2):

$$d = (\varphi(x_1) - \varphi(x_2)) \cdot \left(\frac{\omega}{\|\omega\|} \right) \quad (2)$$

Figura 3 - Representação visual da distância entre hiperplanos.



Fonte: adaptado de (HEARST, 1998)

Com os valores +1 e -1 para rótulos das duas classes de um problema binário, as Equações (1) e (2) são transformadas nas Equações (3) a (6) para se obter a distância entre hiperplanos:

$$f(x_1) = \omega \cdot \varphi(x_1) + b = +1 \quad (3)$$

$$f(x_2) = \omega \cdot \varphi(x_2) + b = -1 \quad (4)$$

$$\omega \cdot (\varphi(x_1) - \varphi(x_2)) = 2 \quad (5)$$

$$d = \frac{2}{\|\omega\|} \quad (6)$$

$\varphi(x_1)$ e $\varphi(x_2)$ são vetores de suporte – os pontos mais próximos à fronteira entre as duas classes. A chamada margem do classificador (ver Equação (7)) é a distância entre os vetores de suporte e o hiperplano que separa as classes, metade da distância d

$$margem = \frac{1}{\|\omega\|} \quad (7)$$

e para maximizar d , elabora-se um problema de otimização que maximize $\|\omega\|^{-1}$, ou equivalentemente, minimize $\|\omega\|^2$ (BISHOP, 2006). Este problema de minimização está representado pela Equação (8) e possui a restrição indicada pela Equação (9):

$$\min_{\omega, b} \frac{1}{2} \|\omega\|^2 \quad (8)$$

$$y_i \cdot f(x_i) \geq 1, \quad \forall i \quad (9)$$

O fator $\frac{1}{2}$ na Equação (8) é um fator de escala adicionado por conveniência, e a restrição garante que não haja pontos classificados erroneamente e nem pontos na região marginal. No caso da classificação de sinais de eletromiografia, assim como, na maioria das situações, a rigidez das Equações (8) e (9) são um problema, pois os dados não são linearmente separáveis, então uma folga é inserida no modelo de forma a permitir erros na classificação (BISHOP, 2006). Assim, o problema de otimização passa a ser representado pela Equação (10):

$$\min_{\omega, b, \xi} \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^m \xi_i \quad (10)$$

sujeito às condições indicadas nas Equações (11) e (12):

$$y_i \cdot f(\mathbf{x}_i) \geq 1 - \xi_i \quad (11)$$

$$\xi_i \geq 0, \quad \forall i \quad (12)$$

onde C é um termo de regularização que impõe peso entre os dois termos da função de minimização: o termo que maximiza a margem, e o termo que penaliza pontos classificados erroneamente. Este é um problema de otimização do tipo convexo, resolvido pelo método dos multiplicadores de Lagrange e deve satisfazer as condições de Karush-Kuhn-Tucker (KKT) (KARUSH, 2014; KUHN; TUCKER, 1951).

O problema não linear representado pela Equação (13) é reescrito na sua forma dual (Equação (14)):

$$(PNL) \begin{cases} \min f(x) \\ \text{s. a. } g_i(x) \leq 0 \\ h_j(x) = 0 \end{cases} \quad (13)$$

$$L(x, \mu, \lambda) = f(x) + \sum_{i=1}^m \mu_i g_i(x) + \sum_{j=1}^n \lambda_j h_j(x) \quad (14)$$

Como $f(x)$ é determinado pela Equação (10), $g_i(x)$ pelas Equações (11) e (12), e não há restrição de igualdade, a forma dual do problema não linear é reescrita pela Equação (15):

$$L = \frac{\|\omega\|^2}{2} + C \sum_{i=1}^m \xi_i + \sum_{i=1}^m \alpha_i (1 - \xi_i - y_i f(\mathbf{x}_i)) - \sum_{i=1}^m \beta_i \xi_i \quad (15)$$

onde o multiplicador de Lagrange α substitui μ da primeira desigualdade e β da segunda. Sabendo-se que é um problema do tipo convexo, a derivada da função L em relação a ω no seu ponto de mínimo deve satisfazer a Equação (16):

$$\frac{\partial L}{\partial \omega} = \omega - \sum_{i=1}^m \alpha_i y_i \varphi(\mathbf{x}_i) = 0 \quad (16)$$

Portanto, os pesos podem ser calculados pela Equação (17):

$$\omega = \sum_{i=1}^m \alpha_i y_i \varphi(\mathbf{x}_i) \quad (17)$$

Da derivada da função em relação a ξ_i , calculam-se as Equações (18) e (19):

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - \beta_i = 0 \quad (18)$$

$$C = \alpha_i + \beta_i \quad (19)$$

e da derivada da função em relação a b , calcula-se a Equação (20):

$$\frac{\partial L}{\partial b} = - \sum_{i=1}^m \alpha_i y_i = 0 \quad (20)$$

Com essa informação, escreve-se o problema dual da Equação (21):

$$\left\{ \begin{array}{l} \min_{\alpha} \frac{1}{2} \sum_{i,j=1}^m \alpha_i \alpha_j y_i y_j \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j) - \sum_{i=1}^m \alpha_i + b \sum_{i=1}^m \alpha_i y_i \\ \text{s. a.} \quad \sum_{i=1}^m \alpha_i y_i = 0 \\ \quad \quad 0 \leq \alpha_i \leq C \quad \forall i \end{array} \right. \quad (21)$$

No problema dual, $\varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j)$ é a função *kernel* que pode assumir, por exemplo, as formas da Tabela 1.

Tabela 1 - Exemplos de Kernel para SVM.

Tipo de Kernel	Equação $K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i) \cdot \varphi(\mathbf{x}_j)$
Linear	$K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \cdot \mathbf{x}_j$
Polinomial	$K(\mathbf{x}_i, \mathbf{x}_j) = (\delta(\mathbf{x}_i \cdot \mathbf{x}_j) + \kappa)^d$
Gaussiano	$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\sigma \ \mathbf{x}_i - \mathbf{x}_j\ ^2)$
Sigmoidal	$K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\delta(\mathbf{x}_i \cdot \mathbf{x}_j) + \kappa)$

O problema de otimização apresentado na Equação (21) só pode ser resolvido analiticamente quando o número de dados de treinamento é pequeno ou quando o problema é separável e se conhece de antemão quais dados são os vetores de suporte (BURGES, 1998). SVMs pequenas podem ser resolvidas por pacotes de otimização de uso geral que resolvam problemas quadráticos convexos com restrições lineares; e quando se trata de problemas maiores, há várias propostas de algoritmos para sua solução (HAYKIN, 2009):

- OSUNA *et al.* (1997) desenvolveram um algoritmo de decomposição que resolve uma sequência de subproblemas muito menores com performance satisfatória em aplicações de até 100.000 pontos;
- PLATT (1999) estendeu o método de Osuna ao introduzir o algoritmo chamado *Sequential Minimal Optimization* (SMO). Este algoritmo quebra o problema em uma série de subproblemas resolvíveis analiticamente;
- JOACHIMS *et al.* (1999) propõem uma série de inovações, sendo as mais relevantes a sugestão de decompor o problema em uma série de problemas menores de uma forma diferente da proposta por Osuna; e o descarte de amostras que não são vetores de suporte por muitas iterações e provavelmente não serão no futuro;
- RIFKIN (2002) desenvolveu o algoritmo SvmFu que sintetiza as ideias propostas pelos três anteriores e adiciona algumas novas proposições;
- DRINEAS; MAHONEY (2005) desenvolveram um algoritmo que calcula uma aproximação para a matriz de *kernel* para acelerar esse cálculo;
- HUSH *et al.* (2006) descreveram algoritmos que produzem soluções aproximadas precisas através do emprego de um processo em dois estágios: o primeiro estágio produz uma solução aproximada para um problema dual de programação quadrática, e o segundo estágio mapeia essa solução para uma solução primal aproximada;

- WESTON; WATKINS (1998) propõem dois algoritmos que estendem SVM para problemas de múltiplas classes: *k-class Support Vector Machines* e *k-class Linear Programming Machines*. Esses métodos não apresentam performance melhor que o uso de múltiplos classificadores binários;
- BREDENSTEINER; BENNETT (1998) propõem outro algoritmo multiclasse porém conclui que múltiplos classificadores binários generalizam melhor que o seu algoritmo;
- CRAMMER; SINGER (2001) e LEE *et al.*, (2004) também trabalharam no problema de SVMs multiclasse, assim como, outros autores.

O algoritmo SMO com função *kernel* linear apresenta uma rotina simples e adequada para implementação em arquiteturas que exploram o paralelismo. Segue os seguintes passos indicados por PLATT (1999):

1. dois exemplos de entrada (\mathbf{x}_1 e \mathbf{x}_2) com respectivas classes (y_1 e y_2) e multiplicadores de Lagrange (α_1 e α_2) são disponibilizados ao algoritmo com os pesos ($\boldsymbol{\omega}$) do classificador binário e *bias* (b);
2. os pesos são renomeados para $\boldsymbol{\omega}^{old}$ e verifica-se se as classes são preditas corretamente ($f^{old}(\mathbf{x}_1) = y_1$ e $f^{old}(\mathbf{x}_2) = y_2$) quando f^{old} é calculado pela Equação (22):

$$\begin{aligned} f^{old}(\mathbf{x}_1) &= \boldsymbol{\omega}^{old} \cdot \mathbf{x}_1 + b \\ f^{old}(\mathbf{x}_2) &= \boldsymbol{\omega}^{old} \cdot \mathbf{x}_2 + b \end{aligned} \quad (22)$$

Caso haja erro na predição, uma perturbação ideal para α é calculada pelas Equações (23) e (24):

$$\begin{aligned} k_{11} &= \mathbf{x}_1^T \cdot \mathbf{x}_1 \\ k_{12} &= \mathbf{x}_1^T \cdot \mathbf{x}_2 \\ k_{22} &= \mathbf{x}_2^T \cdot \mathbf{x}_2 \end{aligned} \quad (23)$$

$$\Delta_2 = \frac{y_2(\boldsymbol{\omega}^{old} \cdot \mathbf{x}_1 - y_1 - \boldsymbol{\omega}^{old} \cdot \mathbf{x}_2 + y_2)}{k_{11} - 2k_{12} + k_{22}} \quad (24)$$

Esta perturbação é aplicada nos multiplicadores se eles não vierem a violar a restrição $0 \leq \alpha_i \leq C$; uma perturbação menor é aplicada se Δ_2 gerar violação (Equação (25)):

$$\begin{aligned} \alpha_1^{new} &= \alpha_1^{old} - \Delta_2^* \\ \alpha_2^{new} &= \alpha_2^{old} + \Delta_2^* \end{aligned} \quad (25)$$

3. pesos são recalculados satisfazendo a Equação (17). No caso de *kernel* linear o cálculo pode ser simplificado conforme indicado pela Equação (26):

$$\boldsymbol{\omega}^{new} = \boldsymbol{\omega}^{old} + y_1 \mathbf{x}_1 (\alpha_1^{new} - \alpha_1^{old}) + y_2 \mathbf{x}_2 (\alpha_2^{new} - \alpha_2^{old}) \quad (26)$$

4. o último passo consiste em calcular o *bias* satisfazendo as condições de Karush-Kuhn-Tucker com mínima alteração em *b* possível (Equação (27)).

$$\begin{cases} b^{new} \geq y_i - \boldsymbol{\omega}^{new} \cdot \mathbf{x}_i & \alpha_i^{new} = 0 \\ b^{new} \leq y_i - \boldsymbol{\omega}^{new} \cdot \mathbf{x}_i & \alpha_i^{new} = C \\ b^{new} = y_i - \boldsymbol{\omega}^{new} \cdot \mathbf{x}_i & 0 < \alpha_i^{new} < C \end{cases} \quad (27)$$

Os pesos e *bias* são atualizados a cada iteração para pares de entrada sendo necessário executar várias iterações até convergir em uma solução. As vantagens deste método é a pequena quantidade de passos, lógica de controle simples e a eliminação do cálculo da matriz $K(\mathbf{x}_i, \mathbf{x}_j)$. Além disso, possibilita o uso de uma arquitetura em *pipeline*. A classificação em mais de duas classes com este método requer o uso de múltiplas SVMs que podem ser organizadas nas topologias um contra todos (OVA – *One Versus All*) ou um contra um (OVO – *One Versus One*).

2.3 TRABALHOS RELACIONADOS À CLASSIFICAÇÃO DE MOVIMENTOS UTILIZANDO SEMG

Muitos trabalhos relacionados à classificação de movimentos utilizando eletromiografia são encontrados na literatura. PEERDEMAN *et al.* (2011) apresentaram uma relação de 44 estudos sobre classificação de movimentos de braço que datam de 1967 a 2010 com diferentes números de eletrodos, classificadores utilizados, características extraídas, números de sujeitos e quantidades de classes. Esses trabalhos, apesar de terem sido considerados estado da arte em 2011, foram realizados com uma baixa quantidade de sujeitos (1 a 12), o que pode gerar resultados com baixa representatividade estatística. Anos depois, ATZORI *et al.* (2014) criaram um conjunto de bases de dados com sinais de sEMG capturados durante a execução de movimentos de mão e braço, facilitando o desenvolvimento de novos estudos, permitindo a comparação entre trabalhos de diferentes autores, e principalmente aumentando a significância estatística dos trabalhos. Essas bases de dados estão expandindo e novas estão surgindo, inclusive uma no Laboratório de Instrumentação Eletro-Eletrônica (IEE-UFRGS) (CENE; SANTOS; BALBINOT, 2018; CENE *et al.*, 2019).

Uma parte fundamental do processo de classificação de movimentos é a extração de características do sinal de sEMG. Segundo ENGLEHART, K. *et al.* (1999), a escolha das

características é frequentemente mais importante do que a escolha do sistema de controle para o desempenho do controle multifuncional. Isso se deve ao fato de os sinais de eletromiografia de superfície serem altamente estocásticos e não estacionários, portanto características como RMS, média móvel e frequência média são extraídas dos sinais como métricas de entrada para o classificador (KONRAD, 2006).

A quantidade de características utilizadas na classificação impacta na complexidade do classificador, pois aumentam seu vetor de entrada e conseqüentemente o vetor de pesos que devem ser ajustados. Neste sentido, TOSIN *et al.* (2017) realizaram um estudo com 21 características do domínio do tempo e 6 características do domínio da frequência com o objetivo de apresentar uma forma de selecionar as características mais importantes para a classificação. Neste estudo verificou-se que uma correta seleção de características impacta na taxa de acerto do classificador além do tempo de processamento necessário para treiná-lo.

Outro ponto importante para aplicações de tempo real é o tamanho da janela (ou segmento) utilizada para extração de características: a janela deve ser grande o bastante para evidenciar as características e pequena o suficiente para não causar desconforto ao usuário por atrasos na resposta da prótese ou exoesqueleto. A literatura tem utilizado 300 ms como tempo limite para a resposta de um sistema de classificação de movimentos, e janelas deslizantes com sobreposição são frequentemente utilizadas na extração das características (ENGLEHART; HUDGINS; CHAN, 2003). Na abordagem com janelas deslizantes, o tamanho da janela pode ser configurado de forma a conter informação útil ao classificador, e a dinâmica do sistema é ditada pelo passo do janelamento (distância temporal entre o início de uma janela e de outra).

Quanto ao uso de classificadores, BELLINGEGNI *et al.* (2017), em um estudo recente, compararam quatro diferentes classificadores em um experimento realizado com 30 voluntários com amputação transradial. Estas pessoas estavam familiarizadas com próteses mioelétricas de mãos no momento dos ensaios. Enquanto os sujeitos executaram cinco movimentos, um eletromiógrafo utilizando seis eletrodos de superfície ativos capturou o sinal mioelétrico para alimentar o sistema de classificação. Os classificadores NLR (*Nonlinear Logistic Regression*), MLP (*Multi-Layer Perceptron*) e SVM foram treinados com o sinal de sEMG sem extração de características e, para o sistema com classificador LDA, as características MAV, RMS, variância, *Waveform Length* (WL) e *Slope Signal Change* (SSC) foram extraídas em janelas de 250 ms e passo de 50 ms. As Máquinas de Vetores de Suporte com função *kernel* de base radial atingiram os melhores resultados neste estudo.

MOURA; FAVIEIRO; BALBINOT (2016) compararam o uso de SVM com diferentes funções *kernel* (linear, base radial e polinomial) tendo como resultado pouca diferença na taxa

de acerto média obtida (83,7%, 85,1% e 80,8% respectivamente). Neste estudo, cinco voluntários não amputados executaram 17 movimentos de mão e braço, e 12 eletrodos de superfície capturaram os sinais de sEMG. Apenas a característica RMS foi extraída dos sinais em janelas com início e fim determinadas pelo início e fim do movimento.

HUANG *et al.* (2016) compararam cinco diferentes formas de combinar SVMs binárias para classificação multiclasse de 13 movimentos de dedos executados por cinco sujeitos não amputados e três sujeitos amputados. Os sinais de sEMG foram capturados por 16 eletrodos de superfície e as características MAV, *Zero Crossing* (ZC), SSC e WL foram extraídas em janelas de 250 ms e passo de 225 ms. Classificadores LDA e SVM organizados nas topologias OVA, OVO, DAG (*Directed Acyclic Graph*), BT-ED (*Evenly Divided Binary Tree*) e BT-KM (*K-Means clustering Binary Tree*) treinaram os movimentos. As SVMs com função *kernel* de base radial atingiram taxas de classificação acima de 83% para todos os métodos propostos, havendo uma diferença de apenas 5% entre a maior taxa de acerto e a menor.

As variáveis envolvidas na classificação de sinais sEMG são tantas que é difícil a comparação entre estudos. Trabalhos diferentes possuem quantidades de voluntários diferentes com graus de amputação diferentes; variam o número de eletrodos, assim como o posicionamento deles; as características extraídas não são as mesmas, bem como a quantidade de características utilizadas e a técnica de janelamento utilizada. Novos classificadores surgem a cada ano e mesmo classificadores bem aceitos pela literatura, como SVM, possuem diferentes funções *kernel* e diferentes topologias. Mesmo com todas essas variáveis, muitos algoritmos avançados de processamento de sinais foram testados com sucesso na classificação de sEMG em computadores. Portanto, o interesse em explorar implementações em *hardware* deve-se às principais aplicações desta área de estudo, onde tipicamente são necessários sistemas de pequeno porte, portáteis e de rápido processamento.

Na revisão dos estudos que envolvem implementações em *hardware* de processamento de sinais de sEMG de SALIM; SALIM; GUAN (2011), foram encontrados poucos trabalhos nesta área: sete trabalhos sobre extração de características utilizando FPGA, DSP (*Digital Signal Processor*), SoC (*System on Chip*) e microprocessadores, e três trabalhos sobre classificação utilizando FPGA e SoC. Os autores concluíram que as FPGAs e os SoCs possibilitam a criação de sistemas que atingem os requisitos de taxa de acerto, tempo de processamento e consumo de potência necessários no processamento de sinais de sEMG.

Em um trabalho mais recente, CHEN *et al.* (2016) propuseram um sistema completo com extração de características, treinamento e classificação em um SoC. Com três canais de entrada para o sinal de sEMG, o sistema extraiu as características MAV, WL e ZC em janelas

de 256 ms para um classificador *Naive Bayes* aprender seis movimentos: flexão e extensão do cotovelo, aperto de mão e extensão da mão, e flexão e extensão horizontal do ombro. O autor citou uma taxa de acerto de 92% atingida pelo sistema, porém não há informações do protocolo do experimento e nem do número de voluntários.

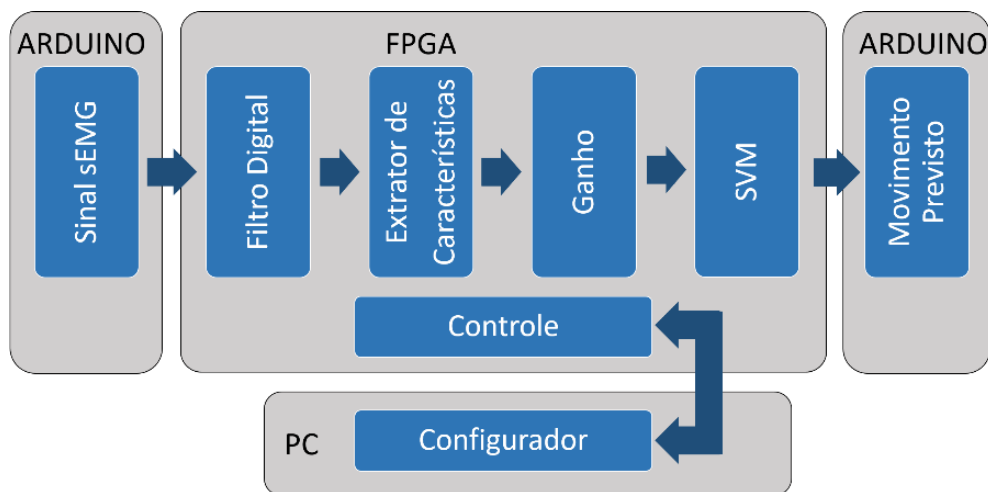
BOSCHMANN *et al.* (2016) também propuseram um sistema de classificação completo em um SoC. No sistema proposto, há um extrator de características embarcado na FPGA do SoC para extração de quatro características, e um classificador LDA embarcado no processador. O sistema foi desenvolvido para sensores de alta densidade, sendo a entrada composta por 256 canais de sEMG. Nessas condições, a implementação da extração de características em FPGA resultou no aumento da velocidade de treinamento em 4,8 vezes. Os ensaios foram realizados apenas por um voluntário amputado e não há informações sobre a taxa de acerto.

3 METODOLOGIA EXPERIMENTAL

3.1 DESCRIÇÃO DO SISTEMA EMBARCADO

O sistema embarcado foi desenvolvido na plataforma Mini-Module Plus Baseboard 2 da AVNET (2012) que contém uma FPGA Xilinx Kintex 7. O sistema proposto neste trabalho pode ser representado por cinco blocos principais: filtro digital, extrator de características, ganho, SVM e controle (ver Figura 4). Cada uma destas partes recebe um subcapítulo específico onde são abordadas a arquitetura e decisões pertinentes, sendo que o bloco SVM foi dividido em três partes: treinamento, validação e classificação.

Figura 4 - Diagrama de blocos do sistema.



Fonte: do autor

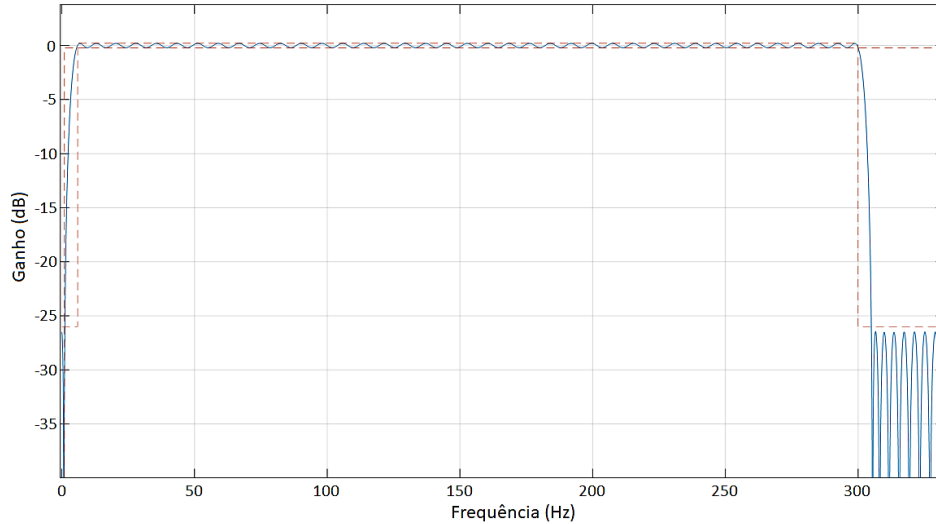
Para simular uma aplicação em tempo real, uma placa auxiliar (Arduino DUE (ARDUINO, 2019)) inseriu no sistema os sinais sEMG disponíveis na base de dados Ninapro (ATZORI *et al.*, 2014) descrita na Seção 3.2.1 à taxa de 2000 amostras por segundo. A captura do resultado da classificação também foi feita pela placa auxiliar que estava ligada à FPGA através de pinos de entrada e saída. Havia ainda uma interface UART ligada a um computador para possibilitar a configuração do sistema.

3.1.1 Implementação do filtro FIR em VHDL

Os doze canais de sinal mioelétrico que são disponibilizados na entrada da FPGA no diagrama de blocos da Figura 4 foram filtrados para eliminar artefatos e ruído fora da banda do sinal de sEMG. Para isso, foi especificado um filtro passa faixa do tipo FIR (*Finite Impulse Response*) com taxa de amostragem de 2 kHz cuja resposta em frequência está apresentada na

Figura 5. A banda de passagem foi definida de 9 Hz a 300 Hz com *ripple* máximo de 5% nessa região. Sinais abaixo de 1 Hz e acima de 400 Hz são rejeitados com atenuação de 26 dB.

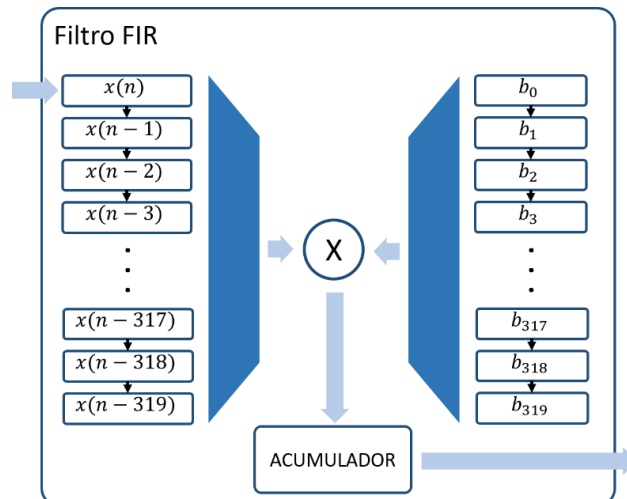
Figura 5 - Resposta em frequência do Filtro FIR implementado.



Fonte: do autor

O filtro FIR descrito em VHDL (*Very High Speed Integrated Circuit Hardware Description Language*) está representado na Figura 6. Este filtro FIR foi implementado com 320 conjuntos de registradores ligados em cadeia (*Shift Registers*) para guardar os dados de entrada com resolução de 12 bits ($x(n) - x(n - 319)$). Esta ligação faz com que a cada nova entrada de dados, todos os registradores sejam atualizados de forma a $x(n)$ guardar sempre o dado mais novo e os próximos 319 conjuntos de registradores guardarem os 319 últimos valores recebidos. O filtro possui 320 coeficientes de 16 bits ($b_0 - b_{319}$) que multiplicam os dados $x(n)$ a $x(n - 319)$, respectivamente.

Figura 6 - Diagrama do filtro FIR implementado.



Fonte: do autor

As multiplicações são realizadas uma a cada ciclo de *clock* por um multiplicador em DSP, pois o *clock* do sistema (125 MHz) é muito mais rápido que a taxa de amostragem de obtenção dos sinais de sEMG (2kHz) e assim se economiza recursos relacionados a programação. Os valores das operações de multiplicação são somados por lógica combinacional e armazenados temporariamente em um acumulador, resultando na operação representada pela Equação (28):

$$y(n) = \sum_{i=0}^{319} x(n-i) \cdot b_i \quad (28)$$

Após a aplicação do filtro FIR, foi realizada a operação de retificação completa do sinal (ver Equação (29)):

$$y_{RET}(n) = \text{abs}(y(n)) \quad (29)$$

Tanto o filtro FIR quanto o retificador foram aplicados aos 12 canais, portanto a lógica foi replicada 12 vezes para compor o bloco filtro digital.

3.1.2 Implementação do bloco de extração de características em VHDL

Quatro características foram extraídas do sinal sEMG filtrado: Média Móvel (MAV), Raiz Média Quadrática (RMS), Potência (PWR) e Comprimento de Onda (WL). As características utilizadas, neste trabalho, são descritas pelas Equações (30) a (33):

$$x_{MAV} = \frac{1}{N} \sum_{i=1}^N x_i \quad (30)$$

$$x_{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2} \quad (31)$$

$$x_{PWR} = \frac{1}{N} \sum_{i=1}^N x_i^2 \quad (32)$$

$$x_{WL} = \frac{1}{N} \sum_{i=1}^N |x_i - x_{i-1}| \quad (33)$$

onde x_i é a amostra i do sinal de EMG filtrado, N o tamanho da janela utilizada na extração, e x_{MAV} , x_{RMS} , x_{PWR} e x_{WL} são as características resultantes das operações. Neste trabalho foi

utilizada a técnica janela deslizante na extração das características, técnica também adotada por CENE; SANTOS; BALBINOT (2018) e SMITH; KUIKEN; HARGROVE (2016) em seus trabalhos.

Na implementação em VHDL do bloco de extração de características, os somatórios no tempo n foram calculados com o auxílio de um acumulador, como apresentado pela Equação (34):

$$\sum_{i=1}^N \alpha_i = ACC(n) = \left[\sum_{i=0}^{N-1} \alpha(n-i) \right] \quad (34)$$

onde α_i e $\alpha(n-i)$ são duas formas diferentes de representar uma amostra de entrada do acumulador, N é o tamanho do acumulador e $ACC(n)$ o valor contido no acumulador no tempo n . Com base na Equação (34), pode-se deduzir as Equações (35) a (37) para o acumulador no tempo $n+1$:

$$ACC(n+1) = \left[\sum_{i=0}^{N-1} \alpha(n+1-i) \right] = \left[\sum_{i=-1}^{N-2} \alpha(n-i) \right] \quad (35)$$

$$= \alpha(n+1) + \left[\sum_{i=0}^{N-1} \alpha(n-i) \right] - \alpha(n+1-N) \quad (36)$$

$$ACC(n+1) = \alpha(n+1) + ACC(n) - \alpha(n+1-N) \quad (37)$$

Portanto, a cada janela ou segmento do sinal de interesse é somado um termo novo (termo do tempo atual) e se subtrai um termo antigo. Um bloco de memória RAM e um registrador foram utilizados para executar esta operação, a memória armazenava os valores de α e o registrador os correspondentes valores do acumulador (ACC). Na extração das características MAV e WL, as memórias possuíam 10 bits de endereçamento e 12 bits de largura de dados, e na extração de PWR e RMS, uma memória compartilhada com 10 bits de endereçamento e 24 bits de largura de dados foi utilizada. Os acumuladores possuíam 24 bits para MAV e WL e 36 bits para PWR e RMS.

As janelas foram configuradas para terem 600 amostras (300 ms na taxa de amostragem de 2kHz) e, para evitar perda de resolução, os pontos de divisão foram substituídos por divisão com ganho configurável determinado pela Equação (38):

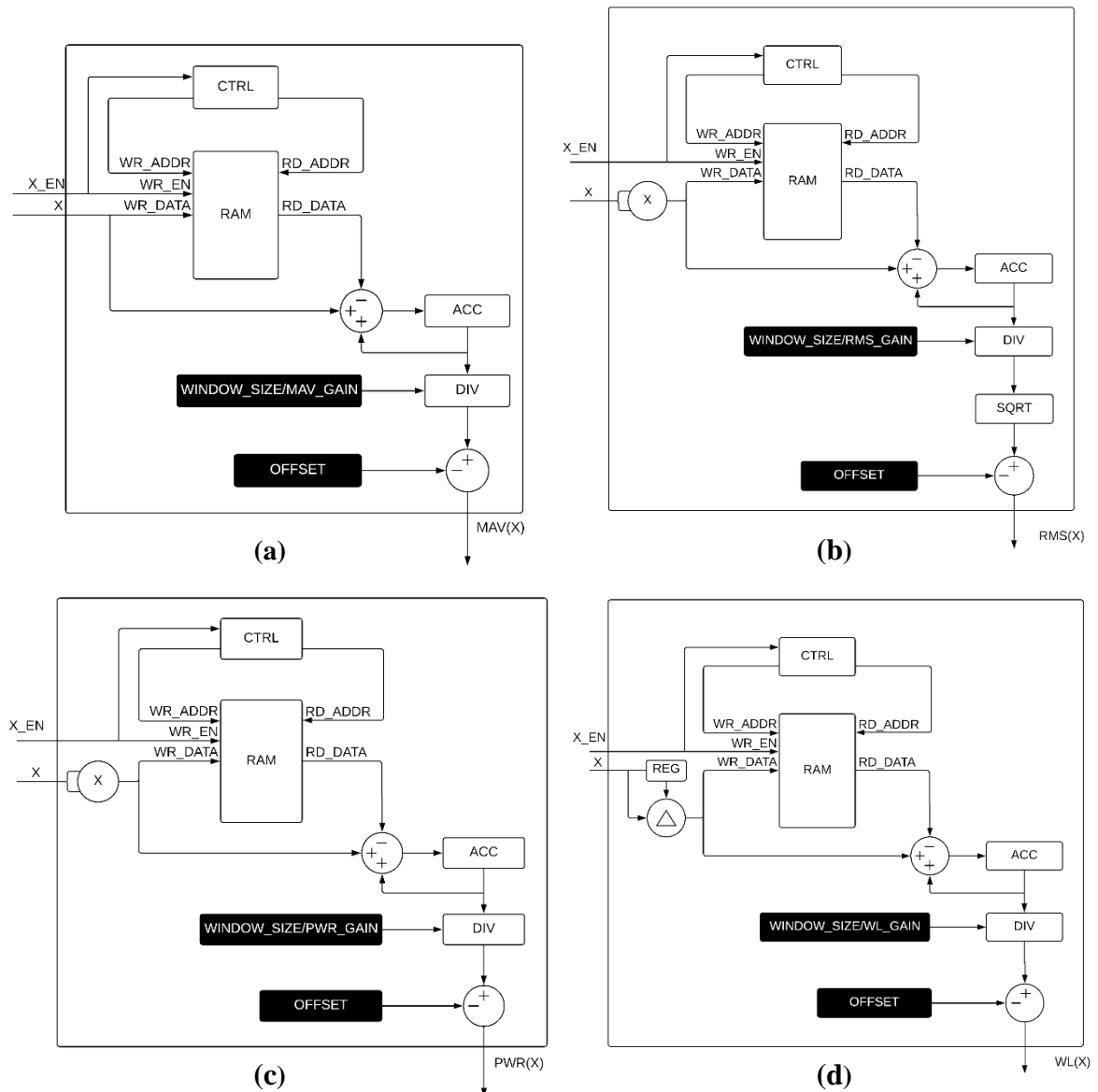
$$\frac{1}{N} \rightarrow \frac{Gain}{N} \quad (38)$$

sendo N o tamanho da janela (600 amostras) e $Gain$ o ganho configurável. Essas divisões são executadas em 34 ciclos de *clock* utilizando apenas operações de comparação, soma e deslocamento. As potências x_i^2 foram calculadas em blocos de DSP em um ciclo de *clock*; a raiz quadrada foi mapeada em RAM e somas, subtrações e módulo utilizaram apenas LUTs.

