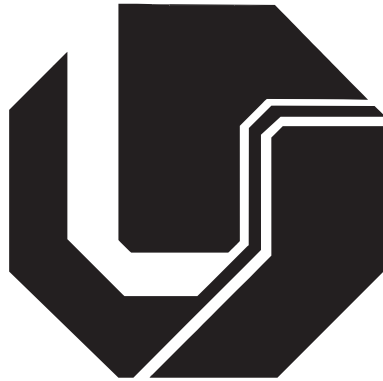


**UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**



**Comparação entre as redes LVQ e MLP no controle de
próteses virtuais para membros superiores**

Daniel Stefany Duarte Caetano

Dezembro

2012

Comparação entre as redes LVQ e MLP no controle de próteses virtuais para membros superiores

Daniel Stefany Duarte Caetano³

Texto apresentado à Universidade Federal de Uberlândia como parte dos requisitos para obtenção do título de Mestre em Ciências.

Prof. Alexandre Cardoso, Dr.

Orientador

Prof. Edgard A.Lamounier Jr, PhD.

Coorientador

Prof. Alexandre Cardoso, Dr.

Coordenador do curso de Pós-Graduação

³Aluno do Curso de Mestrado Engenharia Elétrica UFU

**UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA ELÉTRICA
PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA**

**Comparação entre as redes LVQ e MLP no controle de
próteses virtuais para membros superiores**

Daniel Stefany Duarte Caetano

Texto apresentado à Universidade Federal de Uberlândia, perante a banca de examinadores abaixo, como parte dos requisitos necessários para a obtenção do título de Mestre em Ciências.

Banca Examinadora:

Prof. Alexandre Cardoso, Dr. - Orientador (UFU)

Prof. Edgard A.Lamounier Jr, PhD - Coorientador (UFU)

Profa. Fátima L. Santos Nunes, PhD (USP)

Profa. Ana Claudia Patrocinio, Dra. (UFU)

A Deus, Pai de todos os pais.
Aos que partiram mas que continuam comigo
através do eterno elo de Amor,
Joana minha genitora, Dona Oscarina minha avó,
Madalena minha madrastra, Marlene e Maria Helena
mãezinhas de coração e minha irmã Stella.
Ao meu pai, minhas irmãs Ana Cristina,
Dayane, Priscyla e Virginia pelos momentos
de cumplicidade e de alegrias vividos.

Agradecimentos

Agradeço a Ele que é Pai de todos os pais, Deus, porque sei que um Pai sempre confia em seu filho. Ele tudo me concedeu, me capacitou, para que pudesse vencer a mim mesmo nesta jornada renovadora, que é o viver. Me concedeu familiares e amigos, para que nunca me sentisse sozinho nesta terra e que com exemplo e auxílio de cada um, pudesse aprender um pouquinho mais e que também caminhasse mais firme. Agradeço também a tantos que continuam comigo, através do eterno elo do Amor, minha genitora Joana que tanto amo, minha avó Dona Oscarina, minha madras-tra Maria Madalena, minhas mãezinhas Marlene e Maria Helena Costa e minha irmã Stella. É no seio familiar que vivemos nossas maiores experiências, onde ao mesmo tempo em que aprendemos, nós também ensinamos. Por esta razão agradeço ao meu pai Benedito por sempre ter me dito para estudar, que sem estudo eu não seria ninguém, pelos exemplos e palavras de conselho concedidas. Às minhas irmãs mas, em especial a você Ana Cristina, por ter me revelado que a mamãe sempre desejou que pelo menos um de seus filhos tivesse estudo. Estas palavras me deram muitas vezes forças, para superar obstáculos difíceis e a vocês Dayane, Priscyla, Virginia e Luiza pela companhia de cada dia e por nossa partilha diária de vivências no lar.

Aos meus orientadores Alexandre Cardoso e Edgard Lamounier pela orientação neste caminho e pelas trocas de conhecimentos. Aos amigos Fábio Henrique, Fernando Mattioli, Lucas Garrafa, Kenedy, Keynes, Pedro Cacique, Crisfir, Saulo e Wedson, por tornar a caminhada mais alegre e enriquecida durante os lanches, almoços, confraternizações e também nos momentos de dificuldades.

E por fim, ao CNPq por incentivar através da bolsa oferecida, o fomento de diversas ideias que agora tornar-se-ão realidade.

*"Quando eu desespero,
lembro que ao longo da História o caminho da
verdade e do amor sempre venceu.
Houve tiranos e assassinos e
durante um certo tempo eles pareceram invencíveis,
mas no final, eles caem sempre."
Mahatma Gandhi (1869-1948).*

Resumo

Durante o processo de reabilitação, os indivíduos que tenham sofrido uma perda total ou parcial dos membros superiores estão expostos a muitos riscos. Além disso, um grande esforço mental é exigido durante a fase de treinamento para se adaptar a uma prótese real. Em muitos casos, a utilização de Realidade Virtual na Medicina tem provado ser uma excelente ferramenta para avaliação do risco e apoio, além de reduzir o esforço mental necessário. Para ser útil uma prótese virtual deve ter uma grande semelhança com a real em termos de resposta. Por esta razão, as Redes Neurais Artificiais têm sido exploradas para serem aplicadas na fase de treinamento para proporcionar a resposta em tempo real. O objetivo deste estudo é comparar o desempenho entre a rede neural LVQ e MLP no reconhecimento de padrões EMG, visando melhor controle de próteses virtuais para membros superiores. Para isso, diferentes técnicas de extração de características e métodos de processamento do sinal serão investigadas. Com isto utilizando-se a rede LVQ, a técnica de extração de característica baseada no modelo autoregressivo e uma média de 10% dos padrões de treinamento, alcançou-se 99% de eficiência para os movimentos de mão e 97% de eficiência para os movimentos de braço.

Palavras Chave

Extração de características, Reabilitação, Realidade virtual, Reconhecimento de padrões EMG, Redes neurais

Abstract

During the rehabilitation process, individuals who have experienced a total or partial loss of upper limbs are exposed to many risks. Besides this, a great mental effort is required during the training phase to adapt to a real prosthesis. In many cases, the use of Virtual Reality in Medicine has proven to be an excellent tool for evaluation and support as well as mitigates risk and reduces the mental effort required. In order to be useful, virtual prosthesis must have a great similarity with the real world. For this reason, artificial neural networks have been explored to be applied in the training phase in order to provide real time response. The objective of this study is to compare the performance of the LVQ and MLP neural networks in EMG pattern recognition. To achieve this, different feature extraction techniques for simulation and control of virtual prostheses for upper limbs are investigated. Using the LVQ neural network, autoregressive model as a feature extraction technique and an average of 10% of all training patterns, achieved up 99% of efficiency for the hand movements and 97% of efficiency the arm movements.

Keywords

EMG pattern recognition, Feature extraction, Neural networks, Rehabilitation, Virtual reality

Publicações

A seguir, são apresentadas as publicações resultantes deste trabalho.

Caetano, D.S.D.; Mattioli, F.E.R.; Nogueira, K.L.; Lamounier Jr., E.A.; Cardoso, A. **Comparison between MLP and LVQ neural networks for virtual upper limb prosthesis control.** In: 13th International Conference Intelligent Data Engineering and Automated Learning (IDEAL 2012) , Natal, Brazil, August 29-31, 2012.

Pons, V.C.; Caetano, D.S.D.; Mattioli, F.E.R.; Nogueira, K.L.; Lamounier Jr., E.A.; Cardoso, A. **Comparison of neural networks applied to Augmented Reality for upper limb prosthesis simulation.** In: Anais do VIII Workshop de Realidade Virtual e Aumentada - WRVA'2011, 2011, Uberaba, MG, Brasil.

Mattioli, F.E.R.; Caetano, D.S.D.; Gomes, W.; Lamounier Jr., E.A.; Cardoso, A.; **Utilização de redes neurais para a classificação de sinais EMG aplicados no controle de próteses virtuais de mão.** In: Anais do VII Workshop de Realidade Virtual e Aumentada - WRVA'2010, 2010, São Paulo, SP, Brasil.

Sumário

Lista de Figuras	xi
Lista de Tabelas	xii
Lista de Abreviaturas e Siglas	xiii
1 Introdução	1
1.1 Motivação	1
1.2 Objetivos e Metas	3
1.3 Organização da dissertação	3
2 Fundamentos básicos	5
2.1 Introdução	5
2.2 Fundamentos biomédicos	5
2.2.1 Próteses para membros superiores	6
2.2.1.1 Próteses passivas não-funcionais e ativas-funcionais	6
2.2.1.2 Próteses neuromotoras	6
2.2.1.3 Próteses mioelétricas	7
2.2.2 Controle de próteses mioelétricas	9
2.2.2.1 Controle digital	9

2.2.2.2	Controle proporcional	9
2.2.3	O sinal EMG	9
2.2.4	Processamento do sinal EMG	11
2.2.4.1	Time-domain Features (TDF)	11
2.2.4.2	Modelo Autoregressivo (MAR)	13
2.3	Redes neurais aplicadas no reconhecimento de padrões biomédicos	14
2.3.1	Rede LVQ	15
2.3.2	Rede MLP	16
2.4	Realidade Virtual aplicada à medicina	17
2.4.1	Ambientes virtuais de treinamento	18
2.4.2	Ambientes virtuais para reabilitação	19
2.5	Considerações finais	20
3	Trabalhos Relacionados	21
3.1	Introdução	21
3.1.1	Classificação de sinais EMG no reconhecimento de movimentos utilizando CANFM	21
3.1.2	Controle de prótese virtual baseado em redes neurais para classificação de padrões EMG	23
3.1.3	Classificação de sinais EMG utilizando características no domínio do tempo	25
3.1.4	Classificação de sinais EMG utilizando rede LVQ para controle de próteses virtuais	27
3.2	Considerações finais	28
4	Especificação do sistema proposto	30

4.1	Introdução	30
4.2	Requisitos funcionais	30
4.2.1	Configuração dos parâmetros do classificador LVQ	30
4.2.2	Configuração dos parâmetros do classificador MLP	31
4.2.3	Teste do classificador	31
4.2.4	Controle do ambiente virtual	32
4.3	Requisitos não funcionais	32
4.3.1	Classificação dos movimentos	32
4.3.2	Tempo de resposta	33
4.3.3	Simulação realística	33
4.4	Requisitos de performance em tempo real	33
4.4.1	Aquisição dos dados	34
4.4.2	Janelamento dos sinais	34
4.5	Diagramas de caso de uso	34
4.5.1	Tratamento dos dados	35
4.5.2	Configuração/teste do classificador	37
4.5.3	Processamento em tempo real	38
4.6	Diagramas de classes	39
4.7	Considerações finais	40
5	Detalhes de Implementação	41
5.1	Introdução	41
5.2	Aquisição de sinais	42
5.3	Classificador de sinais	43
5.3.1	Arquitetura da LVQ e MLP	44

5.3.2	Configuração do classificador	44
5.3.3	Algoritmo de treinamento LVQ e MLP	47
5.3.3.1	Algoritmo de treinamento LVQ	47
5.3.3.2	Algoritmo de treinamento MLP	49
5.4	Ambiente virtual	53
5.4.1	Braço virtual	53
5.4.2	Movimentos	53
5.4.3	Comunicação com sistema de classificação	57
5.5	Processamento em tempo real	57
5.5.1	Janelamento	57
5.5.2	Extração de características	58
5.5.3	Classificação	58
5.6	Considerações finais	60
6	Resultados obtidos	61
6.1	Introdução	61
6.2	Metodologia para avaliação de desempenho	61
6.3	Influência dos parâmetros de configuração	62
6.4	Avaliação de performance classificador LVQ	63
6.4.1	Movimentos de mão - MAR	63
6.4.1.1	Taxa de aprendizagem (α)	63
6.4.1.2	Taxa de decaimento de (α)	64
6.4.1.3	Tolerância	65
6.4.1.4	Unidades de saída	66
6.4.2	Movimentos de braço - MAR (Isométrico)	67

6.4.2.1	Taxa de aprendizagem (α)	67
6.4.2.2	Taxa de decaimento de (α)	68
6.4.2.3	Tolerância	69
6.4.2.4	Unidades de saída	69
6.5	Considerações finais	70
7	Conclusões e trabalhos futuros	73
7.1	Introdução	73
7.2	Conclusões	73
7.3	Trabalhos futuros	74
	Referências bibliográficas	75
A	Apêndice de resultados	80
A.1	Avaliação de performance classificador LVQ	80
A.1.1	Movimentos de mão - TDF	80
A.1.1.1	Taxa de aprendizagem (α)	81
A.1.1.2	Taxa de decaimento de (α)	82
A.1.1.3	Tolerância	82
A.1.1.4	Unidades de saída	83
A.1.2	Movimentos de braço - TDF (Isotônico)	84
A.1.2.1	Taxa de aprendizagem (α)	85
A.1.2.2	Taxa de decaimento de (α)	85
A.1.2.3	Tolerância	86
A.1.2.4	Unidades de saída	87
A.2	Avaliação de performance classificador MLP	88

A.2.1	Movimentos de mão - TDF	88
A.2.1.1	Taxa de aprendizagem (α)	88
A.2.1.2	Tolerância	89
A.2.1.3	Número de neurônios na camada oculta	89
A.2.2	Movimentos de mão - MAR	91
A.2.2.1	Taxa de aprendizagem (α)	91
A.2.2.2	Tolerância	91
A.2.2.3	Número de neurônios na camada oculta	92
A.2.3	Movimentos de braço - TDF	92
A.2.3.1	Taxa de aprendizagem (α)	93
A.2.3.2	Tolerância	94
A.2.3.3	Número de neurônios na camada oculta	94
A.2.4	Movimentos de braço - MAR	96
A.2.4.1	Taxa de aprendizagem (α)	96
A.2.4.2	Tolerância	96
A.2.4.3	Número de neurônios na camada oculta	97

Lista de Figuras

2.1	Modelo de prótese passiva não-funcional Ortovan (ORTOVAN, 2012)	6
2.2	Modelo de prótese neuromotora (PLETTENBURG, 2008)	7
2.3	Modelo de prótese antropomórfica (DALLEY et al., 2009)	8
2.4	Amostra de sinal EMG	10
2.5	Exemplos de contrações musculares	10
2.6	Superfície AR - Grupo Isométrico (ANDRADE, 2000)	13
2.7	Arquitetura LVQ para diferentes técnicas de extração de características	16
2.8	Arquitetura MLP para diferentes técnicas de extração de características	17
2.9	Descrição do ambiente virtuais de treinamento (KUTARNIA; PEDERSEN; YUAN, 2010)	18
2.10	Cenário para várias funções dos membros inferiores (VILLIGER et al., 2011)	19
2.11	Visualização do cenário no AV (FEINTUCH et al., 2009)	20
3.1	Diagrama de blocos do sistema de classificação de sinais EMG (HUANG et al., 2003)	22
3.2	Imagem real do DSP baseado na classificação EMG e IV-Mão NTU (HUANG et al., 2003)	23
3.3	Arquitetura do sistema (SOARES et al., 2003)	24

3.4	(a) Arquitetura do sistema de controle; (b) Paciente virtual (SOARES et al., 2003)	25
3.5	Arquitetura do sistema. (HERLE et al., 2008)	25
3.6	Prótese virtual. (HERLE et al., 2008)	26
3.7	Protótipo da arquitetura (MATTIOLI et al., 2011)	27
3.8	Sistema de interface gráfica proposta (MATTIOLI et al., 2011)	28
4.1	Caso de uso: Tratamento de dados	35
4.2	Formatação de arquivos de mão (4.2a, 4.2b) e braço (4.2c, 4.2d)	36
4.3	Exemplos de arquivos com vetores de características	36
4.4	Caso de uso: Configuração/teste do classificador	37
4.5	Caso de uso: Processamento em tempo real	38
4.6	Diagrama de classes	39
5.1	Arquitetura em nível superior	41
5.2	Interface entre o AV e configurações do classificador	44
5.3	Opções de configurações da rede LVQ	45
5.4	Opções de configurações da rede MLP	46
5.5	Configurações de controle do classificador	46
5.6	Modelo 3D do braço	54
5.7	Extensão de cotovelo	54
5.8	Flexão de cotovelo	55
5.9	Pronação do antebraço	55
5.10	Supinação do antebraço	55
5.11	Extensão de mão	56
5.12	Flexão de mão	56

5.13 Fechamento de mão	56
5.14 Metodologia de classificação	59
6.1 Gráficos relacionados à taxa de aprendizagem (α)	64
6.2 Gráficos relacionados à taxa de decaimento de (α)	65
6.3 Gráficos relacionados à tolerância	66
6.4 Gráficos relacionados às unidades de saída	67
6.5 Gráficos relacionados à taxa de aprendizagem (α)	68
6.6 Gráficos relacionados à taxa de decaimento de (α)	69
6.7 Gráficos relacionados à tolerância	70
6.8 Gráficos relacionados às unidades de saída	70
A.1 Gráficos relacionados à taxa de aprendizagem	81
A.2 Gráficos relacionados à taxa de decaimento de (α)	82
A.3 Gráficos relacionados à tolerância	83
A.4 Gráficos relacionados às unidades de saída	84
A.5 Gráficos relacionados à taxa de aprendizagem (α)	85
A.6 Gráficos relacionados à taxa de decaimento de (α)	86
A.7 Gráficos relacionados à tolerância	87
A.8 Gráficos relacionados às unidades de saída	87
A.9 Gráficos relacionados à taxa de aprendizagem	89
A.10 Gráficos relacionados à tolerância	90
A.11 Gráficos relacionados ao número de neurônios na camada oculta	90
A.12 Gráficos relacionados à taxa de aprendizagem	91
A.13 Gráficos relacionados à tolerância	92
A.14 Gráficos relacionados ao número de neurônios na camada oculta	93

A.15 Gráficos relacionados à taxa de aprendizagem	94
A.16 Gráficos relacionados à tolerância	95
A.17 Gráficos relacionados ao número de neurônios na camada oculta . . .	95
A.18 Gráficos relacionados à taxa de aprendizagem	96
A.19 Gráficos relacionados à tolerância	97
A.20 Gráficos relacionados ao número de neurônios na camada oculta . . .	98

Lista de Tabelas

3.1	Características analisadas nos trabalhos relacionados	29
5.1	Padrões de saída da rede MLP	52
6.1	Número de padrões	62
6.2	Resumo dos resultados	71
6.3	Características analisadas nos trabalhos relacionados	72

Lista de Abreviaturas e Siglas

AR	- Autoregressivo
AV	- Ambiente virtual
AVC	- Acidente vascular cerebral
BIOLAB	- Laboratório de Bioengenharia e automática
CANFM	- <i>Cascaded architecture of neural networks with feature map</i>
CRPS	- <i>Complex regional pain syndrome</i>
DSP	- <i>Digital signal processor</i>
DOF	- <i>Degrees of freedom</i>
EMG	- Eletromiográfico
FDD	- <i>Force display devices</i>
GUI	- <i>Graphic user interface</i>
HEMG	- <i>Histogram of EMG</i>
K-NN	- <i>k-nearest neighbor</i>
LVQ	- <i>Learning vector quantization</i>
MAR	- Modelo autoregressivo
MAV	- <i>Mean absolute value</i>
MAVS	- <i>Mean absolute value slope</i>
MLP	- <i>Multi-layer perceptron</i>
NTU	- <i>National Taiwan University</i>
RNA	- Rede neural artificial
RV	- Realidade virtual
sEMG	- Sinal eletromiográfico de superfície
SSC	- <i>Slope sign changes</i>

- SOM** - *Self-organizing maps.*
- TDF** - *Time-domain features.*
- TEO** - *Teager energy operator.*
- UFU** - *Universidade Federal de Uberlândia.*
- USP** - *Universidade de São Paulo.*
- VRML** - *Virtual Reality Modeling Language.*
- WL** - *Waveform length.*
- WAMP** - *Willison amplitude.*
- ZC** - *Zero crossing.*

Capítulo 1

Introdução

1.1 Motivação

O corpo humano possui a capacidade de se adaptar a diversas situações mas, de acordo com a severidade do dano ou distúrbio sofrido, não é possível que naturalmente este sistema se adapte, para novamente restabelecer seu equilíbrio. Quando há a privação de algum sentido, como por exemplo, a Visão, naturalmente um outro sentido se acentuará para suprir a falta do outro. Agora, quando se trata de uma lesão advinda da perda de um membro, por exemplo, uma mão ou um braço, esta recuperação não é feita de maneira natural.

Sabe-se que no mundo, há um aumento de 50.000 indivíduos com amputações por ano, cerca de 30% destes sofrem amputações em membros superiores e pesquisas afirmam ainda que 70% das amputações traumáticas são em membros superiores (TOLEDO et al., 2009).

Uma perda desta magnitude traz consigo tanto danos psicológicos como somáticos. Até o atual momento do desenvolvimento tecnológico, somente é possível restabelecer o dano somático com a utilização de membros artificiais, ou seja, uma prótese. Uma prótese auxiliará o indivíduo em questões estéticas e motoras, substituindo o membro amputado e também em sua reabilitação (NOGUEIRA, 2007). Mas, para que o indivíduo possa utilizar uma prótese real que irá realizar algumas funções do membro amputado, ele deverá passar por um período de treinamento exaustivo até realizar

as contrações musculares corretamente por meio da musculatura remanescente e isto faz com que diversos pacientes desistam no meio do treinamento, além de exporem os mesmos a riscos durante este período (HERLE et al., 2008).

Por meio da musculatura remanescente do membro amputado é que será coletado o sinal eletromiográfico (EMG) advindo das contrações musculares. Através do processamento deste sinal é possível discriminar diferentes movimentos dos membros superiores. Esta aplicação torna-se uma importante interface homem-máquina em diversas áreas, tais como: controle de próteses (proporcionais ou on-off), controle de mãos robóticas e controle de dispositivos FDD (*Force Display Devices*) em Realidade Virtual (NISHIKAWA et al., 2001).

Nos últimos anos, tanto a Medicina como outras áreas relacionadas à saúde humana, têm sido beneficiadas pelos avanços tecnológicos apresentados pela Realidade Virtual (RV) (COSTA; RIBEIRO, 2009). Mais especificamente, quando aplicada à reabilitação humana, a imersão proporcionada pela RV favorece o treinamento de habilidades cognitivas e motoras dos pacientes (WEISS et al., 2004).

A utilização de técnicas de RV no treinamento de usuários de próteses mioelétricas apresenta-se como uma ferramenta complementar que favorece a adaptação dos usuários aos membros artificiais (HERLE et al., 2008). Além de possibilitar a avaliação da eficiência de diferentes sistemas de controle, a utilização de RV na simulação de próteses mioelétricas ameniza o desgaste durante o treinamento, fornecendo um canal de *feedback* visual aos pacientes (SEBELIUS et al., 2005).

As redes neurais artificiais (RNA) são sistemas largamente utilizadas no reconhecimento e classificação de padrões, cujo modelo de aprendizagem é baseado no aprendizado humano (FAUSETT, 1994). Uma característica marcante das RNAs é sua capacidade de generalização: após uma fase de treinamento, na qual alguns padrões de entrada são apresentados e processados pela rede, os pesos da rede são determinados. Na fase de execução, padrões diferentes daqueles utilizados em treinamento, podem ser apresentados para que a rede classifique-os adequadamente. A sua eficiência na classificação dos padrões apresentados, dependem de sua configuração e da complexidade dos padrões apresentados.

Machover (MACHOVER; TICE, 1994) afirma que os sistemas de RV precisam for-

necer uma reação de forma coerente aos movimentos do usuário, tornando a experiência consistente. Desta maneira, torna-se necessária a avaliação de diversas técnicas de pré-processamento do sinal e de extração de características. A fim de que se tenha uma melhor eficiência no reconhecimento do movimento realizado e controle de próteses mioelétricas virtuais.

1.2 Objetivos e Metas

O objetivo deste trabalho é avaliar qual a melhor combinação das técnicas de extração de características juntamente com os métodos de reconhecimento de padrões, nos trabalhos realizados por (MATTIOLI et al., 2011, 2010) no controle de próteses de mão e (NOGUEIRA, 2007; SOARES et al., 2003) no controle de próteses de braço, visando maior confiabilidade no *feedback* visual fornecido pelo ambiente virtual de treinamento, assim como, a eficiência da rede neural na classificação dos padrões EMG.

Com este objetivo foram traçadas as seguintes metas:

- Comparar a performance da rede *Learning Vector Quantization* (LVQ) para os movimentos de braço e mão utilizando: O modelo autoregressivo (MAR) e técnicas no domínio do tempo (TDF);
- Comparar o desempenho da rede *Multi-layer Perceptron* (MLP) para os movimentos de braço e mão utilizando: O modelo autoregressivo (MAR) e técnicas no domínio do tempo (TDF);
- Determinar por meio de análises quantitativas, qual é a melhor combinação de técnicas a ser utilizada, quando o assunto for controle de próteses virtuais para membros superiores.

1.3 Organização da dissertação

A presente dissertação é constituída de 7 capítulos, descritos a seguir.

No Capítulo 1 são apresentados a motivação para o desenvolvimento do trabalho, os objetivos e metas propostos e a estruturação do trabalho.

No Capítulos 2 são apresentados os principais fundamentos biomédicos e tecnológicos relacionados ao trabalho desenvolvido.

No Capítulo 3 é apresentado o estado da arte da linha de pesquisa principal deste trabalho.