Devido à retificação aplicada no passo anterior, os valores de entrada no bloco de extração de características são positivos com resolução de 11 bits (0_d a 2047_d). As saídas possuem 8 bits de resolução e um *offset* foi aplicado para adequar o sinal à faixa de operação do algoritmo SVM (-128_d a 127_d).

A Figura 7 apresenta quatro diagramas de blocos que representam a extração de cada uma das características apresentadas anteriormente.

Figura 7 - Diagramas de blocos das Extrações das Características: (a) Média Móvel, (b) RMS, (c) Potência e (d) Comprimento de Onda.



Fonte: do autor

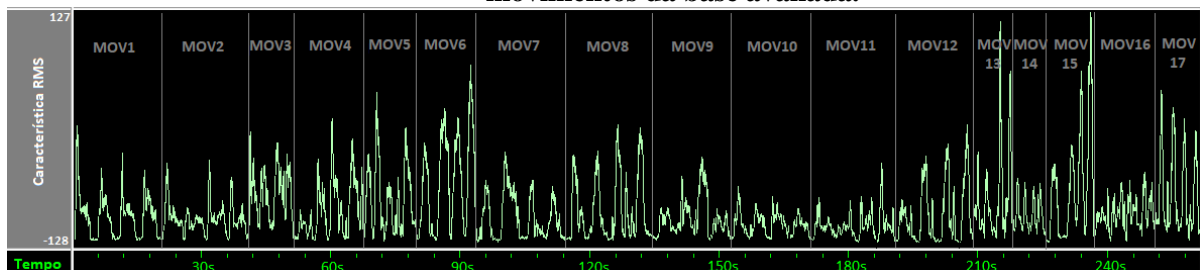
Todos os blocos possuem uma memória RAM, um acumulador e uma pequena lógica de controle (CTRL) que combinados geram a janela deslizante, um divisor (DIV) que permite o ganho configurável (o inverso do ganho é utilizado), e o *offset*. Na extração de PWR a potência do sinal de entrada (x_i^2) deve ser calculada, e na extração de RMS, além da potência, deve-se calcular a raiz quadrada (SQRT). O módulo $|x_i - x_{i-1}|$ da extração da característica PWR foi calculado com o auxílio de um registrador (REG) e lógica combinacional. Ao se utilizar 12 canais de entrada, os circuitos representados foram instanciados 12 vezes, gerando 48 características no total.

As características calculadas foram distribuídas entre os blocos de treinamento, validação e classificação de acordo com uma chave seletora controlada pelo usuário. Quando a chave estava posicionada para classificação, os resultados eram enviados para classificação em passos de 75 ms e, quando a chave estava posicionada para aprendizado, os resultados eram repartidos à proporção 75/25 entre treinamento (75% dos dados) e validação (25% dos dados) em passos de 75 ms. Tanto a chave seletora quanto os ganhos programáveis eram configurados pelo usuário do sistema através da UART.

3.1.3 Implementação em VHDL do bloco de ganho do sistema

As características extraídas dos movimentos diferem muito em amplitude entre um movimento e outro como ilustra o exemplo da Figura 8 ao apresentar o sinal RMS extraído do sinal sEMG de um canal durante a execução de 17 movimentos por um indivíduo da base de dados Ninapro. Essa diferença afeta o treinamento dos movimentos, principalmente na comparação entre dois movimentos de baixa amplitude, o que motivou a implementação de um ganho individual para cada SVM binário. O objetivo deste ganho era amplificar as diferenças dos sinais de entrada ao se utilizar o máximo da faixa de valores aceitos pela SVM binária. Diferentemente dos ganhos do extrator de característica, estes ganhos foram obtidos automaticamente, um por SVM binária.

Figura 8 - Exemplo da característica RMS do sinal de sEMG do canal 1 representando os 17 movimentos da base avaliada.



Fonte: do autor

Então, por exemplo, a entrada da SVM binária que é composta pelas características dos movimentos 10 e 11 da Figura 8 tem um terço de sua faixa utilizada, e um ganho de até 3 vezes pode ser aplicado nestes sinais de entrada, contanto que não haja saturação. Agora, para a SVM binária que diferencia os movimentos 10 e 15, toda a faixa de entrada é utilizada e a aplicação de um ganho saturaria os sinais relativos ao movimento 15. Com 48 sinais de entrada a análise é um pouco mais complexa e optou-se por utilizar o mesmo ganho para todos eles. O cálculo foi realizado da seguinte forma:

1. todos os ganhos foram inicializados com valor máximo;
2. foi realizada a comparação do valor instantâneo das 48 características em busca do maior valor para cada movimento;
3. calculou-se um valor de ganho que levasse este nível para mais perto do limite da faixa de entrada dos SVMs (ver Tabela 2);
4. se o valor calculado fosse menor que o valor anterior, o valor de ganho era atualizado;
5. eram comparados os valores de ganho calculados para os movimentos de entrada e era escolhido o menor para cada uma das 136 SVMs.

Tabela 2 - Cálculo do ganho.

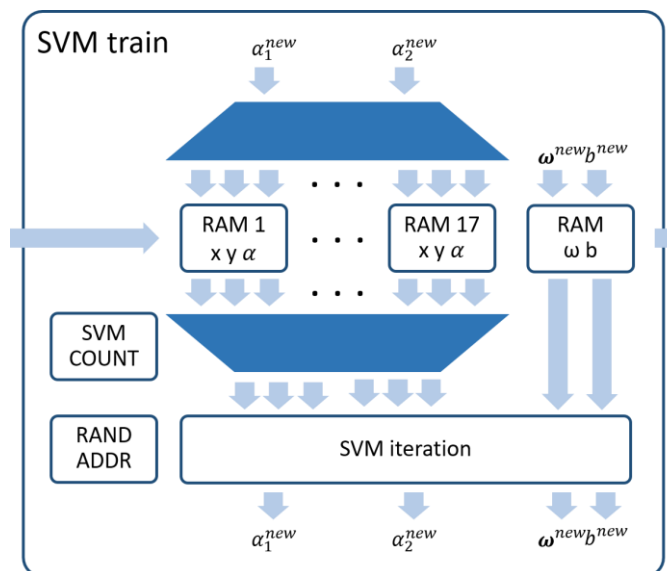
Faixas para enquadramento do maior valor instantâneo entre as características	Faixas com <i>offset</i> de 128 _d (representação por números positivos)	Ganho	Valor após aplicação do ganho	Valor após remoção do <i>offset</i>
[-128 _d , -91 _d]	[0 _d , 37 _d]	6,000	[0 _d , 222 _d]	[-128 _d , 94 _d]
[-90 _d , -72 _d]	[38 _d , 56 _d]	4,000	[152 _d , 224 _d]	[24 _d , 96 _d]
[-71 _d , -53 _d]	[57 _d , 75 _d]	3,000	[171 _d , 225 _d]	[43 _d , 97 _d]
[-52 _d , -36 _d]	[76 _d , 92 _d]	2,500	[190 _d , 230 _d]	[62 _d , 102 _d]
[-35 _d , -17 _d]	[93 _d , 111 _d]	2,125	[197 _d , 235 _d]	[69 _d , 107 _d]
[-16 _d , 7 _d]	[112 _d , 135 _d]	1,750	[196 _d , 236 _d]	[68 _d , 108 _d]
[8 _d , 34 _d]	[136 _d , 162 _d]	1,500	[204 _d , 243 _d]	[76 _d , 115 _d]
[35 _d , 63 _d]	[163 _d , 191 _d]	1,250	[203 _d , 238 _d]	[75 _d , 110 _d]
[64 _d , 89 _d]	[192 _d , 217 _d]	1,125	[216 _d , 244 _d]	[88 _d , 116 _d]
[90 _d , 127 _d]	[218 _d , 255 _d]	1,000	[218 _d , 255 _d]	[90 _d , 127 _d]

A avaliação dos ganhos era realizada em faixas e calculada com somas e deslocamentos para economia de recursos de DSP da FPGA. Esta avaliação deve ser realizada antes do treinamento, durante a alimentação do sistema com os dados de sEMG pois os blocos de treinamento, validação e classificação precisam da informação consolidada para que os valores de entrada sejam multiplicados pelo correspondente ganho.

3.1.4 Implementação em VHDL da etapa de treinamento da SVM

As características extraídas dos doze canais de sinal de sEMG eram disponibilizadas na entrada do bloco de treinamento SVM e armazenadas em blocos de memória de acordo com o movimento ao qual pertencem conforme ilustração da Figura 9. Cada movimento ocupava uma memória específica, ou seja, totalizando 17 memórias com $512 \times 48 \times 8$ bits cada: 512 é a quantidade de amostras que as memórias são capazes de armazenar (equivalente a 38,4 s), 48 é o número de características por amostra e 8 bits a resolução de cada característica (-128_d a 127_d).

Figura 9 - Bloco em VHDL responsável pelo treinamento SVM.



Fonte: do autor

Enquanto as memórias eram preenchidas com as amostras (\mathbf{x}), também era realizada a inicialização dos multiplicadores de Lagrange (α) e pesos (ω) satisfazendo as restrições da Equação (21). Os multiplicadores de Lagrange foram definidos com 8 bits de resolução sendo os primeiros 128 α de cada movimento inicializados em $C/2$ (C é o termo de regularização das SVMs igual a 127_d) e os demais em zero. Estes também foram organizados em 17 memórias e como cada movimento alimentava 16 SVMs binárias diferentes, cada memória possuía 16×512 multiplicadores no total. Os valores de *bias* eram inicializados em zero enquanto que os pesos eram inicializados respeitando as Equações (17) e (27). A Equação (39) indica o procedimento utilizado para atualizar os correspondentes pesos:

$$\omega^{new} = \omega^{old} + yx(\alpha^{new} - 0) \quad (39)$$

sendo y a classe da SVM binária que pode assumir os valores -1 e $+1$, \mathbf{x} o vetor de entrada e α^{new} o multiplicador de Lagrange atribuído à amostra \mathbf{x} . As variáveis $\boldsymbol{\omega}^{old}$ e $\boldsymbol{\omega}^{new}$ representam os vetores de pesos antes e depois da atualização, respectivamente. No cálculo do peso para a primeira amostra, o valor 0 é atribuído à variável $\boldsymbol{\omega}^{old}$. Cada uma das 136 SVMs possuía um vetor de pesos com 48 elementos de 24 bits e um valor de *bias* com 38 bits que eram armazenados em um bloco de memória.

No diagrama de blocos apresentado na Figura 9, além das memórias descritas (RAM 1 a RAM 17 x y e α e RAM ω b), há o módulo *SVM iteration* que executava o algoritmo SMO descrito na Seção 2.2; *SVM COUNT* que realizava o escalonamento do uso das memórias e *SVM iteration*; e *RAND ADDR* que escolhia aleatoriamente as amostras para treinamento. O algoritmo SMO foi transcrito para VHDL em uma arquitetura *pipeline*, ou seja, a cada ciclo de *clock* podia ser alimentado por um dado novo e após um período de latência de 81 ciclos apresentaria uma saída a cada ciclo de *clock* conforme indicado pela Figura 10.

Figura 10 - Operação em pipeline de *SVM iteration* respeitando escalonamento de *SVM COUNT*.

SVM COUNT Entrada	0	1	2	...	81	82	83	...	135	136	137	0	...	78	79	80	81	...
SVM iteration Entrada	SVM 0	SVM 1	SVM 2	...	SVM 81	SVM 82	SVM 83	...	SVM 135			SVM 0	...	SVM 78	SVM 79	SVM 80	SVM 81	...
SVM COUNT Saída					0	1	2	...	54	55	56	57	...	135	136	137	0	...
SVM iteration Saída					SVM 0	SVM 1	SVM 2	...	SVM 54	SVM 55	SVM 56	SVM 57	...	SVM 135			SVM 0	...

Fonte: do autor

O primeiro passo executado pela *SVM iteration* era o cálculo dos cinco produtos escalares indicados pelas Equações (40) e (41):

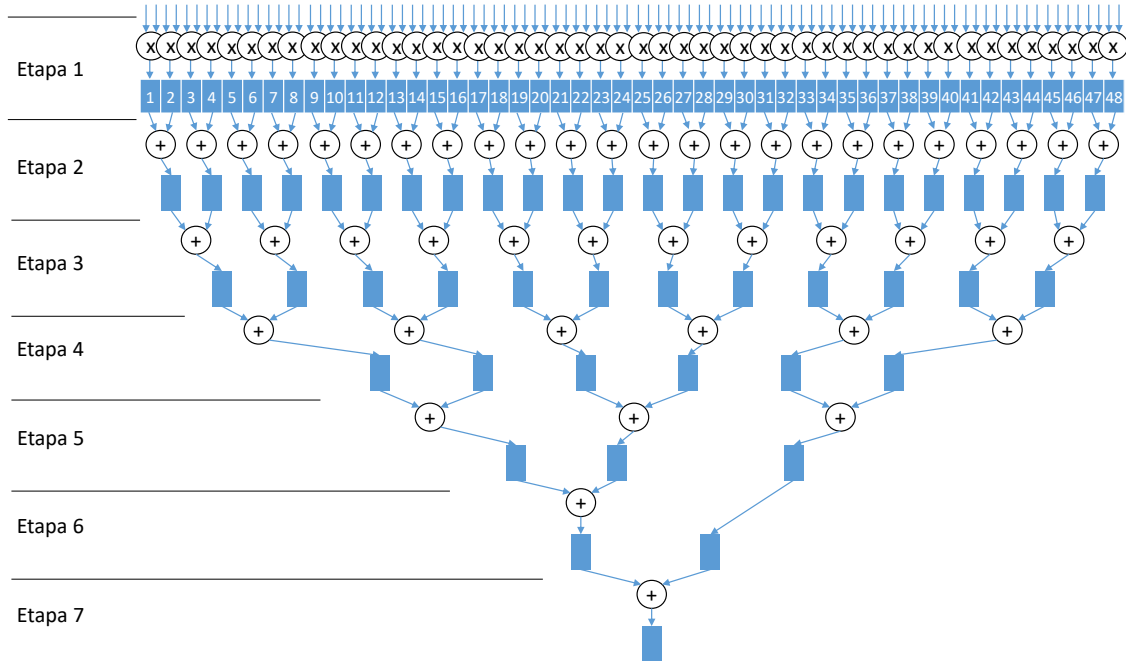
$$\begin{aligned}\omega x_1 &= \boldsymbol{\omega}^{old} \cdot \mathbf{x}_1 \\ \omega x_2 &= \boldsymbol{\omega}^{old} \cdot \mathbf{x}_2\end{aligned}\quad (40)$$

$$\begin{aligned}k_{11} &= \mathbf{x}_1^T \cdot \mathbf{x}_1 \\ k_{12} &= \mathbf{x}_1^T \cdot \mathbf{x}_2 \\ k_{22} &= \mathbf{x}_2^T \cdot \mathbf{x}_2\end{aligned}\quad (41)$$

sendo $\boldsymbol{\omega}^{old}$ os pesos do sistema calculados pela iteração anterior, e \mathbf{x}_1 e \mathbf{x}_2 as amostras de entrada para as quais o sistema está sendo treinado na atual iteração. As variáveis calculadas ωx_1 , ωx_2 , k_{11} , k_{12} e k_{22} são variáveis auxiliares para o cálculo da Equação (24). Para executar estas operações, 96 multiplicadores de 24×8 bits (ver Equação (40)) e 144 multiplicadores de 8×8 bits (ver Equação (41)) nativos da FPGA foram utilizados. Estes consomem recursos de DSP da FPGA e apresentam o resultado em um ciclo de *clock* (etapa

1 da Figura 11). As somas do produto escalar eram executadas aos pares em 6 etapas de um ciclo de *clock* cada (etapas 2 a 7 da Figura 11).

Figura 11 - Representação da operação multiplicação escalar para arquitetura em *pipeline*.



Fonte: do autor

Os multiplicadores de Lagrange eram então atualizados conforme indicado nas Equações (24) e (25). Sabia-se que as classes apresentavam apenas os valores +1 e -1, portanto, para resolver a Equação (24), calculou-se um numerador (NUM) e um denominador (DEN) que alimentavam um divisor conforme indicado na Equação (42):

$$\begin{aligned} \text{NUM} &= y_2(\omega x_1 - \omega x_2) + 1 - y_1 y_2 \\ \text{DEN} &= k_{11} - 2k_{12} + k_{22} \\ \Delta_2 &= \text{NUM}/\text{DEN} \end{aligned} \quad (42)$$

sendo y_1 e y_2 as classes das amostras de entrada, ωx_1 , ωx_2 , k_{11} , k_{12} e k_{22} as variáveis auxiliares calculadas pelas Equações (40) e (41), e Δ_2 a perturbação ideal indicada na Equação (24).

As divisões eram executadas em 34 ciclos de *clock* utilizando operações de comparação, soma e deslocamento, e as multiplicações por y_1 e y_2 apenas alteravam o sinal dos coeficientes e podiam ser trabalhadas com operações condicionais. Com Δ_2 calculado, resolvia-se a Equação (25) de acordo com o seguinte pseudocódigo:

$$\begin{aligned} A &= \alpha_1^{old} - \Delta_2; & B &= \alpha_2^{old} + \Delta_2 \\ C &= \alpha_1^{old} + \alpha_2^{old}; & D &= \alpha_1^{old} + \alpha_2^{old} - C \end{aligned}$$

Se $(B \geq 0)$ e $(B \leq C)$ e $(A \geq 0)$ e $(A \leq C)$	$\alpha_1^{new} = A$; $\alpha_2^{new} = B$
Se $(\Delta_2 < 0)$ e $(C \geq 0)$ e $(C \leq C)$	$\alpha_1^{new} = C$; $\alpha_2^{new} = 0$
Se $(\Delta_2 \geq 0)$ e $(D \geq 0)$ e $(D \leq C)$	$\alpha_1^{new} = D$; $\alpha_2^{new} = C$
Se $(\Delta_2 \geq 0)$ e $(C \geq 0)$ e $(C \leq C)$	$\alpha_1^{new} = 0$; $\alpha_2^{new} = C$
Se $(\Delta_2 < 0)$ e $(D \geq 0)$ e $(D \leq C)$	$\alpha_1^{new} = C$; $\alpha_2^{new} = D$
Se não	$\alpha_1^{new} = \alpha_1$; $\alpha_2^{new} = \alpha_2$

onde Δ_2 é a perturbação ideal calculada pela Equação (42), α_1^{old} e α_2^{old} são os multiplicadores de Lagrange calculados em iteração anterior, e α_1^{new} e α_2^{new} os multiplicadores de Lagrange atualizados.

O próximo passo era calcular a Equação (26) para obtenção dos novos pesos cuja operação era solucionada em apenas 6 ciclos de *clock* com as 48 posições do vetor ω sendo calculadas em paralelo conforme indicação da Equação (43):

$$\begin{aligned} \Delta_1 &= y_1 \mathbf{x}_1 (\alpha_1^{new} - \alpha_1^{old}) \\ \Delta_2 &= y_2 \mathbf{x}_2 (\alpha_2^{new} - \alpha_2^{old}) \\ \omega^{new} &= \omega^{old} + \Delta_1 + \Delta_2 \end{aligned} \quad (43)$$

sendo Δ_1 e Δ_2 perturbações que deviam ser aplicadas aos pesos (ω^{old}) devido à atualização dos valores dos multiplicadores de Lagrange (α_1 e α_2). As variáveis \mathbf{x}_1 e \mathbf{x}_2 são as entradas e y_1 e y_2 os correspondentes rótulos.

Em um ciclo de *clock* calculavam-se as duas subtrações ($\alpha_i^{new} - \alpha_i^{old}$) e outro era necessário para multiplicar o resultado por \mathbf{x}_i . Dois ciclos foram utilizados no tratamento de sinal e multiplicação por y_i e outros dois ciclos de *clock* foram necessários para o correspondente cálculo das duas somas finais.

Finalmente, o valor de *bias* devia ser atualizado para que o sistema continuasse satisfazendo as condições de Karush-Kuhn-Tucker (ver Equação (27)). Para isso, novamente foram utilizados os multiplicadores escalares 24×8 bits para determinar $\omega^{new} \cdot \mathbf{x}_1$ e $\omega^{new} \cdot \mathbf{x}_2$. Então, o seguinte pseudocódigo foi utilizado para se obter o novo valor do *bias*:

Se $y_1 = 1$	$b_1 = (\overline{\omega x_1})$	// $b_1 = -\omega x_1 - 1$
Se não	$b_1 = (\overline{\omega x_1}) + 2$	// $b_1 = -\omega x_1 + 1$
Se $y_2 = 1$	$b_2 = (\overline{\omega x_2})$	// $b_2 = -\omega x_2 - 1$
Se não	$b_2 = (\overline{\omega x_2}) + 2$	// $b_2 = -\omega x_2 + 1$

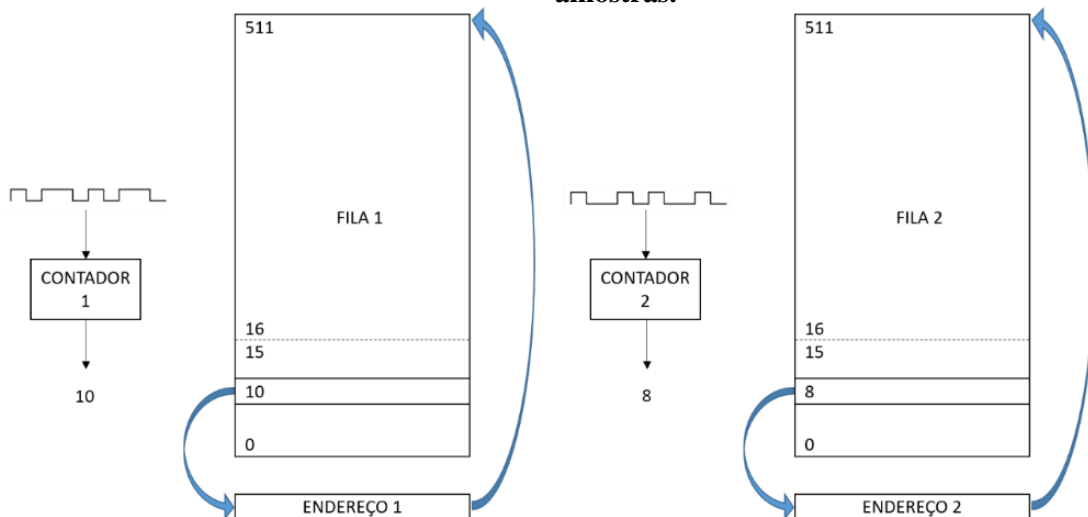
Se $\alpha_1 = 0$ e $\alpha_2 = 0$	$b = b$	// b inalterado
Se $\alpha_1 = C$ e $\alpha_2 = C$	$b = b$	// b inalterado
Se $\alpha_1 = C$ e $\alpha_2 = 0$	$b = (b_1 + b_2)/2$	
Se $\alpha_1 = 0$ e $\alpha_2 = C$	$b = (b_1 + b_2)/2$	
Se $\alpha_2 = 0$ ou $\alpha_2 = C$	$b = b_1$	
Se $\alpha_1 = 0$ ou $\alpha_1 = C$	$b = b_2$	
Se não	$b = b_1$	

Então os valores atualizados de α , ω e b eram guardados em memória, terminando uma iteração do algoritmo SMO.

O bloco *SVM COUNT* é um contador utilizado como referência para o escalonamento dos processos no bloco de treinamento. Incrementado a cada ciclo de 0_d a 137_d, sendo os 136 primeiros ciclos utilizados para leitura dos dados das memórias RAM e os últimos dois para gravação dos dados atualizados nas memórias. As saídas de *SVM iteration* eram armazenadas em registradores até o momento em que o escalonamento permitia sua gravação na memória.

O bloco *RAND ADDR*, indicado na Figura 12, consistiu de 17 filas, 17 contadores e 17 geradores de sinais de habilitação para os contadores para gerar endereços aleatórios e diferentes para cada uma das 17 memórias de amostras. Os endereços de memória ocupados por amostras eram colocados nas filas e cada vez que *SVM COUNT* era zerado, um endereço por movimento era escolhido para ser processado. A escolha das amostras era realizada através dos contadores, o endereço escolhido era o que se encontrava na posição indicada pelo contador e, a cada nova escolha de endereços, os endereços anteriores eram alocados no final das filas. Os sinais de habilitação dos contadores eram pulsos diferentes para cada movimento de modo a evitar que uma amostra de um movimento fosse comparada sempre com a mesma amostra do outro movimento.

Figura 12 - Representação gráfica da geração de endereços randômicos para escolha das amostras.



Fonte: do autor

Para garantir que todas as posições da fila fossem utilizadas, os contadores eram limitados de acordo com as seguintes diretrizes (ver a Tabela 3):

- 497 iterações eram executadas com os contadores limitados em 15_d (iteraões 0_d a 496_d);
- na iteração 497_d, os contadores eram limitados em 14_d, na 498_d em 13_d, e este padrão seguia até a iteração 511_d em que obrigatoriamente o primeiro endereço da fila era usado;
- quando 512 iterações tinham sido executadas, o limite voltava a ser 15_d e a contagem recomeça.

Tabela 3 - Limites para o contador do gerador de endereços.

Atualização da Fila	Valor Mínimo do Contador	Valor Máximo do Contador	Atualização da Fila	Valor Mínimo do Contador	Valor Máximo do Contador
0 _d -496 _d	0 _d	15 _d	505 _d	0 _d	6 _d
497 _d	0 _d	14 _d	506 _d	0 _d	5 _d
498 _d	0 _d	13 _d	507 _d	0 _d	4 _d
499 _d	0 _d	12 _d	508 _d	0 _d	3 _d
500 _d	0 _d	11 _d	509 _d	0 _d	2 _d
501 _d	0 _d	10 _d	510 _d	0 _d	1 _d
502 _d	0 _d	9 _d	511 _d	0 _d	0 _d
503 _d	0 _d	8 _d	512 _d	0 _d	15 _d
504 _d	0 _d	7 _d			

A saída do bloco de treinamento era composta dos pesos e *bias* do sistema para cada um das SVMs treinadas. A cada ciclo de *clock* um diferente conjunto de pesos e *bias* era apresentado juntamente com a indicação de qual SVM se referia. O tempo necessário para rodar uma época, ou seja, apresentar uma vez cada uma das amostras de treinamento ao sistema para cada SVM quando em capacidade máxima (512 amostras por movimento) demandava (ver Equação (44)):

$$\frac{138 \times 512}{125000000} = 565,248 \mu s \quad (44)$$

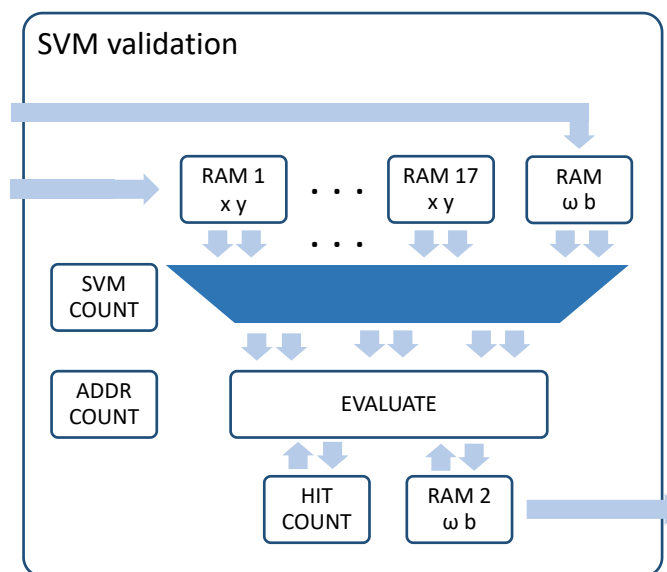
com *clock* de 125MHz desde que observado o tempo de latência. Devido à arquitetura em *pipeline*, para rodar uma época das 136 SVMs, necessitava-se apenas esperar mais 135 ciclos de *clock* após a primeira resposta, ou seja (ver Equação (45)):

$$\frac{138 \times 512 + 135}{125000000} = 566,328 \mu s \quad (45)$$

3.1.5 Implementação em VHDL do bloco de validação da SVM

Os parâmetros ω e b obtidos pelo algoritmo de treinamento eram validados no bloco *SVM validation*. O objetivo deste era garantir a generalização do sistema ao apresentar um conjunto diferente de entradas à SVM e escolher o conjunto de pesos e *bias* que melhor prevê as saídas. O bloco está representado pela Figura 13.

Figura 13 - Bloco em VHDL responsável pela validação dos parâmetros treinados pela SVM.



Fonte: do autor

As entradas do bloco de validação eram armazenadas em memórias de acordo com o movimento ao qual pertenciam da mesma forma que no bloco de treinamento – 17 memórias (RAM1 a RAM17 x y) com $256 \times 48 \times 8$ bits cada: 256 é a capacidade de armazenamento de entradas da memória (equivalente a 19,2 s), 48 é o número de características por amostra e 8 bits a resolução de cada característica. Outra entrada recebia os parâmetros ω e b das 136 SVMs continuamente, e estes também eram armazenados em RAM ($136 \times 48 \times 24$ bits para ω e 136×38 bits para b).

SVM COUNT era o contador que fazia a seleção de qual SVM estava sendo avaliada. Como no bloco de validação não ocorria a realimentação das memórias, não havia necessidade de reservar ciclos para escrita e, portanto, *SVM COUNT* contava de 0_d a 135_d . O bloco *ADDR COUNT* era o contador que varria todos os endereços das memórias de x e y de forma sequencial. As informações endereçadas pelos dois contadores alimentavam o avaliador *EVALUATE*. Este fazia a previsão da saída das SVMs binárias e comparava com a saída desejada conforme o pseudocódigo:

$$f(x_1) = \omega \cdot x_1 + b$$

$$f(x_2) = \omega \cdot x_2 + b$$

Se $y_1 = 1$ e $f(x_1) < 0$	HIT_COUNT(SVM)++
Se $y_1 = 0$ e $f(x_1) > 0$	HIT_COUNT(SVM)++
Se $y_2 = 1$ e $f(x_2) < 0$	HIT_COUNT(SVM)++
Se $y_2 = 0$ e $f(x_2) > 0$	HIT_COUNT(SVM)++

$f(x_i)$ era calculado com o auxílio de um multiplicador igual ao da Figura 11 e caso a previsão acertasse o movimento, o contador *HIT COUNT* era incrementado para a correspondente SVM.

Durante a execução do processo iterativo, o algoritmo de validação escolhia os parâmetros que mais acertavam os conjuntos de entradas e saídas, melhorando a taxa de acerto da SVM. Os parâmetros que geravam as maiores taxas de acerto eram armazenados na RAM 2 para serem utilizados na classificação de sinais. A saída do bloco de validação era composta dos pesos e *bias* guardados na RAM 2 para cada uma das SVMs treinadas. A cada ciclo de *clock* um diferente conjunto de pesos e *bias* era apresentado na saída juntamente com a indicação da SVM a qual se referia. O tempo necessário para validar uma SVM é 256 ciclos de *clock* quando em capacidade máxima e a latência deste bloco era 14 ciclos de *clock*. Para validar as 136 SVMs eram necessários (Equação (46)):

$$\frac{136 \times 256}{125000000} = 278,528 \mu s \quad (46)$$

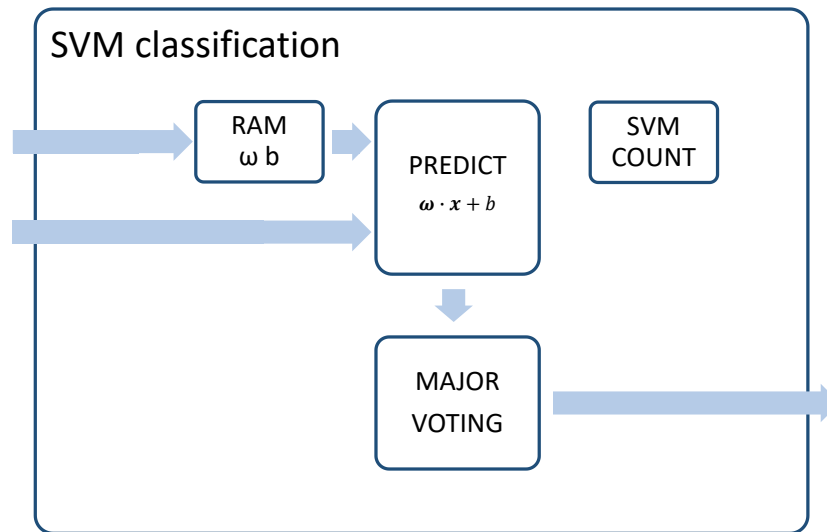
com *clock* de 125 MHz.

3.1.6 Implementação em VHDL da etapa de classificação da SVM

O classificador SVM era o bloco que utilizava os parâmetros escolhidos pelo algoritmo de validação para prever os movimentos executados pelo sujeito. Possuía duas entradas: uma ligada ao bloco de validação para recepção dos pesos e *bias* das SVMs; e uma ligada ao bloco de extração de características para recepção das características extraídas do sinal EMG durante a execução do movimento a ser previsto. As 48 características eram apresentadas às 136 SVMs, uma a cada ciclo de *clock* e o movimento previsto por maior número de SVMs era eleito como a saída do classificador.

A Figura 14 apresenta o diagrama de blocos do módulo desenvolvido para classificar os sinais de sEMG (*SVM classification*). Ele possuía um bloco de memória para armazenar os valores de pesos e *bias* recebidos (RAM ω b), multiplicador escalar e somador (*PREDICT*), contador de 0_d-135_d (*SVM COUNT*), e algoritmo para eleger o movimento previsto (*MAJOR VOTING*).

Figura 14 - Bloco em VHDL responsável pela classificação do sinal de sEMG.



Fonte: do autor

Com a presença de uma entrada de sinais de eletromiografia na entrada do bloco, o contador de SVM era incrementado a cada ciclo de *clock*. Este contador realizava a seleção dos pesos e *bias* que eram lidos da memória RAM, e com isso *PREDICT* calculava um movimento previsto para cada uma das 136 SVMs binárias. O movimento previsto era informado ao *MAJOR VOTING* que consistia em 17 contadores, um para cada movimento. O contador relacionado ao movimento previsto era incrementado e após todas as SVMs binárias preverem suas saídas, os contadores tinham o número de SVMs que previram cada um dos movimentos.

Duas situações podiam ocorrer quando avaliados os contadores:

- um contador contou mais do que todos os outros;
- havia empate de dois ou mais contadores.

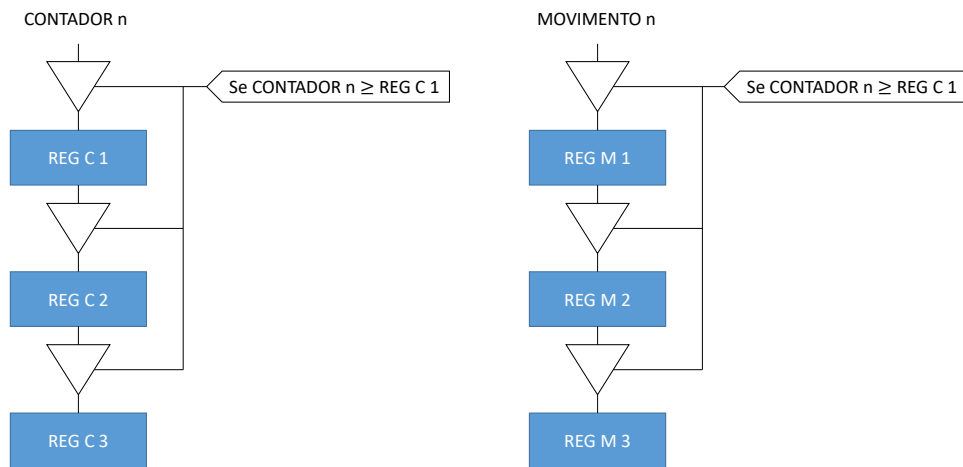
Considerando-se estas duas possibilidades, optou-se por utilizar três registradores para guardar os valores mais altos de contadores e mais três para os respectivos movimentos seguindo a seguinte regra (ver Figura 15):

1. os registradores eram iniciados contendo o primeiro movimento como vencedor, ou seja, REG C 1 continha o valor do contador 1 e REG M 1 continha o movimento 1;
2. comparava-se o valor do segundo contador com REG C1: se o valor do contador 2 era maior, REG C 3 recebia o valor de REG C 2, REG C 2 recebia o valor de REG C 1 e REG C1 recebia o valor do contador 2. Os registradores REG M x seguiam a mesma lógica;
3. as comparações seguiam até o último movimento: se o valor do contador n era maior que o valor de REG C1, REG C 3 recebia o valor de REG C 2, REG C 2 recebia o valor

de REG C 1 e REG C1 recebia o valor do contador n. Os registradores REG M x seguiam a mesma lógica;

- quando todos os movimentos tinham sido comparados, tinha-se como resultado o maior contador no primeiro registrador (REG C 1). No caso de empate de dois movimentos, o mesmo valor era encontrado no segundo registrador, com empate de três movimentos os registradores REG C 1, REG C 2 e REG C 3 tinham o mesmo valor armazenado. No caso de empate de quatro ou mais contadores, havia perda dessa informação.

Figura 15 - Lógica de registro dos contadores para eleição do movimento previsto.



Fonte: do autor

Então, não havendo empate, o movimento contido no primeiro registrador (REG M 1) era passado para a saída; no caso de empate, as SVMs que comparam os movimentos empatados decidiam a saída; e, se nenhum critério fosse satisfeito, o movimento escolhido era o movimento da última previsão:

Se $REG C 1 > REG C 2$
 MOVIMENTO = REG M 1

Se $REG C 2 > REG C 3$
 MOVIMENTO = SVM(REG M 1, REG M 2)

Se SVM(REG M 1, REG M 2) = REG M 1 e SVM(REG M 1, REG M 3) = REG M 1
 MOVIMENTO = REG M 1

Se SVM(REG M 1, REG M 2) = REG M 2 e SVM(REG M 2, REG M 3) = REG M 2
 MOVIMENTO = REG M 2

Se SVM(REG M 1, REG M 3) = REG M 3 e SVM(REG M 2, REG M 3) = REG M 3
 MOVIMENTO = REG M 3

Se não
 MOVIMENTO = MOVIMENTO

O tempo necessário para classificar um movimento foi 174 ciclos de clock, equivalente a 1,392 μ s.

3.1.7 Implementação do bloco de controle em VHDL

Uma interface serial do tipo UART (*Universal Asynchronous Receiver/Transmitter*) permitiu a configuração de parâmetros do sistema conectando-o a um computador. Através desta interface foi possível configurar por quanto tempo o treinamento era executado e valores de ganhos para extração de características, além de adquirir estatísticas da validação SVM. Consistia em um banco de registradores que podia ser acessado através de terminal de comando escrevendo-se “reeee” para leitura ou “waaaa vv” para escrita seguido do comando *enter* onde:

- r é o caractere r em ASCII e w o caractere w em ASCII;
- eeee: endereço de 16 bits, 4 caracteres ASCII na representação hexadecimal;
- vv: valor a ser escrito, 2 caracteres ASCII na representação hexadecimal.

3.2 DESCRIÇÃO DO PROTOCOLO DO EXPERIMENTO

3.2.1 Base de dados utilizada

O experimento para avaliação do sistema proposto foi realizado com o auxílio da base de dados Ninapro (ATZORI *et al.*, 2014) que contém sinais mioelétricos de 40 sujeitos não amputados e 11 sujeitos com amputação transradial capturados durante a execução de 17 movimentos básicos de dedos e pulso (denominado pelos autores de exercício B). Os sujeitos não amputados são homens e mulheres com idade entre 23 e 45 anos com Índice de Massa Corpórea (IMC) entre 18 e 31,5 kg/m² – um indício de que a bioimpedância destes indivíduos é adequada para a aquisição dos sinais. Executaram os movimentos utilizando a mão direita apesar de 34 serem destros e 6 canhotos. Os sujeitos amputados são todos do sexo masculino com idade entre 32 e 67 anos e IMC entre 21,9 e 29,3. Com relação à amputação podem ser observados diferentes graus de perda do antebraço, tempos desde a amputação, intensidades da sensação de membro fantasma, uso de diferentes tipos de próteses, entre outras variáveis (consultar a Tabela 9). Estes sujeitos executaram evidentemente os movimentos utilizando o membro amputado.

Tabela 4 - Descrição dos sujeitos amputados da base Ninapro.

Sujeito	Mão amputada	Lateralidade	Perda do antebraço	Tempo desde a amputação	SMF ¹	DASH ²	Uso de prótese cosmética	Uso de prótese mecânica	Uso de prótese mioelétrica
1	Direita	Destro	50%	13 anos	2	1,67	-	-	13 anos
2	Esquerda	Canhoto	30%	6 anos	5	15,18	6 anos	-	-
3	Direita	Destro	70%	5 anos	2	22,5	-	8 anos	8 anos
4	Direita	Destro	60%	1 ano	1	86,67	-	-	-
5	Esquerda	Canhoto	10%	1 ano	2	11,67	-	0,4 anos	-
6	Esquerda	Destro	60%	13 anos	4	37,5	-	12 anos	-
7	Direita	Destro	100%	7 anos	0	31,67	-	-	6 anos
8	Direita	Destro	50%	5 anos	2	33,33	-	-	4 anos
9	Direita	Destro	10%	14 anos	5	3,33	-	-	14 anos
10	Direita	Destro	50%	2 anos	5	11,67	-	1,7 anos	-
11	Direita	Destro	10%	5 anos	4	12,5	-	5 anos	5 anos

¹Intensidade da Sensação de Membro Fantasma.

²Pontuação DASH (*Disability of the Arm, Shoulder and Hand*)

Fonte: ATZORI et al., 2014

Os ensaios consistiam em os sujeitos reproduzirem os movimentos apresentados em uma tela de computador com 12 eletrodos ativos Delsys Trigno posicionados no antebraço (oito igualmente espaçados em torno do antebraço na altura da junta radio-umeral, um na região mais ativa do músculo *flexor digitorum superficialis* e outro na região mais ativa do extensor *digitorum superficialis*, um na região mais ativa do *biceps brachii* e outro na região mais ativa do *triceps brachii*). Cada movimento tinha duração de 5 segundos e era executado seis vezes por cada sujeito com um período de repouso de 3 segundos entre cada repetição. Durante a execução, os sinais mioelétricos foram capturados por um eletromiógrafo utilizando eletrodos não invasivos ativos com taxa de amostragem de 2 kHz e 16 bits de resolução, sendo suportado na entrada um sinal diferencial de até 11 mV que é readequado a faixa ± 5 V através de um ganho de 909V/V. Os movimentos estão apresentados visualmente na Figura 16 e descritos na Tabela 5.

Figura 16 - Movimentos de dedos e pulsos da base Ninapro.

Fonte: adaptado de (ATZORI et al., 2014)

Tabela 5 - Movimentos de dedos e pulsos da base Ninapro.

Nº do movimento	Movimento
1	polegar pra cima
2	extensão dos dedos indicador e médio e flexão dos demais
3	flexão dos dedos anelar e mínimo e extensão dos demais
4	polegar tocando base do dedo mínimo
5	abdução de todos os dedos
6	dedos flexionados em conjunto
7	apontando com indicador
8	adução de dedos estendidos
9	supinação do pulso (eixo: dedo médio)
10	pronação do pulso (eixo: dedo médio)
11	supinação do pulso (eixo: dedo mínimo)
12	pronação do pulso (eixo: dedo mínimo)
13	flexão do pulso
14	extensão do pulso
15	deslocamento radial do pulso
16	deslocamento ulnar do pulso
17	extensão do pulso com mão fechada
18	posição de repouso

Fonte: ATZORI et al., 2014

3.2.2 Procedimentos executados para avaliar o desempenho do sistema

Os sinais mioelétricos da base de dados foram readequados em Matlab (Versão 2016a) para utilizarem a resolução de 12 bits esperada pelo sistema e gravados em um cartão micro SD antes dos ensaios serem iniciados. Dois arquivos por sujeito foram armazenados, um com quatro repetições dos 17 movimentos para aprendizado (1^a, 2^a, 4^a e 6^a repetição), e outro com as outras duas repetições para classificação (3^a e 5^a). Também em Matlab (Versão 2016a), foram calculados os ganhos que deviam ser configurados no módulo de extração de características para cada um dos sujeitos. Com estas informações obtidas de forma *off-line* disponíveis, os ensaios foram executados da seguinte forma:

- escolhia-se aleatoriamente um sujeito para avaliação;
- configurava-se o ARDUINO para alimentar o sistema com o sujeito escolhido;
- gravava-se a FPGA;
- configurava-se a FPGA com os valores de ganho calculados em Matlab (Versão 2016a) para aquele sujeito e quanto tempo duraria o treinamento;
- iniciava-se a apresentação do arquivo de treinamento ao sistema;

- esperava-se o tempo suficiente para terminar o treinamento e então o arquivo para classificação era apresentado;
- durante a classificação tanto a taxa de acerto observada na validação quanto o movimento previsto pelo sistema eram armazenados para avaliação.

O sistema foi treinado quatro vezes para cada sujeito conforme os ensaios listados na Tabela 6. Os dois primeiros treinamentos duraram 75 milissegundos e dez segundos respectivamente, e foram utilizados para classificar os 17 movimentos citados na Seção 3.2.1. O tempo de 75 ms é o tamanho do passo utilizado no janelamento, ou seja, tempo entre duas amostras disponibilizados ao classificador. Este tempo é suficiente para 132 épocas de treinamento. Os resultados obtidos com o sistema treinado por 10 s tinham como objetivo apresentar a resposta do sistema estabilizado.

Tabela 6 - Ensaios realizados.

Ensaio	Nº de movimentos	Tempo de treinamento
1	17	75 ms
2	17	10 s
3	10	75 ms
4	10	10 s

A partir destes dois primeiros ensaios, um conjunto menor de movimentos foi escolhido para se realizar o terceiro e quarto ensaios. Novamente o sistema foi treinado por 75 ms e 10 s, mas desta vez para classificar 10 movimentos. Os movimentos foram escolhidos por eliminação baseada nas métricas *sensitivity* e preditividade positiva calculadas com o sistema estabilizado (ensaio 2):

1. foram somadas as duas métricas (*sensitivity* e preditividade positiva) calculadas para cada movimento e eliminado o movimento de valor mais baixo;
2. os valores das métricas foram recalculados com um movimento a menos;

Os passos 1 e 2 foram repetidos até sobraem 10 movimentos.

Além das métricas *sensitivity* e preditividade positiva, foram utilizados o Erro Quadrático Médio (EQM) e a taxa de acerto (*accuracy*) para avaliar o desempenho do sistema. As Equações (47) a (50) apresentam o cálculo destas quatro métricas de desempenho:

$$EQM = \frac{1}{N} \sum_{i=1}^N e_i^2 \quad (47)$$

$$accuracy = \frac{VP + VN}{VP + VN + FP + FN} \quad (48)$$

$$\text{valor preditivo positivo} = \frac{VP}{VP + FP} \quad (49)$$

$$\text{sensitivity} = \frac{VP}{VP + FN} \quad (50)$$

Quando se avaliam as SVMs binárias no bloco de validação, o EQM é calculado sobre o erro dessas 136 SVMs, portanto o N da Equação (47) é 136 neste caso. Quando a avaliação é sobre os erros na previsão dos movimentos na etapa de classificação, o parâmetro N pode ser 17 ou 10 dependendo do número total de movimentos sendo analisado. Da mesma forma, a variável e da Equação (47) é o erro de validação das SVMs binárias ou o erro de classificação dos movimentos. As variáveis VP , VN , FP e FN das Equações (48) a (50) são os Verdadeiros Positivos, Verdadeiros Negativos, Falsos Positivos e Falsos Negativos obtidos na classificação dos movimentos.

3.2.3 Método utilizado para validação estatística

Para validar estatisticamente os resultados obtidos para os ensaios, um projeto de experimentos do tipo fatorial completo foi proposto. Partiu-se do pressuposto de que a base de dados Ninapro não viola as premissas desse projeto cujos elementos estão descritos nas Tabela 7. Deseja-se comparar a influência do sujeito que executa o movimento, movimento executado e tempo de treinamento do classificador na taxa de acerto (variáveis controladas). A divisão em dois blocos deve-se ao fato do experimento ser realizado em dois momentos, o primeiro com 17 movimentos e o segundo com 10 movimentos.

Tabela 7 - Projeto do experimento.

Elemento	Nome	Níveis	Valores
Variáveis controladas	Sujeito	51	1 a 51
	Movimento	17	1 a 17
	Tempo	2	75 ms e 10 s
Bloco	Nº de Movimentos	2	17 movimentos e 10 movimentos
Variável de resposta	Taxa de acerto	-	-

O modelo do Projeto de Experimentos segue a Equação (51).

$$y_{ijkl} = \mu + \alpha_i + \beta_j + \gamma_k + \delta_l + (\beta\gamma)_{jk} + (\beta\delta)_{jl} + (\gamma\delta)_{kl} + (\beta\gamma\delta)_{jkl} + \varepsilon_{ijkl} \quad (51)$$

onde y corresponde à resposta relacionada ao sujeito j executando o movimento k no treinamento l e bloco i . A média geral está representada por μ e o efeito relacionado aos blocos

está representado por α_i . As variáveis β_j , γ_k e δ_l representam os efeitos de cada nível de cada variável controlada e ε_{ijkl} corresponde ao erro. Os termos restantes representam os efeitos das interações entre fatores.

4 RESULTADOS E DISCUSSÕES

4.1 VALIDAÇÃO DO SISTEMA EMBARCADO

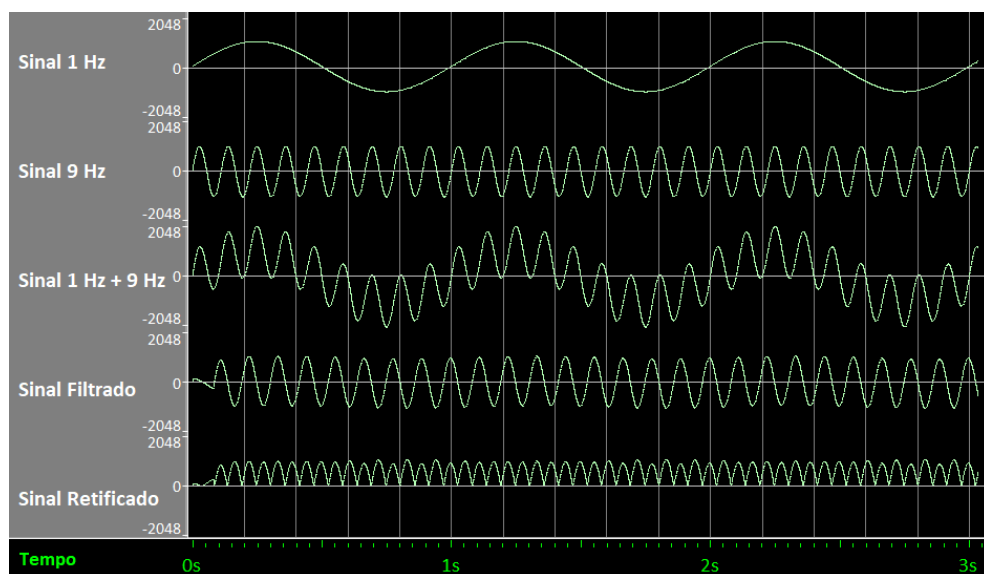
Cada uma das partes do sistema embarcado foi validada individualmente antes da integração ao sistema. Os testes executados e seus resultados são descritos na sequência.

4.1.1 Validação do filtro FIR

O filtro digital foi validado em simulação através da aplicação de senoides de diferentes frequências na sua entrada e análise da resposta em dois pontos: após o filtro FIR e após a retificação. As senoides escolhidas foram: senoide com 1 Hz e 400 Hz que estão fora da banda de passagem e devem ser rejeitados, e senoide com 9 Hz e 300 Hz que estão no limite da banda e não devem sofrer atenuação.

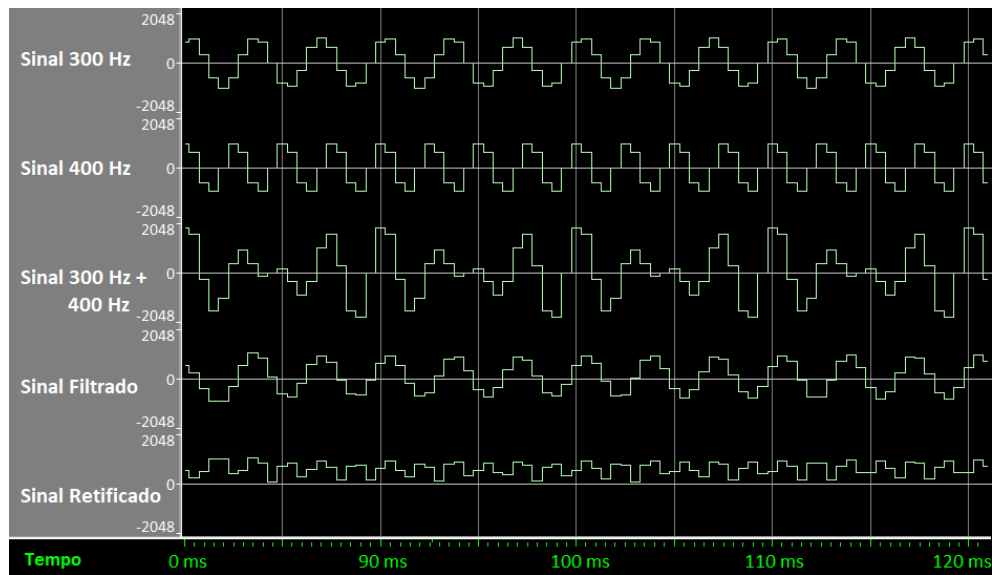
Para facilitar a visualização, primeiramente foi inserido um sinal senoidal com as frequências de 1 Hz e 9 Hz sobrepostas, e depois as frequências de 300 Hz e 400 Hz sobrepostas. A Figura 17 apresenta a composição do primeiro sinal de entrada com as componentes de frequência separadas e somadas, as correspondentes respostas do filtro FIR e da etapa de retificação. A Figura 18 segue o mesmo padrão tendo o segundo sinal como entrada.

Figura 17 - Simulação do filtro digital com entrada composta pelas senoides de 1 Hz e 9 Hz.



Fonte: do autor

Figura 18 - Simulação do filtro digital com entrada composta pelas senoides de 300 Hz e 400 Hz.



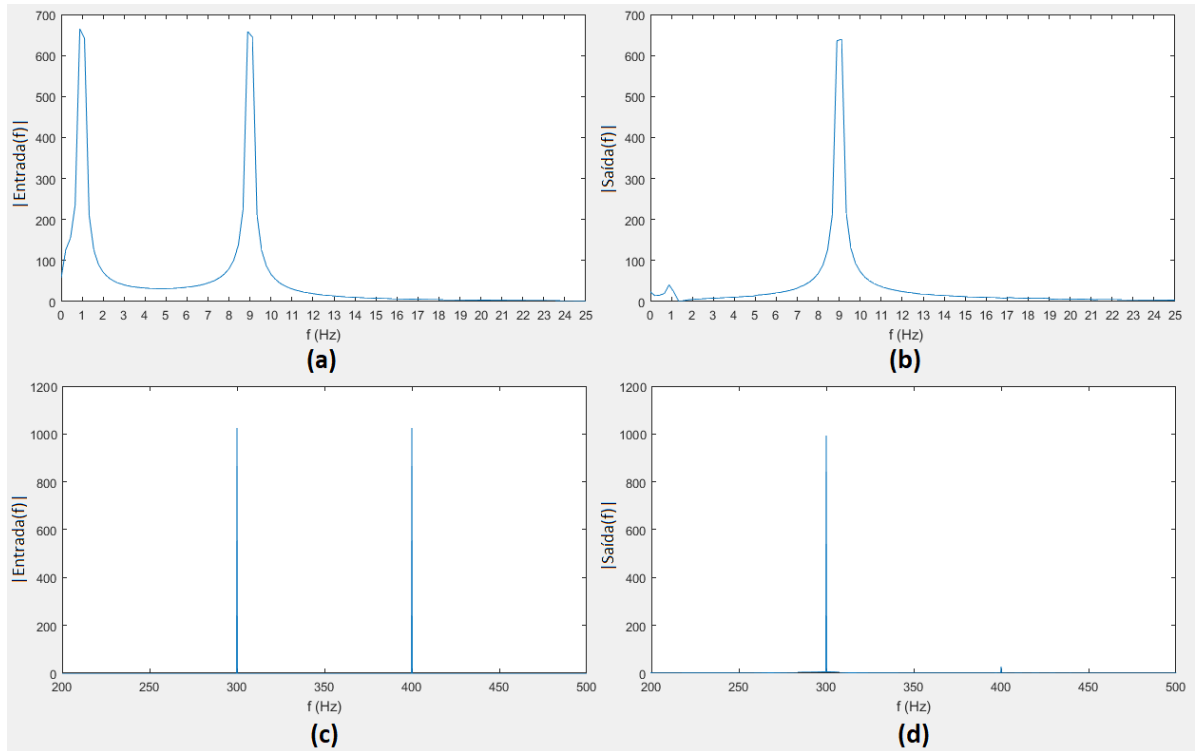
Fonte: do autor

Como é perceptível nas Figuras 17 e 18, o filtro FIR eliminou as componentes senoidais com frequência de 1 Hz e com frequência de 400 Hz ao apresentar no sinal filtrado da Figura 17 uma senoide de 9 Hz, e no sinal filtrado da Figura 18 uma senoide de 300 Hz. A análise em frequência dos sinais utilizados nos testes do filtro é apresentada na Figura 19. Nesta análise pode-se perceber que a componente de 1 Hz presente na entrada do filtro (Figura 19a) foi atenuada em 24 dB pelo filtro e a componente de 9 Hz não sofreu atenuação (Figura 19b). Quando as senoides de 300 Hz e 400 Hz foram apresentadas ao filtro FIR (Figura 19c), o filtro atenuou em 30 dB a componente de 400 Hz (Figura 19d), e não atenuou a componente de 300 Hz, comprovando o seu correto funcionamento.

Após os testes com sinais gerados em simulação, executou-se um teste com sinal de sEMG na entrada. Foi escolhido o conjunto de doze canais com os sinais de eletromiografia adquiridos durante a execução do movimento de flexão do pulso (movimento 13) pelo primeiro sujeito não-amputado da base de dados Ninapro. A

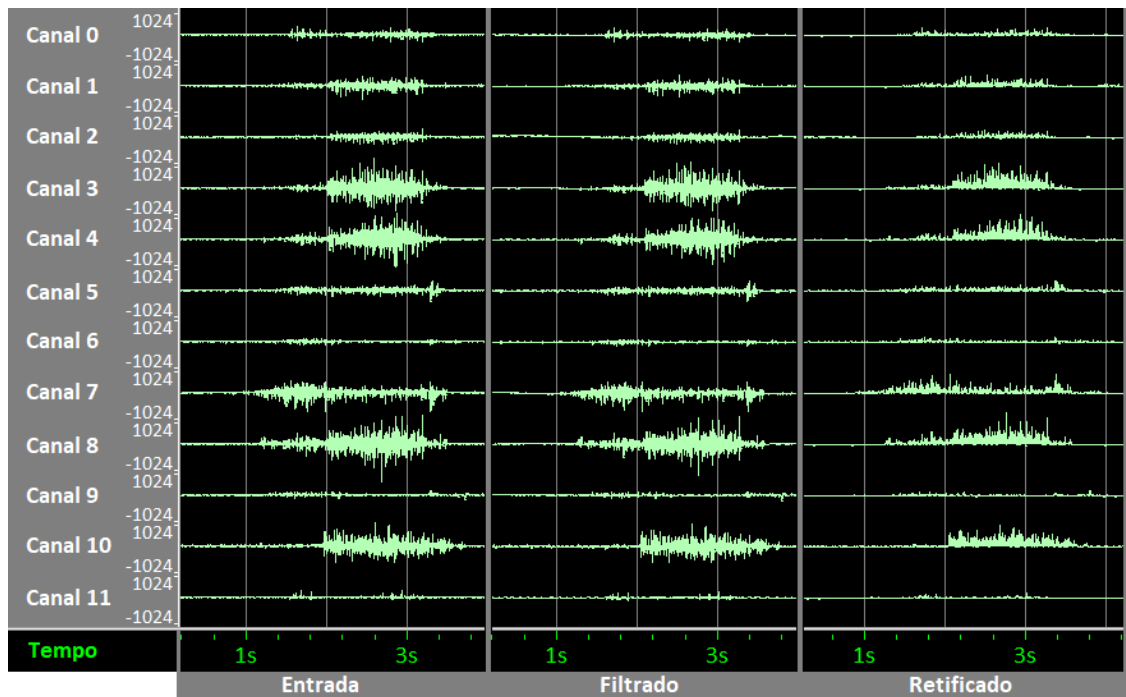
Figura 20 apresenta três janelas obtidas no mesmo período de tempo que ilustram os doze sinais de entrada, sinais após filtro FIR e sinais após retificação.

Figura 19 - Resposta em frequência do filtro FIR: (a) senoides de 1 Hz e 9 Hz na entrada do filtro, (b) saída do filtro para esta entrada, (c) senoides de 300 Hz e 400 Hz na entrada do filtro, (d) saída do filtro para esta entrada.



Fonte: do autor

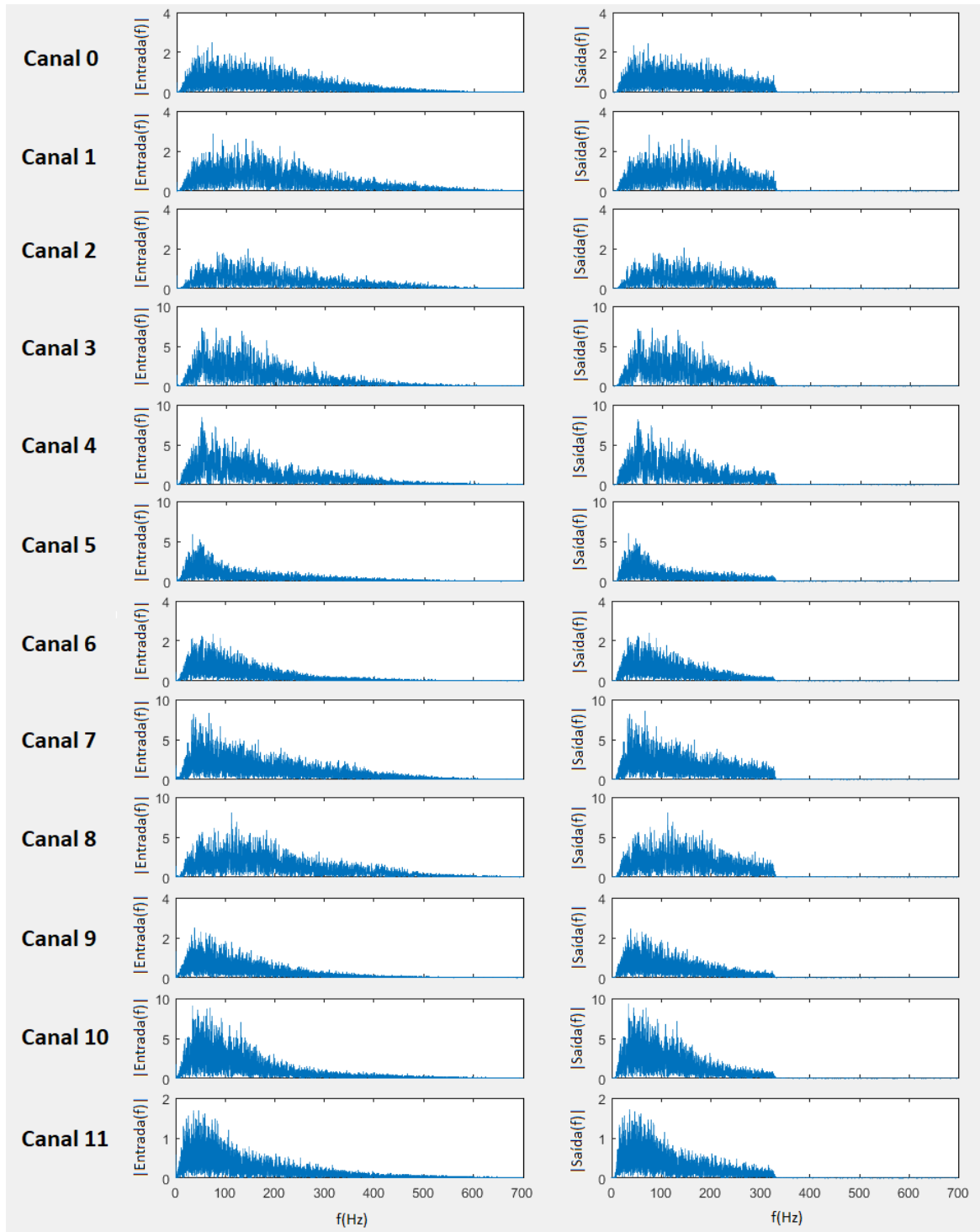
Figura 20 - Resposta do filtro digital com os sinais de sEMG do primeiro sujeito não amputado da base de dados Ninapro executando o movimento 13.