No Capítulo 4 e 5 são apresentados a especificação do sistema proposto e os principais detalhes de implementação.

No Capítulo 6 são apresentados os resultados e a discussão em torno dos mesmos a partir do sistema desenvolvido.

Finalmente, no Capítulo 7, são apresentadas as conclusões e as perspectivas de trabalhos futuros.

Capítulo 2

Fundamentos básicos

2.1 Introdução

Neste capítulo são discutidos alguns conceitos sobre biomedicina e as tecnologias utilizadas neste trabalho.

2.2 Fundamentos biomédicos

As próteses ortopédicas antropomórficas¹ são dispositivos comumente utilizados na reabilitação de pacientes que tiveram a perda parcial ou total de membros superiores. Estas próteses são classificadas de acordo com a sua capacidade funcional em, próteses funcionais ou ativas e próteses não-funcionais ou passivas (BARROS, 2005). A elaboração de modelos matemáticos baseados em conceitos de anatomia, fisiologia e cinesiologia é necessária para que estes modelos possam permitir a análise de movimentos e esforços realizados para a aplicação de conceitos cinemáticos e dinâmicos nos dispositivos mecatrônicos gerados (NOGUEIRA, 2007).

Atualmente, existem diversos modelos de próteses e diferentes maneiras de controlá-las, conforme os exemplos citados nas seções 2.2.1.1, 2.2.1.2 e 2.2.1.3, de maneira que o paciente possa realizar diversas funções do dia a dia.

¹Que é semelhante ou que possui forma humana

2.2.1 Próteses para membros superiores

2.2.1.1 Próteses passivas não-funcionais e ativas-funcionais

As próteses passivas não-funcionais, em geral, são semi-móveis ou totalmente imóveis (BARROS, 2005). Normalmente são utilizadas por pacientes que dão preferência ao aspecto estético da prótese, conforme exibido na Figura 2.1, renunciando às funções ativas da mesma. Por esta razão, este tipo de prótese pode ser utilizado em todos os níveis de amputação.



Figura 2.1: Modelo de prótese passiva não-funcional Ortovan (ORTOVAN, 2012)

As próteses funcionais ou ativas, são comumente utilizadas por pacientes que necessitam, além das questões estéticas, de recuperar os movimentos realizados pelo membro perdido. Próteses mioelétricas e neuromotoras, permitem através de sinais EMG e sinais advindos das atividades cerebrais, ativar as funcionalidades que a prótese realizará (MATTIOLI, 2012).

2.2.1.2 Próteses neuromotoras

Próteses neuromotoras, conforme exibido na Figura 2.2, são normalmente utilizadas para substituir ou restaurar as funções motoras perdidas, em seres humanos paralisados pela interrupção no tráfego dos sinais provenientes do cérebro, responsáveis por acionar a ação de um músculo. Esta técnica é muito indicada na reabilitação de pacientes, que sofreram danos na medula espinhal ou nos próprios nervos (HOCH-

BERG et al., 2006).

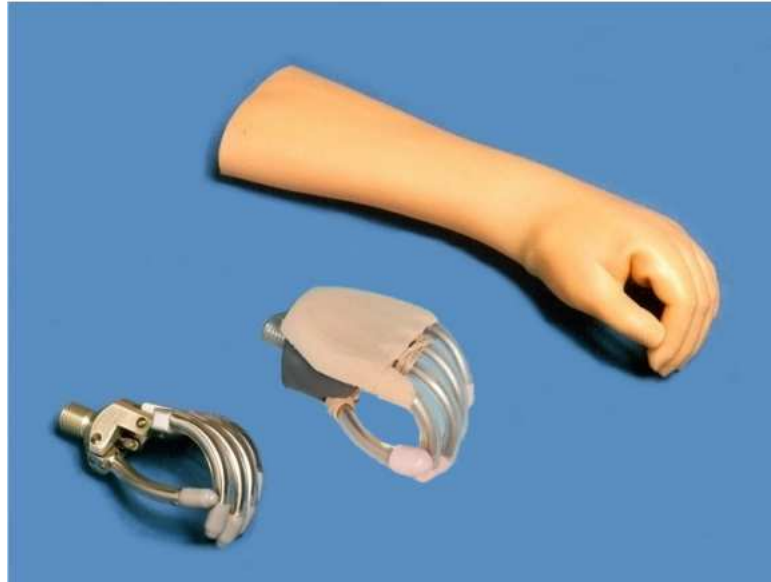


Figura 2.2: Modelo de prótese neuromotora (PLETTENBURG, 2008)

A arquitetura de uma prótese neuromotora se divide em três partes essenciais (HOCHBERG et al., 2006):

- um elemento sensor: capaz de detectar a atividade neuronal;
- um decodificador: que traduz a atividade neuronal em sinais de comando;
- um sistema: que aciona os efetadores de acordo com o movimento desejado.

A coleta do sinal pode ser realizada por meio de métodos invasivos e não-invasivos. As interfaces invasivas apresentam uma série de riscos clínicos ao paciente, além de necessitarem que seja implantado eletrodos no córtex (HOCHBERG et al., 2006). Já os métodos não-invasivos não apresentam estes riscos, porém, é difícil discriminar a atividade a ser realizada, devido aos diversos processamentos que o cérebro realiza simultaneamente.

2.2.1.3 Próteses mioelétricas

A eletromiografia, que teve início com finalidades estritamente clínicas, desenvolveu-se fortemente como uma ferramenta diagnóstica durante a segunda guerra mundial.

Na “engenharia de reabilitação”, a eletromiografia tem sido utilizada no controle das chamadas próteses mioelétricas (BARROS, 2005).

Por meio do sinal EMG descrito na seção 2.2.3, é possível realizar o controle dos movimentos para próteses de membros superiores. As regiões amputadas mais conhecidas são: transcarpal, transradial (abaixo do cotovelo), transumeral (acima do cotovelo) e desarticulação de ombro. O nível de amputação sofrido pelo paciente é um fator altamente restritivo no controle de próteses, limitando a quantidade de canais de coleta de sinais (DALLEY et al., 2009). Na prótese exibida na Figura 2.3, estas restrições causam a limitação nos graus de liberdade (*DOF*), restringindo a realização de atividades do dia a dia (MURGIA, 2005).

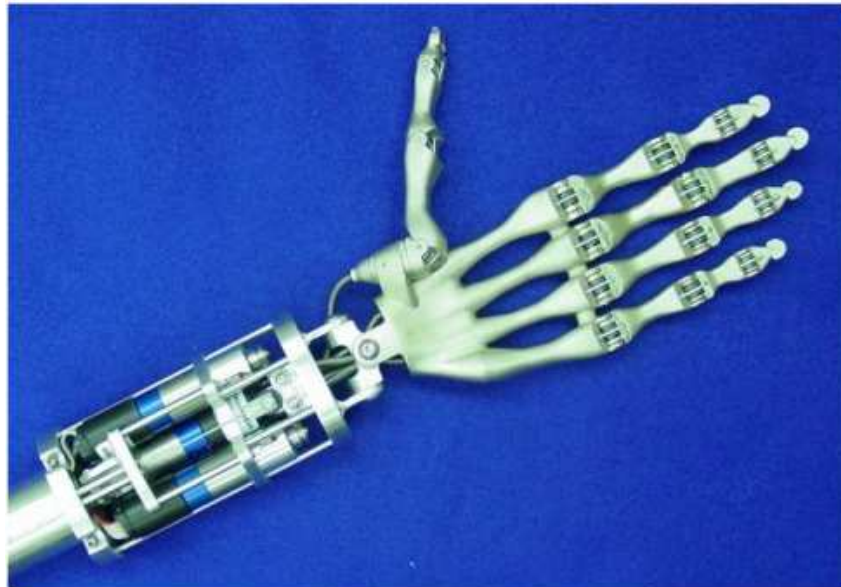


Figura 2.3: Modelo de prótese antropomórfica (DALLEY et al., 2009)

Uma alternativa é a utilização de diferentes técnicas de controle, conforme descrito na seção 2.2.2, que permitam a interpretação de determinadas sequências ou combinações de contrações musculares, permitindo, por exemplo, a rotação do pulso (LI; SCHULTZ; KUIKEN, 2010). Caso seja adotada esta prática de controle, é necessário que o paciente passe por um longo período de treinamento o que pode se tornar muito desgastante e não tão eficiente.

Diversos estudiosos tais como (MATTIOLI et al., 2011; LI; SCHULTZ; KUIKEN, 2010; SOARES et al., 2003; HERLE et al., 2008), estudaram diferentes técnicas de

tratamento e classificação do sinal EMG, a fim de obter um melhor controle de próteses mioelétricas virtuais de mão e braço. Para classificar os padrões de movimentos realizados, técnicas e algoritmos preditivos como redes neurais têm sido estudados mas, a maior parte dos algoritmos são complexos e exigem demasiados esforços computacionais.

2.2.2 Controle de próteses mioelétricas

Nesta seção será descrito um breve resumo sobre 2 tipos de controle de próteses mioelétricas, dentre os existentes.

2.2.2.1 Controle digital

Na técnica de controle digital, independentemente da amplitude do sinal EMG, a velocidade de execução do movimento, seja flexão/extensão de um cotovelo para próteses de braço ou para fechamento/abertura em próteses de mão, será constante (ORTOVAN, 2012).

2.2.2.2 Controle proporcional

A técnica de controle proporcional é normalmente utilizada quando se almeja permitir que a prótese mioelétrica realize os seus movimentos na mesma proporção, que os movimentos reais executados pelo paciente. Neste caso, existe uma relação direta à amplitude da contração do sinal EMG realizado e a força/velocidade do movimento realizado pela prótese (HESSE; HERRMANN, 2010).

2.2.3 O sinal EMG

Apesar da baixa amplitude do sinal EMG exibido na Figura 2.4, sua propagação permite que este seja detectado em determinados pontos na superfície da pele (EMG de superfície ou sEMG), o que viabiliza sua aplicação como sinal de controle de membros artificiais (LI; SCHULTZ; KUIKEN, 2010).

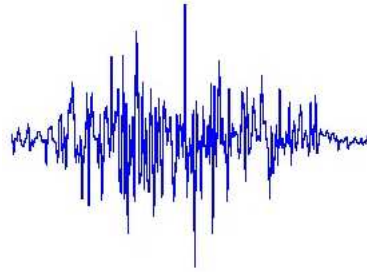
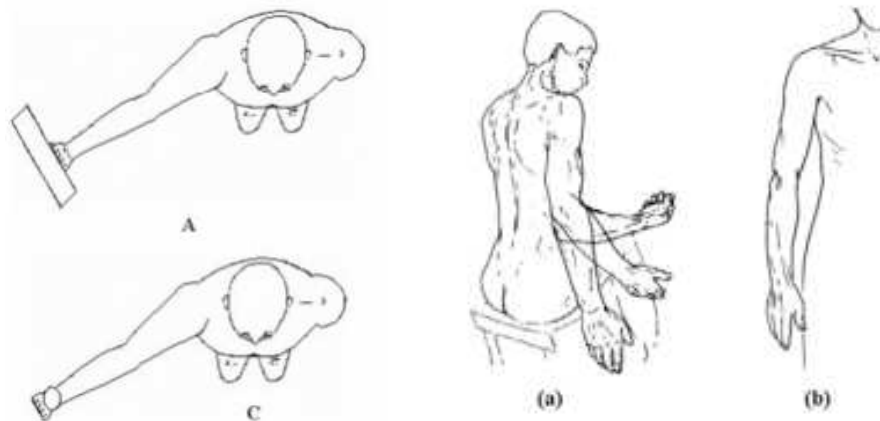


Figura 2.4: Amostra de sinal EMG

Uma contração muscular isométrica ou estática que pode ser chamada também de sustentação, não provoca movimento ou deslocamento articular, sendo que o músculo exerce um trabalho estático. Neste caso não há alteração no comprimento do músculo, mas sim um aumento na tensão máxima do mesmo, conforme demonstra a Figura 2.5a.

Já uma contração muscular isotônica ou dinâmica, é a contração muscular que provoca um movimento articular. Há alteração do comprimento do músculo sem alterar sua tensão máxima, conforme exibido na Figura 2.5b.



(a) Isométricas (OLIVEIRA et al., 2006)

(b) Isotônicas (ANDRADE, 2000)

Figura 2.5: Exemplos de contrações musculares

2.2.4 Processamento do sinal EMG

O sinal EMG é extremamente complexo, sendo afetado pelas propriedades fisiológicas e anatômicas dos músculos, pelo esquema de controle do sistema nervoso periférico, bem como pelas características dos instrumentos utilizados para coletá-los (ANDRADE, 2000).

Com isto é extremamente importante definir quais serão as técnicas utilizadas no processamento do mesmo, a fim de que se possa extrair do sinal as características mais importantes, capazes de representar cada uma das contrações de movimentos realizadas. Nas seções 2.2.4.1 e 2.2.4.2 serão apresentadas as técnicas utilizadas neste trabalho.

2.2.4.1 Time-domain Features (TDF)

Esta abordagem consiste em extrair as características temporais de um conjunto de amostras de sinal EMG, no domínio do tempo, conforme estabelecido por (HUDGINS; PARKER; SCOTT, 1993).

Para cada conjunto de amostras, extrai-se as seguintes características:

- Média dos valores absolutos (*Mean Absolute Value - MAV*);
- Inclinação da média dos valores absolutos (*Mean Absolute Value Slope - MAVS*);
- Cruzamentos em zero (*Zero Crossing - ZC*);
- Alterações no sinal da inclinação (*Slope Sign Changes - SSC*);
- Comprimento de forma de onda (*Waveform Length - WL*);

O *MAV* representa a média dos valores absolutos das amostras do segmento analisado. A Equação 2.1 é utilizada para calcular este valor (HUDGINS; PARKER; SCOTT, 1993).

$$\bar{x}_i = \frac{1}{S} \sum_{m=1}^S |x_m|, \quad (2.1)$$

onde $i = 1 \dots I$ é o número do segmento, S é o número de amostras por segmento e x_m é a m -ésima amostra (amplitude do sinal) no segmento i .

A inclinação da média dos valores absolutos (MAVS) consiste na diferença entre a MAV de dois segmentos adjacentes, e é calculada pela relação (HUDGINS; PARKER; SCOTT, 1993):

$$\Delta \bar{x}_i = \bar{x}_{i+1} - \bar{x}_i, \quad (2.2)$$

onde i e $i + 1$ são dois segmentos adjacentes e $i = 1 \dots I - 1$.

O número de cruzamentos em zero (ZC) é uma medida de frequência que pode ser obtida contando-se o número de vezes em que uma forma de onda corta a reta $y = 0$. Um limiar foi incluído para se filtrar os cruzamentos em zero induzidos por ruídos. Assim como em Herle (HERLE et al., 2008), neste trabalho foi utilizado um limiar $\epsilon = 10^{-6}$. O contador de cruzamentos em zero é incrementado quando a condição

$$\begin{aligned} &\{x_m > 0 \text{ e } x_{m+1} < 0\} \text{ ou} \\ &\{x_m < 0 \text{ e } x_{m+1} > 0\} \text{ e} \\ &|x_m - x_{m+1}| \geq \epsilon \end{aligned} \quad (2.3)$$

é satisfeita para duas amostras consecutivas x_m e x_{m+1} (HERLE et al., 2008).

As alterações no sinal da inclinação (SSC) possibilitam uma outra medida do conteúdo em frequência do sinal. O mesmo limiar utilizado no contador (ZC) foi aplicado ao contador (SSC), que é incrementado quando a condição 2.4 é verdadeira para três amostras consecutivas x_{m-1} , x_m e x_{m+1} (HERLE et al., 2008).

$$\begin{aligned} &x_m > x_{m-1} \text{ e } x_m > x_{m+1} \text{ ou} \\ &x_m < x_{m-1} \text{ e } x_m < x_{m+1} \text{ e} \\ &|x_m - x_{m+1}| \geq \epsilon \text{ ou } |x_m - x_{m-1}| \geq \epsilon \end{aligned} \quad (2.4)$$

O comprimento da forma de onda (WL) é utilizado para se analisar a complexidade da forma de onda em cada segmento. Este parâmetro consiste simplesmente

no comprimento cumulativo da forma de onda dentro do segmento em questão. A equação 2.5 fornece uma medida de amplitude, frequência e duração do segmento em um único parâmetro (HUDGINS; PARKER; SCOTT, 1993), (HERLE et al., 2008):

$$l = \sum_{m=1}^S |\Delta x_m|, \quad (2.5)$$

onde $\Delta x_m = x_m - x_{m-1}$, sendo x_m e x_{m-1} duas amostras adjacentes.

2.2.4.2 Modelo Autoregressivo (MAR)

Um modelo autoregressivo consiste em uma representação de determinado sinal que depende apenas dos valores de saída armazenados anteriormente pelo sistema. Em um modelo autoregressivo, o valor da variável em determinado instante $\hat{y}(n)$, pode ser estimado a partir do valor da variável em instantes anteriores $y(n-1), y(n-2), \dots$. A diferença no número de coeficientes autoregressivos (AR), auxilia na distinção das classes de movimentos, conforme exibido nas Figuras 2.6a e 2.6b.

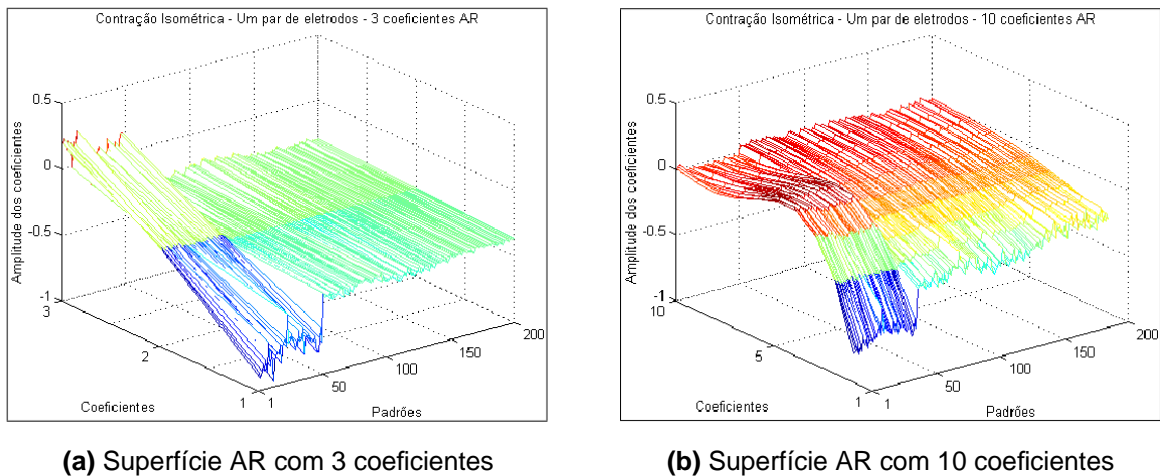


Figura 2.6: Superfície AR - Grupo Isométrico (ANDRADE, 2000)

A equação 2.6 define o modelo autoregressivo (SOARES et al., 2003).

$$\hat{y}(n) = \sum_{m=1}^M a_m(n)y(n-m) + e(n) \quad (2.6)$$

sendo \hat{y} o valor estimado no instante n , a_m o coeficiente autoregressivo (AR) de ordem m , $e(n)$ o erro estimado e M a ordem do modelo (número de coeficientes a_m).

Uma estratégia para o cálculo dos coeficientes autoregressivo e do erro estimado é apresentada a seguir (AKAY, 1994).

1. Inicialização dos coeficientes em 0.
2. Cálculo do valor estimado $\hat{y}(n)$ do sinal de entrada $y(n)$.

$$\hat{y}(n) = \sum_{m=1}^M a_m(n)y(n-m) \quad (2.7)$$

3. Cálculo do erro estimado.

$$e(n) = y(n) - \hat{y}(n) \quad (2.8)$$

4. Atualização dos coeficientes AR.

$$a_m(n+1) = a_m(n) - 2\mu e(n)y(n-m) \quad (2.9)$$

sendo μ uma constante de convergência, normalmente representada por um pequeno valor positivo (da ordem de 10^{-3}) (SOARES et al., 2003), (HEFFTNER; ZUCCHINI; JAROS, 1988).

2.3 Redes neurais aplicadas no reconhecimento de padrões biomédicos

A utilização de redes neurais em áreas como Engenharia, Medicina, e outras, tem colaborado para resolução de problemas, no que tange o reconhecimento de padrões. Devido a sua capacidade de generalização e alta eficiência na classificação, suas aplicações vem crescendo dia após dia.

Vários autores tais como (MATTIOLI et al., 2011; HUANG et al., 2003; LI; SCHULTZ; KUIKEN, 2010) analisaram a aplicação de redes neurais no reconhecimento de padrões de movimentos de mão, enquanto (LAMOUNIER; LOPES; SOARES, 2012; HERLE et al., 2008; SOARES et al., 2003) analisaram padrões de movimentos de braço.

Existem diversos tipos de redes neurais tais como: LVQ, MLP dentre outras. Suas arquiteturas sofrem uma leve variação no algoritmo de treinamento, e reconhecimento de padrões, conforme descrito nas seções 2.3.1 e 2.3.2. Estas variações podem ser o número de neurônios na camada entrada, na camada intermediária/oculta e na camada de saída. A fim de se obter uma melhor eficiência de cada rede, parâmetros como *alpha* (taxa de aprendizado), decaimento da taxa de aprendizado, tolerância e outros são ajustados, permitindo um melhor ajuste dos pesos da rede neural durante a fase de treinamento, otimizando a classificação durante a fase de execução (FAUSETT, 1994).

Durante a fase de treinamento e execução de uma rede neural, informações como tempo de treinamento e eficiência são facilmente mensuradas. Estas informações permitem distinguir qual rede neural é mais indicada. A variação na quantidade de padrões de treinamento² apresentados a rede durante a fase de treinamento, faz com que o tempo de treinamento seja elevado ou reduzido, por aumentar consideravelmente a quantidade de informações a serem processadas para ajustar os valores dos pesos da rede neural. Na fase de execução, pode observar qual é a influência deste aumento ou redução, observando-se a eficiência³ da rede neural. A capacidade de generalização da rede pode ser observada neste momento, levando em consideração a quantidade de padrões utilizados na fase de treinamento e a quantidade de padrões reconhecidos corretamente. Se foi necessário utilizar poucos padrões para se obter uma alta eficiência da rede, então pode-se afirmar que ela possui uma boa característica generalizadora.

2.3.1 Rede LVQ

A arquitetura das redes neurais podem variar conforme dito anteriormente. No caso em estudo, cada técnica de extração de característica utilizada fará com que os números de neurônios na camada de entrada variem, conforme exibido na Figura 2.7a e 2.7b.

Os neurônios na camada entrada da Figura 2.7a terão como informação, cada uma

²Equivalente a cada vetor de característica extraído para cada conjunto de amostras

³Determinada pela quantidade de padrões reconhecidos corretamente durante a fase de execução

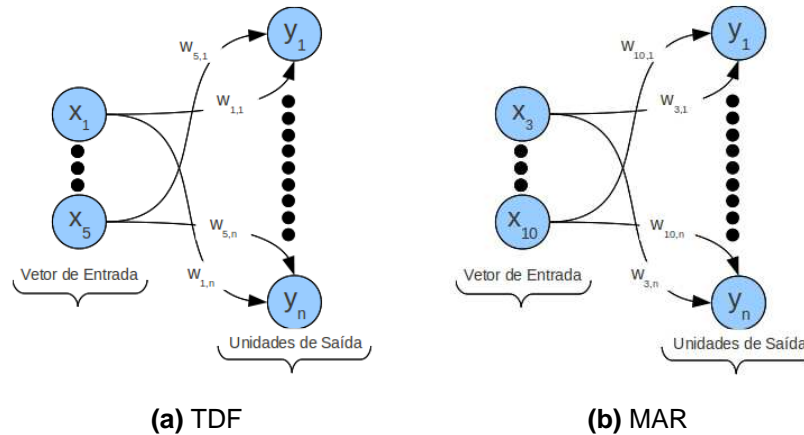


Figura 2.7: Arquitetura LVQ para diferentes técnicas de extração de características

das características da seção 2.2.4.1 que é fixo, pois, conforme os estudos realizados por (MATTIOLI, 2012), não há diferença na performance da rede, se forem utilizadas algumas características isoladamente. Já na Figura 2.7b, o número de neurônios na camada de entrada pode variar em: 3, 4, 6, 8 e 10, de acordo com a ordem do MAR utilizado.

A rede LVQ possui uma técnica de aprendizado supervisionado, na qual as informações referentes a uma dada classe são usadas para se mover os pesos dos vetores de referência, melhorando-se as regiões de decisão do classificador (HAYKIN, 1999). Os vetores de entrada exibidos na Figura 2.7, serão apresentados à rede com suas devidas classificações, um percentual será utilizado para inicializar os pesos da rede na fase de treinamento e os demais na fase de execução. O processo de atualização supervisionada dos pesos, inicia-se com o cálculo da distância euclidiana entre as classes. Após, é verificado qual das classes possui menor distância e aplicados os critérios de atualização dos pesos descritos no *Step 4* do algoritmo de treinamento da (FAUSETT, 1994).

2.3.2 Rede MLP

A rede MLP é uma rede supervisionada, de maneira que a atualização dos seus pesos está associada a uma validação. Sua arquitetura é basicamente dividida nas três partes, conforme exibida Figura 2.8.