Fonte: do autor

Através da aplicação da transformada de Fourier destes sinais (ver Figura 21) pode-se perceber que o filtro elimina as frequências acima de 330 Hz e as frequências abaixo de 1 Hz. Portanto, o filtro funcionou como especificado, não atenuou os sinais na faixa de 9 Hz a 300 Hz.

Figura 21 - Resposta em frequência do filtro FIR com os sinais de sEMG do primeiro sujeito não amputado da base de dados Ninapro executando o movimento 13.



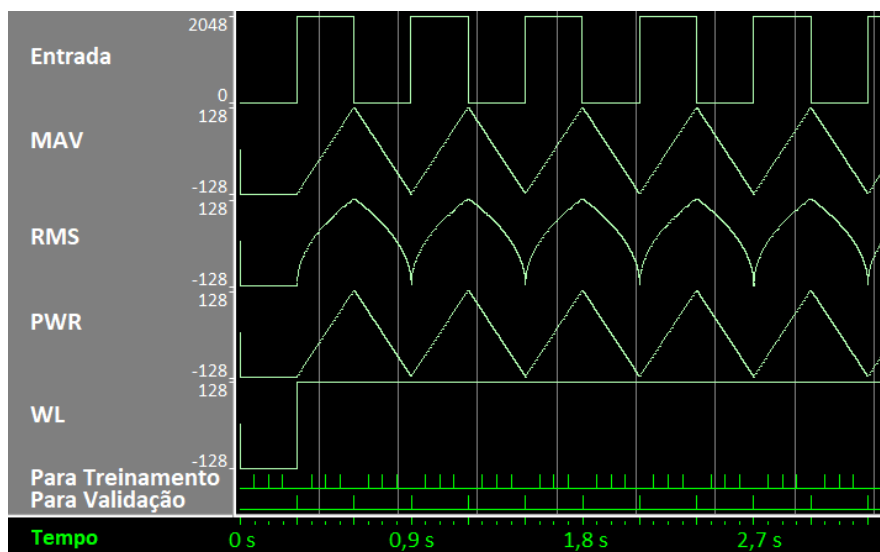
Fonte: do autor

4.1.2 Validação do bloco de extração de características

A extração de características foi validada em simulação através da aplicação de dois padrões diferentes de sinal na entrada do bloco para verificar sua resposta:

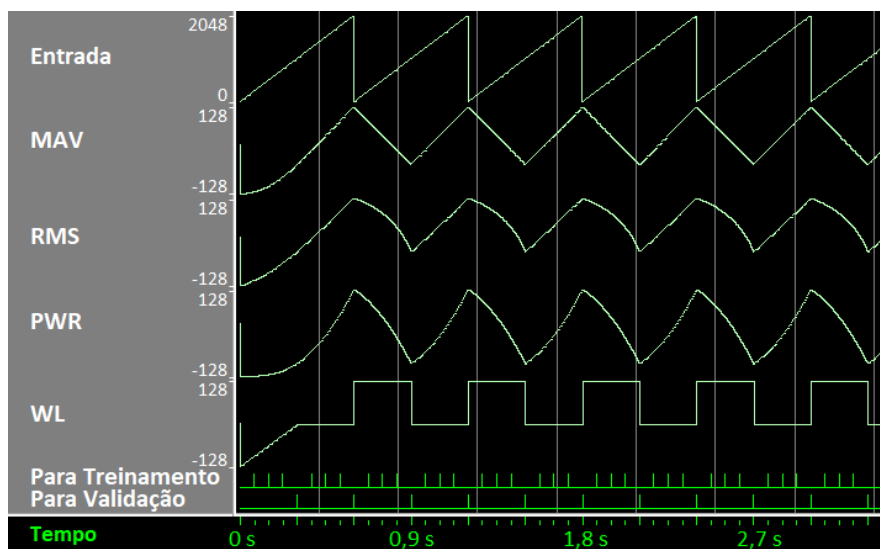
- onda quadrada com período igual a duas vezes o tamanho da janela utilizada para extração de características (600 ms) (ver Figura 22);
- onda dente de serra com período duas vezes maior que a janela para extração de características (600 ms) (ver Figura 23).

Figura 22 - Simulação da extração de características de onda quadrada.



Fonte: do autor

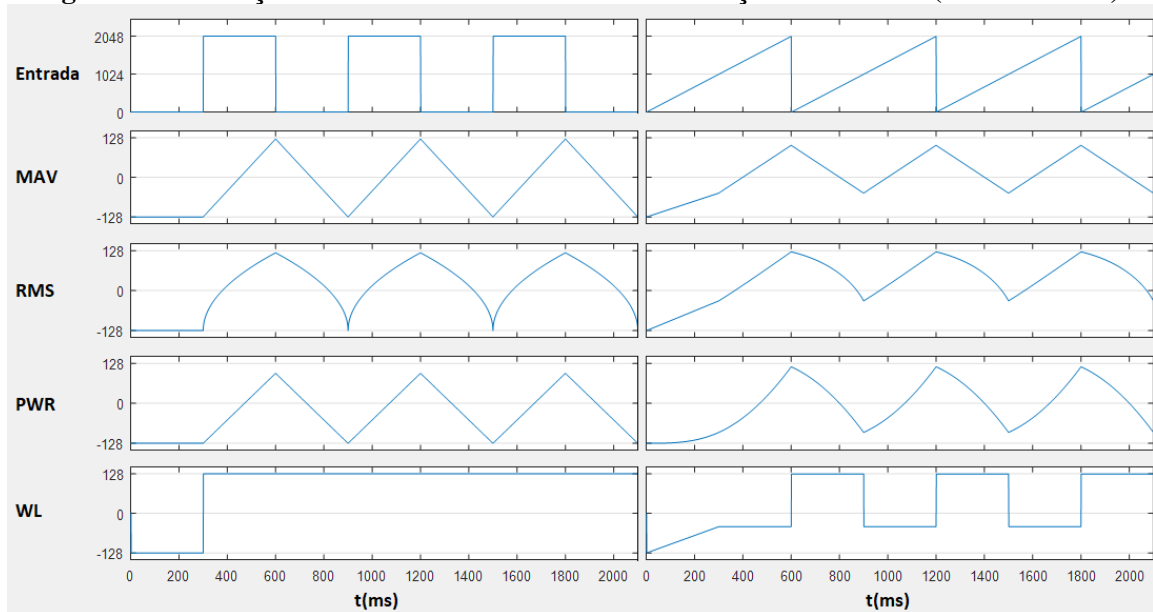
Figura 23 - Simulação da extração de características de dente de serra.



Fonte: do autor

Esses mesmos sinais foram reproduzidos em Matlab (Versão 2016a) e as características foram extraídas com funções disponíveis neste *software* para validar o comportamento do extrator de características por comparação (ver Figura 24). Os ganhos das funções de extração de características foram configurados para que as quatro características extraídas utilizassem o máximo da faixa de saída tanto em VHDL (bloco de extração de características sendo validado) quanto em Matlab (Versão 2016a).

Figura 24 - Extração de características utilizando as funções do Matlab (Versão 2016a).



Fonte: do autor

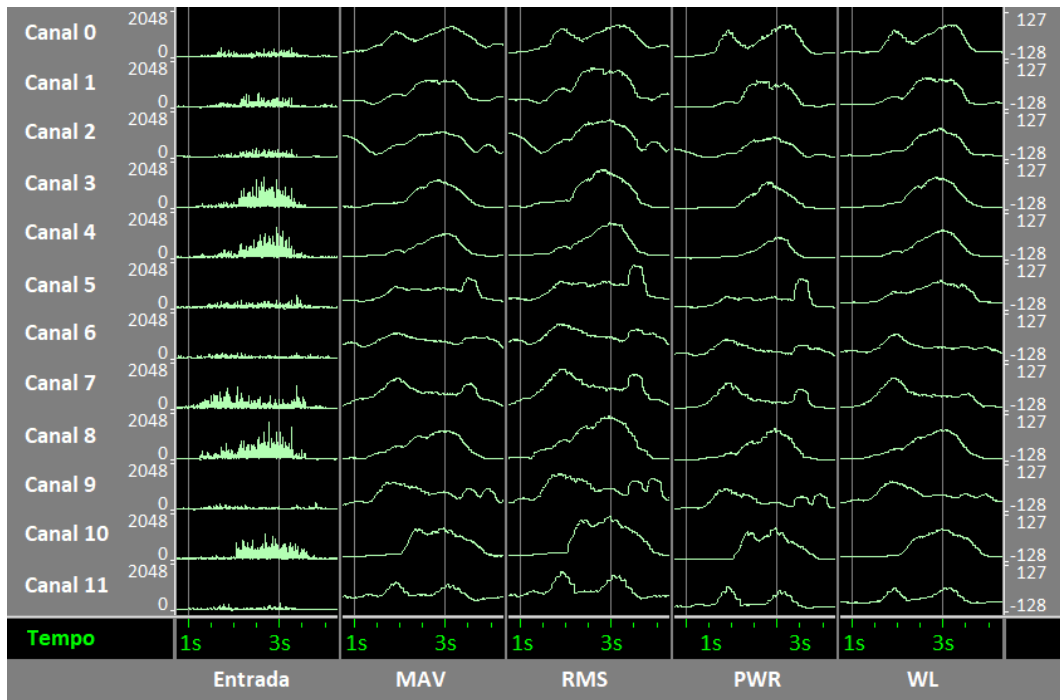
As características apresentadas nas Figuras 22 e 23 são resultado da simulação do bloco de extração de características descrito em VHDL e são iguais às obtidas no *software* Matlab (Versão 2016a) para os mesmos sinais de entrada (Figura 24). Dessa forma, comprovou-se o funcionamento do bloco. Os sinais para treinamento e para validação apresentados nas simulações representam os momentos de amostragem com passo de 75 ms: a cada três amostras para treinamento envia-se uma para validação.

Após a simulação com as formas de onda quadrada e dente de serra, os sinais de sEMG filtrados pelo filtro FIR (Seção 4.1.1) foram inseridos na entrada do bloco de extração de características para verificar a resposta obtida com os sinais mioelétricos. Este teste está representado na Figura 25, onde cinco janelas apresentam os canais de entrada e as quatro características extraídas deles. Os ganhos foram configurados de forma a normalizarem os sinais de entrada utilizando toda a faixa de saída (-128_d a 127_d).

O canal 11 da Figura 25, por exemplo, apresentou na entrada do bloco de extração de características nível máximo de 348_d enquanto que o canal 8 apresentou nível máximo de 2039_d.

A normalização aplicada aos sinais de entrada fez com que os sinais de saída apresentassem todos a mesma amplitude máxima. Portanto o bloco de extração de características, além de extrair as características MAV, RMS, PWR e WL dos sinais de sEMG, permitiu a normalização destes sinais, uma prática comum em sistemas de classificação de sinais de sEMG.

Figura 25 - Respostas da Extração de Características com os sinais de sEMG do primeiro sujeito não amputado da base de dados Ninapro executando o movimento 13.



Fonte: do autor

Para gerar a Figura 25 foram utilizados os ganhos apresentados na Tabela 8. Ao se avaliar outros movimentos, mais movimentos ou outros sujeitos, outros ganhos devem ser utilizados para que não ocorra saturação e para que a normalização respeite os sinais em análise.

Tabela 8 - Ganhos utilizados para gerar Figura 25.

Canal	Ganho(MAV)	Ganho(RMS)	Ganho(PWR)	Ganho(WL)
0	70	0,00829	0,00073	24,5
1	104,5	0,01105	0,00183	43
2	86	0,00691	0,00134	36,5
3	299	0,08770	0,01416	103,5
4	322,5	0,09737	0,01647	102,5
5	103	0,01174	0,00207	31
6	63,5	0,00414	0,00073	19,5
7	164,5	0,03038	0,00524	67,5
8	258	0,06008	0,01037	110
9	51	0,00276	0,00049	15
10	227	0,05110	0,00842	70
11	51	0,00276	0,00049	15

4.1.3 Validação do bloco de ganho do sistema

Os 48 canais com características dos sinais sEMG obtidos do primeiro sujeito não-amputado da base de dados Ninapro foram utilizados para verificação do funcionamento do bloco de ganho. Após a apresentação de 17 movimentos ao algoritmo (ver Seção 3.1.3), obtiveram-se os ganhos apresentados na Tabela 9.

Tabela 9 - Ganhos calculados para os 17 movimentos executados pelo primeiro sujeito não-amputado da base de dados Ninapro.

Movimento	Ganho	Movimento	Ganho
1	2,500	10	1,500
2	2,125	11	1,125
3	1,500	12	1,250
4	1,000	13	1,000
5	1,500	14	1,000
6	1,000	15	1,000
7	2,500	16	1,000
8	1,125	17	1,000
9	1,000		

Isso significa que é possível aplicar um ganho de 2,5 em todos os sinais de entrada do classificador sem saturar os sinais do movimento 1, de 2,125 sem saturar os sinais do movimento 2, e assim por diante. Portanto, para a SVM binária que diferencia o movimento 1 do movimento 2 deste exemplo (ver primeira linha da Tabela 10), é aplicado o menor destes ganhos, 2,125, garantindo que se houver saturação o movimento na entrada não é do movimento 1 nem do movimento 2 e pode ser classificado erroneamente.

A Tabela 10 apresenta as 136 SVMs binárias em sua primeira coluna, os movimentos que elas aprendem nas segunda e terceira colunas, e o ganho obtido pelo algoritmo para o sujeito 1 da base de dados de não amputados na quarta coluna. Pode-se perceber que todas as SVMs que possuem um dos movimentos 4, 6, 9, 13, 14, 15, 16 ou 17 na entrada obtiveram ganho unitário, respeitando a Tabela 9.

A influência do ganho na taxa de acerto foi avaliada em conjunto com os blocos de treinamento e validação. Essa análise está na Seção 4.1.5.

Tabela 10 - Ganhos calculados para as 136 SVMs binárias para o primeiro sujeito não-amputado da base de dados Ninapro.

SVM	Movimento 1	Movimento 2	Ganho	SVM	Movimento 1	Movimento 2	Ganho
1	1	2	2,125	69	3	13	1
2	1	3	1,5	70	4	13	1
3	2	3	1,5	71	5	13	1
4	1	4	1	72	6	13	1
5	2	4	1	73	7	13	1
6	3	4	1	74	8	13	1
7	1	5	1,5	75	9	13	1
8	2	5	1,5	76	10	13	1
9	3	5	1,5	77	11	13	1
10	4	5	1	78	12	13	1
11	1	6	1	79	1	14	1
12	2	6	1	80	2	14	1
13	3	6	1	81	3	14	1
14	4	6	1	82	4	14	1
15	5	6	1	83	5	14	1
16	1	7	2,5	84	6	14	1
17	2	7	2,125	85	7	14	1
18	3	7	1,5	86	8	14	1
19	4	7	1	87	9	14	1
20	5	7	1,5	88	10	14	1
21	6	7	1	89	11	14	1
22	1	8	1,125	90	12	14	1
23	2	8	1,125	91	13	14	1
24	3	8	1,125	92	1	15	1
25	4	8	1	93	2	15	1
26	5	8	1,125	94	3	15	1
27	6	8	1	95	4	15	1
28	7	8	1,125	96	5	15	1
29	1	9	1	97	6	15	1
30	2	9	1	98	7	15	1
31	3	9	1	99	8	15	1
32	4	9	1	100	9	15	1
33	5	9	1	101	10	15	1
34	6	9	1	102	11	15	1
35	7	9	1	103	12	15	1
36	8	9	1	104	13	15	1
37	1	10	1,5	105	14	15	1
38	2	10	1,5	106	1	16	1
39	3	10	1,5	107	2	16	1
40	4	10	1	108	3	16	1
41	5	10	1,5	109	4	16	1
42	6	10	1	110	5	16	1
43	7	10	1,5	111	6	16	1
44	8	10	1,125	112	7	16	1
45	9	10	1	113	8	16	1
46	1	11	1,125	114	9	16	1

Continuação da Tabela 10.

SVM	Movimento 1	Movimento 2	Ganho	SVM	Movimento 1	Movimento 2	Ganho
47	2	11	1,125	115	10	16	1
48	3	11	1,125	116	11	16	1
49	4	11	1	117	12	16	1
50	5	11	1,125	118	13	16	1
51	6	11	1	119	14	16	1
52	7	11	1,125	120	15	16	1
53	8	11	1,125	121	1	17	1
54	9	11	1	122	2	17	1
55	10	11	1,125	123	3	17	1
56	1	12	1,25	124	4	17	1
57	2	12	1,25	125	5	17	1
58	3	12	1,25	126	6	17	1
59	4	12	1	127	7	17	1
60	5	12	1,25	128	8	17	1
61	6	12	1	129	9	17	1
62	7	12	1,25	130	10	17	1
63	8	12	1,125	131	11	17	1
64	9	12	1	132	12	17	1
65	10	12	1,25	133	13	17	1
66	11	12	1,125	134	14	17	1
67	1	13	1	135	15	17	1
68	2	13	1	136	16	17	1

4.1.4 Validação da etapa de treinamento da SVM

A validação do bloco de treinamento foi dividida em duas etapas: primeiro, conjuntos de entradas criados de forma a testar diferentes situações foram apresentados ao *SVM iteration* para verificar se o treinamento é executado corretamente; depois, em conjunto com o bloco de validação, foi verificada a convergência da taxa de acerto das SVMs. A Tabela 11 apresenta as entradas do bloco *SVM iteration* e correspondentes saídas obtidas após uma iteração de treinamento (primeira etapa).

As classes y_1 e y_2 foram fixadas em 1 e -1 respectivamente e as amostras de entrada \mathbf{x}_1 e \mathbf{x}_2 são dois vetores com apenas o primeiro valor diferente de zero no primeiro teste (-16_d e 8_d). O vetor de pesos e valores de *bias* (ω^{old} e b^{old}) foram escolhidos para que na primeira linha as previsões p_1^{old} e p_2^{old} estivessem erradas, na segunda apenas p_2^{old} estivesse errada, na terceira apenas p_1^{old} e na quarta nenhuma errada. Nas linhas seguintes foram testados os limites dos multiplicadores de Lagrange (α_1^{old} e α_2^{old}), 0 e C (127_d). Dessa forma foram testadas as principais situações que o bloco *SVM iteration* deve resolver.

Tabela 11 - Teste do algoritmo de treinamento com padrões gerados em simulação.

ω^{old}	b^{old}	α_1^{old}	α_2^{old}	p_1^{old}	p_2^{old}	ω^{new}	b^{new}	α_1^{new}	α_2^{new}	p_1^{new}	p_2^{new}
(64,0,0,...,0)	0	64	64	-1	1	(80,0,0,...,0)	1281	62	66	1	1
(-64,0,0,...,0)	2048	64	64	1	1	(-80,0,0,...,0)	-1279	66	62	1	-1
(-64,0,0,...,0)	-2048	64	64	-1	-1	(-80,0,0,...,0)	-1279	66	62	1	-1
(-64,0,0,...,0)	0	64	64	1	-1	(-64,0,0,...,0)	0	64	64	1	-1
(-64,0,0,...,0)	2048	127	64	1	1	(-64,0,0,...,0)	2048	127	64	1	-1
(-64,0,0,...,0)	2048	0	64	1	1	(-80,0,0,...,0)	-1279	2	62	1	1
(-64,0,0,...,0)	2048	64	127	1	1	(-80,0,0,...,0)	-1279	66	125	1	-1
(-64,0,0,...,0)	2048	64	0	1	1	(-80,0,0,...,0)	2048	64	0	1	-1

$$y_1 = 1; y_2 = -1; \mathbf{x}_1 = (-16,0,0,\dots,0); \mathbf{x}_2 = (8,0,0,\dots,0)$$

¹Todos os valores estão na base decimal.

Após o treinamento de cada linha, os novos pesos (ω^{new}), *bias* (b^{new}) e multiplicadores (α_1^{new} e α_2^{new}) geraram novas previsões para as classes (p_1^{new} e p_2^{new}). Verificou-se que o treinamento da primeira linha passou a acertar uma previsão ($p_1^{new} = y_1$), da segunda e terceira passaram a acertar as duas previsões ($p_1^{new} = y_1$ e $p_2^{new} = y_2$) e na quarta não houve alteração ($p_1^{new} = p_1^{old} = y_1$ e $p_2^{new} = p_2^{old} = y_2$). Nos testes dos limites dos multiplicadores, verificou-se que quando α precisa ser atualizado de forma a violar as restrições (6ª linha da Tabela 11), não há alteração nos valores de saída; no outro caso (5ª, 7ª e 8ª linhas da Tabela 11), o treinamento ocorreu com sucesso.

A análise com sinais de sEMG foi feita em conjunto com o bloco de validação para que se pudesse observar a convergência da taxa de acerto. Essa análise está na Seção 4.1.5.

4.1.5 Validação do bloco de validação da SVM

Com a confirmação do funcionamento do algoritmo SMO implementado, o bloco de treinamento foi utilizado para aprender a diferenciar movimentos dois a dois, ou seja, para funcionar como um classificador binário de sinais de sEMG disponíveis na entrada. Então, o bloco de validação SVM foi utilizado para selecionar os parâmetros que apresentavam melhora na taxa de acerto. Para este teste foram utilizadas, como exemplo, as 66% das amostras dos sinais mioelétricos do primeiro sujeito não-amputado da base de dados Ninapro.

A Figura 26 apresenta os acertos contabilizados por *HIT_COUNT* para as 20 primeiras SVMs em simulação. Pode-se observar que o número de acertos cresce à medida que o treinamento é executado e os parâmetros que melhor preveem as saídas para as amostras de validação são selecionados.

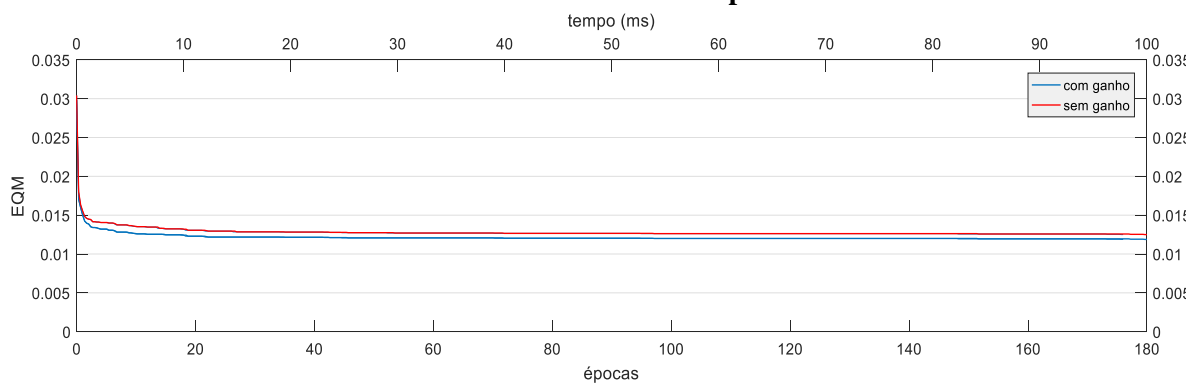
Figura 26 - Evolução do número de acertos das 20 primeiras SVMs para o primeiro sujeito não-amputado da base de dados Ninapro.

Acertos (20)	0	84	108	110			
Acertos (19)	0	101		111			114
Acertos (18)	0	80	111				124
Acertos (17)	0	106	120	136	142		
Acertos (16)	0	134	141				
Acertos (15)	0	55	58	66			
Acertos (14)	0	52	64	67			69
Acertos (13)	0	68	76				77
Acertos (12)	0	96	97				
Acertos (11)	0	100	112				113
Acertos (10)	0	86					
Acertos (9)	0	61	77	78			82
Acertos (8)	0	94	107	108			
Acertos (7)	0	124					
Acertos (6)	0	74					
Acertos (5)	0	95	102				
Acertos (4)	0	95	117			122	123
Acertos (3)	0	97	98				
Acertos (2)	0	87	126	127			131
Acertos (1)	0	87	119	124			125
Tempo		3,0 ms	3,2 ms	3,4 ms	3,6 ms	3,8 ms	4,0 ms

Fonte: do autor

A convergência da taxa de acerto pode ser observada também com o auxílio da métrica Erro Quadrático Médio. A Figura 27 apresenta a evolução do treinamento do sistema com e sem o bloco de ganho para as SVMs binárias descrito e discutido nas Seções 3.1.3 e 4.1.3. Pode-se perceber que o EQM observado no início da validação dos sinais do sujeito 1 da base de dados de não amputados foi 0,0304 para o sistema sem ganho e 0,0298 para o sistema com ganho.

Figura 27 - Evolução do EQM das 136 SVMs para o primeiro sujeito não-amputado da base de dados Ninapro.



Fonte: do autor

Com a evolução do treinamento, rapidamente ambas as abordagens, com e sem ganho, atingem um EQM abaixo de 0,015. Após 75 ms, 132 épocas de treinamento o EQM observado é de 0,0120 para o sistema com ganho e 0,0126 para o sistema sem ganho. A Figura 27 apresenta apenas o período em que o aprendizado é mais evidente, o sistema continuou aprendendo após esse tempo, sendo 0,0116 o EQM observado com ganho após 150 ms de treinamento. Tempos

maiores são observados nas Seções 4.2.1 e 4.4.2, nos ensaios com o sistema completo em operação.

O sistema apresentou melhores resultados com o ganho do que sem o ganho para o sujeito 1 da base de dados de não amputados, portanto optou-se por utilizar o bloco de ganho implementado. A Tabela 12 apresenta os acertos de cada uma das 136 SVMs após 75 ms de treinamento e a taxa de acerto correspondente. Pode-se perceber que a maioria das SVMs tem taxa de acerto acima de 90% após este período de treinamento.

Tabela 12 - Acertos calculados para as 136 SVMs para o primeiro sujeito não-amputado da base de dados Ninapro.

SVM	Movimento 1	Movimento 2	Acertos	SVM	Movimento 1	Movimento 2	Acertos
1	1	2	125 (79%)	69	3	13	86 (86%)
2	1	3	133 (93%)	70	4	13	75 (83%)
3	2	3	98 (77%)	71	5	13	61 (70%)
4	1	4	123 (92%)	72	6	13	67 (93%)
5	2	4	102 (86%)	73	7	13	107 (92%)
6	3	4	74 (72%)	74	8	13	58 (77%)
7	1	5	124 (95%)	75	9	13	69 (77%)
8	2	5	108 (94%)	76	10	13	76 (80%)
9	3	5	82 (83%)	77	11	13	71 (80%)
10	4	5	86 (97%)	78	12	13	70 (76%)
11	1	6	113 (98%)	79	1	14	122 (97%)
12	2	6	97 (97%)	80	2	14	105 (95%)
13	3	6	77 (92%)	81	3	14	79 (84%)
14	4	6	69 (93%)	82	4	14	65 (77%)
15	5	6	66 (93%)	83	5	14	54 (66%)
16	1	7	141 (89%)	84	6	14	63 (95%)
17	2	7	142 (99%)	85	7	14	104 (95%)
18	3	7	124 (97%)	86	8	14	53 (76%)
19	4	7	115 (98%)	87	9	14	70 (84%)
20	5	7	110 (96%)	88	10	14	68 (77%)
21	6	7	98 (98%)	89	11	14	63 (75%)
22	1	8	112 (94%)	90	12	14	66 (77%)
23	2	8	93 (90%)	91	13	14	60 (73%)
24	3	8	70 (80%)	92	1	15	108 (96%)
25	4	8	57 (73%)	93	2	15	95 (98%)
26	5	8	57 (77%)	94	3	15	68 (84%)
27	6	8	59 (99%)	95	4	15	60 (84%)
28	7	8	94 (91%)	96	5	15	65 (96%)
29	1	9	131 (99%)	97	6	15	43 (81%)
30	2	9	112 (96%)	98	7	15	94 (97%)
31	3	9	101 (100%)	99	8	15	57 (100%)
32	4	9	90 (98%)	100	9	15	64 (91%)
33	5	9	61 (69%)	101	10	15	72 (95%)

Continuação da Tabela 12.

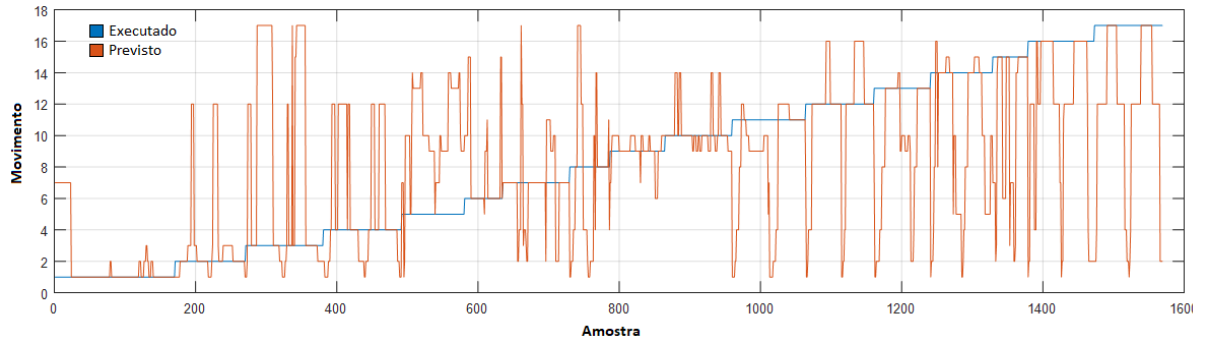
SVM	Movimento 1	Movimento 2	Acertos	SVM	Movimento 1	Movimento 2	Acertos
34	6	9	68 (93%)	102	11	15	64 (91%)
35	7	9	109 (93%)	103	12	15	71 (98%)
36	8	9	72 (94%)	104	13	15	69 (99%)
37	1	10	136 (99%)	105	14	15	53 (83%)
38	2	10	121 (99%)	106	1	16	132 (97%)
39	3	10	105 (99%)	107	2	16	112 (93%)
40	4	10	94 (97%)	108	3	16	85 (81%)
41	5	10	68 (73%)	109	4	16	69 (73%)
42	6	10	76 (97%)	110	5	16	75 (82%)
43	7	10	115 (94%)	111	6	16	76 (99%)
44	8	10	73 (89%)	112	7	16	110 (91%)
45	9	10	67 (70%)	113	8	16	65 (81%)
46	1	11	127 (96%)	114	9	16	91 (97%)
47	2	11	111 (95%)	115	10	16	94 (95%)
48	3	11	95 (94%)	116	11	16	77 (82%)
49	4	11	71 (78%)	117	12	16	62 (64%)
50	5	11	73 (83%)	118	13	16	79 (85%)
51	6	11	67 (92%)	119	14	16	81 (93%)
52	7	11	107 (92%)	120	15	16	74 (100%)
53	8	11	57 (75%)	121	1	17	125 (96%)
54	9	11	64 (71%)	122	2	17	106 (93%)
55	10	11	80 (84%)	123	3	17	72 (73%)
56	1	12	131 (97%)	124	4	17	78 (88%)
57	2	12	112 (94%)	125	5	17	80 (94%)
58	3	12	79 (76%)	126	6	17	70 (99%)
59	4	12	73 (78%)	127	7	17	107 (94%)
60	5	12	80 (89%)	128	8	17	72 (97%)
61	6	12	76 (100%)	129	9	17	73 (83%)
62	7	12	109 (92%)	130	10	17	89 (96%)
63	8	12	64 (81%)	131	11	17	71 (81%)
64	9	12	88 (95%)	132	12	17	67 (74%)
65	10	12	92 (94%)	133	13	17	75 (86%)
66	11	12	72 (78%)	134	14	17	76 (94%)
67	1	13	128 (97%)	135	15	17	68 (100%)
68	2	13	112 (97%)	136	16	17	67 (73%)

4.1.6 Validação da etapa de classificação da SVM

Os pesos e *bias* escolhidos pelo algoritmo de validação após 75 ms de treinamento foram apresentados ao classificador para que os 17 movimentos fossem previstos com sinais de entrada desconhecidos pelo sistema. Essas entradas são as amostras de sinais de sEMG do primeiro sujeito não-amputado da base de dados Ninapro que não foram utilizadas para treinamento e validação (33% restantes). As Figuras 28 e 29 mostram os resultados obtidos para

este teste. A primeira mostra as amostras do movimento executado e correspondente movimento previsto na ordem em que foram apresentadas ao classificador, e a segunda apresenta os resultados na forma de matriz de confusão.

Figura 28 - Classificação dos movimentos utilizados na simulação na ordem da apresentação das amostras para o primeiro sujeito não-amputado da base de dados Ninapro.



Fonte: do autor

Figura 29 - Matriz de confusão resultante da simulação da etapa de classificação para o primeiro sujeito não-amputado da base de dados Ninapro.

1	132	13	6	10	1	0	0	5	0	0	9	4	2	3	0	2	1	70.2%
	8.4%	0.8%	0.4%	0.6%	0.1%	0.0%	0.0%	0.3%	0.0%	0.0%	0.6%	0.3%	0.1%	0.2%	0.0%	0.1%	0.1%	29.8%
2	13	51	15	20	1	0	10	9	0	0	6	3	9	4	3	12	10	30.7%
	0.8%	3.3%	1.0%	1.3%	0.1%	0.0%	0.6%	0.6%	0.0%	0.0%	0.4%	0.2%	0.6%	0.3%	0.2%	0.8%	0.6%	59.3%
3	2	25	43	0	0	0	2	0	0	0	0	0	0	0	3	0	0	57.3%
	0.1%	1.6%	2.7%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.0%	42.7%
4	0	0	0	46	0	0	6	11	0	0	7	7	4	3	0	4	5	49.5%
	0.0%	0.0%	0.0%	2.9%	0.0%	0.0%	0.4%	0.7%	0.0%	0.0%	0.4%	0.4%	0.3%	0.2%	0.0%	0.3%	0.3%	50.5%
5	0	0	0	0	3	1	0	0	0	0	1	0	0	17	0	0	0	13.6%
	0.0%	0.0%	0.0%	0.0%	0.2%	0.1%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	1.1%	0.0%	0.0%	0.0%	36.4%
6	0	0	0	0	0	38	0	0	4	0	0	0	0	0	5	0	0	30.9%
	0.0%	0.0%	0.0%	0.0%	0.0%	2.4%	0.0%	0.0%	0.3%	0.0%	0.0%	0.0%	0.0%	0.3%	0.0%	0.0%	0.0%	19.1%
7	23	0	0	0	9	0	61	0	3	0	1	1	1	0	9	2	0	55.5%
	1.5%	0.0%	0.0%	0.0%	0.6%	0.0%	3.9%	0.0%	0.2%	0.0%	0.1%	0.1%	0.1%	0.0%	0.6%	0.1%	0.0%	44.5%
8	0	0	0	0	0	0	0	19	0	0	3	0	9	7	0	0	0	50.0%
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	1.2%	0.0%	0.0%	0.2%	0.0%	0.6%	0.4%	0.0%	0.0%	0.0%	50.0%
9	0	0	0	0	23	3	4	0	44	21	21	0	7	0	0	0	0	35.8%
	0.0%	0.0%	0.0%	0.0%	1.5%	0.2%	0.3%	0.0%	2.8%	1.3%	1.3%	0.0%	0.4%	0.0%	0.0%	0.0%	0.0%	54.2%
10	0	0	0	0	20	5	3	2	27	61	10	0	8	10	0	0	0	41.8%
	0.0%	0.0%	0.0%	0.0%	1.3%	0.3%	0.2%	0.1%	1.7%	3.9%	0.6%	0.0%	0.5%	0.6%	0.0%	0.0%	0.0%	58.2%
11	0	0	0	0	0	1	6	1	0	0	26	0	0	0	1	0	0	74.3%
	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.4%	0.1%	0.0%	0.0%	1.7%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	25.7%
12	0	11	8	35	0	0	2	3	0	0	20	62	0	0	0	34	50	27.6%
	0.0%	0.7%	0.5%	2.2%	0.0%	0.0%	0.1%	0.2%	0.0%	0.0%	1.3%	4.0%	0.0%	0.0%	0.0%	2.2%	3.2%	72.4%
13	0	0	0	0	22	0	0	0	0	0	0	0	0	37	0	0	0	52.7%
	0.0%	0.0%	0.0%	0.0%	1.4%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	2.4%	0.0%	0.0%	0.0%	37.3%
14	0	0	0	0	10	0	0	2	0	13	0	0	3	29	4	0	0	47.5%
	0.0%	0.0%	0.0%	0.0%	0.6%	0.0%	0.0%	0.1%	0.0%	0.8%	0.0%	0.0%	0.2%	1.8%	0.3%	0.0%	0.0%	52.5%
15	0	0	2	0	0	6	0	0	0	0	0	0	0	12	26	0	0	56.5%
	0.0%	0.0%	0.1%	0.0%	0.0%	0.4%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.8%	1.7%	0.0%	0.0%	43.5%
16	0	0	0	0	0	0	0	0	0	0	20	0	3	0	39	0	0	52.9%
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	1.3%	0.0%	0.2%	0.0%	2.5%	0.0%	0.0%	37.1%
17	0	0	36	0	0	0	1	5	0	0	0	0	0	0	0	0	30	41.7%
	0.0%	0.0%	2.3%	0.0%	0.0%	0.0%	0.1%	0.3%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	1.9%	58.3%
	77.6%	51.0%	39.1%	41.4%	3.4%	70.4%	64.2%	33.3%	56.4%	64.2%	25.0%	63.9%	46.3%	33.0%	52.0%	41.5%	31.3%	47.6%
	22.4%	49.0%	60.9%	58.6%	96.6%	29.6%	35.8%	66.7%	43.6%	35.8%	75.0%	36.1%	53.8%	67.0%	48.0%	58.5%	68.8%	52.4%
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	

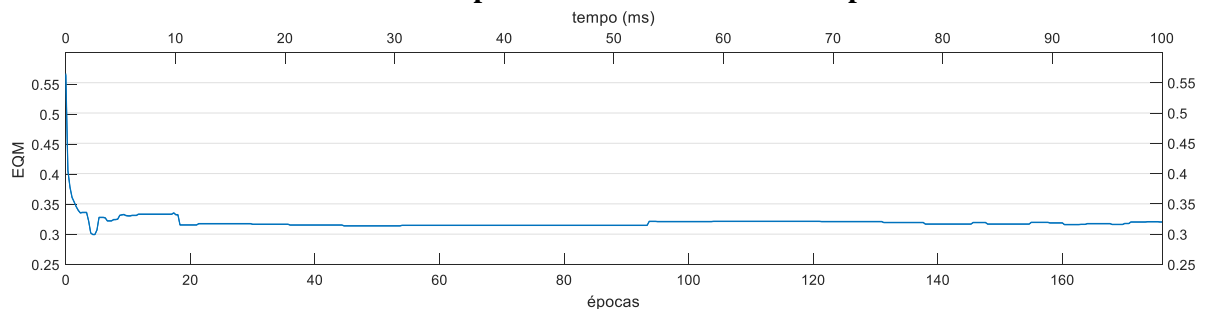
Fonte: do autor

Destes resultados pode-se perceber que com um treinamento de apenas 75 ms o sistema mais erra do que acerta (taxa de acerto de 47,6%) porém este tempo é insuficiente para o sistema convergir a uma solução sendo que apenas 132 épocas foram processadas. Mesmo assim, se

forem analisados os movimentos individualmente, pode-se perceber que as saídas do classificador não são aleatórias, alguns movimentos ou partes deles são confundidos pelo classificador. Isso fica evidente tanto na matriz de confusão quanto na análise das amostras individuais dos movimentos 9 e 10, que o classificador não consegue decidir se prevê como um ou outro movimento.

A Figura 30 apresenta a evolução do EQM para a classificação dos 17 movimentos. Pode-se perceber que o sistema partiu de um ponto com EQM de 0,5651 e após 12 épocas baixou esta métrica para 0,3218. Depois disso, o EQM oscilou por um breve momento e o sistema ficou com o erro quadrático médio estabilizado entre 0,31 e 0,33 por pelo menos 88 ms.

Figura 30 - Evolução do EQM da classificação de 17 movimentos para o primeiro sujeito não-amputado da base de dados Ninapro.



Fonte: do autor

Este gráfico apresenta apenas o início do aprendizado do sistema, a região onde a curva de aprendizado é maior. A Seção 4.2 explora o sistema em treinamentos em períodos maiores de tempo onde se pode perceber que o sistema continua aprendendo após os 100 ms.

4.1.7 Avaliação da ocupação lógica do sistema

Todos os blocos do sistema de classificação de sinais de sEMG em FPGA foram embarcados na plataforma Mini-Module Plus Baseboard 2 da AVNET (2012) que contém uma FPGA Xilinx Kintex 7. Portanto, a ferramenta utilizada para converter o código VHDL no binário interpretado pela FPGA foi a Xilinx Vivado (Versão 2017.2), que apresentou os percentuais de ocupação listados na Tabela 13. Esta mostra que o sistema implementado utilizou os recursos da FPGA de forma a praticamente esgotar as *Lookup Tables* (LUTs), que são os recursos unitários para lógica combinacional. Grande parte dos recursos de memória (Block RAM) também foram utilizados. O bloco que mais utilizou recursos foi o responsável pelo treinamento SVM, pois este possui o algoritmo mais complexo e as maiores memórias

para armazenar as amostras de treinamento. Assim sendo, o projeto do sistema foi adequado à FPGA disponível.

Tabela 13 - Percentual de ocupação lógica dos blocos implementados na correspondente FPGA.

Bloco	Elemento	Ocupação	Disponível
Filtro FIR	LUT	13231 (6,49%)	203800
	Flip-Flop	46584 (11,43%)	407600
	Block RAM	0 (0%)	445
	DSP	1 (0,12%)	840
Extração de Características	LUT	14713 (7,22%)	203800
	Flip-Flop	12958 (3,18%)	407600
	Block RAM	23 (5,17%)	445
	DSP	1 (0,12%)	840
Ganho	LUT	427 (0,21%)	203800
	Flip-Flop	462 (0,11%)	407600
	Block RAM	2 (0,45%)	445
	DSP	0 (0,00%)	840
Treinamento SVM	LUT	116230 (57,03%)	203800
	Flip-Flop	102582 (25,17%)	407600
	Block RAM	188,5 (42,36%)	445
	DSP	192 (22,86%)	840
Validação SVM	LUT	27351 (13,42%)	203800
	Flip-Flop	16997 (4,17%)	407600
	Block RAM	128 (28,76%)	445
	DSP	96 (11,43%)	840
Classificação SVM	LUT	13789 (6,77%)	203800
	Flip-Flop	7979 (1,96%)	407600
	Block RAM	34,5 (7,75%)	445
	DSP	48 (5,71%)	840
Total	LUT	187551 (92%)	203800
	Flip-Flop	187562 (46%)	407600
	Block RAM	376 (84%)	445
	DSP	338 (40%)	840

Os recursos de DSP foram pouco utilizados, o que indica que algumas operações matemáticas descritas em lógica combinacional e memória (divisões e raiz quadrada) poderiam ser descritas de modo a utilizar estes recursos, reduzindo a demanda por LUTs e Block RAMs. Isto facilitaria a fase de roteamento na implementação para a FPGA.

4.2 AVALIAÇÃO DO DESEMPENHO DO SISTEMA COMPLETO

Os resultados obtidos para os experimentos descritos na Seção 3.2.2 foram avaliados estatisticamente conforme descrito na Seção 3.2.3 (ver Tabela 14). Desta avaliação foi possível descartar a hipótese nula para todos os fatores controlados do projeto e suas interações com confiança de 99,8% ($P < 0,002$).

Tabela 14 - Análise de variância.

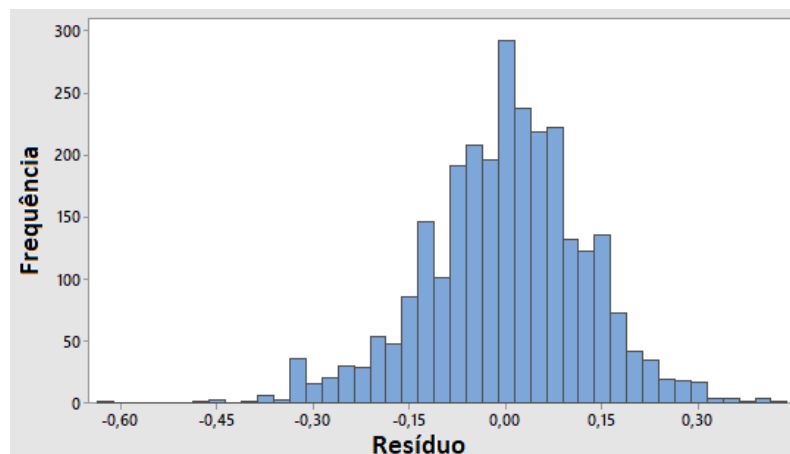
Fonte	GDL	SQ	MQ	Valor F	Valor P
Modelo	1734	168,864	0,09738	4,53	0,000
Nº de Movimentos	1	2,776	2,77637	129,04	0,000
Sujeito (A)	50	29,555	0,59111	27,47	0,000
Movimento (B)	16	9,860	0,61622	28,64	0,000
Tempo (C)	1	8,353	8,35261	388,21	0,000
Interação AB	800	79,766	0,09971	4,63	0,000
Interação AC	50	1,854	0,03708	1,72	0,002
Interação BC	16	0,910	0,05688	2,64	0,000
Interação ABC	800	25,501	0,03188	1,48	0,000
Erro	1019	21,924	0,02152	-	-
Total	2753	190,789	-	-	-

O modelo foi validado através da análise de resíduos. A Figura 31 apresenta o histograma dos resíduos do modelo, onde pode-se perceber que estes seguem uma distribuição normal centrada em zero. A

Figura 32 apresenta os resíduos obtidos na ordem em que os ensaios foram realizados, e a Figura 33, os resíduos *versus* as médias dos níveis dos fatores controlados e dos blocos. Tanto no gráfico da

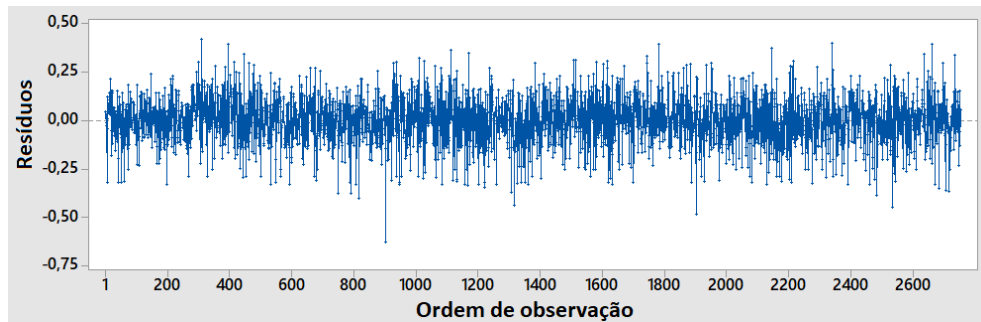
Figura 32 quanto nos gráficos da Figura 33, nenhum padrão conhecido pôde ser identificado, portanto, o modelo foi considerado adequado.

Figura 31 - Histograma dos resíduos do modelo de experimento.



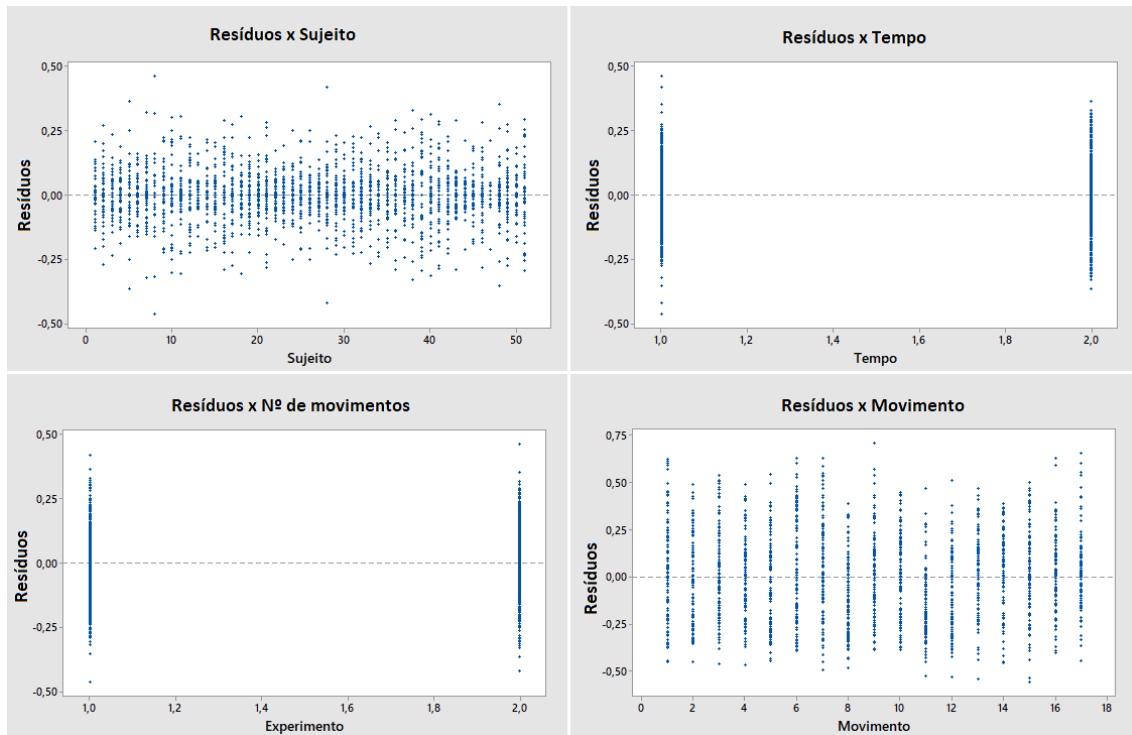
Fonte: do autor

Figura 32 - Distribuição dos resíduos no tempo conforme a ordem de observação.



Fonte: do autor

Figura 33 - Gráficos dos resíduos *versus* a média dos níveis das variáveis controladas do experimento (sujeito, tempo e movimento) e média dos blocos (nº de movimentos).



Fonte: do autor

A influência dos fatores controlados no treinamento do sistema é analisada nas Seções 4.2.1 e 4.2.2 que separam os ensaios do bloco 1 (ensaios com aprendizado de 17 movimentos) e ensaios do bloco 2 (ensaios com aprendizado de 10 movimentos). A Seção 4.2.3 apresenta os resultados obtidos para os sujeitos que geraram as maiores taxas de acerto e a Seção 4.2.4 compara os resultados obtidos com resultados de trabalhos relacionados encontrados na literatura.

4.2.1 Ensaios com aprendizado de 17 movimentos

Executando-se os dois primeiros ensaios propostos na Seção 3.2.2 (treinamento de 75 milissegundos e de dez segundos) com os sinais de sEMG dos 40 sujeitos não amputados da base de dados Ninapro, foram geradas 40 matrizes de confusão para cada ensaio que estão resumidas nas Figuras 34 e 35. Essas duas matrizes de confusão foram geradas a partir da sobreposição das matrizes obtidas para os 40 sujeitos: os valores absolutos apresentados são o somatório dos valores para os sujeitos e as porcentagens são valores médios. As métricas *sensitivity* e preditividade positiva são apresentados na parte inferior e lateral direita das matrizes de confusão respectivamente, e a taxa de acerto no canto inferior direito.

Figura 34 - Sobreposição das matrizes de confusão geradas para sujeitos não amputados após 75 ms de treinamento.

1	382747	037934	47685	32044	18029	22459	68528	24050	17553	20124	30584	19584	4856	5671	40820	12517	11224	44.4%
	3.1%	0.8%	0.4%	0.3%	0.1%	0.2%	0.6%	0.2%	0.1%	0.2%	0.2%	0.2%	0.0%	0.0%	0.3%	0.1%	0.1%	55.6%
2	101721	332430	63187	31581	21313	20538	48235	21994	17773	9873	29997	17115	5175	8258	18304	15918	12709	34.4%
	0.8%	1.9%	0.5%	0.3%	0.2%	0.2%	0.4%	0.2%	0.1%	0.1%	0.2%	0.1%	0.0%	0.1%	0.1%	0.1%	0.1%	55.6%
3	43500	07892	22600	29721	20026	12450	23588	20130	385	1650	3935	3450	5250	11047	1950	2820	4321	43.6%
	0.4%	0.9%	1.8%	0.2%	0.2%	0.1%	0.2%	0.2%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	56.4%
4	22080	33665	40791	32083	09874	34298	40303	44369	18524	19875	25748	11214	15965	21015	5381	11235	8958	33.4%
	0.2%	0.3%	0.3%	1.9%	0.9%	0.3%	0.3%	0.4%	0.1%	0.2%	0.2%	0.1%	0.1%	0.2%	0.0%	0.1%	0.1%	56.6%
5	28662	52804	53412	95380	99154	29887	57564	34416	42076	35927	45102	30922	19105	22966	29828	13228	11757	24.8%
	0.2%	0.4%	0.4%	0.8%	1.6%	0.2%	0.5%	0.3%	0.3%	0.3%	0.4%	0.2%	0.2%	0.2%	0.2%	0.1%	0.1%	75.2%
6	14824	14250	24000	27548	39436	10077	13560	35428	4966	8700	2250	3731	2100	9600	40973	5850	4614	45.5%
	0.1%	0.1%	0.2%	0.2%	0.3%	1.7%	0.1%	0.3%	0.0%	0.1%	0.0%	0.0%	0.0%	0.1%	0.3%	0.0%	0.0%	54.5%
7	78579	78908	52903	43353	68230	62724	15729	82517	34267	36227	59714	63884	22332	32830	63632	24760	26678	33.3%
	0.6%	0.6%	0.4%	0.3%	0.5%	0.5%	3.4%	0.7%	0.3%	0.3%	0.5%	0.5%	0.2%	0.3%	0.5%	0.2%	0.2%	56.7%
8	21363	17026	23597	20262	30550	22891	46842	46950	14163	19479	36099	34089	21449	26799	23808	18404	18717	27.1%
	0.2%	0.1%	0.2%	0.2%	0.2%	0.2%	0.4%	1.2%	0.1%	0.2%	0.3%	0.3%	0.2%	0.2%	0.2%	0.1%	0.2%	72.9%
9	1269	857	2835	11896	14789	0	5676	9958	23377	49914	72924	3851	2357	3016	1670	1950	2700	55.7%
	0.0%	0.0%	0.0%	0.1%	0.1%	0.0%	0.0%	0.1%	1.9%	0.4%	0.6%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	44.3%
10	3847	2367	935	3127	14871	4476	6791	12072	17412	26457	839803	8397	13723	11313	3992	10901	9048	50.1%
	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.1%	0.1%	0.9%	2.1%	0.3%	0.1%	0.1%	0.1%	0.0%	0.1%	0.1%	49.9%
11	43224	15489	8852	18716	17265	20449	30187	19336	78239	67031	81142	28199	10849	10427	21122	22236	16938	29.7%
	0.3%	0.1%	0.1%	0.2%	0.1%	0.2%	0.2%	0.2%	0.6%	0.5%	1.5%	0.2%	0.1%	0.1%	0.2%	0.2%	0.1%	70.3%
12	21026	21110	15791	26633	17151	22356	20144	14308	24795	38255	81898	92387	39295	26390	14690	79538	59103	26.9%
	0.2%	0.2%	0.1%	0.2%	0.1%	0.2%	0.2%	0.1%	0.2%	0.3%	0.7%	1.6%	0.3%	0.2%	0.1%	0.6%	0.5%	73.1%
13	22713	11672	11433	21975	16375	9014	11330	32032	18262	25596	26670	42228	23259	28367	23275	29342	13864	39.3%
	0.2%	0.1%	0.1%	0.2%	0.1%	0.1%	0.1%	0.3%	0.1%	0.2%	0.2%	0.3%	1.8%	0.2%	0.2%	0.2%	0.1%	50.7%
14	13413	46491	53505	47368	40319	22288	17100	55703	24975	30680	53572	35482	50151	22260	658509	18808	12285	27.7%
	0.1%	0.4%	0.4%	0.4%	0.3%	0.2%	0.1%	0.4%	0.2%	0.2%	0.4%	0.3%	0.4%	1.8%	0.5%	0.2%	0.1%	72.3%
15	74799	27897	21673	20666	32355	48745	53734	42634	23425	21382	35094	38958	12487	35280	26673	118336	20639	33.6%
	0.6%	0.2%	0.2%	0.2%	0.3%	0.4%	0.4%	0.3%	0.2%	0.2%	0.3%	0.3%	0.1%	0.3%	2.1%	0.1%	0.2%	56.4%
16	14079	26243	16688	22708	29468	8753	15704	31935	33073	44181	30965	73274	51614	20772	22945	26148	39897	32.6%
	0.1%	0.2%	0.1%	0.2%	0.2%	0.1%	0.1%	0.3%	0.3%	0.4%	0.2%	0.6%	0.4%	0.2%	2.1%	0.8%	57.4%	
17	72181	49445	72666	51425	41837	64058	36124	66983	73006	62807	92390	25012	64435	61093	48859	30362	24910	18.3%
	0.6%	0.4%	0.6%	0.4%	0.3%	0.5%	0.3%	0.5%	0.6%	0.5%	0.7%	1.0%	0.5%	0.5%	0.4%	1.1%	2.0%	81.7%
	39.9%	27.6%	30.7%	31.5%	27.2%	34.1%	45.6%	21.1%	30.1%	35.0%	21.4%	26.3%	39.6%	39.9%	38.9%	38.6%	42.8%	33.4%
	60.1%	72.4%	69.3%	68.5%	72.8%	65.9%	54.4%	78.9%	69.9%	65.0%	78.6%	73.7%	60.4%	60.1%	61.1%	61.4%	57.2%	56.6%
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	

Fonte: do autor

Figura 35 - Sobreposição das matrizes de confusão geradas para sujeitos não amputados após 10 s de treinamento.

1	424620	47424	23741	16933	11185	7521	10905	10162	10134	7482	12214	7819	1184	1575	13562	6588	3157	68.9%
	3.4%	0.4%	0.2%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	0.0%	0.0%	0.1%	0.0%	0.0%	31.1%
2	34551	57580	88456	41182	21807	25910	40259	28479	8698	3985	8277	5249	4178	3008	3600	6872	1871	45.6%
	1.1%	2.9%	0.7%	0.3%	0.2%	0.2%	0.3%	0.2%	0.1%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	54.4%
3	38907	01342	43877	15257	10141	11850	12992	11730	2036	1800	11540	2400	150	2992	7064	1323	1178	59.6%
	0.3%	0.8%	2.8%	0.1%	0.1%	0.1%	0.1%	0.1%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	40.4%
4	26264	42264	39669	99925	57173	25397	20055	27697	3981	11778	15512	13182	8429	19199	5446	6797	6577	45.5%
	0.2%	0.3%	0.3%	2.4%	0.7%	0.2%	0.2%	0.2%	0.0%	0.1%	0.1%	0.1%	0.1%	0.2%	0.0%	0.1%	0.1%	54.5%
5	28043	41162	25135	00350	61320	25250	44388	54665	31623	22064	35810	25912	8654	11329	9621	11021	6624	33.8%
	0.2%	0.3%	0.4%	0.8%	2.1%	0.2%	0.4%	0.4%	0.3%	0.2%	0.3%	0.2%	0.1%	0.1%	0.1%	0.1%	0.1%	66.2%
6	37132	32588	28432	28322	36582	28788	28269	23829	11987	9937	12741	6246	7609	5652	41756	6819	4496	47.2%
	0.3%	0.3%	0.2%	0.2%	0.3%	2.3%	0.2%	0.2%	0.1%	0.1%	0.1%	0.1%	0.1%	0.0%	0.3%	0.1%	0.0%	52.8%
7	35466	42346	24264	26439	62700	65731	76685	53858	11406	22713	29058	29996	4569	17217	22116	6418	8167	50.8%
	0.3%	0.3%	0.2%	0.2%	0.5%	0.5%	3.8%	0.4%	0.1%	0.2%	0.2%	0.2%	0.0%	0.1%	0.2%	0.1%	0.1%	49.2%
8	32538	23871	25722	24717	54815	30047	84132	14319	7417	11652	51355	35250	20025	24150	19656	8613	22016	31.0%
	0.3%	0.2%	0.2%	0.2%	0.4%	0.2%	0.7%	1.7%	0.1%	0.1%	0.4%	0.3%	0.2%	0.2%	0.2%	0.1%	0.2%	69.0%
9	969	15300	3622	5180	11759	3985	5381	11210	10654	62307	84345	2446	5094	4144	5054	3418	750	64.6%
	0.0%	0.1%	0.0%	0.0%	0.1%	0.0%	0.0%	0.1%	3.3%	0.5%	0.7%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	35.4%
10	2912	10271	554	9407	16633	8504	11926	9156	94203	65376	42160	9878	10212	11984	7934	8566	7425	58.3%
	0.0%	0.1%	0.0%	0.1%	0.1%	0.1%	0.1%	0.1%	0.8%	2.9%	0.3%	0.1%	0.1%	0.1%	0.1%	0.1%	0.1%	41.7%
11	32747	20399	14309	11473	21221	13578	28823	34677	97122	85090	50184	26048	11474	12692	24956	14970	12226	39.5%
	0.3%	0.2%	0.1%	0.1%	0.2%	0.1%	0.2%	0.3%	0.8%	0.7%	2.4%	0.2%	0.1%	0.1%	0.2%	0.1%	0.1%	60.5%
12	26538	18054	18415	22762	20853	28403	13461	27822	12652	22091	85763	283596	29123	24979	20890	58899	50192	37.1%
	0.2%	0.1%	0.1%	0.2%	0.2%	0.2%	0.1%	0.2%	0.1%	0.2%	0.7%	2.3%	0.2%	0.2%	0.2%	0.5%	0.4%	62.9%
13	13091	4541	9750	22005	11376	7057	6300	30750	5505	16517	13834	39063	298480	28174	9301	19050	12521	54.5%
	0.1%	0.0%	0.1%	0.2%	0.1%	0.1%	0.1%	0.2%	0.0%	0.1%	0.1%	0.3%	2.4%	0.2%	0.1%	0.2%	0.1%	45.5%
14	6750	16500	9900	33690	39121	11974	17090	38109	11305	23746	24094	23787	64133	30503	366437	22197	15241	41.8%
	0.1%	0.1%	0.1%	0.3%	0.3%	0.1%	0.1%	0.3%	0.1%	0.2%	0.2%	0.2%	0.5%	0.2%	0.5%	0.2%	0.1%	58.2%
15	46126	23610	12165	29886	26081	47056	65069	52535	20715	23283	54323	32509	19386	36554	69350	17385	17394	41.3%
	0.4%	0.2%	0.1%	0.2%	0.2%	0.4%	0.5%	0.4%	0.2%	0.2%	0.4%	0.3%	0.2%	0.3%	3.0%	0.1%	0.1%	58.7%
16	32589	23930	11829	33028	28741	8548	27564	41268	17780	46774	33608	16716	53435	32400	43371	394657	16792	37.1%
	0.3%	0.2%	0.1%	0.3%	0.2%	0.1%	0.2%	0.3%	0.1%	0.4%	0.3%	0.9%	0.4%	0.3%	0.3%	3.2%	0.9%	62.9%
17	40784	21156	26113	15930	9600	6762	17872	24581	19452	19684	31485	71680	18269	16368	16375	84095	94763	40.1%
	0.3%	0.2%	0.2%	0.1%	0.1%	0.1%	0.1%	0.2%	0.2%	0.2%	0.3%	0.6%	0.1%	0.1%	0.1%	0.7%	2.4%	59.9%
	44.2%	42.5%	46.7%	40.7%	35.7%	46.8%	52.3%	30.8%	52.9%	48.3%	35.6%	38.8%	52.9%	54.7%	53.8%	58.2%	50.7%	45.9%
	55.8%	57.5%	53.3%	59.3%	64.3%	53.2%	47.7%	69.2%	47.1%	51.7%	64.4%	61.2%	47.1%	45.3%	46.2%	41.8%	49.3%	54.1%
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	

Fonte: do autor

Destas matrizes é possível perceber o aumento da taxa de acerto média bem como aumento da *sensitivity* média e valores preditivos positivos médios do primeiro para o segundo ensaio, sinal de que 75 ms é tempo insuficiente para o treinamento ser concluído. Mesmo assim, a taxa de acerto 33,4% é superior à chance de acerto de um movimento em dezessete (5,88%), indicando que o algoritmo aprendeu parte do problema no ensaio 1. Outra informação importante que pode ser visualizada nas matrizes de confusão é que, para todos os movimentos, a porcentagem de verdadeiro positivo é sempre maior que a dos falsos positivos e falsos negativos individualmente.

Da mesma forma, foram geradas as matrizes de confusão para os resultados obtidos nos ensaios 1 e 2 com 11 sujeitos amputados (ver Figuras 36 e 37). Novamente é possível observar o aumento da taxa de acerto média, *sensitivity* média e dos valores preditivos positivos médios do primeiro para o segundo ensaio. Novamente, a taxa de acerto (20,9%) obtida no ensaio 1 é

superior à chance de acerto de um movimento em dezessete (5,88%), indicando aprendizado parcial do problema em 75 ms. No treinamento de 10 s, a porcentagem dos verdadeiros positivos é maior que as porcentagens individuais dos falsos positivos e dos falsos negativos.