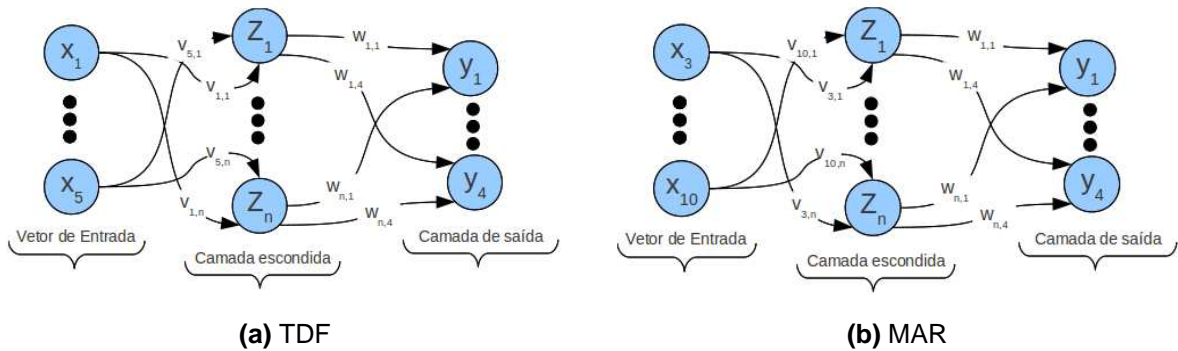


Figura 2.8: Arquitetura MLP para diferentes técnicas de extração de características

O número de neurônios na camada de entrada será variável para cada tipo de técnica de extração de característica utilizada, conforme descrito nas Figuras 2.8a e 2.8b. No entanto, diferentemente da rede LVQ, o número de neurônios na camada de saída é fixo. A rede MLP pode possuir mais de uma camada intermediária e o número de neurônios nesta camada é variável. O método de treinamento das redes MLP consiste em um método de gradiente descendente, baseado na redução do erro quadrático total da saída da rede (FAUSETT, 1994).

2.4 Realidade Virtual aplicada à medicina

O termo Realidade Virtual (RV) surgiu em meados dos anos 70 e pesquisadores sentiram a necessidade de uma definição, para diferenciar as simulações computacionais tradicionais dos mundos digitais que começavam a ser criados (COSTA; RIBEIRO, 2009). Kirner afirma que, a RV configura-se como uma interface avançada de terceira geração para aplicações computacionais, na qual o usuário pode interagir, em tempo real, a partir de um ambiente tridimensional sintético, utilizando dispositivos multi sensoriais (COSTA; RIBEIRO, 2009). A RV também pode ser caracterizada pela coexistência integrada de três ideias básicas: imersão, interação e envolvimento (COSTA; RIBEIRO, 2009). Com o objetivo de tornar válido para usuário as sensações fornecidas por este tripé, é que os sistemas de RV integram em suas arquiteturas, sofisticados dispositivos tais como luvas de dados, capacetes imersivos, headphones aumentando a imersão fornecida pelo ambiente virtual (AV) (COSTA; RIBEIRO, 2009).

As técnicas de RV têm sido largamente utilizadas em sistemas de simulações, ambientes de treinamento, medicina, educação dentre outros. Nas seções 2.4.1 e 2.4.2 serão apresentados alguns trabalhos na área de medicina.

2.4.1 Ambientes virtuais de treinamento

Ambientes virtuais de treinamento são largamente utilizados na área de medicina, pois, em diversos casos se torna inviável realizar tais treinamentos, por razões financeiras ou éticas. No trabalho realizado por (KUTARNIA; PEDERSEN; YUAN, 2010), foi desenvolvido um AV para simulação de ultrassom de diagnóstico de baixo custo, totalmente adaptativo e interativo. De acordo com a posição do transdutor de movimento, é exibido no AV uma imagem 2D sobre a área em análise, conforme exibido na Figura 2.9.

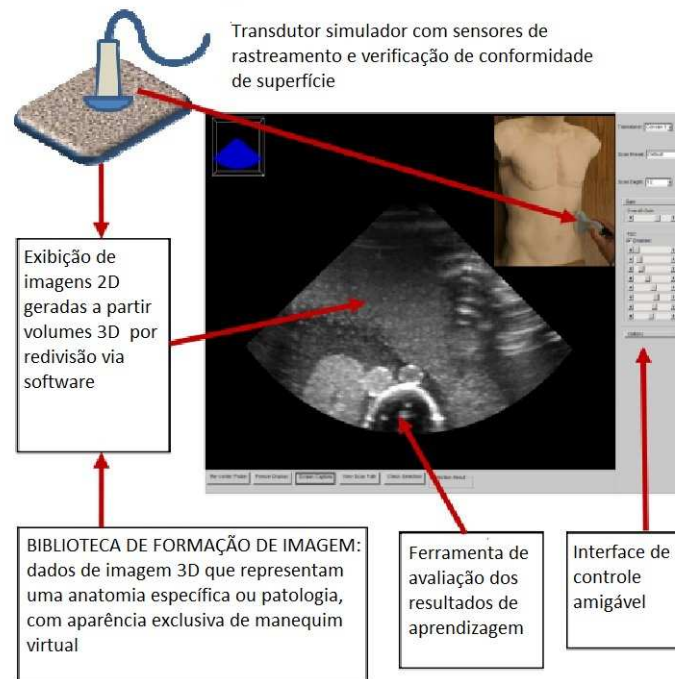


Figura 2.9: Descrição do ambiente virtuais de treinamento (KUTARNIA; PEDERSEN; YUAN, 2010)

Outros trabalhos como realizado por (MATTIOLI et al., 2011; CAETANO et al., 2012) são utilizados em pacientes, que sofreram a perda parcial ou total de membros superiores, de maneira que os mesmos possam utilizar o AV de treinamento, visando

a reabilitação dos movimentos através de uma prótese virtual, não expondo o paciente a riscos e reduzindo o esforço mental necessário.

2.4.2 Ambientes virtuais para reabilitação

O desenvolvimento de AV para reabilitação é uma das grandes aplicações de técnicas de RV. No trabalho realizado por (VILLIGER et al., 2011), estas técnicas foram utilizadas para reabilitar pacientes que sofreram uma lesão medular (LM), que causa disfunção dos membros inferiores e dor neuropática associada. Por meio do treinamento intensivo utilizando cenários em RV divertidos, os autores afirmam que é possível reformular redes corticais reduzindo a dor neuropática e melhorar a função motora. A Figura 2.10 ilustra o primeiro sistema desenvolvido, que combina ações de observação, execução e de endereçamento a função dos membros inferiores, bastando para isto a utilização de sapatos de tamanho ajustável com sensores de movimento integrado. Os pacientes submetidos a terapia relataram a melhora no bem-estar físico.



Figura 2.10: Cenário para várias funções dos membros inferiores (VILLIGER et al., 2011)

Uma outra aplicação é o AV desenvolvido exibido pela Figura 2.11, com objetivo de reabilitar pacientes com acidente vascular cerebral (AVC) e complexo síndrome de dor regional (CRPS) (FEINTUCH et al., 2009).



Figura 2.11: Visualização do cenário no AV (FEINTUCH et al., 2009)

Devido aos estímulos fornecidos pela RV, desenvolveu-se um sistema baseado em uma plataforma de captura de movimento que atualmente está voltado para o tratamento da dor e comprometimento de membros superiores. O sistema emprega algoritmos de processamento de imagem. O paciente se vê em uma tela dentro de um ambiente virtual. O braço lesionado é substituído por um braço virtual, com o qual o paciente pode interagir através da interface do sistema.

2.5 Considerações finais

Este capítulo forneceu elucidações sobre fundamentações biomédicas tais como: tipos de próteses, suas formas de controle, sinal EMG e diferentes maneiras de processá-lo, redes neurais e fundamentações sobre RV assim como algumas aplicações na área de medicina.

No próximo capítulo será apresentado um estudo de alguns trabalhos correlatos ao estudo em questão.

Capítulo 3

Trabalhos Relacionados

3.1 Introdução

Com o intuito de encontrar trabalhos que auxiliem na fundamentação teórica relacionada a sistemas de controle de próteses utilizando RV, foi realizado um estudo do estado da arte procurando avaliar suas principais características (técnicas de processamento do sinal EMG e classificações de padrões EMG), assim como suas limitações visando auxiliar no desenvolvimento de um ambiente de treinamento virtual que ofereça maior confiabilidade.

3.1.1 Classificação de sinais EMG no reconhecimento de movimentos utilizando CANFM

O trabalho apresentado por (HUANG et al., 2003) introduz uma arquitetura que visa reduzir principalmente o tempo de treinamento on-line e obter uma alta taxa de classificação.

A arquitetura CANFM é composta por dois tipos de redes neurais: uma sem supervisão que são os mapas auto organizáveis de Kohonen, e uma camada de rede supervisionada MLP. A rede SOM que utiliza o algoritmo k vizinhos mais próximos, baseia-se na distância (por exemplo, a distância euclidiana) que possui o mérito de simplicidade. No entanto, a rede SOM não pode lidar com os recursos de complexi-

dade, pois não existem técnicas de aprendizagem de máquina envolvidos. Além disso, a rede MLP é mais robusta por ter maior capacidade de separabilidade dos padrões não linearmente separáveis.

Com a arquitetura proposta na Figura 3.1, o autor visa reconhecer oito classes diferentes de movimentos realizados do antebraço a baixo. Foram utilizados 3 canais de coleta, um foi colocado no palmar longus, extensor dos dedos e flexor carpo e para melhor distinguir as características de cada movimento, foram adotadas as técnicas MAR e HEMG (que é a extensão da ZC e WAMP) (HUANG et al., 2003).

O sistema de classificação proposto é detalhado na Figura 3.1, os módulos de extração de características e o CANFM estão encapsulados dentro do DSP chip TMS320C31 produzido pela *Texas Instruments*.



Figura 3.1: Diagrama de blocos do sistema de classificação de sinais EMG (HUANG et al., 2003)

O sinal de cada canal de coleta é aplicado diretamente a um dos 3 conjuntos de rede SOM, cada um com 13 entradas (9 coeficientes HEMG e 4 coeficientes AR) responsáveis por realizar filtragem das informações. Posteriormente, a resposta final de cada conjunto é aplicado a rede MLP, que possui 6 neurônios na camada de entrada, 8 neurônios na camada de saída e 10 neurônios na camada escondida. Suas configurações de treinamento se baseiam no algoritmo de treinamento *backpropagation*, com taxa de aprendizado 0.6 e momentum¹ 0.9.

No primeiro teste realizado, foram utilizados 20 padrões de treinamento e 20 padrões de execução. Nestas configurações alcançou-se 98,75% de eficiência geral para as 8

¹Valor positivo entre 0 e 1 que auxilia na convergência da rede, reduzindo os efeitos dos mínimos locais.

classes de movimento.

Para cada movimento realizado, um *feedback* visual é fornecido ao usuário, conforme exibido na Figura 3.2, através do IV-Mão NTU (*National Taiwan University*) que exhibe ao usuário uma imagem em 2D do movimento reconhecimento pela CANFM.

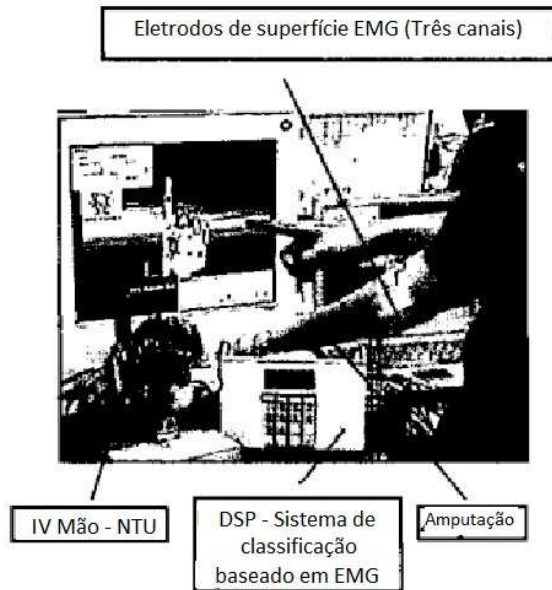


Figura 3.2: Imagem real do DSP baseado na classificação EMG e IV-Mão NTU (HU-ANG et al., 2003)

3.1.2 Controle de prótese virtual baseado em redes neurais para classificação de padrões EMG

O objetivo do trabalho proposto por (SOARES et al., 2003) visa reduzir o grande esforço mental dos pacientes, no estágio inicial de treinamento na utilização de uma prótese de braço. A proposta é que seja utilizada uma prótese virtual, que será controlada por meio dos sinais EMGs advindos dos músculos remanescentes daquele paciente, que tenha sofrido a perda de um membro superior, como no caso em estudo, um braço.

Para realizar a coleta dos sinais EMG, foram utilizadas dois tipos que contrações musculares que são contração isométrica e contração isotônica.

Antes de realizar reconhecimento do movimento executado pelo paciente, foram

adotadas algumas técnicas de processamento do sinal EMG, conforme descrito na Figura 3.3. A janela de sinal é fixa em 200ms, o que equivale a um conjunto de 1000 amostras. Após determinado as fronteiras da janela, é realizada a extração das características utilizando o MAR. O valor atual de uma amostra de sinal EMG sempre dependerá dos seus n valores anteriores, de acordo com a ordem do MAR. Neste caso somente os coeficientes AR serão utilizados para alimentar a rede neural MLP.

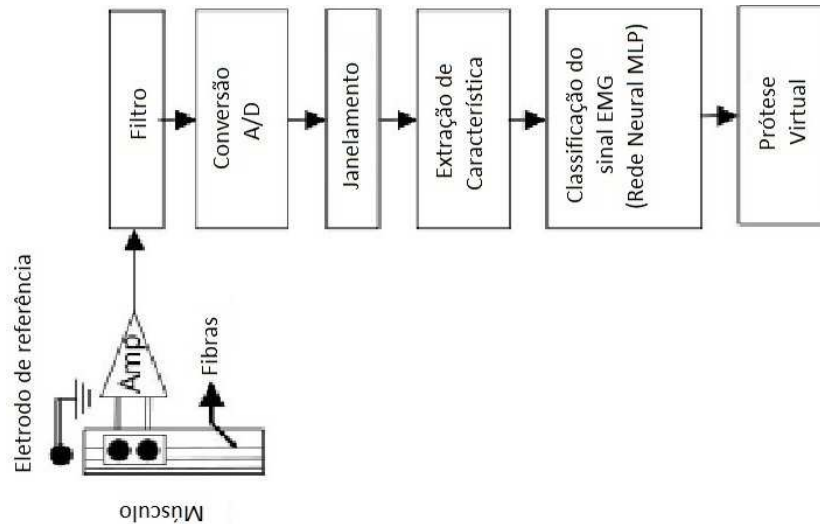


Figura 3.3: Arquitetura do sistema (SOARES et al., 2003)

A rede neural MLP foi implementada com algoritmo de treinamento *backpropagation* e momentum igual a 0. A parametrização da rede MLP é de: taxa de aprendizado igual a 0.01, função sigmoide binária, quatro neurônios na camada de saída, oitenta neurônios na camada oculta e o número de neurônios na camada de entrada pode variar em 3, 4, 6, 8 e 10, de acordo com a ordem do MAR utilizado. Tendo-se classificado os coeficientes apresentados a rede neural MLP, será enviado via *socket* um sinal ao AV criado em VRML, que reproduz exatamente o movimento classificado pela rede (SOARES et al., 2003).

Finalmente, utilizando 25 padrões de movimento na fase de treinamento para cada movimento (flexão/extensão de cotovelo e pronação/supinação pulso) e 25 padrões de movimento de cada na fase de execução, alcançou-se 100% de eficiência para todos os movimento com MAR de ordem 10. A Figura 3.4 exibe a arquitetura que oferece um *feedback* visual ao paciente.

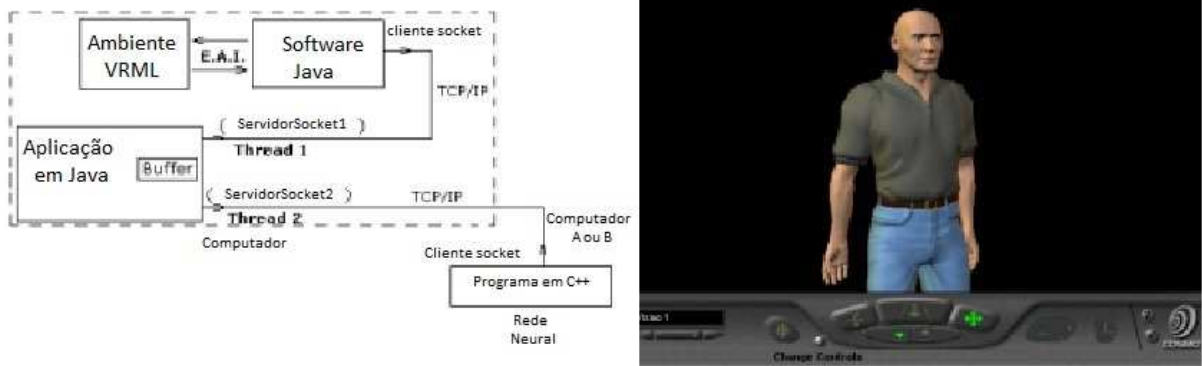


Figura 3.4: (a) Arquitetura do sistema de controle; (b) Paciente virtual (SOARES et al., 2003)

3.1.3 Classificação de sinais EMG utilizando características no domínio do tempo

No mesmo sentido do trabalho anterior, (HERLE et al., 2008) propôs o desenvolvimento de uma arquitetura, conforme exibido na Figura 3.5, que utiliza um número reduzido de características no domínio do tempo (TDF) que são apresentados a rede MLP, com dupla camada de neurônios na camada escondida, a fim de se obter uma melhor resposta em menor tempo no reconhecimento de movimentos de braço.



Figura 3.5: Arquitetura do sistema. (HERLE et al., 2008)

A coleta dos sinais sEMG foi realizada por meio de dois canais, um posicionado na musculatura dos bíceps a 50 milímetros do cotovelo e o outro no tríceps. Com isto, foi possível coletar uma série de repetições dos seguintes movimentos (flexão e extensão de braço, pronação/supinação de antebraço) dos indivíduos em estudo. A taxa de amostragem durante a coleta dos sinais EMG é de 1000 amostras por segundo. Cada janela de movimento possui apenas 200 amostras e cada janela é segmentada em 5

conjuntos com 40 amostras, os quais são extraídas as características determinadas por (HUDGINS; PARKER; SCOTT, 1993) que serão apresentadas a rede neural para classificação.

A rede MLP com retroalimentação de erros foi implementada com 10 neurônios na camada de entrada. São apresentados por vez um total de 10 características no domínio do tempo, sendo as 5 características extraídas dos segmentos de cada canal, 10 neurônios em cada camada oculta e 4 neurônios na camada de saída. A função utilizada foi a sigmoideal binária.



Figura 3.6: Prótese virtual. (HERLE et al., 2008)

O simulador da prótese virtual, não foi implementado para ser utilizado em tempo real. Após realizado a coleta dos movimentos, estes são previamente armazenados em arquivos, de maneira que através das opções fornecidas pela interface com usuário, ele possa selecionar um dos movimentos previamente salvos, para que o sistema classifique e reproduza corretamente o movimento. Através das técnicas estabelecidas por (HERLE et al., 2008), utilizando-se 94% do total de padrões gerados na fase de treinamento da rede e o restante na fase de execução, alcançou-se 96,97% de eficiência na classificação dos padrões com tempo de treinamento menor que 1 minuto.

3.1.4 Classificação de sinais EMG utilizando rede LVQ para controle de próteses virtuais

Como sistemas de treinamento baseado em computação, tem sido largamente estudado no campo da reabilitação (MATTIOLI et al., 2011) apresentam a proposta do desenvolvimento de uma arquitetura, capaz de suportar um ambiente de treinamento virtual utilizando técnicas de RV, para reabilitar pacientes que tenham sofrido a perda de membros superiores, como no caso em estudo, uma mão.

A arquitetura da proposta tem uma similaridade com algumas técnicas utilizadas no trabalho realizado por Herle et al. na seção 3.1.3, no que se refere a técnicas de extração de características utilizadas e segmentação da janela. A Figura 3.7 descreve em um nível mais alto, a arquitetura proposta pelo mesmo.

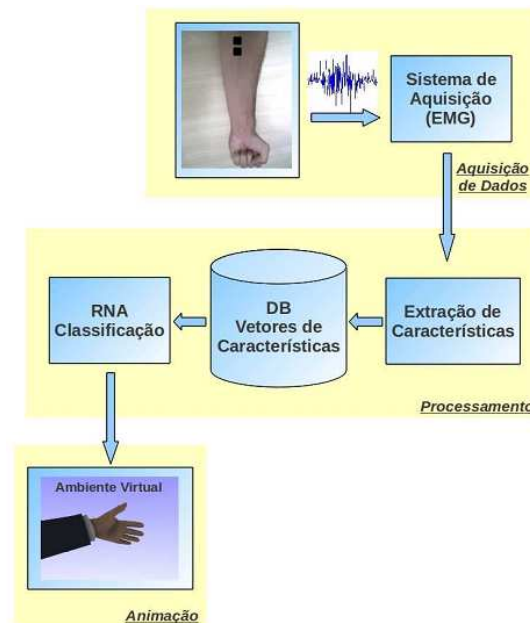


Figura 3.7: Protótipo da arquitetura (MATTIOLI et al., 2011)

No bloco *data acquisition* que é arquiteturalmente um *client socket*, ou seja, um provedor dos sinais EMG advindos das contrações isométricas dos movimentos realizados pelo paciente, serão enviados posteriormente para o *server socket*, que é composto por um conjunto de processos responsáveis por realizar processamento do sinal. Dentro do bloco *processing*, primeiramente é realizado um janelamento automático do sinal utilizando o detector de bordas implementado por (PERETTA, 2010).

As amostras que compõem esta janela são encaminhadas ao bloco *feature extraction*, onde é segmentada em grupos de 40 amostras cada, análogo ao processo realizado por (HERLE et al., 2008). A cada um destes segmentos, é aplicada a técnica de extração de características utilizada no trabalho da seção 3.1.3, que são por sua vez armazenadas em arquivos de dados, para utilização futura, e apresentados a rede LVQ para reconhecimento do movimento realizado. Posteriormente será encaminhado ao AV um sinal para engatilhar a animação correta, fornecendo ao paciente um *feedback* visual em tempo real do movimento realizado ou selecionado.

Através das técnicas utilizadas por (MATTIOLI et al., 2011), foi possível utilizando-se apenas 72% do total de padrões de movimentos gerados na fase de treinamento e o restante, 28% na fase de execução, alcançar uma eficiência de 97% na classificação correta dos movimentos realizados, com menos de 1 segundo de treinamento. Isto garante um bom *feedback* visual ao usuário em tempo real através da interface gráfica proposta na Figura 3.8.

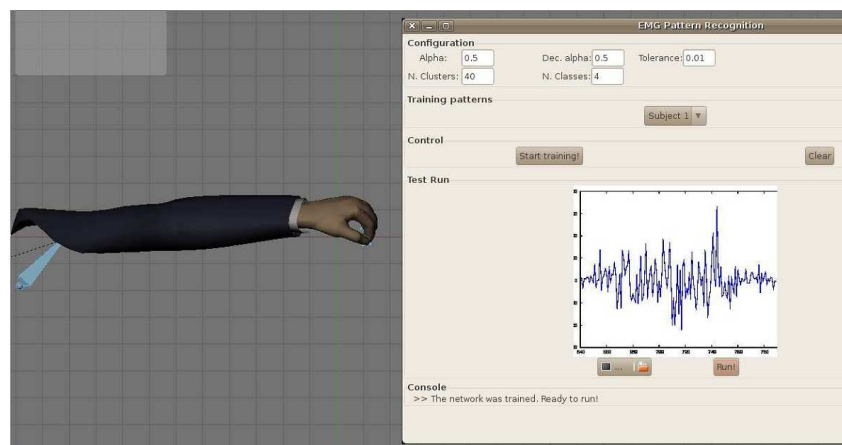


Figura 3.8: Sistema de interface gráfica proposta (MATTIOLI et al., 2011)

3.2 Considerações finais

Nos trabalhos realizados nas seções 3.1.1, 3.1.2, 3.1.3 e 3.1.4 procurou-se evidenciar quais movimentos de prótese abordados, as técnicas utilizadas no processamento do sinal EMG, arquitetura, redes neurais utilizadas e suas configurações, percentuais de padrões de movimentos utilizados, tanto na fase de treinamento quanto execução

da rede neural, a fim de se obter cada uma das eficiências alcançadas, além da quantidade de tempo de treinamento durante os testes realizados.

A Tabela 3.1 demonstra um comparativo entre os trabalhos analisados.