As taxas de acerto médias do sistema treinado para sujeitos amputados e não amputados apresentam uma diferença de 15,3% após 10 s de treinamento, sendo a maior taxa de acerto obtida pelos sujeitos não amputados. Resultados piores eram esperados para sujeitos amputados devido à falta de algum músculo ou dificuldade por parte do voluntário em ativar alguns músculos.

Figura 36 - Sobreposição das matrizes de confusão geradas para sujeitos amputados após 75 ms de treinamento.

Movimento Previsto	1	16831 0.4%	7286 0.2%	7398 0.2%	1950 0.1%	189 0.0%	12693 0.3%	1650 0.0%	1763 0.0%	947 0.0%	600 0.0%	1800 0.0%	0 0.0%	1500 0.0%	0 0.0%	1527 0.0%	1200 0.0%	0 0.0%	29.4%
	2	12862 0.3%	43339 1.1%	17706 0.5%	6521 0.2%	11808 0.3%	9235 0.2%	8756 0.2%	12509 0.3%	450 0.0%	3000 0.1%	6609 0.2%	3981 0.1%	9904 0.3%	7900 0.2%	9930 0.3%	7945 0.2%	1350 0.0%	24.9%
	3	34275 0.9%	51279 1.3%	51648 1.6%	21692 0.6%	7650 0.2%	2495 0.1%	4950 0.1%	9300 0.2%	3478 0.1%	1500 0.0%	2400 0.1%	2322 0.1%	4745 0.1%	10205 0.3%	4500 0.1%	1350 0.0%	2672 0.1%	27.2%
	4	9550 0.2%	22245 0.6%	16076 0.4%	43160 1.1%	16050 0.4%	3654 0.1%	1650 0.0%	5636 0.1%	5589 0.1%	600 0.0%	9118 0.2%	12612 0.3%	3486 0.1%	7383 0.2%	4756 0.1%	4948 0.1%	2400 0.1%	25.6%
	5	4025 0.1%	8700 0.2%	16712 0.4%	23388 0.6%	28464 0.7%	10751 0.3%	16779 0.4%	19459 0.5%	8100 0.2%	15251 0.4%	20346 0.5%	13723 0.4%	11834 0.3%	12749 0.3%	7804 0.2%	2885 0.1%	1362 0.0%	12.8%
	6	19385 0.5%	17084 0.4%	18089 0.5%	19741 0.5%	38986 1.0%	53563 1.6%	25309 0.7%	23990 0.6%	12544 0.3%	12409 0.3%	3209 0.1%	6505 0.2%	18924 0.5%	8523 0.2%	16414 0.4%	9666 0.2%	11737 0.3%	19.5%
	7	10950 0.3%	23392 0.6%	7279 0.2%	4298 0.1%	10134 0.3%	14113 0.4%	72136 1.9%	16606 0.4%	4140 0.1%	5464 0.1%	4826 0.1%	3025 0.1%	8661 0.2%	6956 0.2%	3086 0.1%	2310 0.1%	1021 0.0%	36.4%
	8	3150 0.1%	729 0.0%	2850 0.1%	4200 0.1%	5939 0.2%	7950 0.2%	10923 0.3%	21556 0.6%	2975 0.1%	10901 0.3%	3000 0.1%	3039 0.1%	3000 0.1%	1800 0.0%	1350 0.0%	2850 0.1%	1650 0.0%	24.5%
	9	11712 0.3%	5148 0.1%	6409 0.2%	20921 0.5%	7736 0.2%	8053 0.2%	12346 0.3%	19421 0.5%	57333 1.5%	17121 0.4%	28383 0.7%	8854 0.2%	5127 0.1%	5118 0.1%	6713 0.2%	4999 0.1%	19187 0.5%	23.4%
	10	1800 0.0%	5042 0.1%	1262 0.0%	3279 0.1%	4950 0.1%	945 0.0%	6097 0.2%	2700 0.1%	14425 0.4%	28399 0.7%	4050 0.1%	6585 0.2%	2100 0.1%	7050 0.2%	450 0.0%	934 0.0%	987 0.0%	31.2%
	11	7319 0.2%	7473 0.2%	11364 0.3%	6812 0.2%	900 0.0%	750 0.0%	3150 0.1%	3399 0.1%	14671 0.4%	14508 0.4%	32513 0.8%	5460 0.1%	3872 0.1%	6397 0.2%	7650 0.2%	5060 0.1%	2318 0.1%	24.3%
	12	10904 0.3%	3150 0.1%	10350 0.3%	13350 0.3%	7999 0.2%	5100 0.1%	6961 0.2%	18132 0.5%	14550 0.4%	20421 0.5%	32682 0.8%	54054 1.4%	20540 0.5%	12860 0.3%	3150 0.1%	23821 0.6%	19085 0.5%	19.5%
	13	8850 0.2%	6783 0.2%	4861 0.1%	7111 0.2%	7050 0.2%	7457 0.2%	10800 0.3%	11250 0.3%	11682 0.3%	1765 0.0%	17626 0.5%	28862 0.7%	33929 0.9%	6234 0.2%	13438 0.3%	5622 0.1%	4984 0.1%	18.0%
	14	6836 0.2%	943 0.0%	3672 0.1%	7800 0.2%	25079 0.6%	19363 0.5%	20976 0.5%	20711 0.5%	19414 0.5%	4052 0.1%	17285 0.4%	8009 0.2%	23149 0.6%	38479 1.0%	9288 0.2%	4800 0.1%	2850 0.1%	16.5%
	15	12029 0.3%	5172 0.1%	5740 0.1%	8003 0.2%	15505 0.4%	15268 0.4%	26756 0.7%	10919 0.3%	28540 0.7%	9198 0.2%	18976 0.5%	7091 0.2%	9019 0.2%	27605 0.7%	51857 1.3%	7649 0.2%	2173 0.1%	19.8%
	16	36573 0.9%	29970 0.8%	29109 0.7%	26720 0.7%	31274 0.8%	26241 0.7%	27775 0.7%	37301 1.0%	34980 0.9%	42785 1.1%	32548 0.8%	46503 1.2%	32118 0.8%	42205 1.1%	56591 1.5%	92953 2.4%	55162 1.4%	13.7%
	17	28154 0.7%	7350 0.2%	12371 0.3%	16088 0.4%	9000 0.2%	10200 0.3%	600 0.0%	7500 0.2%	16500 0.4%	20024 0.5%	13200 0.3%	13314 0.3%	19063 0.5%	12621 0.3%	23219 0.6%	37135 1.0%	71366 1.8%	22.5%
	7.2%	17.7%	26.5%	18.4%	12.4%	29.2%	28.0%	8.9%	22.9%	13.7%	13.1%	24.1%	16.1%	18.0%	23.4%	43.0%	35.6%	20.9%	
	92.8%	82.3%	73.5%	81.6%	87.6%	70.8%	72.0%	91.1%	77.1%	86.3%	86.9%	75.9%	83.9%	82.0%	76.6%	57.0%	64.4%	79.1%	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17		

Movimento Executado

Fonte: do autor

Figura 37 - Sobreposição das matrizes de confusão geradas para sujeitos amputados após 10 s de treinamento.

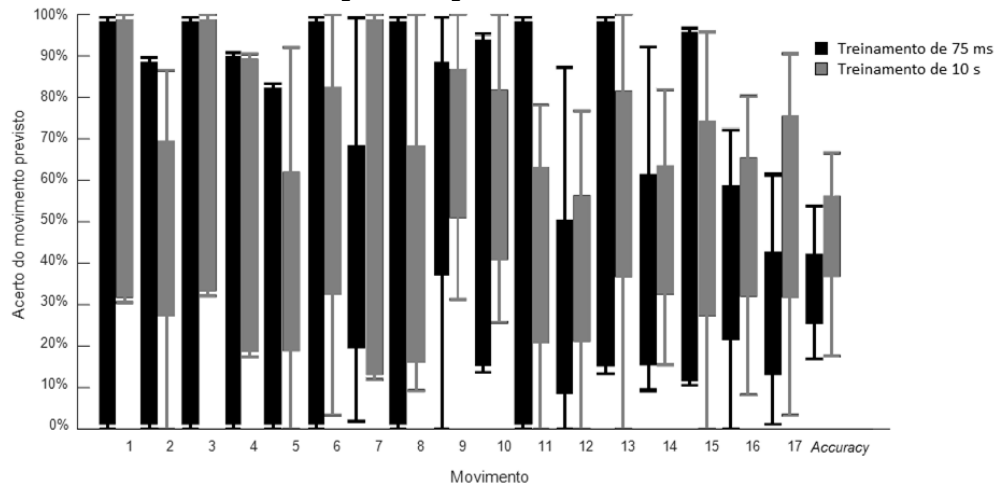
Movimento Previsto	1	57425 1.5%	20239 0.5%	25607 0.7%	3218 0.1%	300 0.0%	750 0.0%	1714 0.0%	2627 0.1%	150 0.0%	450 0.0%	450 0.0%	450 0.0%	3118 0.1%	410 0.0%	3000 0.1%	417 0.0%	8131 0.2%	44.7%
	2	25204 0.6%	77447 2.0%	17986 0.5%	8513 0.2%	5155 0.1%	450 0.0%	7756 0.2%	6240 0.2%	300 0.0%	600 0.0%	5850 0.2%	2654 0.1%	7731 0.2%	7675 0.2%	300 0.0%	1045 0.0%	2100 0.1%	43.8%
	3	28607 0.7%	17140 0.4%	32034 0.8%	11621 0.3%	16669 0.4%	1050 0.0%	3750 0.1%	5700 0.1%	2299 0.1%	3750 0.1%	1800 0.0%	2700 0.1%	7874 0.2%	2064 0.1%	750 0.0%	1200 0.0%	8927 0.2%	21.7%
	4	12356 0.3%	25425 0.7%	27523 0.7%	83520 2.1%	5846 0.2%	6806 0.2%	1050 0.0%	5445 0.1%	8336 0.2%	7752 0.2%	8129 0.2%	7884 0.2%	2755 0.1%	11839 0.3%	4521 0.1%	1901 0.0%	900 0.0%	37.6%
	5	900 0.0%	5769 0.1%	17850 0.5%	27578 0.7%	59805 1.5%	5700 0.1%	2486 0.1%	3900 0.1%	3000 0.1%	4650 0.2%	7350 0.2%	6300 0.2%	9316 0.2%	4350 0.1%	2550 0.1%	1350 0.0%	600 0.0%	36.6%
	6	10500 0.3%	5257 0.1%	4272 0.1%	6222 0.2%	24764 0.6%	69479 1.8%	8032 0.2%	7267 0.2%	3862 0.1%	4957 0.1%	287 0.0%	648 0.0%	8100 0.2%	2012 0.1%	10871 0.3%	9538 0.2%	4128 0.1%	38.6%
	7	900 0.0%	8954 0.2%	3657 0.1%	1320 0.0%	13831 0.4%	19761 0.5%	75583 1.9%	15728 0.4%	3450 0.1%	1022 0.0%	2025 0.1%	900 0.0%	450 0.0%	0 0.0%	0 0.0%	300 0.0%	55 0.0%	51.1%
	8	5286 0.1%	4437 0.1%	6390 0.2%	2702 0.1%	6690 0.2%	13369 0.3%	28279 0.7%	50947 1.3%	5850 0.2%	10114 0.3%	6750 0.2%	3938 0.1%	5550 0.1%	4950 0.1%	5100 0.1%	1200 0.0%	1800 0.0%	31.2%
	9	7862 0.2%	2339 0.1%	4726 0.1%	7129 0.2%	3130 0.1%	3465 0.1%	20736 0.5%	14652 0.4%	75904 2.0%	12432 0.3%	24612 0.6%	5565 0.1%	3835 0.1%	4728 0.1%	8549 0.2%	3293 0.1%	10202 0.3%	35.6%
	10	0 0.0%	2250 0.1%	4122 0.1%	9300 0.2%	5100 0.1%	11177 0.3%	15628 0.4%	6000 0.2%	29510 0.8%	83296 2.1%	11054 0.3%	9672 0.2%	4637 0.1%	7499 0.2%	5850 0.2%	750 0.0%	2359 0.1%	40.0%
	11	9150 0.2%	4437 0.1%	5928 0.2%	6497 0.2%	8376 0.2%	7523 0.2%	8325 0.2%	10372 0.3%	13472 0.3%	5198 0.1%	68606 1.8%	20937 0.5%	9977 0.3%	6750 0.2%	8945 0.2%	7040 0.2%	4896 0.1%	33.2%
	12	2996 0.1%	8227 0.2%	12580 0.3%	8700 0.2%	8403 0.2%	2100 0.1%	8837 0.2%	16660 0.4%	12902 0.3%	13149 0.3%	23542 0.6%	66467 1.7%	16790 0.4%	4936 0.1%	1950 0.1%	11409 0.3%	6000 0.2%	29.5%
	13	12758 0.3%	16528 0.4%	4650 0.1%	9257 0.2%	17086 0.4%	16895 0.4%	15854 0.4%	19600 0.5%	15070 0.4%	1950 0.1%	28177 0.7%	21062 0.5%	62540 1.6%	15564 0.4%	16154 0.4%	10650 0.3%	5505 0.1%	21.6%
	14	3805 0.1%	2343 0.1%	23694 0.6%	13384 0.3%	11013 0.3%	5400 0.1%	12114 0.3%	13512 0.3%	8972 0.2%	1903 0.0%	9882 0.3%	7469 0.2%	21890 0.6%	58627 1.5%	13739 0.4%	9878 0.3%	3535 0.1%	26.5%
	15	17112 0.4%	6906 0.2%	5897 0.2%	4365 0.1%	13219 0.3%	27333 0.7%	23705 0.6%	38649 1.0%	23299 0.6%	11650 0.3%	13824 0.4%	10617 0.3%	14568 0.4%	45773 1.2%	388622 2.3%	26090 0.7%	13469 0.3%	23.0%
	16	21470 0.6%	32070 0.8%	28074 0.7%	21939 0.6%	15356 0.4%	19823 0.5%	17843 0.5%	19501 0.5%	31328 0.8%	31246 0.8%	23436 0.6%	30984 0.8%	17363 0.4%	25561 0.7%	34694 0.9%	88077 2.3%	36206 0.9%	17.8%
	17	18874 0.5%	5317 0.1%	7905 0.2%	9803 0.3%	14100 0.4%	6750 0.2%	5991 0.2%	5352 0.1%	12614 0.3%	13947 0.4%	12797 0.3%	25723 0.7%	14477 0.4%	11347 0.3%	16201 0.4%	42024 1.1%	91592 2.4%	29.1%
	24.4%	31.6%	13.8%	35.5%	26.1%	31.9%	29.3%	21.0%	30.3%	40.0%	27.6%	29.7%	29.6%	27.4%	40.0%	40.7%	45.7%	30.6%	
	75.6%	68.4%	86.2%	64.5%	73.9%	68.1%	70.7%	79.0%	69.7%	60.0%	72.4%	70.3%	70.4%	72.6%	60.0%	59.3%	54.3%	69.4%	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17		
	Movimento Executado																		

Fonte: do autor

As Figuras 38 e 39 apresentam respectivamente a comparação dos valores preditivos positivos e a comparação dos valores de *sensitivity* observados para os sujeitos não amputados após treinamento de 75 ms e treinamento de 10 s. Pode-se observar nos gráficos os valores máximos e mínimos obtidos e a faixa de um desvio padrão para cima e para baixo em torno da média. Nestas comparações é possível perceber que os valores preditivos positivos para um mesmo movimento pode variar de 0% a quase 100% em treinamentos distintos aos 75 ms. Com a evolução do treinamento há uma tendência desta dispersão diminuir e a média aumentar. A *sensitivity* também apresenta um aumento em suas médias com a evolução do treinamento, mas a diminuição da dispersão não foi observada na maioria dos casos.

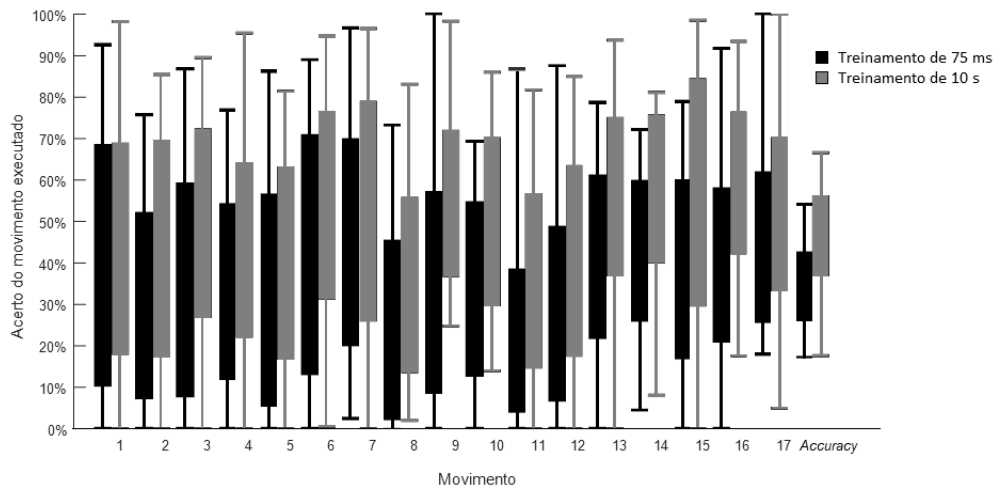
As mesmas comparações foram feitas para sujeitos amputados e encontram-se nas Figuras 40 e 41. Para estes sujeitos o aumento das médias na maioria dos movimentos é clara, mas nada pode ser concluído quanto a dispersão.

Figura 38 - Valores preditivos positivos para cada movimento executado por sujeito não amputado após 75 ms e 10 s de treinamento.



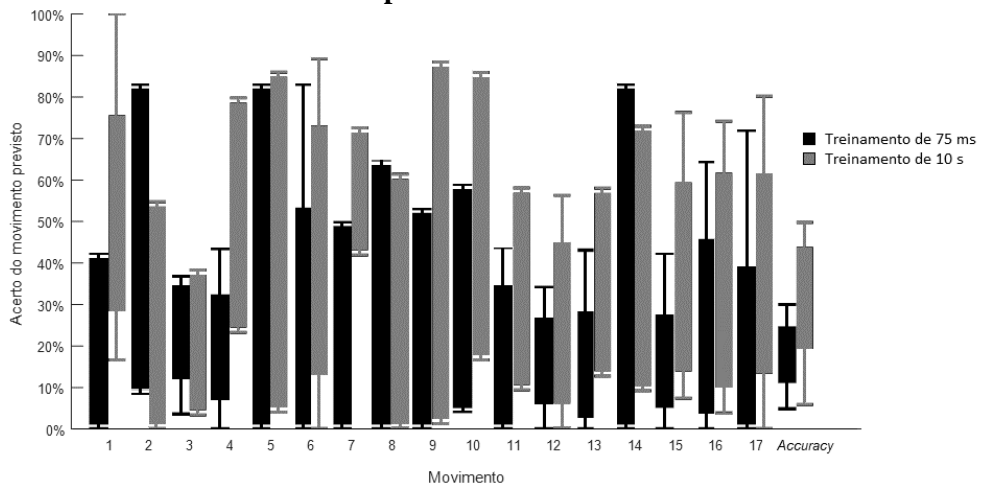
Fonte: do autor

Figura 39 - Sensitivity para cada movimento executado por sujeito não amputado após 75 ms e 10 s de treinamento.



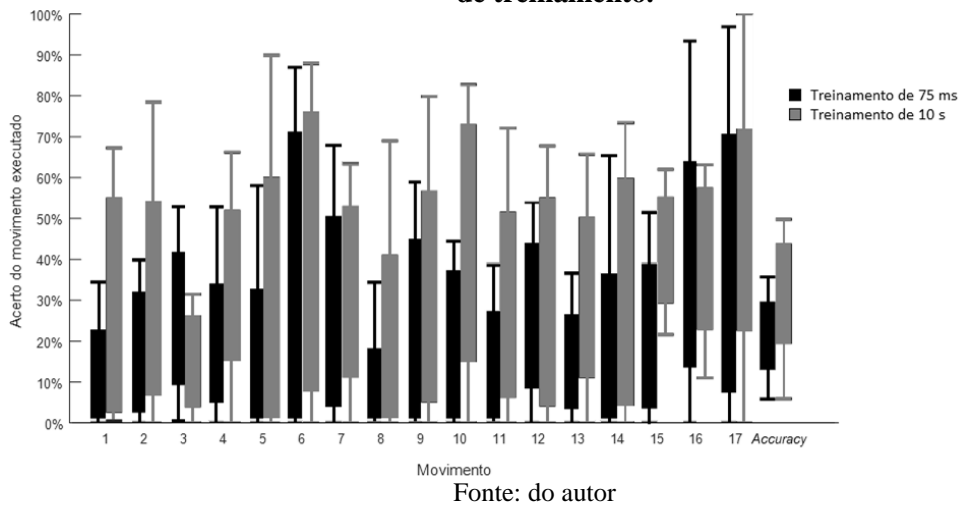
Fonte: do autor

Figura 40 - Valores preditivos positivos para cada movimento executado por sujeito amputado após 75 ms e 10 s de treinamento.



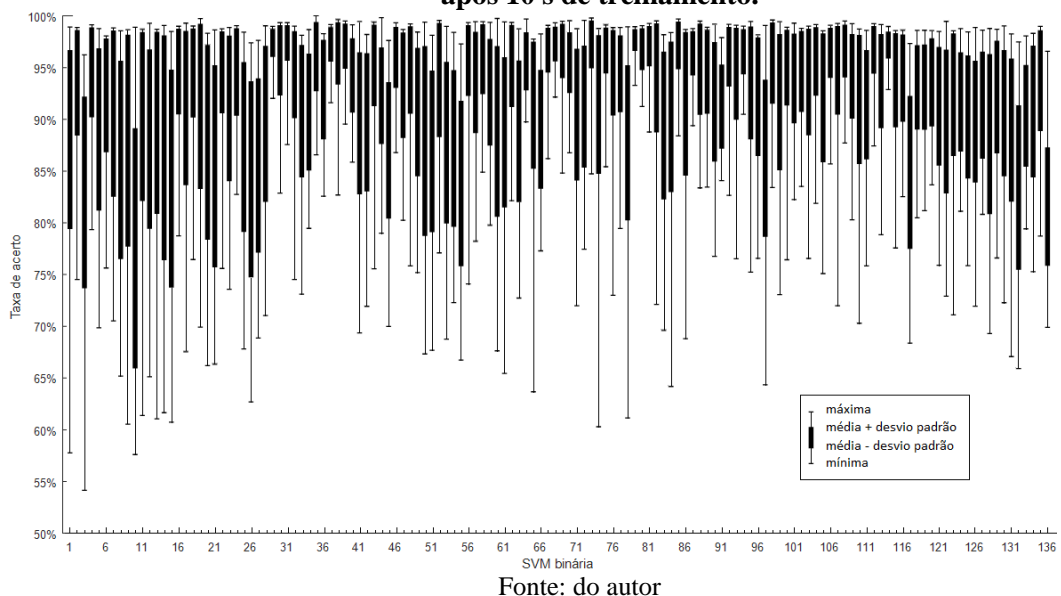
Fonte: do autor

Figura 41 - Sensitivity para cada movimento executado por sujeito amputado após 75 ms e 10 s de treinamento.



A alta dispersão apresentada tanto no valor de *sensitivity* quanto no valor de preditividade positiva em todos os movimentos pode estar relacionada com o sujeito que executou os movimentos, movimentos com características parecidas, aprendizado das SVMs ou método de classificação. Essa dificuldade em classificar os movimentos refletiu na taxa de acerto que apresentou média de 46,5%, valor máximo de 66,6% e valor mínimo de 17,6% para sujeitos não-amputados e valores ainda menores para sujeitos amputados (31,6%, 49,8% e 5,9%) após 10 segundos de treinamento. Informações de como foi o treinamento das SVMs que geram estes resultados, podem ser extraídas das taxas de acerto do algoritmo de validação. A Figura 42 ilustra a taxa de acerto das 136 SVMs binárias na etapa de validação com valores máximos e mínimos observados e faixa de um desvio padrão para cima e para baixo em torno da média após 10 s de treinamento.

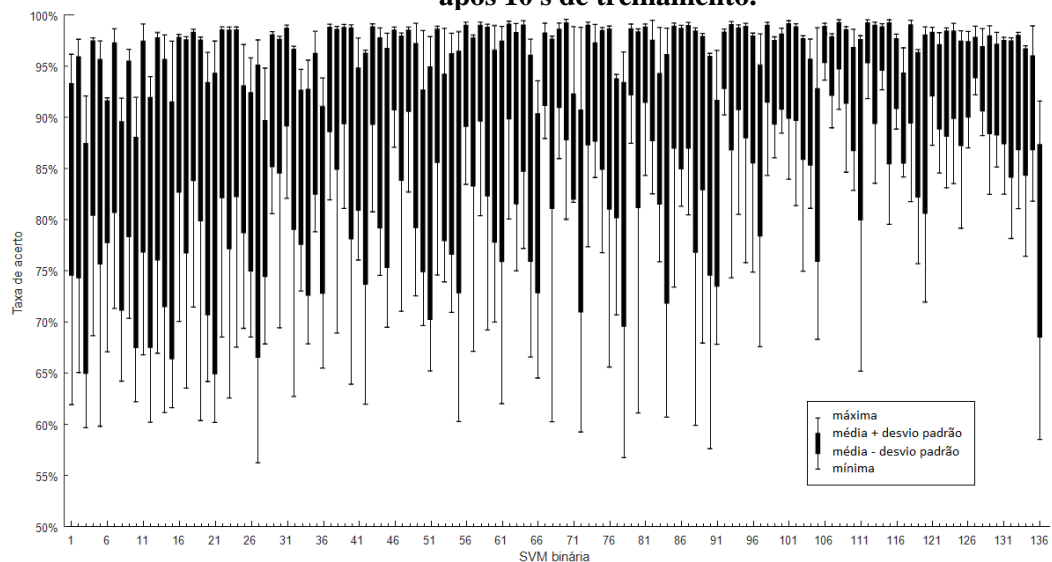
Figura 42 - Taxa de acerto de cada uma das 136 SVMs no ensaio com 40 sujeitos não-amputados após 10 s de treinamento.



Na etapa de validação, movimentos puderam ser diferenciados com taxa de acerto média de 92,1% sendo que a taxa da SVM 10 (movimento 4 × movimento 5) apresentou a menor média (77,5%) e a SVM 79 (movimento 1 × movimento 14) apresentou a maior média (97,7%). Agrupando-se as SVMs em SVMs que compararam movimentos de dedos, SVMs que compararam movimentos de pulso e SVMs que compararam movimentos de dedos com movimentos de pulso, verificou-se que o algoritmo teve maior dificuldade em diferenciar os movimentos de dedos (taxa de acerto média de 89,5%), e maior facilidade em diferenciar movimentos de dedos de movimentos de pulso (taxa de acerto média de 93,5%). A comparação de movimentos de pulso atingiu taxa de acerto média de 92,0%. Acredita-se que esse resultado se deve principalmente à diferença de amplitude dos sinais de pulso e dedos, e aos músculos responsáveis por movimentos de dedos serem músculos mais internos, de mais difícil captação pelos eletrodos de superfície.

Quando os 11 sujeitos amputados da base de dados Ninapro foram analisados, a taxa de acerto média foi menor, 89,9% sendo que a taxa da SVM 3 (movimento 2 × movimento 3) apresentou a menor média (76,2%) e a SVM 112 (movimento 7 × movimento 16) apresentou a maior média (97,5%). A Figura 43 ilustra a taxa de acerto das 136 SVMs binárias na etapa de validação com informação de valores máximos e mínimos observados e faixa de um desvio padrão para cima e para baixo em torno da média para os sujeitos amputados. Para estes sujeitos também foi verificado que a tarefa mais difícil é comparar movimentos de dedos (taxa de acerto média de 85,5%), e a tarefa mais fácil é comparar movimentos de dedos com movimentos de pulso (taxa de acerto média de 91,8%). A comparação de movimentos de pulso atingiu taxa de acerto média de 90,0%.

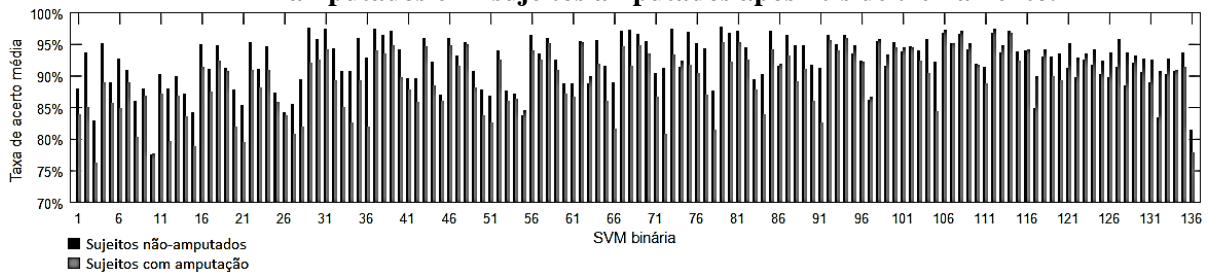
Figura 43 - Taxa de acerto de cada uma das 136 SVMs no ensaio com 11 sujeitos amputados após 10 s de treinamento.



Fonte: do autor

Na comparação das médias obtidas para sujeitos amputados e não-amputados (Figura 44) verificou-se que a maioria dos movimentos foram melhor classificados quando o sinal foi adquirido de um sujeito não-amputado, porém em algumas comparações os sujeitos amputados geraram resultados melhores.

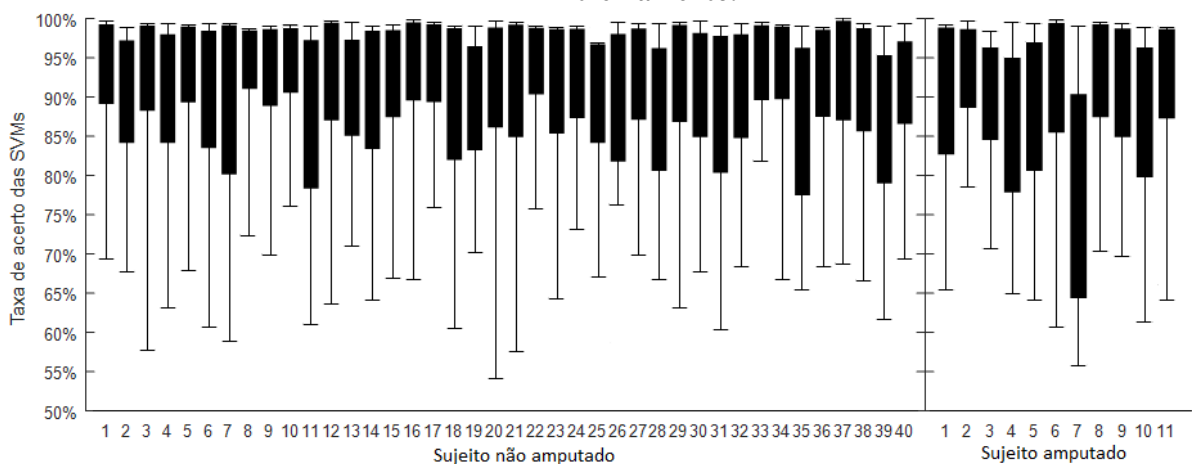
Figura 44 - Taxa de acerto média de cada uma das 136 SVMs no ensaio com 40 sujeitos não amputados e 11 sujeitos amputados após 10 s de treinamento.



Fonte: do autor

Pôde-se observar também que as SVMs tiveram taxas de acerto diferentes em cada treinamento. Para descobrir se essa variação pode estar relacionada com os sujeitos que executaram os movimentos, foram comparadas as taxas de acerto das SVMs para cada sujeito (Figura 45).

Figura 45 - Taxa de acerto das SVMs obtida para cada um dos 51 sujeitos após 10 s de treinamento.

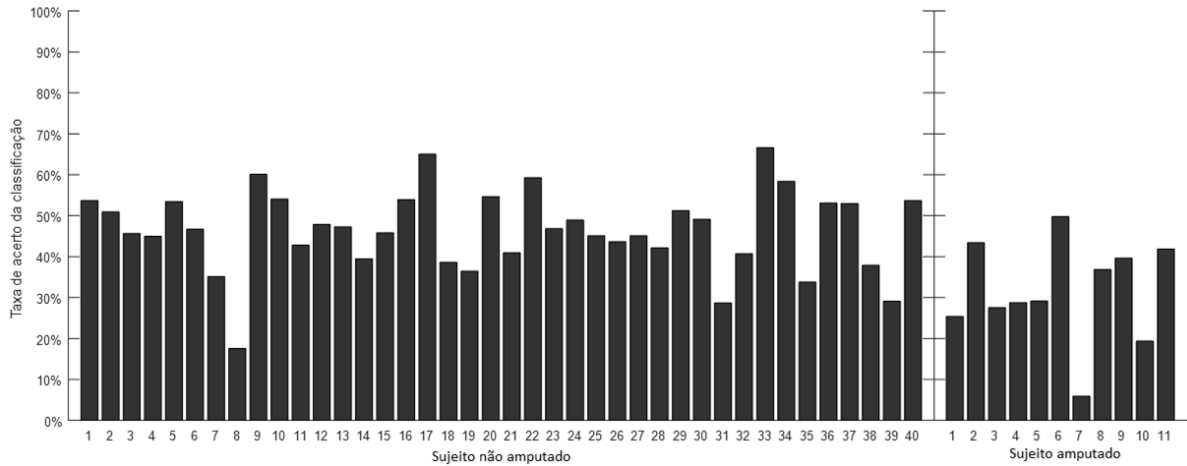


Fonte: do autor

Desta comparação é possível perceber que existem sujeitos que comprometem a média das taxas de acerto das SVMs por apresentarem taxa média baixa: o sujeito com menor média entre os não amputados é o sujeito 35 (86,9%) e dos amputados é o sujeito 7 (77,5%). Mas este não é o único fator, todos os sujeitos possuem pelo menos uma SVM gerando taxa de acerto abaixo de 80%, o que sugere ser possível melhorar a taxa de acerto removendo alguns movimentos e consequentemente algumas SVMs. Também foi observado que aumentar a taxa

de acerto de média das SVMs não necessariamente refletirá no aumento da taxa de acerto da classificação dos movimentos: o treinamento para o sujeito não amputado 35, por exemplo, possui a menor média na taxa de acerto de SVMs (Figura 45) porém a classificação para esse sujeito é melhor que a classificação para outros 3 sujeitos, como mostra a Figura 46.

Figura 46 - Taxa de acerto da classificação para cada um dos 51 sujeitos após 10 s de treinamento.



Fonte: do autor

4.2.2 Ensaios com aprendizado de 10 movimentos

A partir das análises da Seção 4.2.1, optou-se por utilizar no terceiro e quarto treinamento um conjunto de dez movimentos escolhidos por eliminação baseada nas métricas *sensitivity* e valores preditivos positivos calculados como descrito na Seção 3.2.2. O grupo de movimentos reduzido é formado por:

Tabela 15 - Conjunto reduzido de movimentos de dedos e pulsos utilizados nos ensaios com 10 movimentos.

Nº do movimento	Movimento
1	polegar pra cima
3	flexão dos dedos anelar e mínimo e extensão dos demais
4	polegar tocando base do dedo mínimo
6	dedos flexionados em conjunto
7	apontando com indicador
9	supinação do pulso (eixo: dedo médio)
10	pronação do pulso (eixo: dedo médio)
13	flexão do pulso
14	extensão do pulso
16	deslocamento ulnar do pulso

Nos ensaios com número reduzido de movimentos, novamente foram realizadas as sobreposições das matrizes de confusão resultando nas Figuras 47 a 50.

Figura 47 - Sobreposição das matrizes de confusão geradas para sujeitos não amputados após 75 ms de treinamento do conjunto reduzido de movimentos.

Movimento Previsto	1	495373 6.8%	84410 1.2%	59019 0.8%	46981 0.6%	71013 1.0%	33209 0.5%	25483 0.3%	13636 0.2%	18356 0.3%	33059 0.5%	56.3% 43.7%
	3	73131 1.0%	286979 3.9%	49703 0.7%	48960 0.7%	96742 1.3%	14267 0.2%	11379 0.2%	16332 0.2%	27041 0.4%	12023 0.2%	45.1% 54.9%
	4	43823 0.6%	85061 1.2%	304207 4.2%	55387 0.8%	70399 1.0%	31249 0.4%	34819 0.5%	34239 0.5%	49795 0.7%	30437 0.4%	41.1% 58.9%
	6	41341 0.6%	38258 0.5%	36634 0.5%	236191 3.2%	48006 0.7%	25549 0.4%	25834 0.4%	11288 0.2%	17155 0.2%	10881 0.1%	48.1% 51.9%
	7	115308 1.6%	62964 0.9%	67786 0.9%	70632 1.0%	463285 6.4%	36425 0.5%	41223 0.6%	28441 0.4%	65173 0.9%	33856 0.5%	47.0% 53.0%
	9	2864 0.0%	2086 0.0%	27581 0.4%	1946 0.0%	21606 0.3%	305500 4.2%	110687 1.5%	15010 0.2%	13299 0.2%	17012 0.2%	59.0% 41.0%
	10	6361 0.1%	1368 0.0%	12814 0.2%	10424 0.1%	17265 0.2%	148224 2.0%	320969 4.4%	19661 0.3%	16181 0.2%	23946 0.3%	55.6% 44.4%
	13	65138 0.9%	37429 0.5%	42125 0.6%	33928 0.5%	39646 0.5%	47035 0.6%	42453 0.6%	247296 3.4%	37047 0.5%	65031 0.9%	37.6% 62.4%
	14	26144 0.4%	69198 0.9%	81484 1.1%	57292 0.8%	20700 0.3%	57461 0.8%	44831 0.6%	69392 1.0%	265754 3.6%	35767 0.5%	36.5% 63.5%
	16	90544 1.2%	68200 0.9%	55133 0.8%	53719 0.7%	62508 0.9%	77751 1.1%	98601 1.4%	109107 1.5%	47649 0.7%	415676 5.7%	38.5% 61.5%
		51.6% 48.4%	39.0% 61.0%	41.3% 58.7%	38.4% 61.6%	50.8% 49.2%	39.3% 60.7%	42.4% 57.6%	43.8% 56.2%	47.7% 52.3%	61.3% 38.7%	45.8% 54.2%
		1	3	4	6	7	9	10	13	14	16	
		Movimento Executado										

Fonte: do autor

Figura 48 - Sobreposição das matrizes de confusão geradas para sujeitos amputados após 75 ms de treinamento do conjunto reduzido de movimentos.

Movimento Previsto	1	20467 0.9%	5732 0.3%	17414 0.8%	21857 1.0%	6900 0.3%	4209 0.2%	1721 0.1%	2707 0.1%	7518 0.3%	6860 0.3%	21.5% 78.5%
	3	52832 2.3%	80521 3.5%	29019 1.3%	4200 0.2%	17100 0.8%	5073 0.2%	2400 0.1%	12974 0.6%	6922 0.3%	7906 0.3%	36.8% 63.2%
	4	11913 0.5%	29430 1.3%	57997 2.5%	13484 0.6%	8939 0.4%	12991 0.6%	11169 0.5%	14158 0.6%	18926 0.8%	9957 0.4%	30.7% 69.3%
	6	23119 1.0%	17889 0.8%	16058 0.7%	67586 3.0%	24626 1.1%	15244 0.7%	12609 0.6%	18654 0.8%	15038 0.7%	7200 0.3%	31.0% 69.0%
	7	13032 0.6%	12565 0.6%	8622 0.4%	14158 0.6%	77984 3.4%	5656 0.2%	13898 0.6%	12355 0.5%	8622 0.4%	3549 0.2%	45.8% 54.2%
	9	19337 0.8%	8235 0.4%	18435 0.8%	6519 0.3%	24867 1.1%	71465 3.1%	13330 0.6%	5795 0.3%	4777 0.2%	7943 0.3%	39.5% 60.5%
	10	5700 0.3%	12157 0.5%	19791 0.9%	12641 0.6%	22514 1.0%	32730 1.4%	75105 3.3%	14167 0.6%	18754 0.8%	10545 0.5%	33.5% 66.5%
	13	27701 1.2%	21871 1.0%	20112 0.9%	23896 1.0%	26100 1.1%	30697 1.3%	6322 0.3%	57858 2.5%	33237 1.5%	21081 0.9%	21.5% 78.5%
	14	17240 0.8%	5497 0.2%	9533 0.4%	17865 0.8%	12450 0.5%	22723 1.0%	14262 0.6%	31240 1.4%	45400 2.0%	14780 0.6%	23.8% 76.2%
	16	43826 1.9%	38999 1.7%	38087 1.7%	35625 1.6%	36170 1.6%	49530 2.2%	57181 2.5%	41063 1.8%	54786 2.4%	126302 5.5%	24.2% 75.8%
	8.7% 91.3%	34.6% 65.4%	24.7% 75.3%	31.0% 69.0%	30.3% 69.7%	28.5% 71.5%	36.1% 63.9%	27.4% 72.6%	21.2% 78.8%	58.4% 41.6%	29.9% 70.1%	
	1	3	4	6	7	9	10	13	14	16		
	Movimento Executado											

Fonte: do autor

Figura 49 - Sobreposição das matrizes de confusão geradas para sujeitos não amputados após 10 s de treinamento do conjunto reduzido de movimentos.

1	515241 7.1%	54944 0.8%	32191 0.4%	32390 0.4%	47844 0.7%	15353 0.2%	11867 0.2%	12154 0.2%	7934 0.1%	18835 0.3%	68.8% 31.2%
3	43309 0.6%	360008 4.9%	34240 0.5%	23517 0.3%	40288 0.6%	7769 0.1%	4650 0.1%	4266 0.1%	4085 0.1%	7305 0.1%	68.0% 32.0%
4	23287 0.3%	64326 0.9%	360792 4.9%	48151 0.7%	83821 1.1%	18370 0.3%	27045 0.4%	33442 0.5%	37578 0.5%	16959 0.2%	50.5% 49.5%
6	44393 0.6%	41363 0.6%	34466 0.5%	310015 4.3%	28294 0.4%	33433 0.5%	20914 0.3%	7440 0.1%	7350 0.1%	19382 0.3%	56.7% 43.3%
7	103338 1.4%	70459 1.0%	63399 0.9%	71886 1.0%	510625 7.0%	31422 0.4%	30126 0.4%	13187 0.2%	32533 0.4%	16687 0.2%	54.1% 45.9%
9	9412 0.1%	4769 0.1%	32321 0.4%	12234 0.2%	19642 0.3%	416448 5.7%	86991 1.2%	7212 0.1%	10955 0.2%	14592 0.2%	67.8% 32.2%
10	15619 0.2%	20583 0.3%	30209 0.4%	19878 0.3%	37027 0.5%	151759 2.1%	460880 6.3%	37909 0.5%	38233 0.5%	41275 0.6%	54.0% 46.0%
13	81471 1.1%	40692 0.6%	24022 0.3%	39113 0.5%	44626 0.6%	32003 0.4%	25673 0.4%	287791 3.9%	43239 0.6%	56692 0.8%	42.6% 57.4%
14	40356 0.6%	29879 0.4%	51969 0.7%	30428 0.4%	41949 0.6%	16102 0.2%	37322 0.5%	81344 1.1%	338436 4.6%	41921 0.6%	47.7% 52.3%
16	83601 1.1%	48930 0.7%	72877 1.0%	27848 0.4%	57054 0.8%	54011 0.7%	50811 0.7%	79657 1.1%	37107 0.5%	444040 6.1%	46.5% 53.5%
	53.7% 46.3%	48.9% 51.1%	49.0% 51.0%	50.4% 49.6%	56.0% 44.0%	53.6% 46.4%	60.9% 39.1%	51.0% 49.0%	60.7% 39.3%	65.5% 34.5%	54.9% 45.1%
	1	3	4	6	7	9	10	13	14	16	

Fonte: do autor

Figura 50 - Sobreposição das matrizes de confusão geradas para sujeitos amputados após 10 s de treinamento do conjunto reduzido de movimentos.

1	66967 2.9%	39155 1.7%	12109 0.5%	8444 0.4%	10597 0.5%	2700 0.1%	5284 0.2%	3721 0.2%	450 0.0%	2926 0.1%	44.0% 56.0%
3	15939 0.7%	58455 2.6%	19332 0.8%	4500 0.2%	3300 0.1%	1050 0.0%	4500 0.2%	9000 0.4%	2044 0.1%	10350 0.5%	45.5% 54.5%
4	31808 1.4%	36043 1.6%	94560 4.2%	21347 0.9%	13260 0.6%	12999 0.6%	13716 0.6%	4800 0.2%	12245 0.5%	5097 0.2%	38.5% 61.5%
6	25350 1.1%	11534 0.5%	7458 0.3%	77469 3.4%	23043 1.0%	5032 0.2%	3450 0.2%	11400 0.5%	5063 0.2%	6450 0.3%	44.0% 56.0%
7	9312 0.4%	3995 0.2%	7950 0.3%	18237 0.8%	73429 3.2%	8785 0.4%	2846 0.1%	750 0.0%	2837 0.1%	900 0.0%	56.9% 43.1%
9	14758 0.6%	5576 0.2%	8403 0.4%	4311 0.2%	43742 1.9%	74797 3.3%	23366 1.0%	14819 0.7%	18673 0.8%	11174 0.5%	34.1% 65.9%
10	2700 0.1%	4950 0.2%	6450 0.3%	5518 0.2%	8521 0.4%	42187 1.9%	95446 4.2%	11280 0.5%	11100 0.5%	7274 0.3%	48.8% 51.2%
13	10569 0.5%	12009 0.5%	21383 0.9%	23692 1.0%	22117 1.0%	20261 0.9%	7759 0.3%	77021 3.4%	29548 1.3%	16170 0.7%	32.0% 68.0%
14	12076 0.5%	11337 0.5%	16365 0.7%	28938 1.3%	23991 1.1%	33106 1.5%	17749 0.8%	32682 1.4%	71570 3.1%	28671 1.3%	25.9% 74.1%
16	45726 2.0%	49736 2.2%	41025 1.8%	25375 1.1%	35617 1.6%	49401 2.2%	33886 1.5%	45466 2.0%	60555 2.7%	127150 5.6%	24.7% 75.3%
	28.5% 71.5%	25.1% 74.9%	40.2% 59.8%	35.6% 64.4%	28.5% 71.5%	29.9% 70.1%	45.9% 54.1%	36.5% 63.5%	33.4% 66.6%	58.8% 41.2%	35.9% 64.1%
	1	3	4	6	7	9	10	13	14	16	

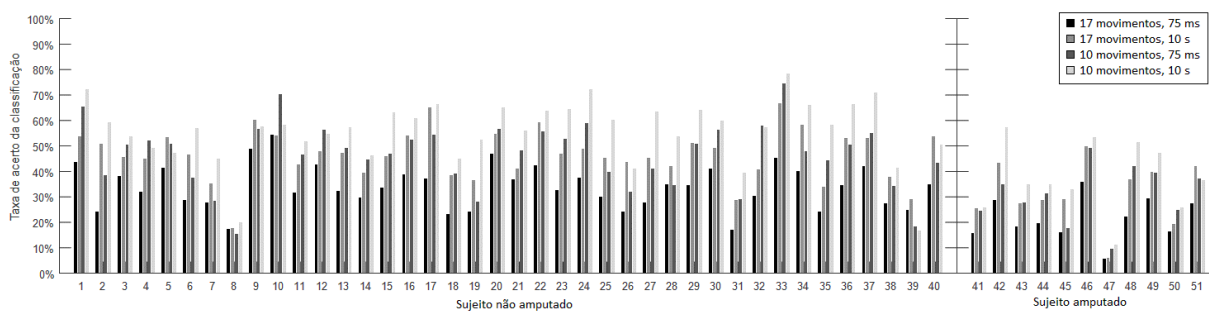
Fonte: do autor

Como nos ensaios anteriores (ensaio 1 e ensaio 2), a taxa de acerto obtida para sujeitos não amputados é mais alta que a taxa de acerto obtida para sujeitos amputados, e os treinamentos de 10 s apresentam taxas de acerto superiores aos treinamentos de 75 ms. A redução de movimentos, porém, aumentou todas estas taxas de acerto: sujeitos não amputados atingiram taxa de acerto média de 45,8% com treinamento de 75 ms e 54,9% para treinamento de 10 s. Para os sujeitos amputados, os valores obtidos foram 29,9% e 35,9% respectivamente.

Cinco sujeitos apresentaram taxas de acerto acima de 70% (ver Figura 51): sujeitos 1, 24 e 37 no treinamento de 10 s, sujeito 10 no treinamento de 75 ms e sujeito 33 em ambos os treinamentos. A Figura 51 mostra ainda que em praticamente todos os casos o treinamento de 10 segundos apresenta taxa de acerto superior ao treinamento de 75 milissegundos, e os melhores resultados são obtidos com o conjunto reduzido de movimentos. Para alguns sujeitos, os movimentos escolhidos não foram os mais adequados, é o caso dos sujeitos 39, 40 e 41 que apresentaram diferença muito pequena na taxa de acerto ou até piora.

Alguns sujeitos apresentaram taxa de acerto muito abaixo da média em todos os ensaios (sujeitos 8, 39 e 47), um indício de que os movimentos podem ter sido mal executados por esses voluntários. Da mesma forma, o sujeito 33 apresenta a melhor taxa de acerto em todos os ensaios, um indício de que a execução dos movimentos por este voluntário gerou sinais mais fáceis de detectar.

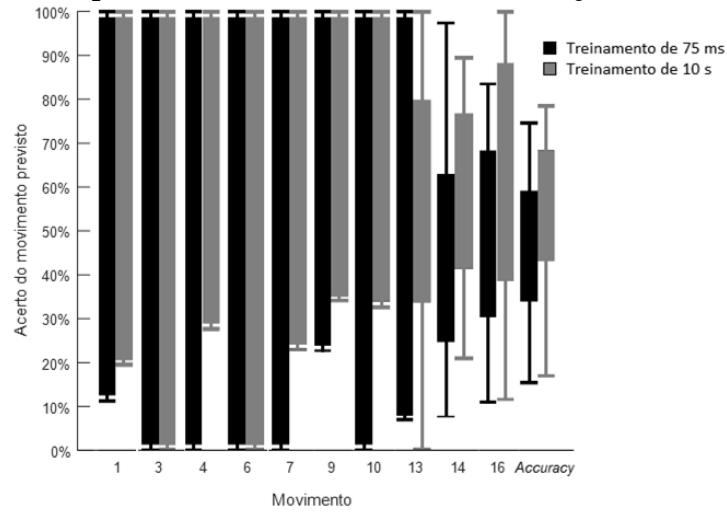
Figura 51 - Taxa de acerto da classificação por sujeito.



Fonte: do autor

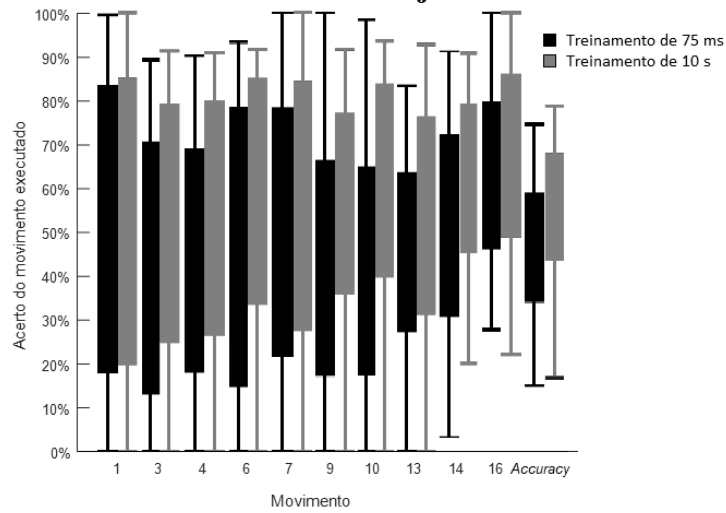
As Figuras 52 a 55 apresentam os gráficos de valores preditivos positivos e *sensitivity* para os ensaios com 10 movimentos. A mesma tendência de diminuição das dispersões dos valores preditivos positivos observadas nos ensaios com 17 movimentos executados por sujeitos não amputados foram observadas nestes gráficos. Também o aumento das médias de *sensitivity* e valores preditivos positivos tanto para sujeitos amputados como não amputados foi observado.

Figura 52 - Valores preditivos positivos para cada movimento executado por sujeito não amputado após 75 ms e 10 s de treinamento do conjunto reduzido de movimentos.



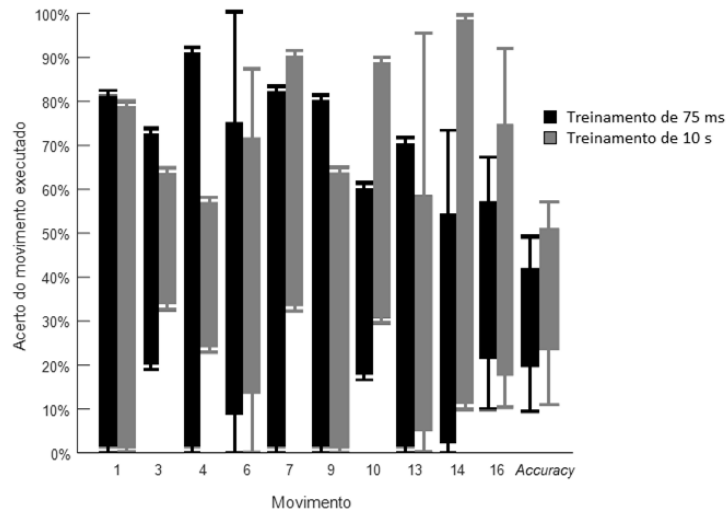
Fonte: do autor

Figura 53 - Sensitivity para cada movimento executado por sujeito não amputado após 75 ms e 10 s de treinamento do conjunto reduzido de movimentos.



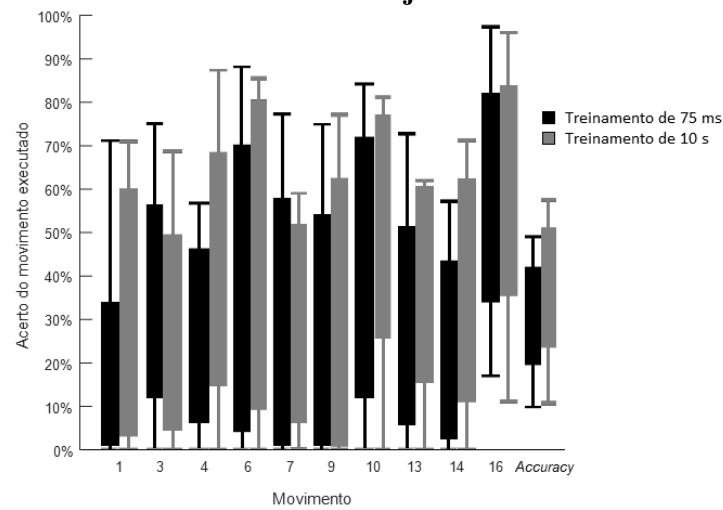
Fonte: do autor

Figura 54 - Valores preditivos positivos para cada movimento executado por sujeito amputado após 75 ms e 10 s de treinamento do conjunto reduzido de movimentos.



Fonte: do autor

Figura 55 - *Sensitivity* para cada movimento executado por sujeito amputado após 75 ms e 10 s de treinamento do conjunto reduzido de movimentos.



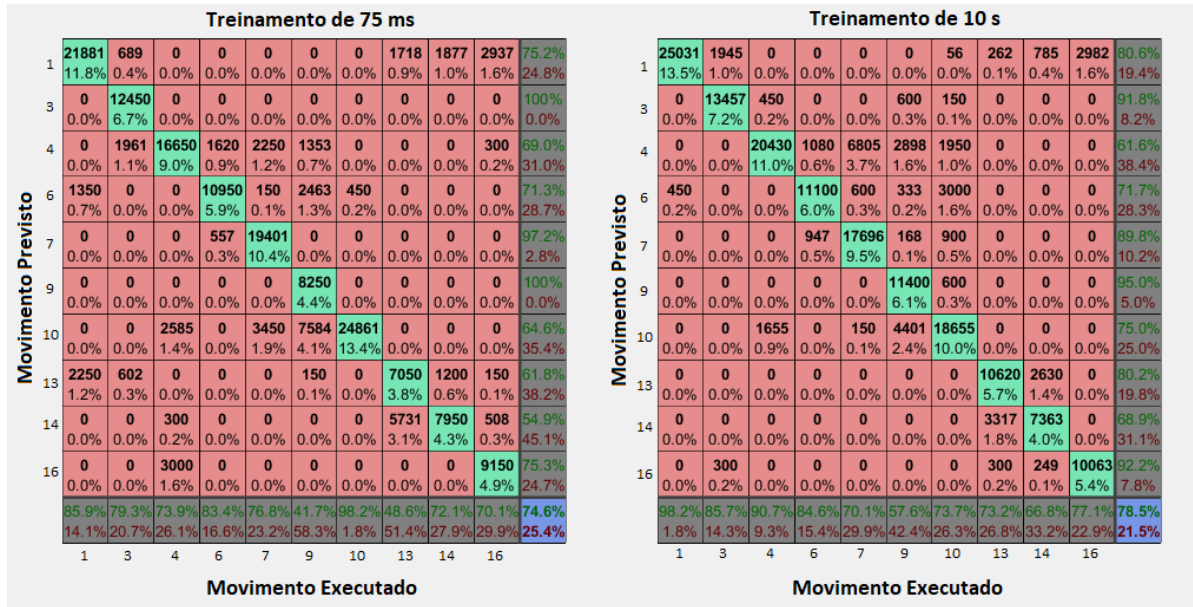
Fonte: do autor

Nas Figuras 53 e 55 são observados vários movimentos com valor mínimo de *sensitivity* igual a zero. Uma possível razão para isso é a baixa taxa de acerto verificada na Figura 51 para os sujeitos 8, 39 e 47. Pode-se perceber ainda que o movimento 16 é o que apresenta maior valor mínimo, média e valor máximo, portanto, o movimento mais fácil de distinguir entre os 10 escolhidos.

4.2.3 Análise dos sujeitos com maior taxa de acerto

O sujeito que apresentou as maiores taxas de acerto foi o sujeito 33 nos ensaios com 10 movimentos. A Figura 56 apresenta a matriz de confusão obtida na classificação dos movimentos executados por este sujeito tanto no treinamento de 75 ms quanto no treinamento de 10 s. Nessas matrizes de confusão é possível perceber uma grande quantidade de zeros nas posições de falsos positivos e falsos negativos, e uma concentração das previsões em verdadeiros positivos. Os erros de previsão estão concentrados em grupos na matriz: movimentos 4, 6, 7, 9 e 10, e movimentos 1, 13, 14 e 16.

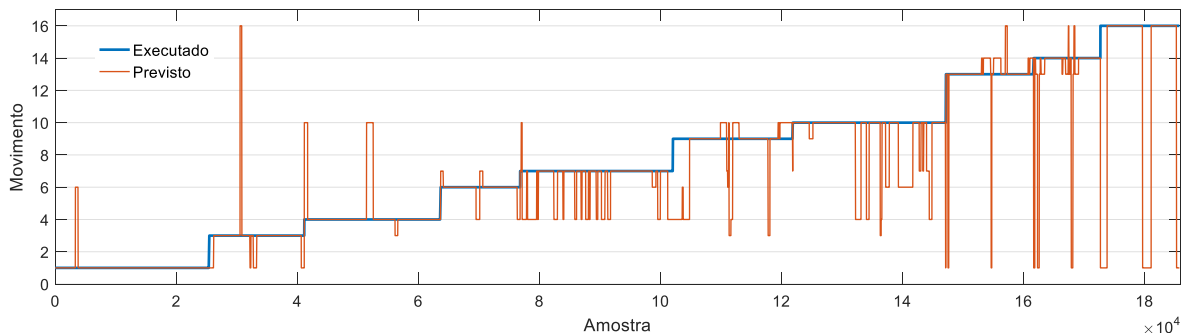
Figura 56 - Matrizes de confusão da classificação do sujeito 33 da base de dados de não amputados com treinamento de 75 ms e com treinamento de 10 s.



Fonte: do autor

Como a execução dos movimentos gerou uma série temporal de amostras que entraram no classificador, a análise dessa classificação no tempo também apresenta resultados interessantes. A Figura 57 ilustra o movimento executado e o movimento previsto após o sistema ter treinado por 10 s na ordem da apresentação das amostras ao classificador. Pode-se perceber que para a maioria dos movimentos houveram três momentos em que os erros se concentraram: início meio e fim.

Figura 57 - Classificação dos movimentos na ordem da apresentação das amostras para o sujeito 33 da base de dados de não-amputados.



Fonte: do autor

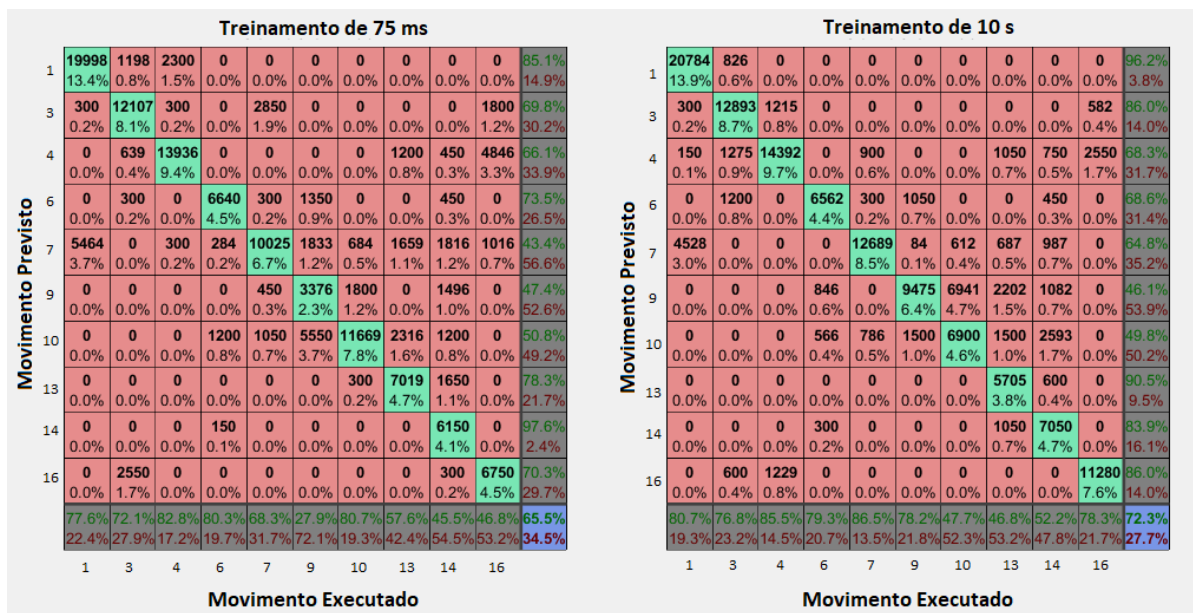
A explicação para este comportamento é que duas repetições de cada movimento foram utilizadas para classificação, uma após a outra. Assim sendo, os movimentos apresentados no gráfico consistem em sair da posição de repouso para executar o movimento, voltar para a

posição de repouso, sair da posição de repouso novamente e finalmente voltar ao repouso. Os erros de classificação estão concentrados nessa parte dinâmica dos movimentos descrita: saída e volta para repouso.

O sujeito 1 apresentou taxa de acerto de 65,5% após 75 ms de treinamento e 72,3% após 10 s de treinamento. A Figura 58 apresenta as matrizes de confusão obtidas para este sujeito. Assim como para o sujeito 33, pode-se observar uma grande quantidade de falsos positivos e falsos negativos com valor zero e alto número de verdadeiros positivos.

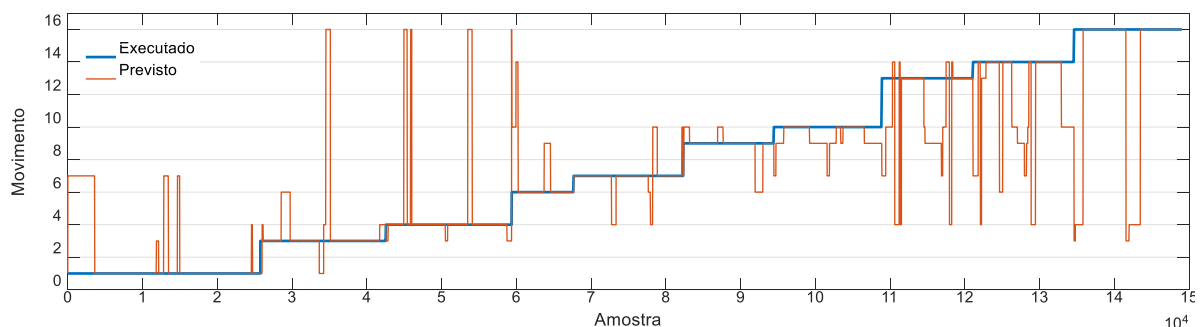
A análise no tempo dos movimentos executados e previstos para o sujeito 1 ilustrada pela Figura 59 apresenta uma concentração dos erros nas regiões dinâmicas do movimento da mesma forma que foi discutido para o sujeito 33. Para este sujeito, porém o número de amostras em erro a cada repetição foi maior, sugerindo uma dinâmica mais lenta ou menos precisa.

Figura 58 - Matrizes de confusão da classificação do sujeito 1 da base de dados de não amputados com treinamento de 75 ms e com treinamento de 10 s.



Fonte: do autor

Figura 59 - Classificação dos movimentos na ordem da apresentação das amostras para o sujeito 1 da base de dados de não-amputados.