Tabela 3.1: Características analisadas nos trabalhos relacionados

Trabalhos relacionados	Movimentos mão e braço	Janelamento dinâmico	Pré-processamento TDF	Pré-processamento MAR	Classificador LVQ	Classificador MLP	Teste performance tempo real	Realidade virtual
(HUANG et al., 2003)	✗	✗	✗	✓	✗	✓	✓	✗
(SOARES et al., 2003)	✗	✗	✗	✓	✗	✓	✓	✓
(HERLE et al., 2008)	✗	✗	✓	✗	✗	✓	✗	✓
(MATTIOLI et al., 2011)	✗	✓	✓	✗	✓	✗	✓	✓

Através da Tabela 3.1 pode-se perceber que em nenhum dos trabalhos apresentados, foi abordado o desenvolvimento de um AV, que permita a reabilitação do paciente durante o treinamento de movimentos de mão e braço ao mesmo tempo. Na maioria dos trabalhos optou-se, por utilizar a rede MLP com backpropagation no reconhecimento de padrões de sinais EMG, além de ter-se utilizado em dois trabalhos as técnicas de processamento no domínio do tempo (TDF) e em outros dois as técnicas de processamento baseadas no modelo autoregressivo (MAR).

No capítulo subsequente será determinada a especificação do sistema, com o qual será possível avaliar, quais das combinações citadas na Tabela 3.1 oferecerá melhor eficiência e confiabilidade no controle de próteses virtuais para membros superiores.

Capítulo 4

Especificação do sistema proposto

4.1 Introdução

Neste capítulo, são discutidos os principais requisitos do sistema desenvolvido no presente trabalho. Primeiramente são apresentados os requisitos funcionais (funcionalidades) e não funcionais (requisitos de desempenho e resposta) do sistema proposto. Em seguida, são apresentados os principais requisitos de performance em tempo real. Ao final do capítulo, são apresentados os diagramas de caso de uso e classes referentes ao sistema em desenvolvimento.

4.2 Requisitos funcionais

Os principais requisitos de um sistema em RV para o treinamento de usuários de próteses envolvem a correta classificação dos movimentos realizados pela prótese, a configuração dos parâmetros do classificador e o controle do AV de treinamento.

4.2.1 Configuração dos parâmetros do classificador LVQ

Com o objetivo de melhorar a performance do classificador, o sistema deve possuir uma interface que permita a configuração dos diversos parâmetros do mesmo. Os parâmetros considerados neste trabalho são:

- Taxa de aprendizagem (*alpha*);
- Decaimento da taxa de aprendizagem (*dec_alpha*);
- Número de unidades de classificação (*n_output_units*);
- Número de classes (*n_classes*);
- Tolerância (*tolerance*);

A definição de cada um destes parâmetros será apresentada na seção 5.3.2 e sua influência será discutida na seção de resultados.

4.2.2 Configuração dos parâmetros do classificador MLP

Com o objetivo de melhorar a performance do classificador, o sistema deve possuir uma interface que permita a configuração dos diversos parâmetros do mesmo. Os parâmetros considerados neste trabalho são:

- Taxa de aprendizagem (*alpha*);
- Número de neurônios na camada oculta (*n_hidden_neurons*);
- Tolerância (*tolerance*);

A definição de cada um destes parâmetros será apresentado na seção 5.3.2 e sua influência será discutida na seção de resultados.

4.2.3 Teste do classificador

Para que o usuário possa testar o classificador, o sistema deve possibilitar que um sinal de teste seja processado e classificado pelo sistema. É desejável ainda que o sistema realize a classificação de diversos sinais, procedendo posteriormente a uma avaliação estatística dos resultados obtidos nesta classificação.

4.2.4 Controle do ambiente virtual

O AV desempenha um importante papel no contexto da aplicação proposta. É o AV que atuará como canal de comunicação entre o sistema e o paciente, fornecendo o *feedback* visual necessário para o treinamento deste. Assim sendo, deve ser concebida uma arquitetura que possibilite a manipulação de uma prótese virtual a partir dos movimentos detectados pelo classificador.

4.3 Requisitos não funcionais

Os principais requisitos não funcionais do sistema em questão, estão relacionados à performance de classificação dos movimentos abordados, à resposta em tempo real do sistema e à qualidade da simulação realizada.

4.3.1 Classificação dos movimentos

Um dos objetivos principais deste trabalho está na comparação entre a aplicação de diferentes classificadores e técnicas de processamento do sinal EMG, a fim de que se tenha uma melhor eficiência no reconhecimento dos padrões de movimentos realizados pelos pacientes. Machover (MACHOVER; TICE, 1994) afirma que os sistemas de RV precisam fornecer uma reação de forma coerente aos movimentos do usuário, tornando a experiência consistente. Sendo assim, é necessário que o classificador tenha a menor margem de erro possível aumentando a eficiência no *feedback* visual fornecido pelo AV. Nos trabalhos realizados por (HUANG et al., 2003; SOARES et al., 2003; HERLE et al., 2008; MATTIOLI et al., 2011) a margem de erro foi de mais ou menos 4% na classificação correta dos padrões. Por esta razão, neste trabalho espera-se que a margem de erro possa ser menor que os demais trabalhos realizados. Os detalhes a respeito dos percentuais de classificação alcançado neste trabalho serão discutidas no capítulo resultados.

4.3.2 Tempo de resposta

Baseado no trabalho realizado por (CHU; MOON; MUN, 2006), o tempo de resposta de um sistema de controle para próteses mioelétricas de mão não deve ultrapassar 300ms. Tendo em vista as diferentes técnicas e classificadores utilizados neste trabalho, foi verificado que em nenhum dos casos, o tempo de resposta entre a classificação e a resposta da prótese virtual foi maior que o limite, estabelecido por (CHU; MOON; MUN, 2006).

4.3.3 Simulação realística

Conforme as citações realizadas na seção 2.4 é necessário que o AV que faz uso das técnicas de RV, permita que o usuário possa interagir e se sentir imerso no ambiente. Os diversos trabalhos apontados deixam claro que a utilização de técnicas de RV auxiliam e muito na reabilitação do paciente. Por esta razão o sistema proposto neste trabalho, deve favorecer ao usuário uma fácil interação com sistema, de maneira que ele não venha a abandonar o treinamento nas fases preliminares. Nos trabalhos realizados por (MATTIOLI et al., 2011; HERLE et al., 2008), fica claro e evidente, a importância da eficiência que o sistema para que o paciente não venha a optar pela utilização de uma prótese passiva, devido as limitações encontradas durante o treinamento para utilização de uma prótese ativa.

4.4 Requisitos de performance em tempo real

O tempo de resposta do movimento realizado pelo paciente, será determinado pela eficiência entre os processos desde a coleta de dados até a classificação do movimento realizado, pois, após esta etapa será enviado para o AV um sinal que determinará qual animação de movimento será realizado. Isto deve ocorrer de maneira que os requisitos temporais (resposta em tempo real) estejam dentro de um limite aceitável. Isto dependerá da eficiência do sistema de aquisição de dados e o algoritmo de janelamento utilizado, que determinará quais serão os intervalos de dados que possuem informações relevantes para a classificação do sinal, excluindo assim

os ruídos. O sistema deverá ser capaz de classificar e reconhecer com eficiência 7 diferentes movimentos tais como flexão/ extensão de punho, fechamento de mão, pronação/supinação de ante-braço e flexão/extensão de braço.

4.4.1 Aquisição dos dados

Para realizar o controle de próteses EMG é necessário possuir um sistema que monitore constantemente o estado dos músculos que participam dos movimentos desejados. Sendo assim, se faz necessário que a aplicação que irá simular uma prótese, tenha a capacidade de capturar estas informações e salvá-las em disco, a fim de que possam ser processados e discriminados pelo classificador. Este sistema de armazenamento de dados será utilizado como interface entre o sistema de aquisição e o sistema de classificação.

4.4.2 Janelamento dos sinais

O algoritmo para janelamento de sinais é uma das partes mais importantes em um sistema, por ser responsável em discriminar qual parte do sinal é significativa e relevante para a classificação do movimento, e qual parte é ruído. Este processo afeta diretamente a classificação do sinal, pois, se não for corretamente configurado, pode realizar o corte em parte da atividade muscular que seria importante na classificação do sinal ou até mesmo, inserir muitos ruídos dentro de uma janela, aumentando a quantidade de informação a ser processada tornando o tempo de processamento longo, não respondendo aos critérios temporais estabelecidos entre a realização do movimento e a resposta da prótese.

4.5 Diagramas de caso de uso

Nesta seção são apresentados os principais casos de uso do sistema desenvolvido.

4.5.1 Tratamento dos dados

O caso de uso exibido pela Figura 4.1 descreve todas as ações que o usuário deve realizar no tratamento dos arquivos de dados tais como:

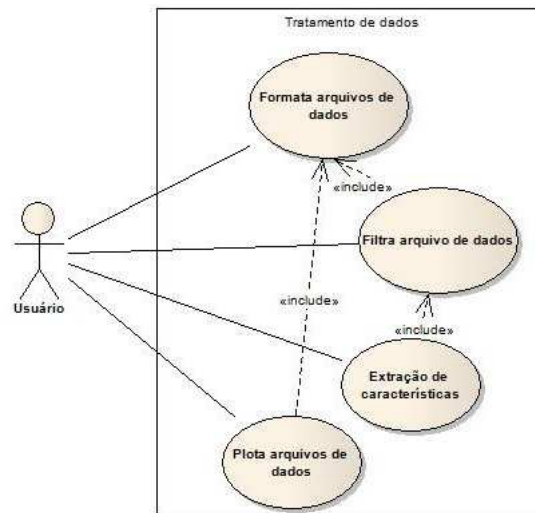


Figura 4.1: Caso de uso: Tratamento de dados

1. *Formatar os arquivos de dados:* Por meio desta ação o usuário pode executar uma automação (*script*) capaz de analisar cada um dos arquivos de dados fornecidos pelo eletromiógrafo e extrair dos mesmos apenas o valor de amplitude do sinal EMG.

As Figuras 4.2a e 4.2c exibem os arquivos na sua forma original. As Figuras 4.2b e 4.2d exibem os arquivos já formatados. Observe que cada linha do arquivo contém uma amostra sinal EMG já normalizada.

2. *Filtrar os arquivos de dados:* O objetivo desta ação é fornecer ao usuário uma ferramenta que seja capaz de filtrar as amostras de sinais contidas em cada arquivo, separando as amostras de sinais EMG em janelas (intervalos de dados) que contenham somente as informações relevantes do arquivo, excluindo os possíveis ruídos. O algoritmo utilizado para implementar o filtro de detecção automática de borda de início e fim de um movimento, ou seja, uma janela e segmentá-la será apresentado na seção 5.5.1.
3. *Extrair as características dos arquivos de dados:* A fim de reduzir a quantidade de informações advindas de cada segmento que contém um conjunto de amos-

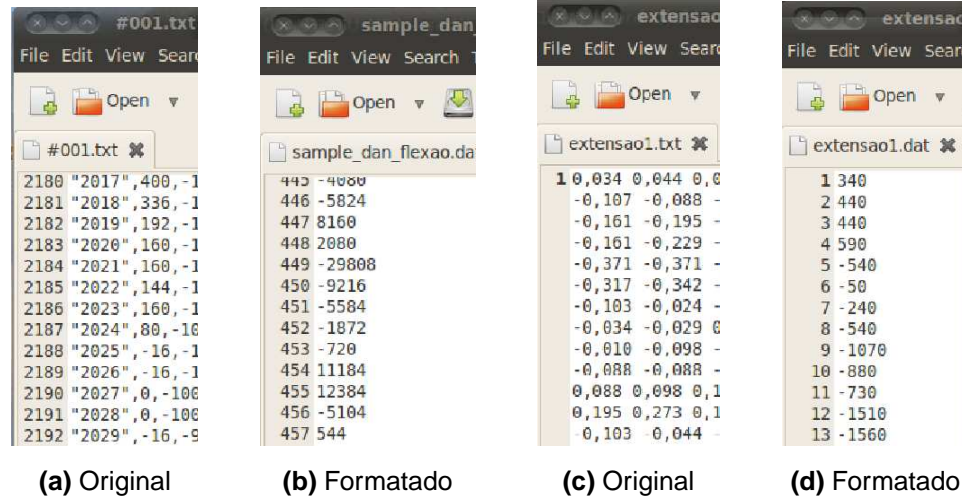


Figura 4.2: Formatação de arquivos de mão (4.2a, 4.2b) e braço (4.2c, 4.2d)

tras de sinais EMG, foi implementado um módulo capaz de realizar a extração das características mais importantes, reduzindo o tempo de processamento e otimizando a performance do classificador. O módulo implementado extrai e salva em arquivos os vetores com as características exibidas pelas Figuras 4.3a e 4.3b. Cada vetor de característica contendo um conjunto determinado de informações extraídas, pode ser denominado como um padrão de treinamento.

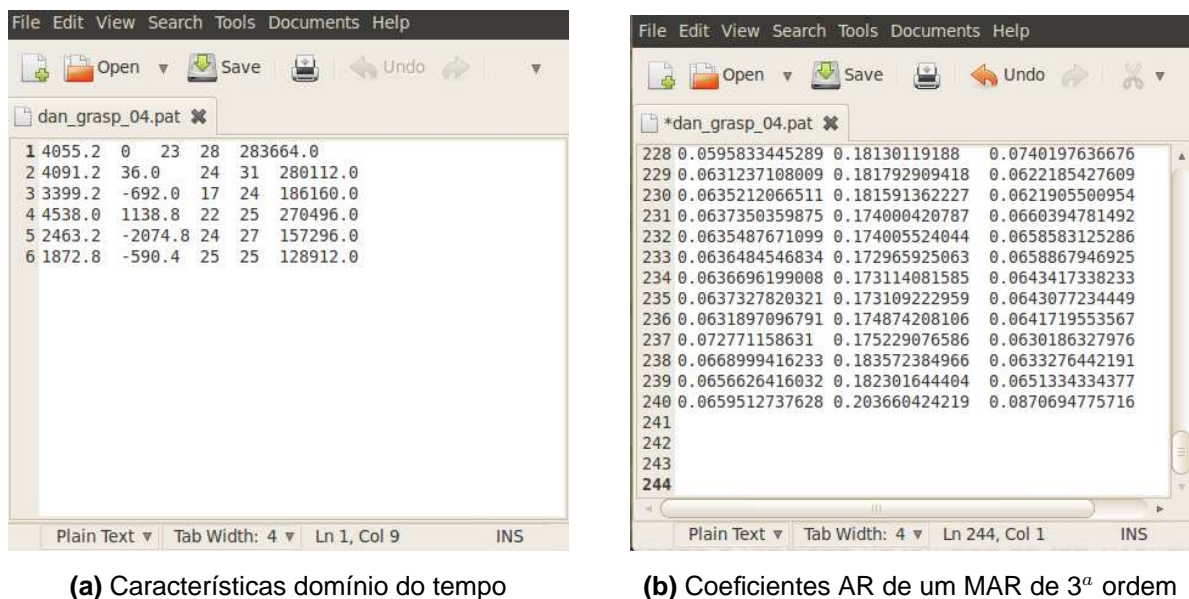


Figura 4.3: Exemplos de arquivos com vetores de características

Sendo assim, por meio da Figura 4.3a pode-se observar que foram extraídos 6 vetores de características que equivalem a 6 padrões de treinamento.

4. *Plotar os arquivos de dados*: Cada um dos sinais armazenados em disco deverá estar associado com seu devido gráfico, que deverá ser plotado e armazenado em disco.

4.5.2 Configuração/teste do classificador

O caso de uso exibido pela Figura 4.4 descreve todas as opções de interações que o usuário poderá realizar através da GUI do sistema implementado.

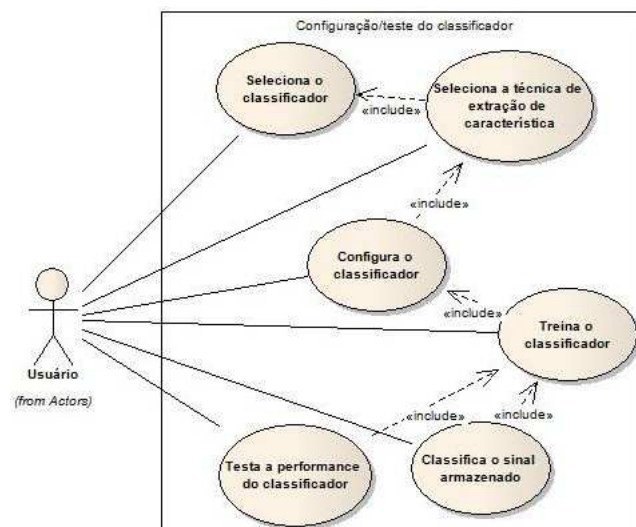


Figura 4.4: Caso de uso: Configuração/teste do classificador

1. *Selecionar o classificador*: O usuário deve escolher uma das redes neurais (LVQ ou MLP) para utilização do sistema.
2. *Selecionar técnica de extração de característica*: O usuário deve escolher umas das técnicas de extração de características (TDF ou MAR) a ser utilizada.
3. *Configurar o classificador*: O usuário deve configurar os parâmetros do classificador apresentados na seção 4.2.1 e 4.2.2, respeitando seus limites.
4. *Treinar o classificador*: O usuário deve treinar a rede neural utilizando uma das bases de padrões de treinamento armazenadas em disco.
5. *Classificar o sinal armazenado*: O usuário pode selecionar um dos padrões de treinamento armazenados em disco, para testar a eficiência da classificação da rede neural.

6. *Testar a performance do classificador*: O usuário pode desejar que seja realizado a avaliação de performance da rede, quanto a classificação de um conjunto de padrões de treinamento armazenados em disco. Para isto, cada padrão estará acompanhado de sua classificação correta, e no fim, o sistema exibirá um relatório geral sobre a eficiência do classificador para todos os padrões selecionados.

4.5.3 Processamento em tempo real

O caso de uso exibido pela Figura 4.5 descreve todas as ações que o usuário deverá realizar para visualizar a resposta do sistema em tempo real.

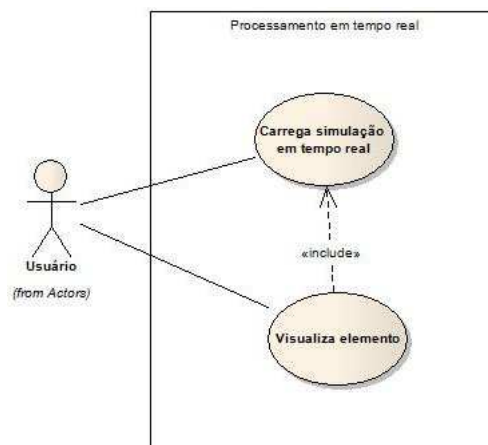


Figura 4.5: Caso de uso: Processamento em tempo real

1. *Carregar simulação em tempo real*: Para que o sistema realize a classificação em tempo real de um movimento, o usuário, deve carregar o sistema com um dos arquivos armazenados em disco. Com isto, as amostras serão enviadas periodicamente ao sistema classificador, com a frequência utilizada durante a coleta das informações por meio do eletromiógrafo.
2. *Visualizar resposta em tempo real*: Uma vez recebido as amostras, o sistema classificador retornará qual é a classificação para aquele conjunto de amostra recebido, e este retorno será enviado ao AV que irá reproduzir o movimento informado, oferecendo ao paciente um *feedback* visual.

4.6 Diagramas de classes

Com base nos requisitos apresentados nas seções anteriores, foram estabelecidos os diagramas de classes exibidos na Figura 4.6 para o trabalho desenvolvido.

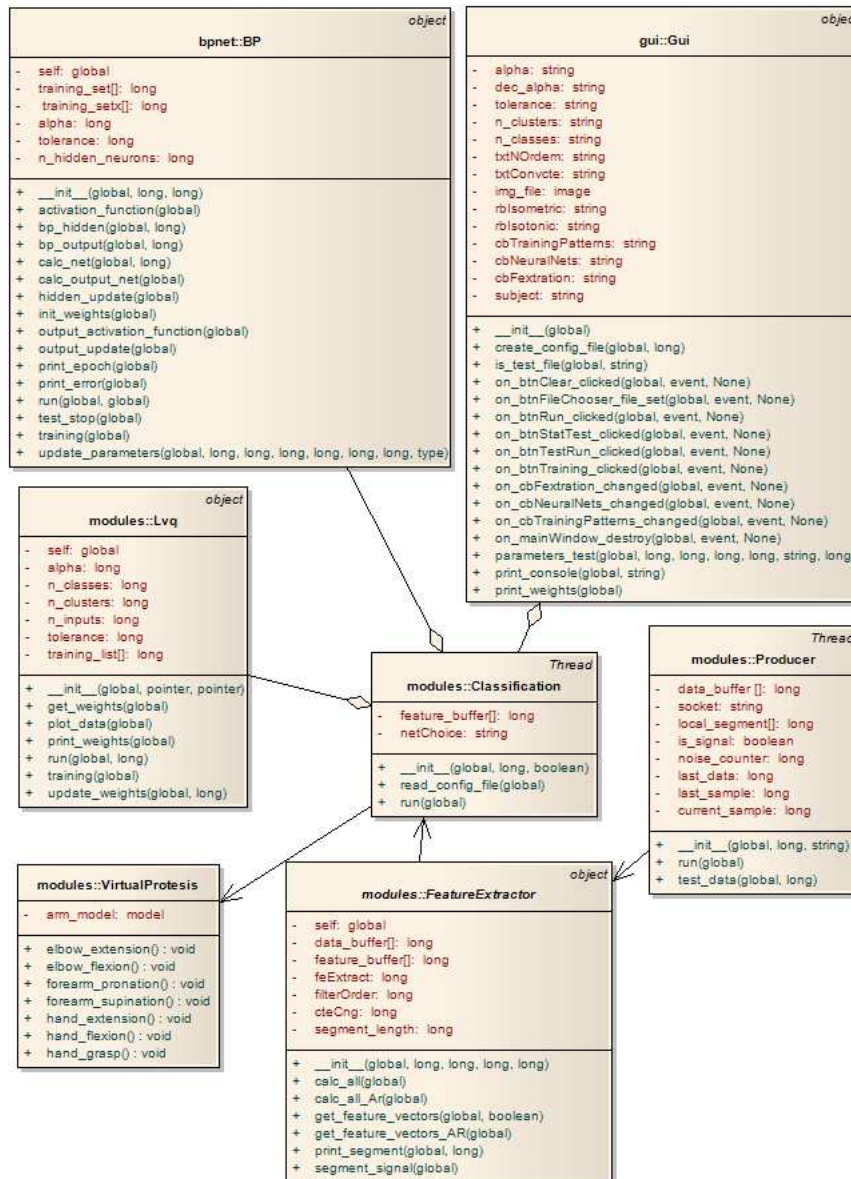


Figura 4.6: Diagrama de classes

Todos os métodos definidos dentro de cada classe, são acionados durante a execução das ações realizadas pelo usuário durante a simulação em tempo real, principalmente os métodos dos módulos **Producer**, **FeatureExtractor**, **Classification** e **VirtualProthesis**. Os métodos dos três primeiros módulos são associados a *Threads*

e toda classificação realizada gerará um sinal de controle que será interpretado pelo módulo **VirtualProthesis**.

Durante este processo é estabelecida uma conexão entre *server socket* e o *client socket*, onde um arquivo de dado é selecionado e suas amostras são enviadas a uma dada frequência para o lado servidor. A *Thread Producer* recebe as amostras que são armazenadas em um *buffer* e posteriormente janeladas através do método *test_data* implementado. Estas janelas de amostras são armazenadas no *data_buffer* a *Thread FeatureExtractor* primeiramente segmenta a janela, por meio do método (*segment_signal*), após, é aplicada uma das técnicas de extração de características por meio dos métodos (*get_features_vectors* ou *get_features_vectorsAR*) e armazenado no *feature_buffer*. A *Thread Classification* executa um dos classificadores (**LVQ** ou **MLP**) que receberá um *feature_buffer* que será classificado conforme as configurações e parâmetros estabelecidos previamente. A cada classificação realizada é enviada ao módulo **VirtualProthesis** um sinal de controle responsável por engatilhar um dos métodos no AV, responsável por executar a animação respectiva.

Os arquivos contendo os padrões de treinamento, ou seja, as *features* extraídas, são armazenadas em arquivos de dados, pois, o módulo **GUI** que ilustra todas as operações e eventos que podem ser realizados através da interface com usuário, utilizará estes arquivos para executar todas as ações com exceções dos requisitos estabelecidos para tempo real.

4.7 Considerações finais

Neste capítulo foi apresentada a especificação do sistema desenvolvido, contendo seus principais requisitos. No próximo capítulo, serão apresentados detalhes da implementação assim como um refinamento da especificação inicial.

Capítulo 5

Detalhes de Implementação

5.1 Introdução

Neste capítulo são apresentados os detalhes sobre a implementação do sistema e sua plataforma de desenvolvimento, assim como, a arquitetura estabelecida exibida na Figura 5.1 baseado nos requisitos estabelecidos no capítulo anterior:

- *Sistema operacional:* Ubuntu Linux, 10.04 (kernel) 2.6.32-38.
- *Hardware:* Processador Intel[®] Core[™] 2 Quad 2.6GHz; 2GB de memória RAM.
- *Linguagem de programação:* Python 2.6.5, GTK 2.20.1-0.
- *Ambiente de modelagem:* Blender 2.49b.

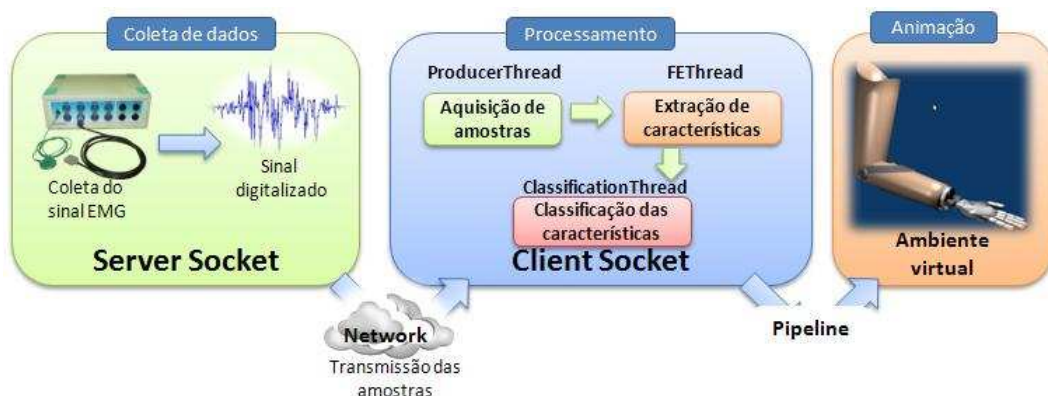


Figura 5.1: Arquitetura em nível superior

O primeiro bloco “*Coleta de dados*”, realiza a aquisição dos sinais EMG advindos das contrações musculares realizadas e estas amostras podem ser enviadas via *socket* para uma simulação de tempo real ou lidas diretamente por meio da interface gráfica.

No segundo bloco “*Processamento*”, a primeira fase a ser realizada é “**Aquisição de amostras**” que aplicará a cada uma das amostras recebidas o operador (TEO) que é um método de janelamento dinâmico utilizado por Peretta (PERETTA, 2010) e Mattioli (MATTIOLI et al., 2011). Na segunda fase “**Extração das características**”, cada janela recebida será segmentada, e para cada segmento recebido será realizada a extração de características. Na terceira fase “**Classificação das características**” cada vetor de característica recebido será classificado por um dos classificadores implementados.

Finalmente no terceiro bloco “*Animação*”, cada classificação realizada pelas redes neurais, será enviada via *pipeline* para o Blender3DTM que executará a animação respectiva.

Nas próximas seções serão comentados os detalhes de implementação para cada um dos blocos realizados.

5.2 Aquisição de sinais

Para avaliar as técnicas de pré-processamento, extração de características e reconhecimento de padrões adotadas neste trabalho, foram utilizadas duas bases de dados distintas:

1. A base de dados com movimentos de mão, utilizada por (MATTIOLI et al., 2010, 2011) foi fornecida pelo laboratório de Engenharia Biomédica da Universidade Federal de Uberlândia ¹ por meio do sistema de aquisição neuropack MEB-2200 (KOHDEN, 2011). A base de dados aborda os seguintes movimentos: flexão, extensão e pronação do pulso e flexão dos dedos com as seguintes características:

- Cinco repetições para cada classe de movimento;

¹<http://www.biolab.ufu.br>

- Foram coletados sinais EMG de três indivíduos diferentes com um total de 60 movimentos;
2. A base de dados com movimentos de braço, utilizada por (NOGUEIRA, 2007; SOARES et al., 2003) aborda os seguintes movimentos: flexão, extensão de cotovelo e pronação/supinação de antebraço com as seguintes características:
- Cinquenta repetições para cada classe de movimento;
 - Os sinais EMG foram coletados de um único indivíduo com um total de 200 movimentos;

Cada base de dado foi coletada utilizando-se apenas um par de eletrodos, no entanto, suas características de coleta são diferentes, conforme descritas abaixo:

1. Movimentos de mão:

- Tipo de contração muscular: Isométrica;
- Músculo flexor radial do carpo;
- Frequência de corte inferior: 20Hz;
- Frequência de corte superior: 10KHz;
- Frequência de amostragem: 2KHz;

2. Movimentos de braço:

- Tipo de contração muscular: Isométrica e Isotônica;
- Músculos: cabeça curta do bíceps (braquial) e cabeça longa do bíceps (braquial);
- Frequência de amostra: 5KHz;
- Frequência de Nyquist: 1KHz;

5.3 Classificador de sinais

Com intuito de avaliar qual classificador possui melhor eficiência e menor tempo de treinamento na classificação dos padrões de sinais EMG, foram implementados

os classificadores LVQ e MLP (FAUSETT, 1994). Nesta seção são apresentados os detalhes sobre a implementação de ambos classificadores. Os resultados advindos da utilização de ambos classificadores serão apresentados no Capítulo 6.

5.3.1 Arquitetura da LVQ e MLP

Conforme explicado na seção 2.3 ambas redes neurais possuem diferentes arquiteturas, o que normalmente implicará em variações, tanto na maneira como a rede neural classificará cada padrão de treinamento apresentado, quanto na maneira como será realizado seu treinamento e no tempo de treinamento realizado.

Nas seções 5.3.3.1 e 5.3.3.2 são detalhadas informações sobre algoritmos de treinamento de cada rede neural.

5.3.2 Configuração do classificador

Por meio da integração da interface com Blender3DTM exibida pela Figura 5.2, o paciente visualizará em tempo real, cada *feedback* visual do sinal processado pelo sistema. A interface se divide basicamente em 5 sessões distintas. Por meio destas seções o usuário definirá quais os parâmetros e quais operações do sistema serão utilizadas são elas:

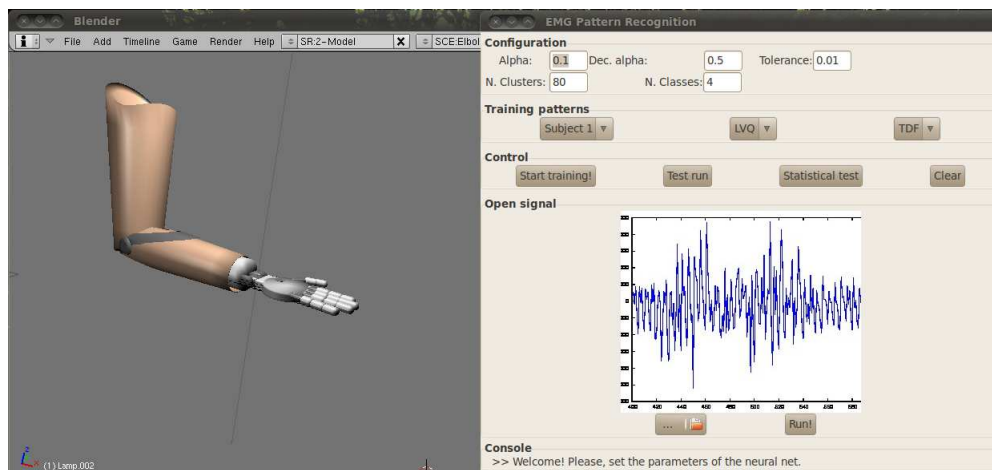


Figura 5.2: Interface entre o AV e configurações do classificador

1. **Seção Configuration:** Configuração dos parâmetros inerentes a cada rede neural tais como:

- Taxa de aprendizado (*Alpha*);
- Decaimento da taxa de aprendizado (*Dec. alpha*);
- Tolerância;
- Número de clusters;
- Número de classes;
- Ordem do AR;
- Constante de convergência;
- Número de neurônios na camada escondida;

2. **Seção Training patterns:** Permite a seleção de indivíduos, rede neural e técnica de extração de características conforme descrito abaixo:

- *Subjects*: são quatro bases de dados distintas, ou seja, os sinais foram coletados de quatro indivíduos diferentes;
- *LVQ*: Seleção de um dos dois classificadores implementados (LVQ ou MLP);
- *AR*: Seleção de uma das duas técnicas de extração de características implementadas (TDF e AR);



(a) Utilizando TDF

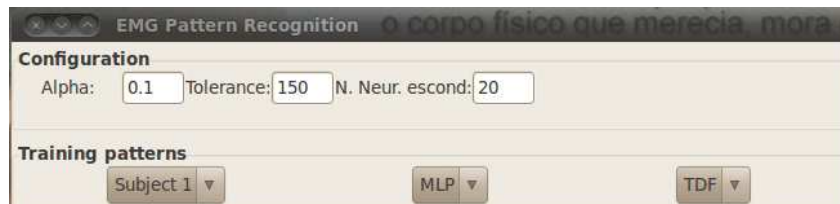


(b) Utilizando MAR

Figura 5.3: Opções de configurações da rede LVQ

Antes de realizar qualquer operação existente na interface gráfica, deve-se primeiramente configurar os parâmetros existentes nas seções anteriores. As Figuras 5.3a e 5.3b exibem quais parâmetros podem ser modificados para a rede LVQ.

As Figuras 5.4a e 5.4b exibem os parâmetros para configuração da rede MLP.



(a) Utilizando TDF



(b) Utilizando MAR

Figura 5.4: Opções de configurações da rede MLP

A Figura 5.5 exibe as operações existentes dentro das seções *Control*, *Open signal* e *Console* detalhadas mais abaixo.

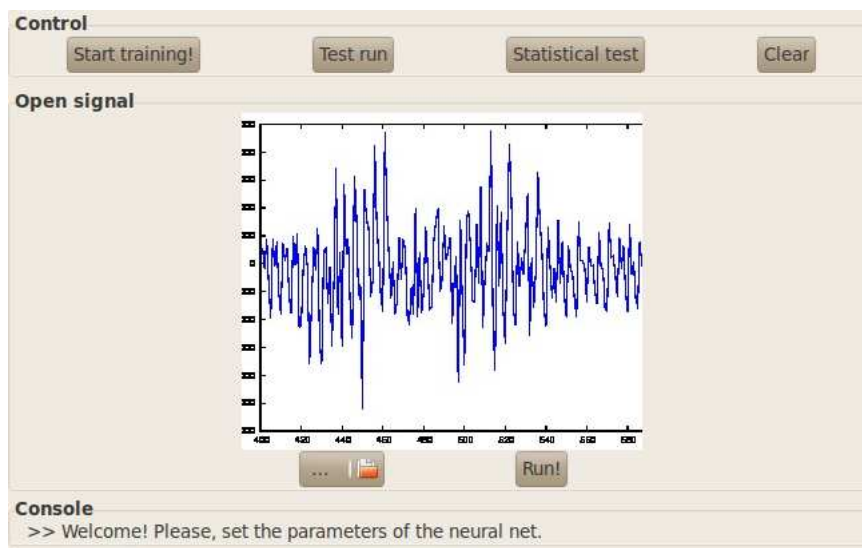


Figura 5.5: Configurações de controle do classificador

3. **Seção Control:** Seleção das possíveis ações que o usuário poderá executar:

- *Start training*: Inicialização do treinamento da rede selecionada de acordo com os parâmetros determinados;
 - *Test run*: Execução de um teste estatístico de acordo com a base de dados e parâmetros selecionados;
 - *Statistical test*: Execução de testes estatísticos para todos os padrões de treinamento da base selecionada, variando-se cada um dos parâmetros dentro da seção *Configuration*.
 - *Clear*: Limpa todas as configurações realizadas na interface;
4. **Seção *Open signal***: Permite selecionar um dos sinais armazenados em disco para testar a rede;
- *Run*: Executa o teste do movimento selecionado na base de dados;
5. **Seção *Console***: Permite ao usuário, visualizar o status das operações /informações realizadas através da interface;

5.3.3 Algoritmo de treinamento LVQ e MLP

Nesta seção são abordados os detalhes sobre a implementação dos algoritmos de treinamento, critérios de inicialização dos pesos, função de ativação (para a rede MLP) e quais os critérios de parada adotadas para ambas as redes.

Uma característica comum a cada algoritmo de treinamento é que será sempre gravado um arquivo de texto, contendo as atualizações realizadas nos pesos durante o treinamento da rede para ser reutilizado, caso se mantenha as mesmas configurações de rede.

5.3.3.1 Algoritmo de treinamento LVQ

Algumas características relacionadas à arquitetura da rede LVQ já foram previamente discutidas na seção 2.3.1. Seguem os detalhes de implementação na ordem mencionada na seção anterior.

1. Passo 0: Inicialização dos pesos

- Inicializa-se os pesos (W_j) de acordo com as diretivas discutidas (FAUSETT, 1994);
- Determine qual o valor da taxa de aprendizado;

2. Passo 1: Enquanto a condição de parada for falsa

- Execute os passos de 2-6;

3. Passo 2: Para padrão do vetor de entrada

- Execute os passos de 3-5;

4. Passo 3: Para cada J, calcule:

$$D_j = \sum_i (W_{ij} - X_i)^2 \quad (5.1)$$

Encontre o índice no qual D_j é mínimo.

5. Passo 4: Critérios de atualização dos pesos

- **Se $T = C_j$ faça:**

Aproxima o peso:

$$W_j(\text{Novo}) = W_j(\text{Anterior}) + \alpha[X_i - W_j(\text{Anterior})] \quad (5.2)$$

- **Se $T \neq C_j$ faça:**

Repele o peso:

$$W_j(\text{Novo}) = W_j(\text{Anterior}) - \alpha[X_i - W_j(\text{Anterior})] \quad (5.3)$$

6. Passo 5: Redução do valor da taxa de aprendizado**7. Passo 6: Testar a condição de parada**

Pode-se observar que matematicamente, ao contrário do algoritmo de treinamento da rede MLP descrita na próxima seção, o algoritmo de treinamento da rede LVQ é simples. Por se tratar apenas do cálculo da distância euclidiana entre os vetores de entrada.

Baseado no trabalho realizado por (HERLE et al., 2008), os padrões de treinamento selecionados para inicializar os pesos da rede foram selecionados aleatoriamente. É importante ressaltar que os padrões de movimento utilizados no treinamento não serão utilizados posteriormente durante a fase de execução.

O critério de parada estabelecido é baseado em um limite inferior para a taxa de aprendizado, uma vez que a taxa de aprendizado é reduzida a cada ciclo de treinamento. Sendo assim, os ciclos de treinamento serão interrompidos quando o valor da taxa de aprendizado for menor que o valor de tolerância estabelecido anteriormente.

Todas as informações sobre os resultados alcançados, variando-se os parâmetros da rede para as diversas classes de movimento, serão discutidas no Capítulo 6.

5.3.3.2 Algoritmo de treinamento MLP

Assim como na rede LVQ, algumas características relacionadas à arquitetura da rede MLP já foram discutidas anteriormente na seção 2.3.2. Seguem todos os detalhes de implementação de acordo com a ordem estabelecida no início desta seção.

1. Passo 0: Inicialização dos pesos.

- Foi utilizado o algoritmo de *Nguyen-Widrow* para inicializar os pesos da rede, conforme descrito na (FAUSETT, 1994).

2. Passo 1: Enquanto a condição de parada for falsa, execute os passos 2-9.

3. Passo 2: Para cada padrão de treinamento, execute os passos 3-8.

- *FeedFoward*:

4. Passo 3: Para vetor de entrada recebido, distribuir cada elemento contido $(X_i, i = 1, \dots, n)$, para cada unidade na camada escondida

5. **Passo 4: Para cada unidade na camada escondida ($Z_j, j = 1, \dots, p$), calcular o somatório das entradas líquidas.**

$$z_in_j = \sum_{i=1}^n X_i \cdot v_{ij} + v_{0j} \quad (5.4)$$

- Para cada valor de z_in_j calculado, aplicar a função de ativação e enviar este sinal para as unidades na camada de saída:

$$z_j = f(z_in_j) \quad (5.5)$$

6. **Passo 5: Para cada unidade da camada de saída ($Y_k, k = 1, \dots, m$), calcular o somatório das entradas líquidas:**

$$y_in_k = \sum_{j=1}^p z_j \cdot w_{jk} + w_{0k} \quad (5.6)$$

- Para cada valor de y_in_k calculado, aplicar a função de ativação e calcular seu sinal de saída.

$$y_k = f(y_in_k) \quad (5.7)$$

Retropropagação do erro:

7. **Passo 6: Para cada unidade da camada de saída ($Y_k, k = 1, \dots, m$) recebe um padrão alvo (t_k) correspondente ao padrão de treinamento, calcula o termo de informação do erro:**

$$\delta_k = (t_k - y_k) \cdot f'(y_in_k) \quad (5.8)$$

- Para cada valor de δ_k determinado, calcula-se Δw_{jk} para a correção dos pesos e envia δ_k para as unidades da camada escondida:

$$\delta w_{jk} = \alpha \delta_k z_j \quad (5.9)$$

8. **Passo 7: Para cada unidade da camada escondida ($Z_j, j = 1, \dots, p$), calcula-se a soma das entradas “delta” ($\delta_i n_j$) recebido da camada de saída:**

$$\delta_i n_j = \sum_{k=1}^m \delta_k \cdot w_{jk}, \quad (5.10)$$

- A seguir, calcula-se o valor da informação do erro (δ_j) e o vetor de correção dos pesos (δv_{ij}):

$$\delta_j = \delta_i n_j \cdot f'(z_i n_j) \quad (5.11)$$

$$\delta v_{ij} = \alpha \cdot \delta_j \cdot X_i \quad (5.12)$$

Atualização dos pesos:

9. **Passo 8: Para cada unidade da camada de saída ($Y_k, k = 1, \dots, m$) atualiza-se ($j = 0, \dots, p$):**

- $w_{jk}(\text{novos}) = w_{jk}(\text{antigo}) + \Delta w_{jk}$;
- Para cada unidade da camada de escondida ($Z_j, j = 1, \dots, p$) atualiza-se ($i = 0, \dots, n$):
- $v_{ij}(\text{novos}) = v_{ij}(\text{antigo}) + \Delta v_{ij}$;

10. **Passo 9: Testar a condição de parada**

Baseado no trabalho realizado por (HERLE et al., 2008), os padrões de treinamento selecionados para inicializar os pesos da rede, foram selecionados aleatoriamente. Visando a redução no tempo de treinamento foi utilizada a inicialização dos pesos de Nguyen-Widrow (ANDRADE, 2000). Após alguns testes iniciais foi verificado que a rede necessitava de muitos ciclos para convergir. Sendo assim, foi implementado o algoritmo *Momentum* (FAUSETT, 1994), que visa reduzir o número de ciclos de treinamento, permitindo uma convergência mais rápida da rede. É importante ressaltar que os padrões de movimento utilizados no treinamento, não serão utilizados posteriormente durante a fase de execução.

Visando evitar a saturação da rede, todos os padrões de treinamento utilizados, foram normalizados no intervalo de $[-1, 1]$. Sendo assim, foi utilizada a função de ativação sigmoide bipolar, descrito pela equação 5.13.

$$f(x) = \frac{1}{1 + e^{-x}} - 1 \quad (5.13)$$

Desta maneira, como tem-se duas classes de padrões de movimentos advindos do braço e mão, os valores de saída para cada movimento são descritos na Tabela 5.1 abaixo.

Tabela 5.1: Padrões de saída da rede MLP

Movimento	y_1	y_2	y_3	y_4
Flexão do punho	1	-1	-1	-1
Extensão do punho	-1	1	-1	-1
Pronação do antebraço	-1	-1	1	-1
Flexão dos dedos	-1	-1	-1	1
Flexão do cotovelo	1	-1	-1	-1
Extensão do cotovelo	-1	1	-1	-1
Supinação do antebraço	-1	-1	-1	1

Baseado nas recomendações da (FAUSETT, 1994), foram utilizadas três diferentes abordagens como critério de parada:

- *Erro quadrático total máximo:* quando o somatório dos erros quadráticos de cada neurônio de saída da rede é inferior à tolerância estabelecida, o treinamento é interrompido;
- *Número máximo de ciclos:* quando um número máximo de ciclos de treinamento é atingido, este é interrompido. Experimentalmente, verificou-se que com mais de 10000 ciclos de treinamento, dificilmente era observada a convergência da rede. Desta forma, foi utilizado este valor (10000) como número máximo de ciclos para o treinamento da rede;

- *Aumento do erro*: o último critério de parada corresponde ao aumento do erro em sucessivos ciclos de treinamento. Experimentalmente, verificou-se que quando o erro aumenta por mais de 100 ciclos consecutivos, a capacidade de convergência da rede é perdida. Desta forma, foi utilizado este valor como referência para este critério de parada.

Todas as informações sobre os resultados alcançados, variando-se os parâmetros da rede para as diversas classes de movimento, serão discutidos no Capítulo 6.

5.4 Ambiente virtual

Nesta seção são abordadas todas as informações relacionadas ao AV tais como: detalhes de modelagem do braço virtual, movimentos realizados e a forma de comunicação com o sistema de classificação.

5.4.1 Braço virtual

O braço modelado por (LAMOUNIER; LOPES; SOARES, 2012) foi adaptado, com intuito de tornar possível a realização de todos os movimentos descritos na Tabela 5.1 e visualizados na seção 5.4.2.

A Figura 5.6a apresenta o modelo 3D do braço, com algumas adaptações para o Blender3DTM. Através da malha do braço apresentado, foi possível atribuir ao mesmo os juntas, que permitem que a malha do objeto se deforme acompanhando os movimentos de cada osso, gerando o movimento. A Figura 5.6b apresenta a armadura do braço, que é composta de 27 juntas.

5.4.2 Movimentos

Visando oferecer ao paciente um *feedback* visual de cada um dos sete movimentos, que podem ser coletados a partir de um sEMG advindo da contração muscular e reconhecidos pelo sistema de classificação, foram geradas sete diferentes animações.

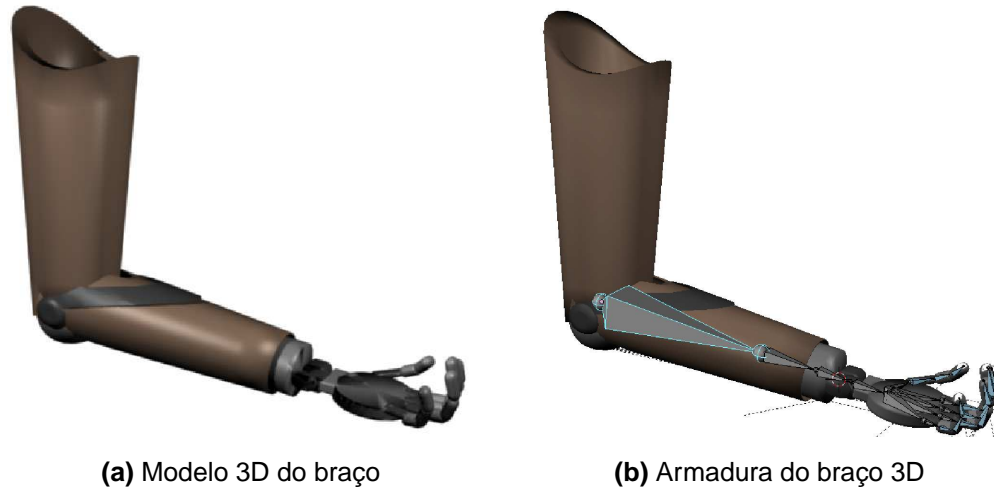


Figura 5.6: Modelo 3D do braço

Para se criar cada uma das mesmas, deve-se inicialmente definir em qual *frame* cada animação irá começar, e qual será a posição dos ossos neste instante. Após é selecionado qual será o ultimo *frame* para a animação e qual a posição dos ossos neste ponto. Com isto ao inicializar a animação o Blender3DTM realizará a interpolação nos *frames* intermediários, fornecendo ao paciente a visualização do movimento classificado.

As Figuras 5.7, 5.8, 5.9, 5.10 , 5.11, 5.12 e 5.13 apresentam as posições iniciais, intermediárias e finais de cada movimento conforme descrito anteriormente.