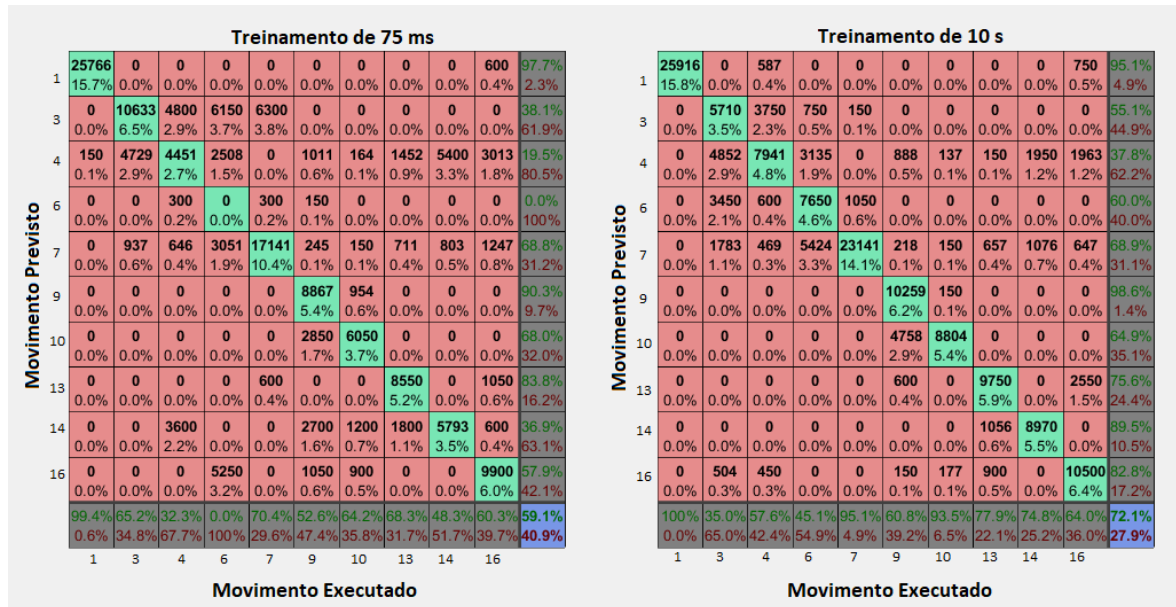
Fonte: do autor

O sujeito 24 apresentou taxa de acerto de 59,1% após 75 ms de treinamento e 72,1% após 10 s de treinamento. A Figura 60 apresenta as matrizes de confusão obtidas para este sujeito que, assim como as matrizes obtidas para os sujeitos 1 e 33, indicam uma grande quantidade de falsos positivos e falsos negativos com valor zero, e alto número de verdadeiros positivos.

A análise no tempo dos movimentos executados e previstos para o sujeito 24 ilustrada pela Fonte: do autor

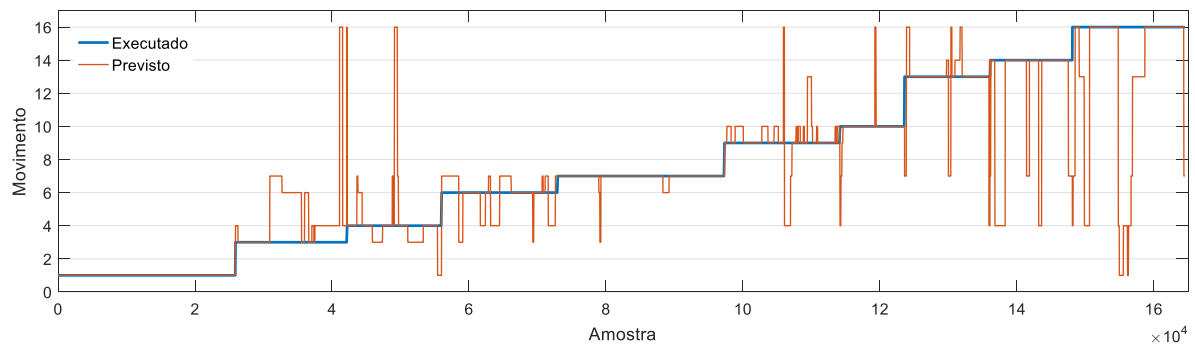
Figura 61 apresenta uma concentração dos erros em alguns movimentos. O movimento que se destacou pelos erros foi o movimento 3 com 35% de acerto. Por outro lado, o movimento 1 foi previsto com 100% de acerto e gerou poucos falsos positivos (ver Figura 60).

Figura 60 - Matrizes de confusão da classificação do sujeito 24 da base de dados de não amputados com treinamento de 75 ms e com treinamento de 10 s.



Fonte: do autor

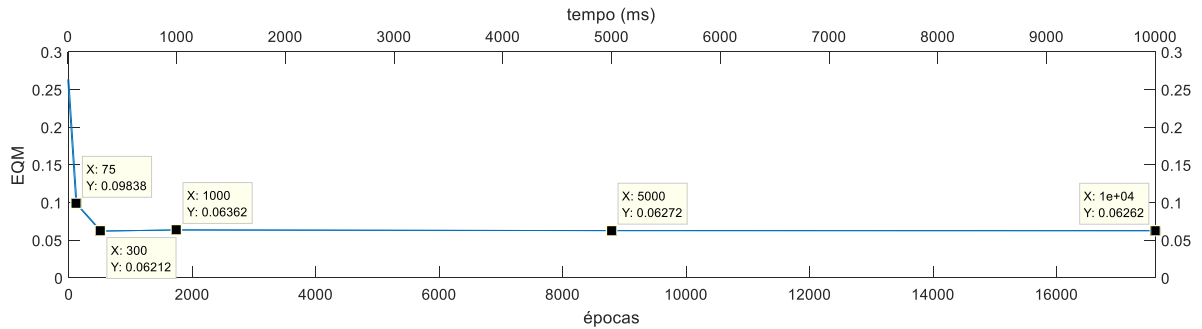
Figura 61 - Classificação dos movimentos na ordem da apresentação das amostras para o sujeito 24 da base de dados de não-amputados.



Fonte: do autor

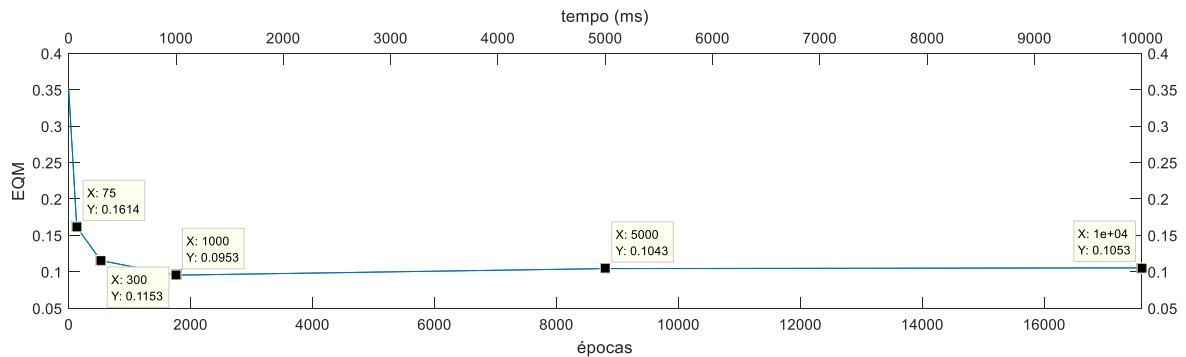
Em várias partes do texto é comentado que após 10 segundos de treinamento, o sistema está estável. As Figuras 62, 63 e 64 apresentam a evolução do EQM no treinamento do sistema para os sujeitos 33, 1 e 24 a fim de confirmar essa afirmação. Os pontos em que o EQM foi avaliado estão destacados, sendo X o tempo em milissegundos e Y o valor do EQM.

Figura 62 - EQM observado na classificação de 10 movimentos para o sujeito 33 da base de dados de não-amputados.



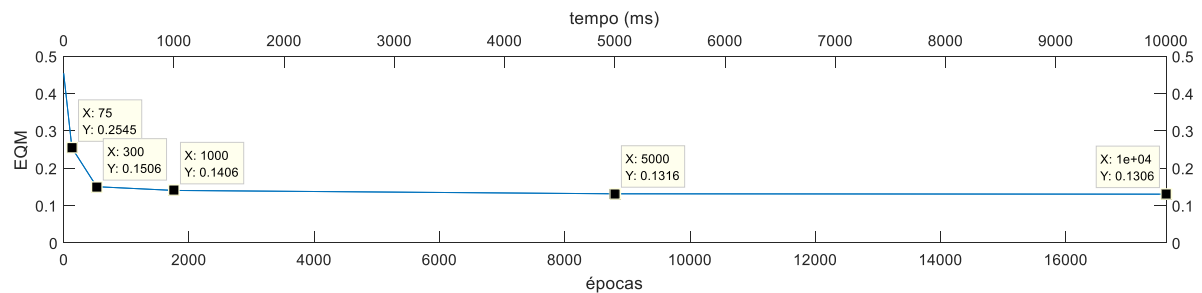
Fonte: do autor

Figura 63 - EQM observado na classificação de 10 movimentos para o sujeito 1 da base de dados de não-amputados.



Fonte: do autor

Figura 64 - EQM observado na classificação de 10 movimentos para o sujeito 24 da base de dados de não-amputados.



Fonte: do autor

4.2.4 Comparação dos resultados com trabalhos relacionados da área

Na revisão de PEERDEMAN *et al.* (2011) sobre o estado da arte dos sistemas de próteses de braço utilizando o sinal mioelétrico, foram citados vários trabalhos com taxas de acerto acima de 90%, e HUANG *et al.* (2016) apresentaram uma taxa de acerto de 83% na difícil tarefa de classificar 13 movimentos de dedos. Estes resultados, porém, foram obtidos com sistemas desenvolvidos para diferentes sujeitos e diferentes movimentos e não permitem uma comparação direta e justa com o sistema desenvolvido neste trabalho.

TOSIN *et al.* (2017), em seu estudo com extração de 21 características do domínio do tempo e 6 características do domínio da frequência, obtiveram uma taxa de acerto média de 85,67% na classificação dos mesmos 17 movimentos utilizando os sinais de sEMG de 10 voluntários não amputados da base de dados Ninapro; e MOURA; FAVIEIRO; BALBINOT (2016) obtiveram taxa de acerto média de 85,1% em um sistema com a extração de apenas uma característica (RMS) de 5 voluntários da mesma base de dados. Apesar de estes trabalhos terem sido desenvolvidos em computadores, onde os recursos de memória e processamento são abundantes, são os que melhor se comparam ao sistema proposto nesta dissertação por utilizarem os mesmos movimentos e mesma base de dados.

Algumas taxas de acerto atingidas pelo sistema desenvolvido neste trabalho se aproximam das taxas encontradas pelos dois sistemas citados mesmo sem a extração de 27 características como é feito no primeiro, e utilizando janelas deslizantes, o que não é feito no segundo. Com taxa de acerto máxima de 78,5% na classificação de 10 movimentos do segmento mão-braço e 68,1% na classificação de 17 movimento executados por 40 sujeitos não amputados, o sistema proposto confirma que é possível a criação de sistemas de classificação de sinais de sEMG em FPGAs, como anteriormente afirmado por SALIM; SALIM; GUAN (2011).

Quando a taxa média de acerto do sistema é observada (54,9% na classificação de 10 movimentos do segmento mão-braço executados por voluntários não amputados), percebe-se que o sistema não se adaptou bem a todos os sujeitos. Além do fato da execução dos movimentos pelos voluntários ter impacto na taxa de acerto, o sistema foi projetado com foco em velocidade de processamento e não na alta taxa de acerto, havendo uma necessidade de otimização caso uma maior assertividade seja desejada.

O foco em velocidade se deve às necessidades do usuário. Com a visão de que o usuário provavelmente necessitará calibrar e atualizar os parâmetros de uma prótese diariamente devido a remoção dos eletrodos para higienização, por exemplo, desejou-se que o sistema aprendesse

rapidamente e que este aprendizado fosse embarcado com o classificador. Para que isso fosse possível com a FPGA disponível, algumas decisões que comprometeram a taxa de acerto tiveram de ser tomadas, como a determinação da resolução das operações matemáticas (maioria fixada em 8 bits).

Poucos trabalhos apresentaram o tempo necessário para treinar seus classificadores. BELLINGEGNI *et al.* (2017) comentaram que os treinamentos duraram mais de quatro horas por sujeito porém não extraía características dos sinais. Tempos mais baixos foram apresentados em dois dos estudos analisados:

- HUANG *et al.* (2016) apresentaram tempos de treinamento acima de 621 ms nos seus ensaios com classificadores LDA de 16 e 64 entradas. Para treinamento de SVMs com função *kernel* de base radial, o tempo necessário foi de 51,7 s;
- O sistema em FPGA desenvolvido por BOSCHMANN *et al.* (2016), com entrada composta por 256 canais de sEMG e com extração de quatro características, precisava de 120 ms para treinar uma LDA e 79 ms para prever um movimento com o classificador. Estes tempos compreendiam o tempo desde a aquisição dos sinais de sEMG até a conclusão da fase de treinamento ou classificação.

O sistema desenvolvido neste trabalho prevê movimentos em microssegundos, sendo o tempo de atraso do sistema determinado pelo passo escolhido no janelamento, 75 ms. Este tempo é equivalente ao tempo necessário pelo sistema desenvolvido por BOSCHMANN *et al.* (2016). O tempo necessário para terminar o treinamento, porém, não foi verificado. Sabe-se que após 10 segundos o sistema está estável. Este tempo é superior ao tempo necessário pelo sistema de BOSCHMANN *et al.* (2016) e ao sistema com LDA de HUANG *et al.* (2016).

Mas, na arquitetura desenvolvida, o treinamento acontece em um processo paralelo ao de classificação que está constantemente se atualizando e que a qualquer momento pode receber novas entradas e apresentar novos parâmetros ao classificador. Dessa forma, o sistema de classificação de movimentos apresentado está preparado para treinar e classificar no tempo de um passo do janelamento. A velocidade de treinamento é aproximadamente 1,75 épocas por milissegundo e, após o período de 75 ms, 132 épocas foram treinadas.

5 CONCLUSÕES

O trabalho proposto tinha como objetivo desenvolver um sistema em *hardware* para classificar movimentos do segmento mão-braço utilizando sinais de sEMG. Este classificador devia ser compatível com aplicações em tempo real onde resultados devem ser apresentados em menos de 300 ms. Para isso, uma FPGA foi escolhida e todas as etapas de um sistema de classificação de sinais mioelétricos foram embarcadas nela: filtragem, extração de características e uma SVM completa (treinamento, validação e classificação).

A velocidade de processamento atingida pelo sistema de classificação desenvolvido ficou acima da esperada e, portanto, optou-se por fazer ensaios com treinamento de apenas 75 ms de duração, tempo equivalente a 132 épocas. Outros ensaios foram realizados com o sistema estabilizado, após 10 s de treinamento.

Na entrada do sistema, optou-se por utilizar uma base de dados de sinais de sEMG também adotada por outros trabalhos, a base de dados Ninapro (ATZORI *et al.*, 2014). Os sinais de eletromiografia de superfície resultantes da execução de 17 movimentos por 11 sujeitos amputados e 40 sujeitos sem amputação foram apresentados à entrada do sistema nos dois primeiros ensaios. A partir destes ensaios, 10 movimentos foram escolhidos para compor um conjunto reduzido de movimentos que o sistema devia aprender em mais dois ensaios.

O sistema foi desenvolvido com base em algoritmos já testados por outros autores em diferentes contextos. Mesmo assim, durante o desenvolvimento decisões tiveram de ser tomadas devido a limitações do *hardware* e objetivos principais. Os resultados dos testes do sistema mostraram que ele é funcional em todas as suas etapas e obteve taxa de acerto de 78,5% para o sujeito 33 na classificação de 10 movimentos mesmo não sendo este o foco do trabalho. Como já mencionado, o objetivo principal era a velocidade, portanto os recursos da FPGA foram utilizados de forma a se ter o sistema mais rápido possível e não o mais assertivo.

Portanto a principal recomendação para trabalhos futuros é, com base no sistema já desenvolvido, focar em melhorar a taxa de acerto. Várias melhorias podem ser feitas neste sentido:

- aumentar resolução dos blocos de extração de características, treinamento, validação e classificação. Algoritmos utilizados pela literatura são implementados com ponto flutuante enquanto que o classificador desenvolvido por este trabalho possui resolução de 8 bits na maior parte de suas operações;
- aumentar o número de características extraídas do sinal de sEMG. Sabe-se que características do domínio da frequência possuem informações importantes da

intenção de movimento e é interessante utilizar este tipo de característica no classificador;

- alterar a arquitetura OVO para OVA ou outra arquitetura. Verificou-se uma ótima taxa de acerto média nas SVMs binárias, porém, ao combinar os 136 classificadores para identificar um movimento, a taxa despencou. Acredita-se que uma arquitetura com menos comparações, OVA por exemplo, pode apresentar taxas de acerto menores nas SVMs binárias, mas taxas de acerto maiores na identificação do movimento;
- alterar a função *kernel* do sistema. Apesar de estudos afirmarem que bons resultados são obtidos com a função *kernel* linear, a maioria dos estudos utiliza a função de base radial.

Essas melhorias propostas impactam na utilização de recursos da FPGA. Para serem possíveis, uma FPGA maior deve ser utilizada ou recursos devem ser compartilhados possivelmente impactando em uma pequena redução da velocidade de processamento das SVMs.

Uma outra abordagem recomendada para trabalhos futuros é desenvolver o sistema em um SoC, componente que integra uma FPGA a um processador. O SoC possui barramentos de comunicação entre FPGA e processador de alta velocidade e permite um classificador com a lógica de controle programada no processador e o processamento dedicado gravado na FPGA. Este componente permite que se utilize o melhor da programação sequencial em conjunto com o melhor do *hardware* dedicado.

Também se recomenda para o futuro a aquisição de sinais de sEMG de voluntários para apresentação ao sistema sem o intermédio de um computador. Propõe-se nesse sentido realizar ensaios com sujeitos executando o protocolo da base de dados e aquisição dos sinais em duas abordagens: aquisição com realimentação e sem realimentação. Na abordagem sem realimentação uma tela apresenta os movimentos que devem ser repetidos pelo voluntário. Na abordagem com realimentação, é apresentado ao voluntário o movimento previsto pelo sistema permitindo que ele melhore a sua execução. Nesta abordagem, usuário e sistema aprendem juntos a executarem os movimentos.

REFERÊNCIAS BIBLIOGRÁFICAS

AL OMARI, F. et al. Pattern Recognition of Eight Hand Motions Using Feature Extraction of Forearm EMG Signal. **Proceedings of the National Academy of Sciences India Section A - Physical Sciences**, Berlin, v. 84, n. 3, p. 473–480, 2014.

ANGUITA, D.; BONI, A.; RIDELLA, S. A Digital Architecture for Support Vector Machines: Theory, Algorithm, and FPGA Implementation. **IEEE Transactions on Neural Networks**, New York, v. 14, n. 5, p. 993–1009, 2003.

ARDUINO. **Arduino Due: Reference Design**. 2019. 2 p. Disponível em: <https://content.arduino.cc/assets/DUE_V02g_sch.pdf>

ASGHARI OSKOEI, M.; HU, H. Myoelectric control systems-A survey. **Biomedical Signal Processing and Control**, Amsterdam, v. 2, n. 4, p. 275-294, 2007.

ATZORI, M. et al. Electromyography data for non-invasive naturally-controlled robotic hand prostheses. **Scientific Data**, London, v. 1, p. 1–13, 2014.

ATZORI, M.; MULLER, H. The Ninapro database: A resource for sEMG naturally controlled robotic hand prosthetics. In: ANNUAL INTERNATIONAL CONFERENCE OF THE IEEE ENGINEERING IN MEDICINE AND BIOLOGY SOCIETY, 37, 2015, Milan. **Proceedings...** New York: IEEE, 2015. p. 7151-7154.

AVNET. **Mini-Module Plus Baseboard 2: User Guide**. 2012. 1162 p. Disponível em: <https://www.avnet.com/opasdata/d120001/medias/docus/58/AES-MMP-BB2-G_UG_RevC_v1_0.pdf>

BASMAJIAN, J. V.; DE LUCA, C. J. **Muscles alive: their functions revealed by electromyography**. 5ª ed. Baltimore: Williams & Wilkins, 1985.

BELLINGEGNI, A. D. et al. NLR, MLP, SVM, and LDA: A comparative analysis on EMG data from people with trans-radial amputation. **Journal of NeuroEngineering and Rehabilitation**, Berlin, v. 14, n. 1, p. 1–16, 2017.

BISHOP, C. M. **Pattern Recognition and Machine Learning**. [s.l.: s.n.]. v. 4 Disponível em: <<http://www.library.wisc.edu/selectedtoes/bg0137.pdf>>

BOSCHMANN, A. et al. FPGA-based acceleration of high density myoelectric signal processing. In: INTERNATIONAL CONFERENCE ON RECONFIGURABLE COMPUTING AND FPGAS, 10 , 2015, Mexico City. **Anais...** New York: IEEE, 2016, p. 1-8.

BREDENSTEINER, E. J.; BENNETT, K. P. Multicategory Classification by Support Vector Machines. **Computational Optimization and Applications**, Boston, v. 79, p. 53–79, 1998.

BURCK, J. M.; BIGELOW, J. D.; HARSHBARGER, S. D. Revolutionizing prosthetics: Systems engineering challenges and opportunities. **Johns Hopkins APL Technical Digest (Applied Physics Laboratory)**, Washington, v. 30, n. 3, p. 186–197, 2011.

BUREAU OF LABOR STATISTICS. **Occupational Outlook Handbook - Biomedical**

Engineers. 2015. Disponível em: <<http://www.bls.gov/ooh/architecture-and-engineering/biomedical-engineers.htm>>.

BURGES, C. J. C. A Tutorial on Support Vector Machines for Pattern Recognition. **Data Mining and Knowledge Discovery**, [s. l.], v. 2, n. 2, p. 121–167, 1998. Disponível em: <<http://www.springerlink.com/index/Q87856173126771Q.pdf%5Cnhttp://link.springer.com/10.1023/A:1009715923555>>

CENE, V. H.; SANTOS, R. R. Dos; BALBINOT, A. Using Antonyan Vardan Transform and Extreme Learning Machines for Accurate sEMG Signal Classification. In: ANNUAL INTERNATIONAL CONFERENCE OF THE IEEE ENGINEERING IN MEDICINE AND BIOLOGY SOCIETY, 40, 2018, Honolulu. **Anais...** New York: IEEE, 2018, p. 5224–5227.

CHEN, X. et al. SoC-based architecture for robotic prosthetics control using surface electromyography. In: INTERNATIONAL CONFERENCE ON INTELLIGENT HUMAN-MACHINE SYSTEMS AND CYBERNETICS, 8, 2016, Hangzhou. **Anais...** New York: IEEE, 2016, p. 134–137.

CRAMMER, K.; SINGER, Y. On The Algorithmic Implementation of Multiclass Kernel-based Vector Machines. **Journal of Machine Learning Research (JMLR)**, Cambridge, v. 2, p. 265–292, 2001.

DRINEAS, P.; MAHONEY, M. W. Approximating a Gram Matrix for Improved Kernel-Based Learning. In: INTERNATIONAL CONFERENCE ON COMPUTATIONAL LEARNING THEORY, 23, 2010, Haifa. **Anais...** Berlin: Springer, 2010, v. 6, p. 323–337.

ENGLEHART, K. et al. Classification of the myoelectric signal using time-frequency based representations. **Medical Engineering and Physics**, Amsterdam, v. 21, n. 6–7, p. 431–438, 1999.

ENGLEHART, K.; HUDGINS, B.; CHAN, A. Continuous multifunction myoelectric control using pattern recognition. **Technology and Disability**, v. 15, p. 95–103, 2003. Disponível em: <<http://iospress.metapress.com/index/bt9k494qw999bgdu.pdf>>

FISHER, R. A. The use of multiple measurements in taxonomic problems. **Annals of Eugenics**, [s. l.], v. 7, n. 2, p. 179–188, 1936. Disponível em: <<http://doi.wiley.com/10.1111/j.1469-1809.1936.tb02137.x>>

GROLEAT, T.; ARZEL, M.; VATON, S. Hardware acceleration of SVM-based traffic classification on FPGA. In: INTERNATIONAL WIRELESS COMMUNICATIONS AND MOBILE COMPUTING CONFERENCE, 8, 2012, Limassol, **Anais...** New York: IEEE, 2012, p. 443–449.

HALL, J. E. **Guyton E Hall Tratado De Fisiologia Médica**. 13^a ed. Rio de Janeiro: Elsevier, 2017.

HAYKIN, S. **Neural Networks and Learning Machines**. Upper Saddle River: Pearson education, 2009.

HEARST, Marti A.. et al. Support vector machines. **IEEE Intelligent Systems and their applications**, [s. l.], v. 13, n. 4, p. 18–28, 1998..

- HUANG, H. et al. EMG pattern recognition using decomposition techniques for constructing multiclass classifiers. In: IEEE RAS AND EMBS INTERNATIONAL CONFERENCE ON BIOMEDICAL ROBOTICS AND BIOMECHATRONICS, 6, 2016, Singapore. **Proceedings...** New York: IEEE, 2016, p. 1296–1301.
- HUSH, D. et al. QP Algorithms with Guaranteed Accuracy and Run Time for Support Vector Machines. **Journal of Machine Learning Research**, [s. l.], v. 7, p. 733–769, 2006. Disponível em: <<http://portal.acm.org/citation.cfm?id=1248573>>
- ISON, M. et al. Simultaneous myoelectric control of a robot arm using muscle synergy-inspired inputs from high-density electrode grids. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, 30, 2015, Seattle. **Proceedings...** New York: IEEE, 2015, p. 6469–6474.
- ISON, M.; ARTEMIADIS, P. The role of muscle synergies in myoelectric control: Trends and challenges for simultaneous multifunction control. **Journal of Neural Engineering**, [s. l.], v. 11, n. 5, 2014.
- JO, H. et al. **Biomedical Engineering: Frontier Research and Converging Technologies**. Berlin: Springer, 2015. v. 9
- JOACHIMS, T.; DORTMUND, U.; JOACHIMSCSUNI-DORTMUNDDE, T. Making Large-Scale SVM Learning Practical. **Advances in Kernel Methods - Support Vector Learning**, [s. l.], p. 41–56, 1999. Disponível em: <<http://svmlight.joachims.org/%5Cnhttps://eldorado.uni-dortmund.de/handle/2003/2596>>
- KARUSH, W. Minima of functions of several variables with inequalities as side conditions. In: GIORGI, Giorgio; KJELDSEN, Tinne Hoff. **Traces and Emergence of Nonlinear Programming**. Basel: Birkhäuser, 2014. p. 217–245.
- KONRAD, P. **The ABC of EMG**. [s.l.: s.n.]. Disponível em: <<http://www.noraxon.com/docs/education/abc-of-emg.pdf>>
- KUHN, H. W.; TUCKER, A. Nonlinear Programming. In: SYMPOSIUM ON MATHEMATICAL STATISTICS AND PROBABILITY, 2, 1951, Berkeley. **Anais...** Berkeley: University of California Press, 1951, p. 481–492. Disponível em: <<http://projecteuclid.org/euclid.bsmsp/1200500249>>
- LEE, Y.; LIN, Y.; WAHBA, G. Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data. **Journal of the American Statistical Association**, [s. l.], v. 99, n. 465, p. 67–81, 2004.
- MOURA, K. O. A.; FAVIEIRO, G. W.; BALBINOT, A. Support vectors machine classification of surface electromyography for non-invasive naturally controlled hand prostheses. In: ANNUAL INTERNATIONAL CONFERENCE OF THE IEEE ENGINEERING IN MEDICINE AND BIOLOGY SOCIETY, 38, 2016, Orlando. **Proceedings...** New York: IEEE, 2016, p. 788–791.
- NEBEKER, F. Golden accomplishments in biomedical engineering. **IEEE Engineering in Medicine and Biology Magazine**, [s. l.], v.21, n. 3, p. 17-47, 2002.
- OSUNA, E.; FREUND, R.; GIROSI, F. An improved training algorithm for support vector

machines. In: NEURAL NETWORKS FOR SIGNAL PROCESSING, 7, 1997, Amelia Island. **Proceedings...** New York: IEEE, 1997, p. 276–285. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=622408>

PAPADONIKOLAKIS, M.; BOUGANIS, C.-S. A Heterogeneous FPGA Architecture for Support Vector Machine Training. In: IEEE ANNUAL INTERNATIONAL SYMPOSIUM ON FIELD-PROGRAMMABLE CUSTOM COMPUTING MACHINES (FCCM), 18, 2010, Charlotte. **Proceedings...** New York: IEEE, 2010, p. 6–9.

PEERDEMAN, B. et al. Myoelectric forearm prostheses: State of the art from a user-centered perspective. **The Journal of Rehabilitation Research and Development**, [s. l.], v. 48, n. 6, p. 719, 2011. Disponível em: <<http://www.rehab.research.va.gov/jour/11/486/pdf/peerdeman486.pdf>>

PETERNEL, L. et al. Adaptive control of exoskeleton robots for periodic assistive behaviours based on EMG feedback minimisation. **PLoS ONE**, [s. l.], v. 11, n. 2, p. 1–26, 2016.

PIÑA-RAMÍREZ, O.; VALDÉS-CRISTERNA, R.; YÁÑEZ-SUÁREZ, O. An FPGA implementation of linear kernel support vector machines. In: IEEE INTERNATIONAL CONFERENCE ON RECONFIGURABLE COMPUTING AND FPGA'S, 2006, San Luis Potosi. **Proceedings...** Piscataway: IEEE, 2, 2006, p. 314–319.

PLATT, J. C. Fast training of support vector machines using sequential minimal optimization. In: SCHÖLKOPF, Bernhard et al. **Advances in kernel methods**. Cambridge: MIT Press. p. 185–208.

RIFKIN, R. M. **Everything Old Is New Again: A Fresh Look At Historical Approaches in Machine Learning**. 2002. Cambridge: Massachusetts Institute of Technology. Disponível em: <<http://hdl.handle.net/1721.1/17549>>

RODRIGUEZ-CARRENO, I.; GILA-USEROS, L.; MALANDA-TRIGUEROS, A. Motor Unit Action Potential Duration: Measurement and Significance. In: AJEENA, Ihsan M. **Advances in Clinical Neurophysiology**. London: Intechopen. Disponível em: <<https://www.intechopen.com/books/advances-in-clinical-neurophysiology/motor-unit-action-potential-duration-measurement-and-significance>>

SALIM, S. I. M.; SALIM, A. J.; GUAN, S. Y. Hardware implementation of surface electromyogram signal processing: A survey. In: IEEE CONTROL AND SYSTEM GRADUATE RESEARCH COLLOQUIUM, 2, 2011, Shah Alam. **Anais...** New York: IEEE, 2011, p. 27 - 31.

SHIMA, K.; TSUJI, T. FPGA implementation of a probabilistic neural network using delta-sigma modulation for pattern discrimination of EMG signals. In: IEEE/ICME INTERNATIONAL CONFERENCE ON COMPLEX MEDICAL ENGINEERING, 2, 2007, Beijing. **Proceedings...** New York: IEEE, 2007, p. 402–407. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4381765>

SMITH, L. H.; KUIKEN, T. A.; HARGROVE, L. J. Myoelectric control system and task-specific characteristics affect voluntary use of simultaneous control. **IEEE Transactions on Neural Systems and Rehabilitation Engineering**, [s. l.], v. 24, n. 1, p. 109–116, 2016.

SUBASI, A. et al. Surface EMG based classification of basic hand movements using rotation

forest. In: ADVANCES IN SCIENCE AND ENGINEERING TECHNOLOGY INTERNATIONAL CONFERENCES, 8, 2018, Abu Dhabi. **Proceedings...** New York: IEEE, 2018, p. 1–5.

TORTORA, G. J.; DERRICKSON, B. **Princípios de Anatomia e Fisiologia**. 12^a ed. São Paulo: Grupo Gen - Guanabara Koogan, 2010.

TOSIN, M. C. et al. SEMG feature selection and classification using SVM-RFE. In: ANNUAL INTERNATIONAL CONFERENCE OF THE IEEE ENGINEERING IN MEDICINE AND BIOLOGY SOCIETY, 39, 2017, Jeju. **Proceedings...** New York: IEEE, 2017, p. 390–393.

VAPNIK, V. N. An overview of statistical learning theory. **IEEE transactions on neural networks**, [s. l.], v. 10, n. 5, p. 988–999, 1999. Disponível em: <<http://www.ncbi.nlm.nih.gov/pubmed/18252602>>

WANG, J. Z. et al. EMG signal classification for myoelectric teleoperating a dexterous robot hand. In: ANNUAL INTERNATIONAL CONFERENCE OF THE IEEE ENGINEERING IN MEDICINE AND BIOLOGY SOCIETY, 27, 2005, Shanghai. **Proceedings...** New York: IEEE, 2005, p. 5931-5933.

WESTON, J.; WATKINS, C. Multi-class Support Vector Machines. **Pattern Recognition**, [s. l.], p. 1–10, 1998. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.50.9594>>

WOLBERG, W. H.; MANGASARIAN, O. L. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. **Proceedings of the National Academy of Sciences**, [s. l.], v. 87, n. 23, p. 9193–9196, 1990. Disponível em: <<http://www.pnas.org/cgi/doi/10.1073/pnas.87.23.9193>>

XILINX. Vivado. **Vivado Design Suite User Guide, Design Flows Overview**. Versão 2017.2. San Jose: Xilinx, 2017. Conjunto de programas.