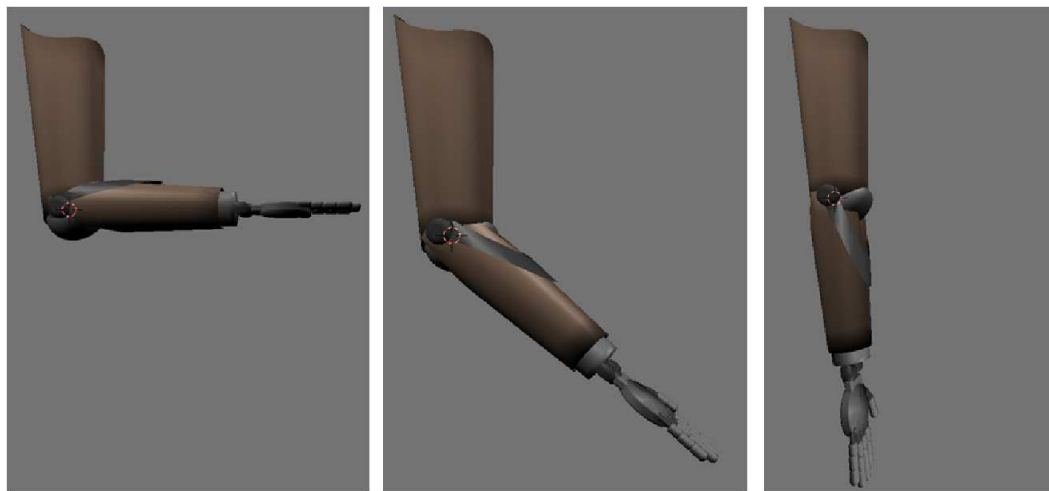


Figura 5.7: Extensão de cotovelo

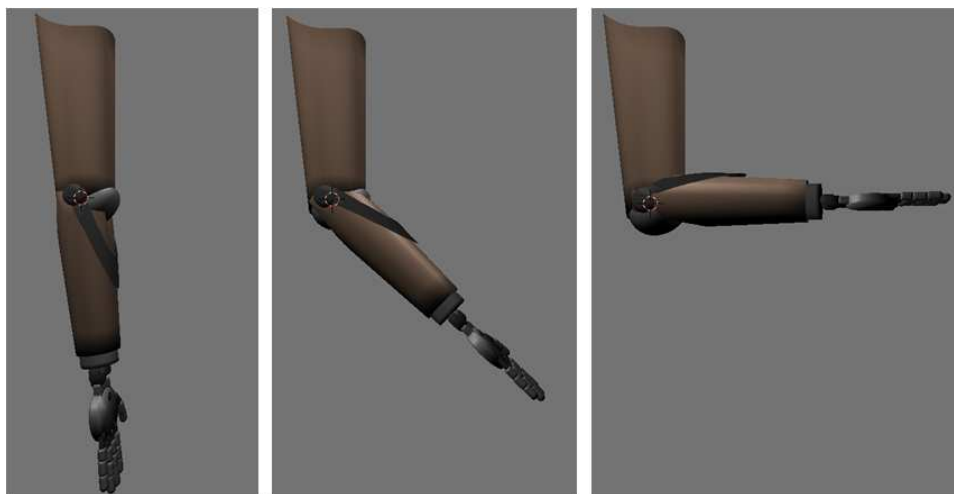


Figura 5.8: Flexão de cotovelo

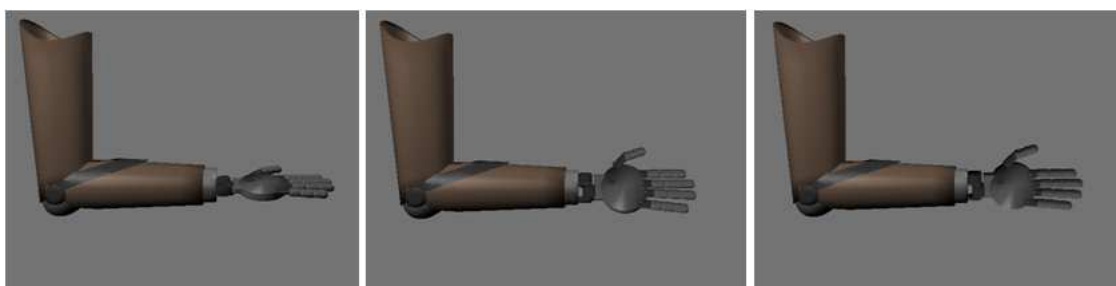


Figura 5.9: Pronação do antebraço

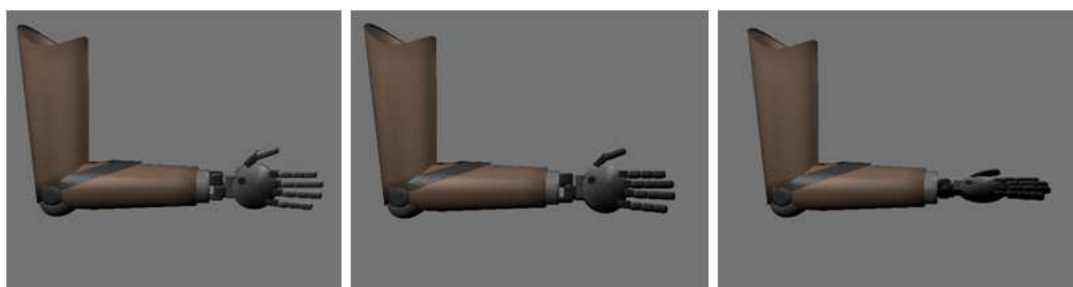


Figura 5.10: Supinação do antebraço



Figura 5.11: Extensão de mão

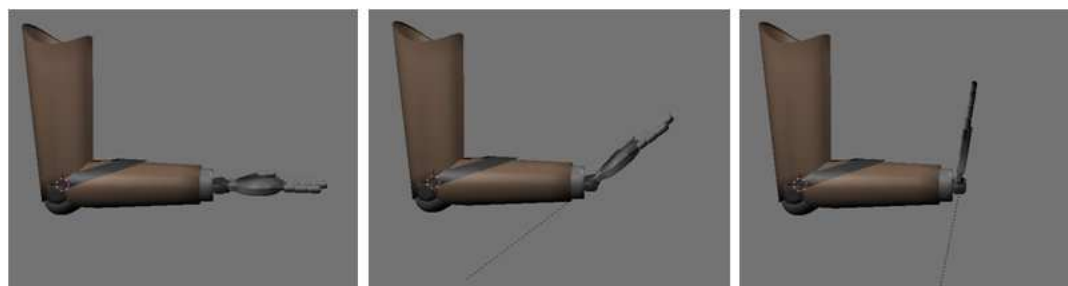


Figura 5.12: Flexão de mão



Figura 5.13: Fechamento de mão

5.4.3 Comunicação com sistema de classificação

Visando atender os requisitos determinados é necessário que exista um meio de comunicação entre o sistema de classificação e o AV. Baseado nos conceitos de sistemas operacionais, é possível que processos distintos troquem informações entre si através de um canal denominado *pipeline*. Sendo assim, o AV instancia o sistema de classificação como um subprocesso do mesmo, com isto, tem-se uma via de comunicação unidirecional entre o sistema de classificação e o AV. Cada segmento classificado pelo sistema, é enviado via *pipeline* para o AV que está sendo suportado pelo Blender3DTM, que interpreta o sinal e engatilha uma das cenas que possui a animação desejada.

5.5 Processamento em tempo real

Baseado na Figura 5.1, pode-se observar que o protótipo desenvolvido possui uma arquitetura de *socket* cliente/servidor, conforme estabelecido por (MATTIOLI et al., 2011). Esta arquitetura foi desenvolvida para que se pudesse avaliar a eficiência do sistema em uma situação de tempo real. Por meio da instância servidor da arquitetura, pode-se selecionar um arquivo que contém amostras referentes a um dos movimentos estudados, e enviar suas amostras há uma frequência fixa para a instância cliente da arquitetura, responsável por detectar as bordas de início e fim de um movimento, extrair características dos segmentos e classificar em tempo real.

Estas três ações são realizadas paralelamente por meio de três *Threads* que alternam entre si. Nas seções seguintes serão detalhadas cada uma das ações realizadas pelas *Threads*.

5.5.1 Janelamento

Conforme explicado anteriormente, o método de janelamento utilizado, impacta diretamente o processamento em tempo real, pois, é ele que determinará quando uma janela se inicia e termina, separando quais amostras são relevantes ou quais são

ruídos.

O Operador de Energia de Teager (TEO), é um método para detecção de fronteira em tempo real criado por Peretta (PERETTA, 2010) usado por Mattioli et. al (MATTIOLI et al., 2011). Este operador permite determinar quais os intervalos com maior e menor energia do sinal, através de uma análise simultânea de amplitude e frequência do sinal.

$$\Psi[x(n)] = x_n^2 - x_{n-1}.x_{n+1} \quad (5.14)$$

A equação 5.14 define o TEO no domínio discreto no qual todas as amostras recebidas serão aplicadas.

Onde: Ψ é o operador TEO, $x(n)$ é a n_{esima} amostra de sinal.

Todas as amostras são armazenadas temporariamente em um *buffer* até fechar a janela. Após, todas as amostras armazenadas no *buffer* são liberadas para que a *Thread* responsável pela extração das características possa realizar suas funções.

5.5.2 Extração de características

Detectado que um novo *buffer* de dados contendo as novas amostras que compõem uma janela já se encontra disponível, a primeira função desta *Thread* é segmentar em grupos de 40 amostras cada um, similar ao utilizado por (MATTIOLI, 2012). A seguir, a cada um dos segmentos são aplicadas uma das técnicas de extração de características detalhadas na seção 2.2.4.1 e 2.2.4.2, gerando os vetores de características que são armazenados em um outro *buffer*. Logo após a finalização da extração das características o mesmo é liberado, para que a *Thread* de classificação possa realizar suas funções.

5.5.3 Classificação

Liberado o novo vetor de características, a *Thread* inicializa sua classificação, utilizando umas das redes neurais implementadas (LVQ ou MLP).

Caso tenha sido escolhida a técnica TDF para extração das características, o vetor apresentará 5 posições distintas, contendo em cada uma delas as cinco características determinadas que serão apresentadas a rede como entrada, conforme exemplificado na letra (a) da Figura 5.14. No entanto, caso tenha sido o MAR escolhido como técnica para extração das características, ao invés de uma redução no volume de informação por segmento, teremos um aumento, pois, cada amostra será representada por M números de AR coeficientes, ou seja, $40 \times M$ coeficientes para cada segmento serão apresentados a rede escolhida.

Independentemente da técnica escolhida, foi adotada a metodologia de classificação utilizada por (MATTIOLI et al., 2011), conforme visto nas letras (b) e (c) da Figura 5.14 abaixo. Ao fim da classificação de todos segmentos, é verificado qual aquele obteve maior repetição conforme descrito na letra (c) da Figura 5.14 o que determina qual é o movimento reconhecido.

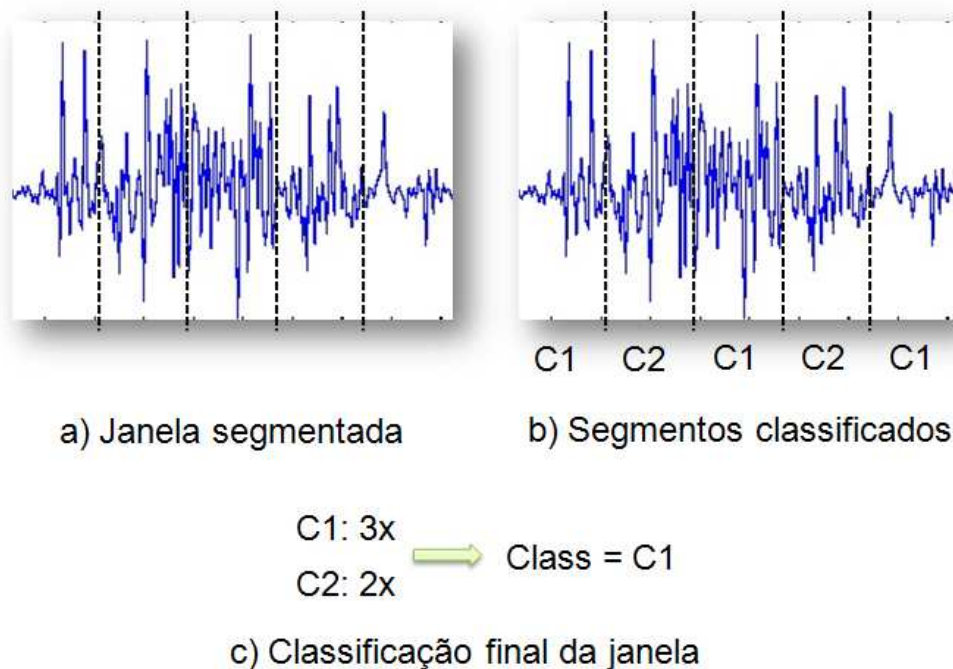


Figura 5.14: Metodologia de classificação

5.6 Considerações finais

Neste capítulo, foram abordados os principais detalhes de implementação do sistema desenvolvido. No próximo capítulo, será apresentado os resultados obtidos e a metodologia utilizada para alcançá-los.

Capítulo 6

Resultados obtidos

6.1 Introdução

Neste capítulo, será abordada a metodologia utilizada nos testes, assim como, os resultados obtidos para cada uma das redes neurais implementadas com as diferentes técnicas de extração de características.

6.2 Metodologia para avaliação de desempenho

Objetivando validar a melhor combinação formada entre os diferentes classificadores e técnicas de extração de características, capazes de oferecer melhor desempenho em termos de tempo de treinamento e eficiência na classificação dos padrões de sinais EMG utilizados, provendo ao AV, maior confiabilidade é que se propõe a metodologia descrita nas seções seguintes.

Resalta-se que há duas bases de dados distintas, coletadas de formas diferentes. Visando encontrar uma forma comum para avaliação das mesmas, apesar das diferenças, definiu-se que, para todos os testes realizados, a quantidade de padrões de sinais utilizados na fase de treinamento será variada, a fim de encontrar qual é o valor mínimo de padrões a serem utilizados para ambos os casos, permitindo um alcance máximo de eficiência. Baseado no trabalho realizado por (HERLE et al., 2008),

serão utilizados no máximo 90% do total de padrões de movimento durante a fase de treinamento. A Tabela 6.1 exibe qual foi o total de padrões de treinamento adquiridos, para cada técnica de extração de característica utilizada.

Tabela 6.1: Número de padrões

TDF			MAR		
Isométrico		Isotônico	Isométrico		Isotônico
Mão	Braço	Braço	Mão	Braço	Braço
71	276	271	3560	11040	10840

Devido à variância do sinal EMG sempre é levado em consideração uma margem de confiança de 95% para todos os testes realizados, e também serão extraídos gráficos que permitam a análise da eficiência (conforme equação 6.1) e do tempo de treinamento para cada variação de teste (LI; SCHULTZ; KUIKEN, 2010).

$$E = 100x \frac{N_{acertos}}{N_{total}} \quad (6.1)$$

6.3 Influência dos parâmetros de configuração

Conforme apresentado na seção anterior, todos os parâmetros de configuração de cada um dos classificadores serão variados. Com os gráficos extraídos, será possível observar quais são os parâmetros ótimos, com a variação de um parâmetro por vez. Segue abaixo a configuração padrão de cada classificador:

- Os valores a serem utilizados na configuração padrão da rede LVQ são:
 - Taxa de aprendizado (α) = 0.1;
 - Taxa de decaimento de (α) = 0.5;
 - Tolerância = 0.01;
 - O número de clusters padrão será diferente para cada base de dados, pois, cada base possui uma quantidade de padrões de treinamento diferente;

2. Os valores padrões a serem utilizados na configuração padrão da rede MLP são:

- Taxa de aprendizado (α) = 0.1;
- Tolerância = 150;
- Número de neurônios na camada de oculta para os padrões de braço e mão é igual a 20;

Serão realizados 100 testes para a variação de cada parâmetro utilizando a técnica de extração de características TDF e 25 testes para a técnica MAR. A fim de avaliar a influência da variação de cada parâmetro no tempo de treinamento da rede, implementou-se um contador/*timer*, que mensurará o intervalo entre o tempo de início e fim de cada treinamento.

Na próxima seção são apresentados os gráficos relacionados aos testes realizados, que demonstram os melhores resultados obtidos, assim como, as elucidações em torno dos mesmos. Para mais detalhes em torno dos resultados obtidos e demais testes realizados, verificar a seção de Apendice (A).

6.4 Avaliação de performance classificador LVQ

6.4.1 Movimentos de mão - MAR

6.4.1.1 Taxa de aprendizagem (α)

É possível verificar na Figura 6.1a que, para todas as ordens do MAR e para os diferentes valores de taxa de aprendizagem (α), a forma da curva foi a mesma. Apenas a amplitude do tempo de treinamento sofreu alterações, pois, quanto maior a ordem do MAR, maior é o número de coeficientes AR e maior é o tempo de treinamento. Pequenos valores de taxa de aprendizagem (α) implicam em menos ciclos de treinamento, ou seja, à medida que a taxa de aprendizagem (α) aumenta, aumenta-se também o número de ciclos e o tempo de treinamento.

Conforme pode-se observar no gráfico apresentado pela Figura 6.1b, à medida que o valor da taxa de aprendizagem (α) é aumentada, a eficiência da rede é reduzida

para todas as ordens de MAR.

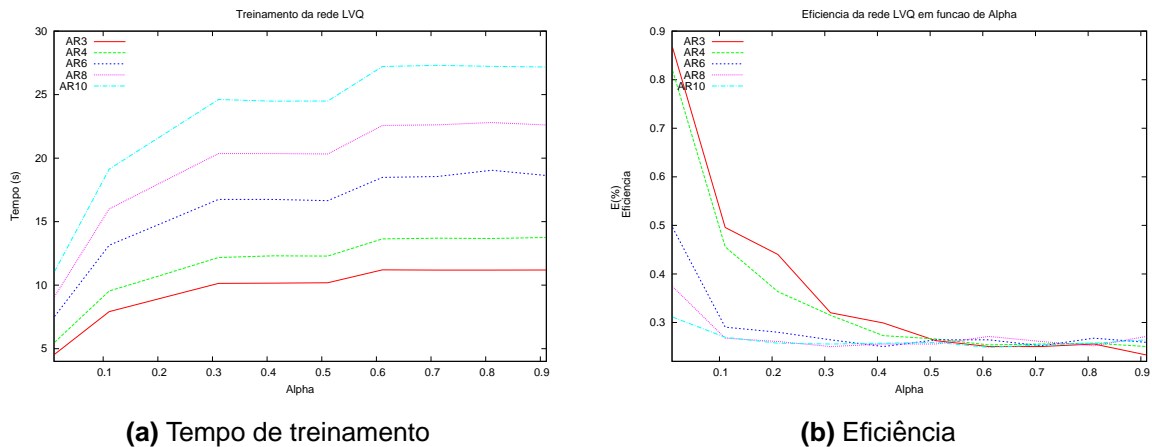


Figura 6.1: Gráficos relacionados à taxa de aprendizagem (α)

Observa-se que para valores de taxa de aprendizagem (α) entre 0.01 a 0.5, a eficiência da rede para os modelos de ordem de 3 e 4 foram os melhores. Já para os modelos de ordem entre 6 e 10, não apresentaram a mesma eficiência, pois, quanto mais se aumenta a ordem do modelo, menor é a capacidade de generalização da rede, fazendo com que sua eficiência reduza.

No entanto, é possível verificar que à medida que o valor da taxa de aprendizagem (α) é aumentada, inclusive para os valores da taxa de aprendizagem (α) maior que 0.5, há uma convergência na redução da eficiência da rede, pois, conforme dito anteriormente, o valor da taxa de aprendizagem (α) influencia consideravelmente o ajuste dos pesos das unidades de saída.

6.4.1.2 Taxa de decaimento de (α)

Observa-se na Figura 6.2a que elevados valores de decaimento de (α) geram uma redução mais lenta nos valores da taxa de aprendizado (α), o que resulta em um aumento na quantidade de ciclos de treinamento. É importante ressaltar que, a variação da taxa de decaimento de (α) teve relevante significância apenas para os modelos de ordem 3 e 4, nos demais, a variação do decaimento de (α) não influenciou tanto no tempo de treinamento da rede.

A Figura 6.2b apresenta a variação da eficiência da rede em função da variação

do decaimento de (α), para cada ordem do MAR. Conforme pode-se observar, para elevados valores de decaimento de (α) ocorre uma redução significativa na eficiência da rede, principalmente para os MAR de ordem 3 e 4. Isto pode ser explicado pelo fato de elevados valores de decaimento de (α), geram uma lenta redução nos valores da taxa de aprendizado (α), que resulta em uma considerável atualização nos valores dos pesos a cada ciclo de treinamento.

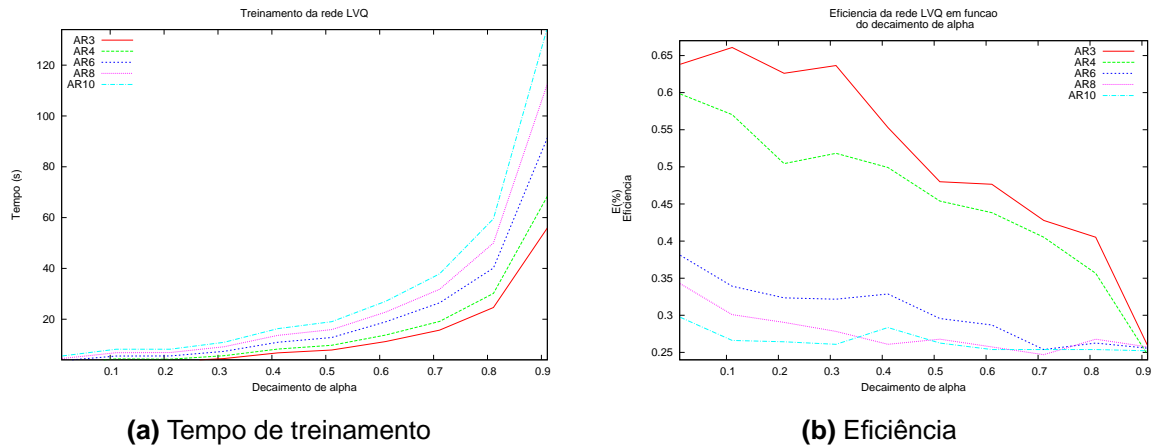


Figura 6.2: Gráficos relacionados à taxa de decaimento de (α)

6.4.1.3 Tolerância

Para os valores de tolerância entre 0.001 e 0.05 exibidos na Figura 6.3a, pode-se observar que o tempo de treinamento da rede reduz consideravelmente e proporcionalmente para todas as ordens de MAR, devido a redução nos ciclos de treinamento executados.

Avaliando os valores de tolerância, apresentados na Figura 6.3b pode-se observar que para os valores entre 0.001 e 0.03, a eficiência da rede reduz consideravelmente para os modelos de ordem 3 e 4, isto ocorre devido a grande quantidade de ciclos de treinamento executados (*over training*), que faz com que os valores dos pesos da rede sejam alterados muitas vezes, reduzindo a capacidade de generalização da mesma. Porém, a eficiência da rede cresce consideravelmente para os MAR de ordem 3 e 4, para valores de tolerância entre 0.03 e 0.06, o que determina uma redução na quantidade de ciclos de treinamento, evitando assim o *over training*.

Já para os valores de tolerância maiores que 0.06, novamente a eficiência da rede

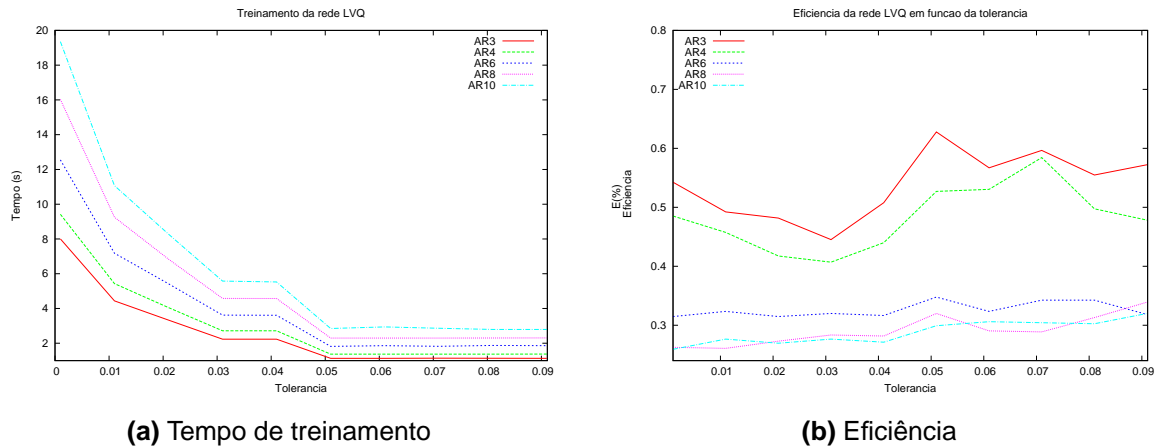


Figura 6.3: Gráficos relacionados à tolerância

volta a reduzir e isto se deve, pelo fato da quantidade de ciclos de treinamento serem reduzidas, de maneira que a rede não é treinada corretamente, ou seja, seus pesos não são devidamente atualizados.

Ressalta-se que a variação da tolerância gerou uma alteração específica em cada curva, para cada ordem do MAR que pode ser facilmente interpretada, a partir do esclarecimento anterior.

6.4.1.4 Unidades de saída

É possível verificar na Figura 6.4a que, quanto menor a quantidade de unidades de saída menor é o tempo de treinamento, isto explica-se pelo fato de ter-se inicialmente poucos pesos da rede para serem atualizados e os cálculos são realizados mais rapidamente. No entanto, à medida que este valor é aumentado até mais ou menos 70% do total de padrões, o tempo cresce proporcionalmente, pois a quantidade de cálculos aumenta, aumentando também o tempo de treinamento. À medida que se aumenta a ordem do modelo também aumenta-se o tempo de treinamento, gastando-se até 130 segundos para treinar a rede.

Inicializando a rede com uma quantidade de unidades de saída igual a 400, conforme visto na Figura 6.4b, todos os MAR conseguiram alcançar uma eficiência de 100% na classificação dos padrões.

Para uma quantidade de unidades de saída menor que 400, pode-se observar que,

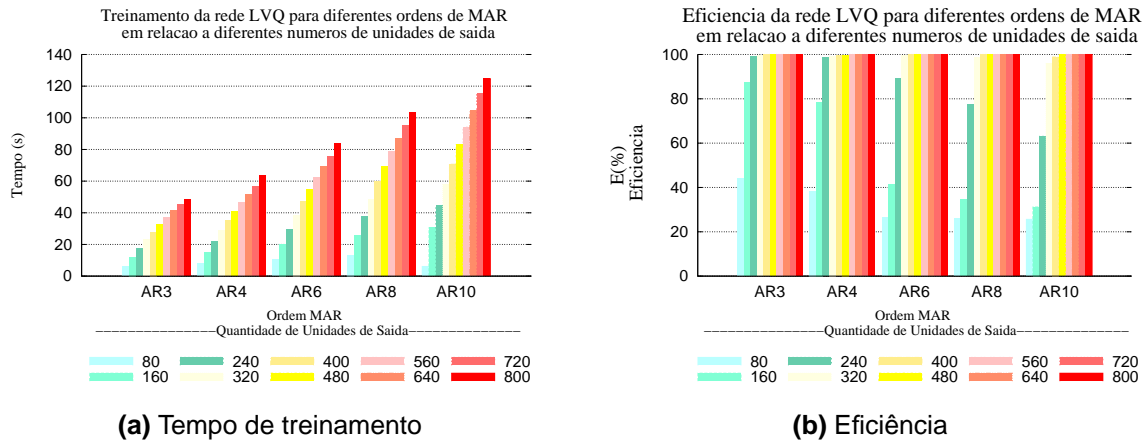


Figura 6.4: Gráficos relacionados às unidades de saída

à medida que é aumentado a ordem do MAR, a eficiência da rede reduz, e isto pode ser explicado, pois, à medida que é aumentado a ordem do modelo a rede perde a capacidade de generalização por não conseguir distinguir bem os padrões de movimentos.

Conclui-se que, para se obter uma boa resposta da rede, basta utilizar um MAR de ordem 3 com 400 unidades de classificação. Importante ressaltar que, de 3560 padrões de treinamento, somente 400 padrões foram utilizados para inicializar o treinamento da rede, ou seja, apenas 11,23% do total de padrões é suficiente para se obter uma boa resposta.

6.4.2 Movimentos de braço - MAR (Isométrico)

6.4.2.1 Taxa de aprendizagem (α)

Pode-se observar na Figura 6.5a que, para todas as ordens do MAR e para os diferentes valores de taxa de aprendizagem (α), a forma da curva foi a mesma. Apenas a amplitude do tempo de treinamento sofreu alterações, pois, quanto maior a ordem do MAR, maior é o número de coeficientes AR e maior é o tempo de treinamento. Conforme dito anteriormente, pequenos valores de taxa de aprendizagem (α) implicam em menos ciclos de treinamento, ou seja, à medida que a taxa de aprendizagem (α) aumenta, aumenta-se também os número de ciclos e o tempo de treinamento.

Conforme pode-se observar no gráfico apresentado pela Figura 6.5b, à medida que o valor da taxa de aprendizagem (α) é aumentada, a eficiência da rede é reduzida para todas as ordens de MAR.

Avaliando os valores de taxa de aprendizagem (α) apresentados na Figura 6.5b pode-se observar que, para valores entre 0.01 a 0.3, a eficiência da rede para MAR de ordem 8 foi o melhor. Já as demais ordens de MAR não apresentaram a mesma eficiência.

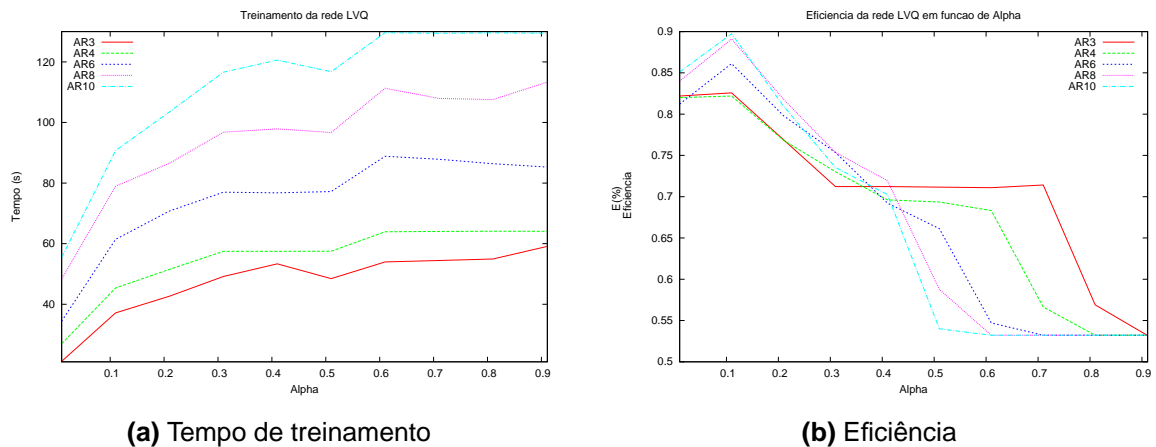


Figura 6.5: Gráficos relacionados à taxa de aprendizagem (α)

6.4.2.2 Taxa de decaimento de (α)

É possível verificar na Figura 6.6a que elevados valores de decaimento de (α) geram uma redução mais lenta nos valores da taxa de aprendizado (α), o que resulta em um aumento na quantidade de ciclos de treinamento. Ressalta-se que a variação da taxa de decaimento de (α) teve relevante significância apenas para o modelo de ordem 10, nos demais, a variação do decaimento de (α) não influenciou tanto no tempo de treinamento da rede.

A Figura 6.6b apresenta a variação da eficiência da rede em função da variação do decaimento de (α) para cada ordem do MAR. Observa-se que durante o aumento da taxa de decaimento de (α) a eficiência da rede para os MAR de ordem 8 e 10 foram as melhores. No entanto, para todas as ordens de MAR a eficiência foi reduzindo à medida que a taxa de decaimento de (α) aumentou, porque gerarem uma lenta

redução nos valores da taxa de aprendizado (α), que resulta em uma considerável atualização nos valores dos pesos a cada ciclo de treinamento.

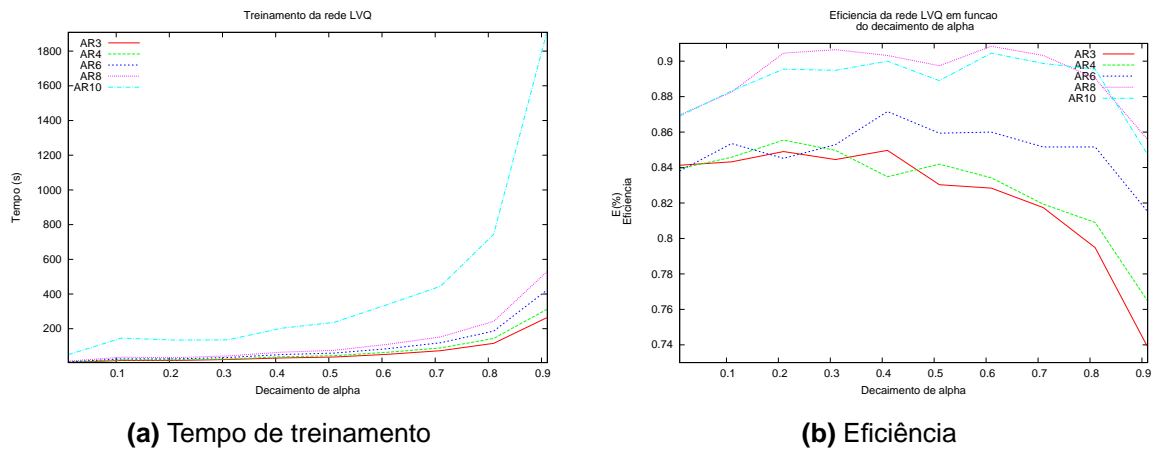


Figura 6.6: Gráficos relacionados à taxa de decaimento de (α)

6.4.2.3 Tolerância

Para os valores de tolerância entre 0.001 e 0.05 na Figura 6.7a, observa-se que o tempo de treinamento da rede reduz consideravelmente e proporcionalmente para todas as ordens de MAR, devido à redução nos ciclos de treinamento executados.

É possível verificar na Figura 6.7b que, para cada ordem de MAR a curva de eficiência teve uma característica diferente. Isto explica-se facilmente, pois, cada conjunto de coeficientes AR permitem à rede neural, um melhor ou pior discernimento de cada padrão durante a classificação.

6.4.2.4 Unidades de saída

Observa-se na Figura 6.8a que, quanto menor a quantidade de unidades de saída menor é o tempo de treinamento, isto explica-se pelo fato de ter-se inicialmente poucos pesos da rede para serem atualizados e os cálculos são realizados mais rapidamente. Uma outra observação que pode ser realizada é que à medida que se aumenta a ordem do modelo também aumenta-se o tempo de treinamento, gastando-se até 380 segundos para treinar a rede para um MAR de ordem 3.

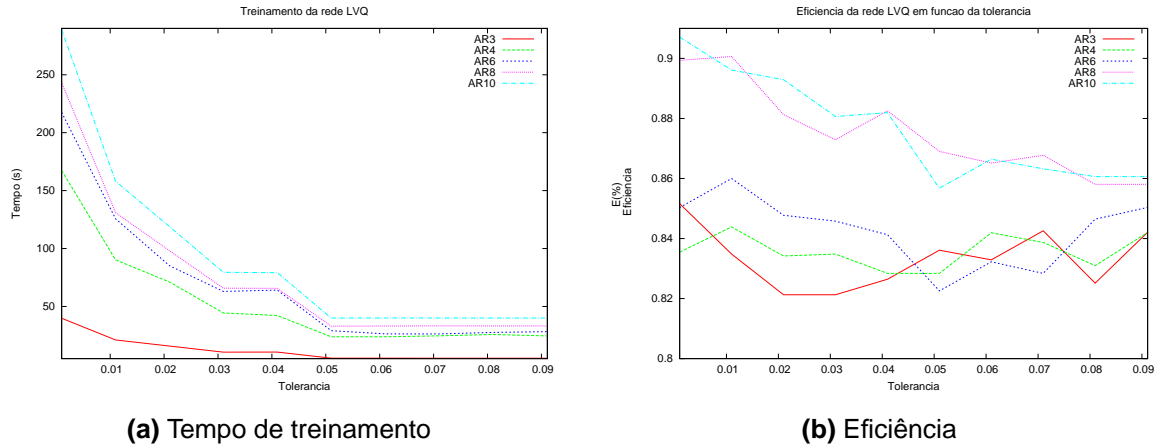


Figura 6.7: Gráficos relacionados à tolerância

Inicializando a rede com uma quantidade de unidades de saída igual a 1200 que é equivalente a 1200 padrões de treinamento utilizados e um MAR de ordem 3, observa-se na Figura 6.8b que todos os MAR conseguiram alcançar uma eficiência maior que 97% na classificação dos padrões, utilizando-se apenas 10% do total de padrões de treinamento.

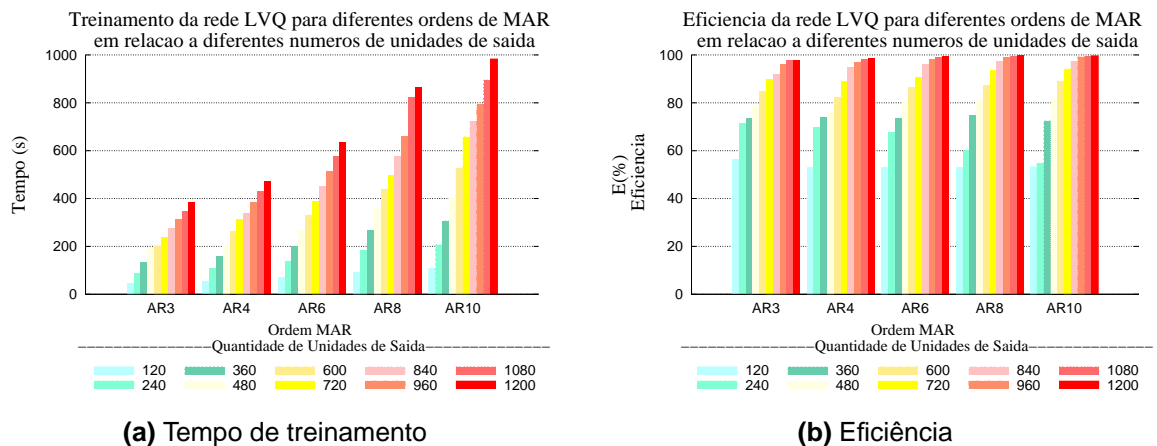


Figura 6.8: Gráficos relacionados às unidades de saída

6.5 Considerações finais

A motivação principal deste trabalho é avaliar qual a combinação de redes neurais e técnica de extração de característica, que garante maior confiabilidade no controle de próteses virtuais para membros superiores para um paciente que tenha sofrido

algum dano em um braço, antebraço ou mão.

Os testes realizados foram utilizados para que fosse possível detectar qual a combinação de classificadores e técnicas de extração de características, que permitisse alcançar maior performance em termos de eficiência e tempo de treinamento.

A Tabela 6.2 exibe um resumo de todos os testes realizados assinalando em **negrito** aqueles em que a performance foi melhor.

Tabela 6.2: Resumo dos resultados

Movimento	Rede	Técnica	% Padrões	% Eficiência	Tempo Treinam.
Mão	LVQ	TDF	72%	97%	≤ 1
		MAR	11%	99%	27
	MLP	TDF	90%	85%	100
		MAR	2%	27%	25
Braço	LVQ	TDF	90%	80%	≤ 1
		MAR	10%	97%	380
	MLP	TDF	90%	68%	1500
		MAR	2%	80%	500

Analisando os valores apresentados na Tabela 6.2, é possível verificar que a combinação da rede LVQ com a técnica de extração de característica MAR apresentou-se como a melhor opção em ambos os casos, para os movimentos de braço e mão. Além disto, pode-se observar que o tempo de treinamento para o movimento de braço foi elevado, mas algumas sugestões de trabalhos futuros podem atenuar/reduzir o tempo de treinamento da rede.

A Tabela 6.3 apresenta a contribuição deste trabalho, em relação aos trabalhos correlatos apresentados anteriormente no Capítulo 3 e no próximo capítulo, serão apresentadas as conclusões e as propostas de trabalhos futuros.

Sendo assim, a combinação da rede LVQ com a técnica de extração de característica baseada no MAR, apresentou uma elevada taxa de eficiência que reduz a possibilidade do ambiente virtual, retornar ao paciente um movimento diferente daquele coletado pela contração muscular realizada pelo mesmo, fornecendo maior con-

Tabela 6.3: Características analisadas nos trabalhos relacionados

Trabalhos relacionados	Movimentos mão e braço	Janelamento dinâmico	Pré-processamento TDF	Pré-processamento MAR	Classificador LVQ	Classificador MLP	Teste performance tempo real	Realidade virtual
(HUANG et al., 2003)	✗	✗	✗	✓	✗	✓	✓	✗
(SOARES et al., 2003)	✗	✗	✗	✓	✗	✓	✓	✓
(HERLE et al., 2008)	✗	✗	✓	✗	✗	✓	✗	✓
(MATTIOLI et al., 2011)	✗	✓	✓	✗	✓	✗	✓	✓
(CAETANO et al., 2012)	✓	✓	✓	✓	✓	✓	✓	✓

fiabilidade.

Capítulo 7

Conclusões e trabalhos futuros

7.1 Introdução

Neste capítulo, destacam-se os aspectos da pesquisa e conclusões do trabalho apresentado nesta dissertação. Além disso, serão apresentadas algumas sugestões para trabalhos futuros e a contribuição científica do presente trabalho.

7.2 Conclusões

Analisando os resultados alcançados com o classificador LVQ devido à combinação de técnicas de extração de características autoregressivas o aponta como uma boa alternativa que oferecerá uma boa confiabilidade aos *feedbacks* visuais que o AV virtual oferecerá ao paciente, respeitando a afirmação realizada por Machover (MACHOVER; TICE, 1994) de que os sistemas em RV precisam fornecer uma reação de forma coerente aos movimentos do usuário, tornando a experiência consistente.

Em termos de eficiência esta combinação é totalmente indicada se comparada aos índices alcançados nos trabalhos realizados por (HERLE et al., 2008; MATTIOLI, 2012; SOARES et al., 2003). Utilizando-se apenas 10% do total dos padrões de treinamento foi possível alcançar uma eficiência maior que 97% para quase todos os casos. O que não se pode concluir para a rede MLP, por ter apresentado durante os testes,

uma baixa eficiência e um elevado tempo de treinamento, devido as características arquiteturais da própria.

Pode-se perceber que a concepção da arquitetura proposta baseada no trabalho realizado por (MATTIOLI et al., 2011), é muito indicada para a simulação de controle de um AV em tempo real, graças a combinação do detector de bordas TEO utilizados por (PERETTA, 2010), as técnicas de segmentação utilizadas por (HERLE et al., 2008) e as técnicas de classificação utilizadas por (MATTIOLI et al., 2011), permitindo que o usuário possa realizar diversos testes, tendo-se para isto apenas uma base de dados propriamente preparada.

Este trabalho reforça a boa aplicação de RV em processos de reabilitação, por apresentar uma interface com a qual o usuário pode interagir de forma natural, recebendo do sistema *feedbacks* visuais em tempo real de todas as ações realizadas, permitindo que o mesmo em diversas situações, não seja exposto em risco durante uma fase de treinamento e adaptação à prótese e reduzindo o seu esforço mental, que pode evitar que o mesmo venha a optar pela utilização de uma prótese passiva ao invés de utilizar uma prótese ativa, devido as dificuldades encontradas nesta fase. Além de permitir a abstração de diversas contribuições para a área em estudo. Uma delas é que mesmo independente da técnica utilizada, um ponto crítico na hora da coleta dos sinais é a quantidade de canais utilizados, pois, nos estudos realizados foram utilizados, apenas um par de eletrôdos para ambos os casos. E, baseado nos testes realizados, como por exemplo com a rede LVQ utilizando TDF, uma boa alternativa para tentar elevar a eficiência da rede para movimento de braços, é aumentar o número de canais de coleta, permitindo assim ter-se mais informações para que a rede possa separar melhor os movimentos.

7.3 Trabalhos futuros

Baseado no trabalho realizado, propoe-se algumas colaborações para trabalhos futuros tais como:

- Realizar novamente os testes, mas com bases de movimentos com características

de coleta igualmente realizadas;

- A fim de reduzir o tempo de treinamento apresentado para a rede LVQ para ambos movimentos, utilizar durante a fase de detecção de janela apenas os primeiros 200ms coletados, assim como estudado por (SOARES et al., 2003; HERLE et al., 2008), para verificar qual a redução no tempo de treinamento;
- Avaliar o desempenho desta arquitetura elaborada em tempo real em um paciente real e um profissional de saúde;
- Avaliar a utilização de outras técnicas diferentes de redes neurais, como por exemplo, *wavelet*, para o reconhecimento de padrões de treinamento em tempo real.

Referências bibliográficas

AKAY, M. *Biomedical signal processing*. [S.l.]: Academic Press San Diego, 1994.

ANDRADE, A. O. de. *Metodologia para Classificação de Sinais EMG no Controle de Membros Artificiais*. Dissertação (Mestrado) — Universidade Federal de Uberlândia, Uberlândia, Janeiro 2000.

BARROS, K. R. de. *Metodologia para Classificação de Sinais EMG para Controle de Próteses com Baixo Esforço Computacional*. Dissertação (Mestrado) — Universidade Federal de Uberlândia, Uberlândia, Agosto 2005.

CAETANO, D. et al. Comparison between mlp and lvq neural networks for virtual upper limb prosthesis control. In: YIN, H.; COSTA, J.; BARRETO, G. (Ed.). *Intelligent Data Engineering and Automated Learning - IDEAL 2012*. [S.l.]: Springer Berlin Heidelberg, 2012, (Lecture Notes in Computer Science, v. 7435). p. 382–389. ISBN 978-3-642-32638-7.

CHU, J.; MOON, I.; MUN, M. A real-time emg pattern recognition system based on linear-nonlinear feature projection for a multifunction myoelectric hand. *Biomedical Engineering, IEEE Transactions on, IEEE*, v. 53, n. 11, p. 2232–2239, 2006.

COSTA, R. M.; RIBEIRO, M. W. (Ed.). *Aplicações de Realidade Virtual e Aumentada - Livro do pré-simpósio do SVR2009*. Porto Alegre: SBC, 2009.

DALLEY, S. et al. Design of a multifunctional anthropomorphic prosthetic hand with extrinsic actuation. *Mechatronics, IEEE/ASME Transactions on, IEEE*, v. 14, n. 6, p. 699–706, 2009.

FAUSETT, L. (Ed.). *Fundamentals of neural networks: architectures, algorithms, and applications*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1994. ISBN 0-13-334186-0.

FEINTUCH, U. et al. Virhab-a virtual reality system for treatment of chronic pain and disability. In: IEEE. *Virtual Rehabilitation International Conference, 2009*. [S.l.], 2009. p. 83–86.

HAYKIN, S. *Neural networks: a comprehensive foundation*. [S.l.]: Prentice Hall PTR Upper Saddle River, NJ, USA, 1999.

HEFFTNER, G.; ZUCCHINI, W.; JAROS, G. The electromyogram (emg) as a control signal for functional neuromuscular stimulation. i. autoregressive modeling as a means of emg signature discrimination. *Biomedical Engineering, IEEE Transactions on, IEEE*, v. 35, n. 4, p. 230–237, 1988.

HERLE, S. et al. Classification of surface electromyographic signals for control of upper limb virtual prosthesis using time-domain features. *Proceedings of the 2008 IEEE International Conference on Automation, Quality and Testing, Robotics*, p. 160–165, 2008.

HESSE, F.; HERRMANN, J. Homeokinetic proportional control of myoelectric prostheses. In: IEEE. *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*. [S.l.], 2010. p. 1786–1791.

HOCHBERG, L. et al. Neuronal ensemble control of prosthetic devices by a human with tetraplegia. *Nature*, Nature Publishing Group, v. 442, n. 7099, p. 164–171, 2006.

HUANG, H. et al. Emg classification for prehensile postures using cascaded architecture of neural networks with self-organizing maps. In: IEEE. *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*. [S.l.], 2003. v. 1, p. 1497–1502.

HUDGINS, B.; PARKER, P.; SCOTT, R. A new strategy for multifunction myoelectric control. *IEEE Transactions on Biomedical Engineering*, v. 40, n. 1, p. 82–94, 1993.

KOHDEN, N. Nihon kohden homepage. *Disponível em*: < <http://nihonkohden.de> >, 2011.

KUTARNIA, J.; PEDERSEN, P.; YUAN, C. Virtual reality training system for diagnostic ultrasound. In: IEEE. *Ultrasonics Symposium (IUS), 2010 IEEE*. [S.l.], 2010. p. 1652–1656.

LAMOUNIER, E.; LOPES, K.; SOARES, A. Using augmented reality techniques to simulate myoelectric upper limb prostheses. In: BIOENGINEERING & BIOMEDICAL SCIENCE. *J Bioengineer & Biomedical Sci S1:010*. doi:10.4172/2155-9538.S1-010. [S.l.], 2012.

LI, G.; SCHULTZ, A.; KUIKEN, T. Quantifying pattern recognition-based myoelectric control of multifunctional transradial prostheses. *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, IEEE, v. 18, n. 2, p. 185–192, 2010.

MACHOVER, C.; TICE, S. Virtual reality. *Computer Graphics and Applications, IEEE*, v. 14, n. 1, p. 15–16, jan. 1994. ISSN 0272-1716.

MATTIOLI, F. et al. Utilização de redes neurais para a classificação de sinais emg aplicados no controle de próteses virtuais de mão. *Anais WRVA2010*, p. 230–235, 2010.

MATTIOLI, F. et al. Classification of emg signals using artificial neural networks for virtual hand prosthesis control. In: IEEE. *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*. [S.l.], 2011. p. 7254–7257.

MATTIOLI, F. E. R. *Utilização de Técnicas de Realidade Virtual na Simulação de Próteses de Mão*. Dissertação (Mestrado) — Universidade Federal de Uberlândia, Uberlândia, Fevereiro 2012.

MURGIA, A. *A Gait Analysis Approach To The Study Of Upper Limb Kinematics Using Activities Of Daily Living*. Dissertação (Mestrado) — The University of Reading, United Kingdom, September 2005.

NISHIKAWA, D. et al. On-Line Learning Method for EMG Prosthetic Hand Control. *Electronics and Communications in Japan*, p. 1510–1519, 2001.

NOGUEIRA, K. L. *O Uso de Técnicas de Realidade Virtual e Aumentada na Simulação de Prótese de Membros Superiores*. Dissertação (Mestrado) — Universidade Federal de Uberlândia, Uberlândia, Julho 2007.

OLIVEIRA, A. et al. Avaliação eletromiográfica de músculos da cintura escapular e braço durante exercícios com carga axial e rotacional. *Rev Bras Med Esporte*, SciELO Brasil, v. 12, n. 1, p. 11–5, 2006.

ORTOVAN, O. Ortovan ortopedia - técnica ortopédica. Disponível em: <
[http : //www.ortovan.com.br/index.php?option = com_contentview = articleid = 46Itemid = 67](http://www.ortovan.com.br/index.php?option=com_contentview=articleid=46Itemid=67)>, Campinas-SP, 2012.

PERETTA, I. S. *A novel Word Boundary Detector based on the Teager Energy Operator for Automatic Speech Recognition*. Dissertação (Mestrado) — Universidade Federal de Uberlândia, Uberlândia, Dezembro 2010.

PLETTENBURG, D. The wilmer passive hand prosthesis for toddlers. *JPO: Journal of Prosthetics and Orthotics*, v. 21, n. 2, p. 97, 2008.

SEBELIUS, F. et al. Real-time control of a virtual hand. *Technology and Disability*, IOS Press, v. 17, n. 3, p. 131–141, 2005.

SOARES, A. et al. The development of a virtual myoelectric prosthesis controlled by an EMG pattern recognition system based on neural networks. *Journal of Intelligent Information Systems*, Springer, v. 21, n. 2, p. 127–141, 2003.

TOLEDO, C. et al. Upper limb prostheses for amputations above elbow: A review. In: IEEE. *Health Care Exchanges, 2009. PAHCE 2009. Pan American*. [S.l.], 2009. p. 104–108.

VILLIGER, M. et al. Virtual reality rehabilitation system for neuropathic pain and motor dysfunction in spinal cord injury patients. In: IEEE. *Virtual Rehabilitation (ICVR), 2011 International Conference on*. [S.l.], 2011. p. 1–4.

WEISS, P. et al. Video capture virtual reality as a flexible and effective rehabilitation tool. *Journal of NeuroEngineering and Rehabilitation*, BioMed Central Ltd, v. 1, n. 1, p. 12, 2004.

Apêndice A

Apêndice de resultados

São apresentados neste capítulo os resultados relacionados com os demais testes previamente estabelecidos.

A.1 Avaliação de performance classificador LVQ

A.1.1 Movimentos de mão - TDF

Para observar qual a influência da modificação dos parâmetros de configuração da rede neural (taxa de aprendizagem (α), taxa de decaimento de (α) e tolerância), tanto na performance quanto no tempo de treinamento, adotou-se a seguinte abordagem:

1. Mantendo-se os demais parâmetros fixos nos valores padrões, o parâmetro em análise é alterado, dentro da faixa de valores estabelecidos abaixo:

- Os valores de $alpha(\alpha)$ (taxa de aprendizado) serão de 0.01 a 0.99;
- Os valores de dec_alpha (decaimento de alpha) serão de 0.01 a 0.99;
- Os valores de $tolerance$ (tolerância) serão de 0.001 a 0.099;

2. Número de unidades de saída (clusters):

- Serão variados de 7 a 70, com 10 intervalos diferentes utilizando TDF;

- Serão variados de 80 a 800, com 10 intervalos diferentes utilizando MAR, para diferentes ordens do MAR;

A.1.1.1 Taxa de aprendizagem (α)

Na Figura A.1a, pode-se observar que à medida que o valor da taxa de aprendizado (α) é aumentado, o tempo de treinamento da rede também aumenta, devido à variação do valor da taxa de aprendizado (α) que afeta diretamente a atualização dos pesos. A rede LVQ efetua uma redução gradual na taxa de aprendizado (α) a cada ciclo de treinamento. Conforme estabelecido em um dos critérios de parada para a rede LVQ, quando o valor da taxa de aprendizado (α) for menor que a tolerância estabelecida, a fase de treinamento é interrompida. Por esta razão, quanto maior for o valor da taxa de aprendizado (α), maior será o número de ciclos de treinamento e proporcionalmente o tempo de treinamento.

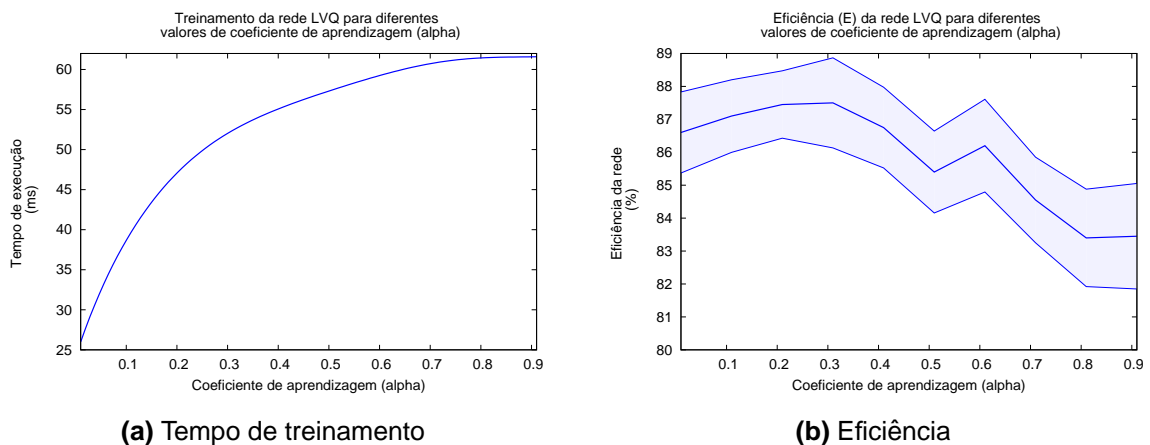


Figura A.1: Gráficos relacionados à taxa de aprendizagem

Já na Figura A.1b observa-se que, para valores de taxa de aprendizado (α) entre 0.01 e 0.3, a eficiência da rede aumenta gradualmente, por permitir um melhor ajuste nos valores dos pesos das unidades de saída. No entanto, para valores maiores que 0.3, observa-se a redução na eficiência da rede, devido aos elevados valores de taxa de aprendizado (α), o que implica em uma atualização grosseira dos pesos, ou a um *over training*, que faz com que os valores dos pesos da rede sejam alterados muitas vezes, reduzindo a capacidade de generalização (FAUSETT, 1994).

A.1.1.2 Taxa de decaimento de (α)

Na Figura A.2a, observa-se que, elevados valores de decaimento de (α) geram uma redução mais lenta nos valores da taxa de aprendizado (α), o que resulta em um aumento na quantidade de ciclos de treinamento. Uma vez que, a cada ciclo de treinamento o valor da taxa de aprendizado (α) é reduzido proporcionalmente ao valor da taxa de decaimento de (α).

A Figura A.2b apresenta a variação da eficiência da rede em função da variação da taxa de decaimento de (α). Pode-se observar que houve picos de eficiência próximos a 90%, para os valores de decaimento de (α) igual a 0.3 e 0.8. Sempre antes a cada um destes valores houve um aumento e após uma redução na eficiência. Isto pode ser explicado pelo fato de que, elevados valores de decaimento de (α), geram uma lenta redução nos valores de taxa de aprendizado (α), que resulta em uma considerável atualização nos valores dos pesos.

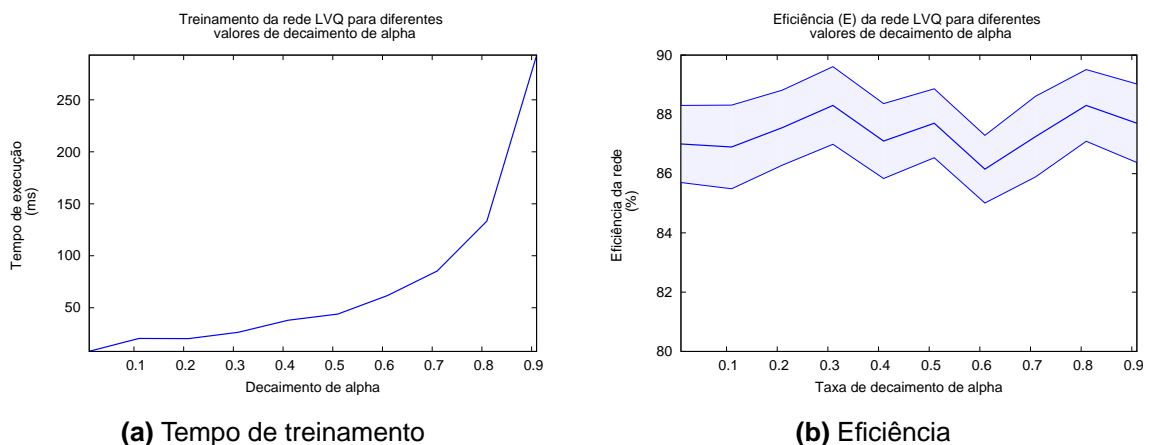


Figura A.2: Gráficos relacionados à taxa de decaimento de (α)

A.1.1.3 Tolerância

Na Figura A.3a pode-se perceber que o tempo de treinamento tende a ser elevado para valores pequenos de tolerância, devido ao aumento no número de ciclos de treinamento, que influencia diretamente o tempo de treinamento. À medida que este valor é aumentado, gradativamente o tempo é reduzido.

Avaliando os resultados apresentados na Figura A.3b pode-se observar que para

pequenos valores de tolerância a rede apresenta uma boa eficiência, porém à medida que este valor é aumentado, a eficiência da rede reduz gradativamente. Isso explica-se pela relação existente entre a taxa de aprendizado (α) e a tolerância, que determinam a condição de parada para o treinamento da rede. Quanto maior for a tolerância, menor é a quantidade de ciclos de treinamento, reduzindo o ajuste nos pesos da rede.

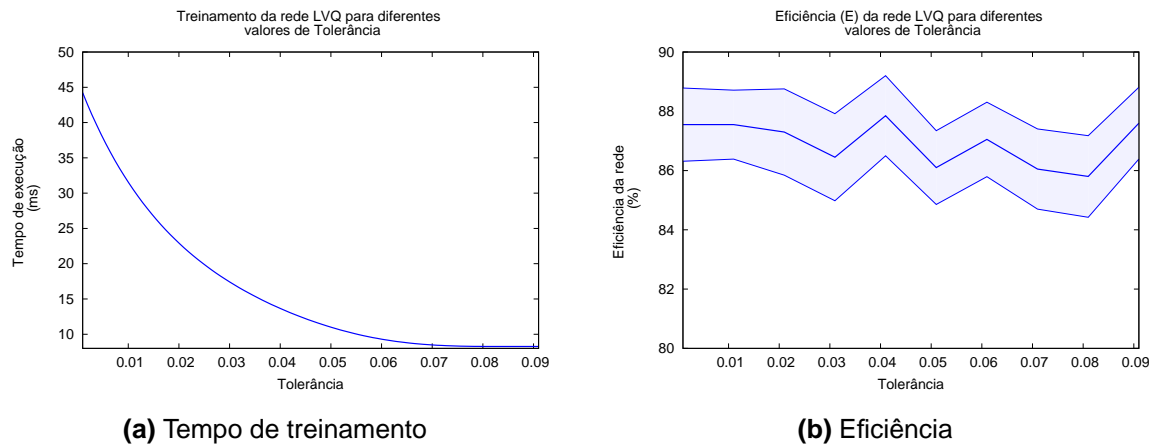


Figura A.3: Gráficos relacionados à tolerância

A.1.1.4 Unidades de saída

Ressalta-se que para as análises relacionadas à variação da quantidade de unidades de saída na eficiência da rede, os padrões de movimentos utilizados na fase de treinamento e execução são os mesmos. No entanto, os padrões utilizados no treinamento não são utilizados na fase de execução. Por esta razão, se a rede for inicializada com um número elevado de unidades de classificação, isto implicará diretamente na perda de generalização da rede. O ideal é que, quanto menor for o número de unidades de classificação (que também está diretamente relacionada com a quantidade de padrões de treinamento utilizada), se tenha uma excelente taxa de eficiência o que determina uma ótima capacidade de generalização da rede.

Pode-se observar na Figura A.4a que o tempo de treinamento aumenta até 35 unidades de classificação e para valores maiores que este, o tempo de treinamento decresce. Inicialmente, não há muitas unidades de classificação o que reduz a quantidade de padrões utilizados para inicializar os pesos da rede. À medida que é aumentada a quantidade de unidades de classificação, aumenta-se a quantidade de padrões

utilizados no treinamento, reduzindo a lista de padrões usados na execução, o que gradativamente torna os cálculos mais complexos.

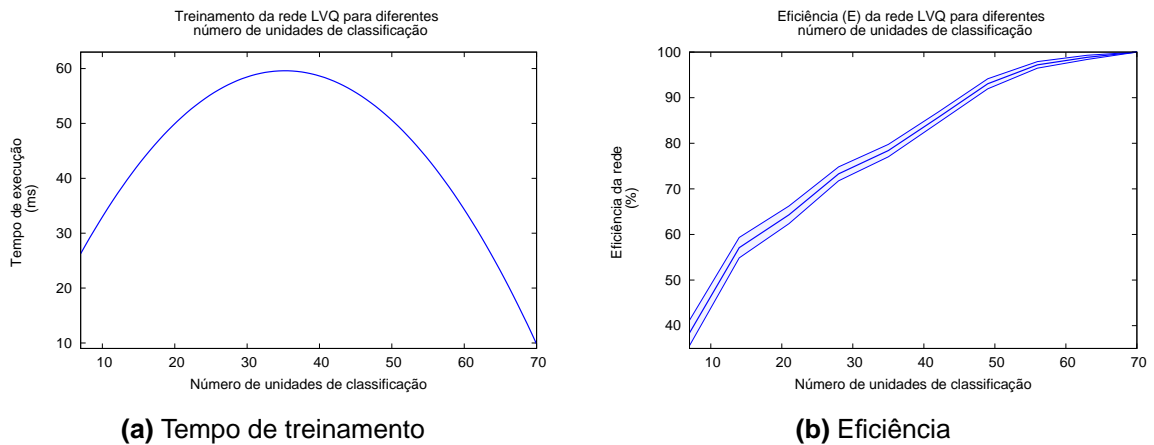


Figura A.4: Gráficos relacionados às unidades de saída

Pode-se avaliar por meio da Figura A.4a que a eficiência da rede aumenta à medida que as unidades de classificação são aumentadas. Utilizando-se 90% dos padrões de treinamento a rede alcançou 97% de eficiência.

A.1.2 Movimentos de braço - TDF (Isotônico)

Para observar qual a influência da modificação dos parâmetros de configuração da rede neural (taxa de aprendizagem (α), taxa de decaimento de (α) e tolerância), tanto na performance quanto no tempo de treinamento, foi adotada a seguinte abordagem:

1. Mantendo-se os demais parâmetros fixos nos valores padrões, o parâmetro em análise é alterado, dentro da faixa de valores estabelecidos abaixo:

- Os valores de $alpha(\alpha)$ (taxa de aprendizado) serão de 0.01 a 0.99;
- Os valores de $dec.alpha$ (decaimento de alpha) serão de 0.01 a 0.99;
- Os valores de $tolerance$ (tolerância) serão de 0.001 a 0.099;

2. Número de unidades de saída (clusters):

- Serão variados de 24 a 240, com 10 intervalos diferentes utilizando TDF;

- Serão variados de 120 a 1200, com 10 intervalos diferentes utilizando MAR, para diferentes ordens do MAR;

A.1.2.1 Taxa de aprendizagem (α)

Observa-se na Figura A.5a que, à medida que o valor da taxa de aprendizado (α) é aumentada, o tempo de treinamento da rede também aumenta. Devido à redução gradual na taxa de aprendizado (α) a cada ciclo de treinamento. Quanto maior for o valor da taxa de aprendizado (α), maior será o número de ciclos de treinamento e proporcionalmente o tempo de treinamento.

É possível verificar na Figura A.5b que, para valores de taxa de aprendizado (α) entre 0.01 e 0.42, a eficiência da rede aumenta gradualmente, com uma leve oscilação. No entanto, para valores maiores que 0.42, observa-se a redução na eficiência da rede devido aos elevados valores de taxa de aprendizado (α).

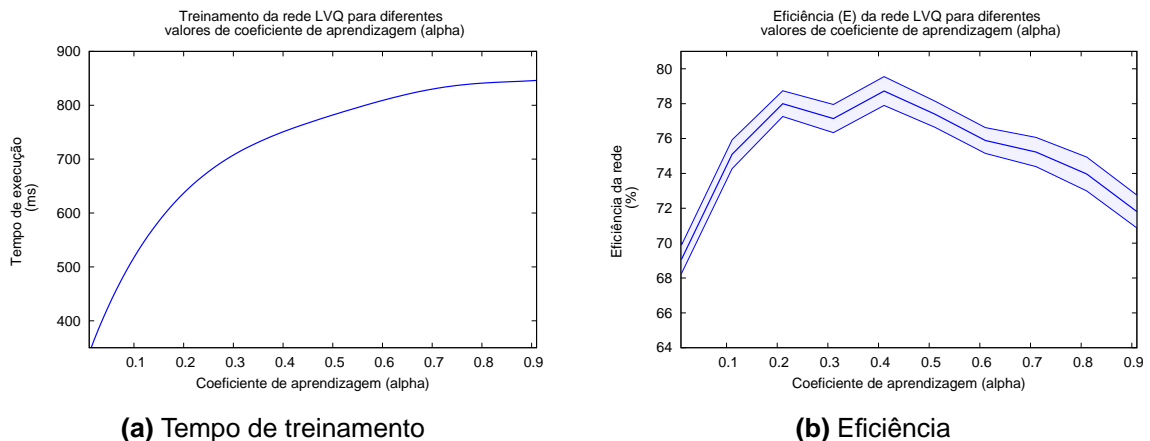


Figura A.5: Gráficos relacionados à taxa de aprendizagem (α)

A.1.2.2 Taxa de decaimento de (α)

Avaliando os resultados apresentados na Figura A.6b pode-se observar que elevados valores de decaimento de (α) geram uma redução mais lenta nos valores da taxa de aprendizado (α), o que resulta em um aumento na quantidade de ciclos de treinamento. Uma vez que, a cada ciclo de treinamento o valor da taxa de aprendizado (α) é reduzida proporcionalmente ao valor da taxa de decaimento de (α).

A Figura A.6b apresenta a variação da eficiência da rede em função da variação da taxa de decaimento de (α). Pode-se observar que a eficiência da rede aumenta gradualmente com aumento da taxa de decaimento de (α), pois, há mais ciclos de treinamento e os pesos da rede são melhores ajustados, permitindo uma melhor classificação da rede.

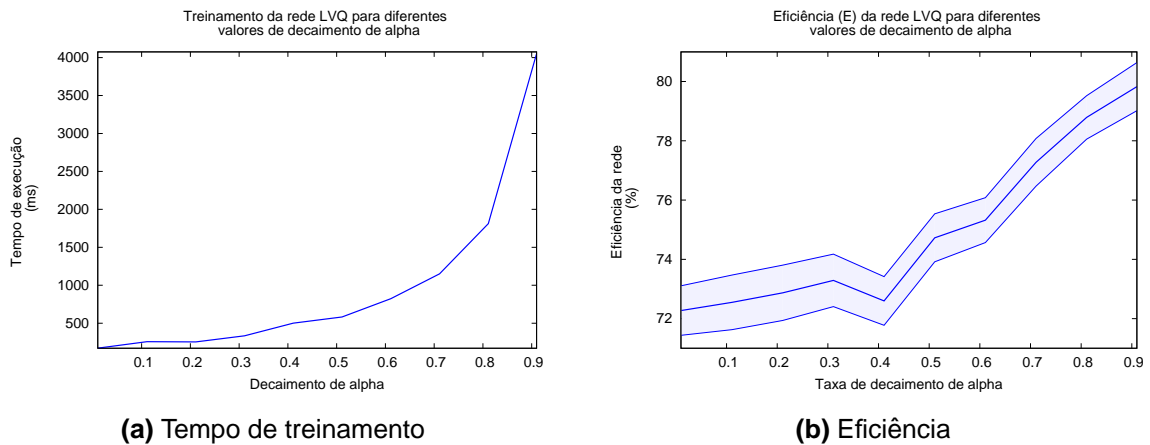


Figura A.6: Gráficos relacionados à taxa de decaimento de (α)

A.1.2.3 Tolerância

Observa-se na Figura A.7a que, o tempo de treinamento tende a ser elevado para valores pequenos de tolerância, devido ao aumento no número de ciclos de treinamento, que diretamente influencia o tempo de treinamento. À medida que este valor é aumentado, gradativamente o tempo é reduzido.

É possível verificar na Figura A.7b que, para pequenos valores de tolerância a rede apresenta uma boa eficiência, porém, à medida que este valor é aumentado, a eficiência da rede reduz gradativamente, devido à relação existente entre a taxa de aprendizado (α) e a tolerância, que determinam a condição de parada para o treinamento da rede. Quanto maior for a tolerância, menor é a quantidade de ciclos de treinamento, reduzindo o ajuste nos pesos da rede.

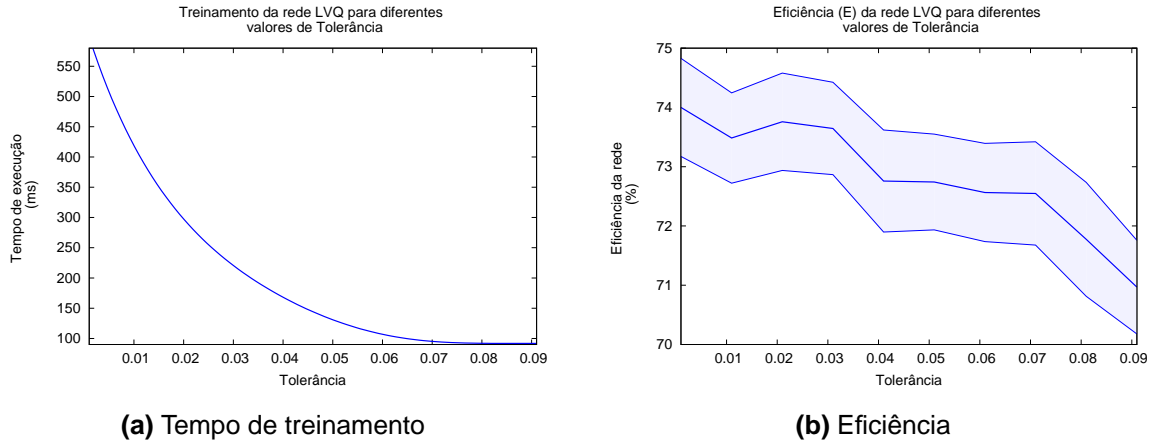


Figura A.7: Gráficos relacionados à tolerância

A.1.2.4 Unidades de saída

É possível observar na Figura A.8a que, o tempo de treinamento aumenta até 120 unidades de classificação e para valores maiores que estes o tempo de treinamento decresce.

Observa-se na Figura A.8b que a eficiência da rede aumenta à medida que as unidades de classificação são aumentadas. Utilizando-se 90% dos padrões de treinamento a rede alcançou 80% de eficiência.

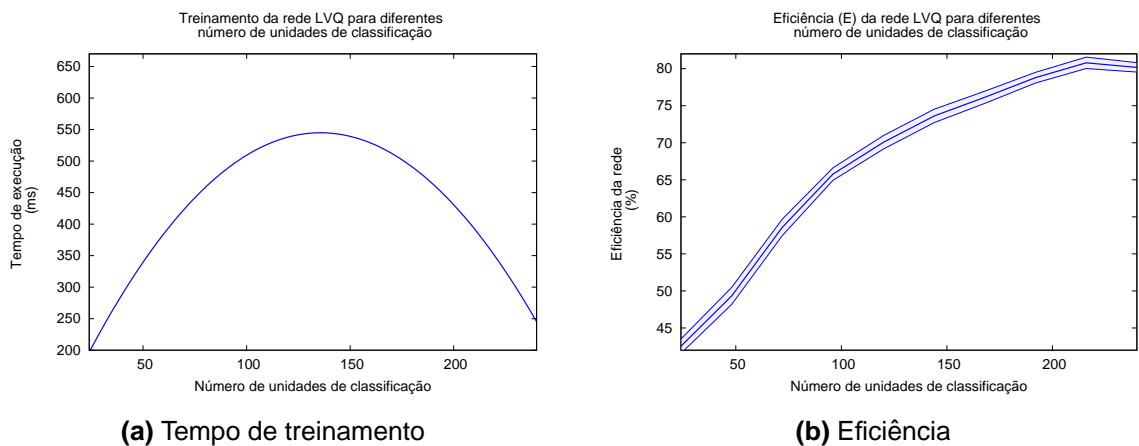


Figura A.8: Gráficos relacionados às unidades de saída

A.2 Avaliação de performance classificador MLP

A.2.1 Movimentos de mão - TDF

Para observar qual a influência da modificação dos parâmetros (taxa de aprendizagem, tolerância e número de neurônios na camada oculta) de configuração da rede neural tanto na performance quanto no tempo de treinamento foi adotado a seguinte abordagem:

1. Mantendo-se os demais parâmetros fixos nos valores padrões, o parâmetro em análise é alterado, dentro da faixa de valores estabelecidos abaixo:

- Os valores de $alpha(\alpha)$ (taxa de aprendizado) serão de 0.1 a 0.7;
- Os valores de $tolerance$ (tolerância) serão de 100 a 300;
- Os valores de neurônios na camada oculta serão de 1 a 30;

2. Número de padrões de treinamento:

- Para os padrões de movimentos de mão será 90% do total de padrões usando TDF e 2% do total de padrões usando MAR;

A.2.1.1 Taxa de aprendizagem (α)

É possível verificar na Figura A.9a que, à medida que o valor da taxa de aprendizagem (α) é aumentada, o tempo de treinamento da rede também aumenta. Como é sabido, a variação do valor da taxa de aprendizagem (α) afeta diretamente a atualização dos pesos o que levar a um aumento do erro quadrático a cada ciclo, levando a rede a divergir.

Observa-se na Figura A.9b que a rede alcançou uma eficiência de 45% para valores de taxa de aprendizagem (α) igual a 0.2.

Conclui-se que, para valores de taxa de aprendizagem (α) maiores que 0.2 a rede não convergiu, devido ao crescimento contínuo do erro quadrático.

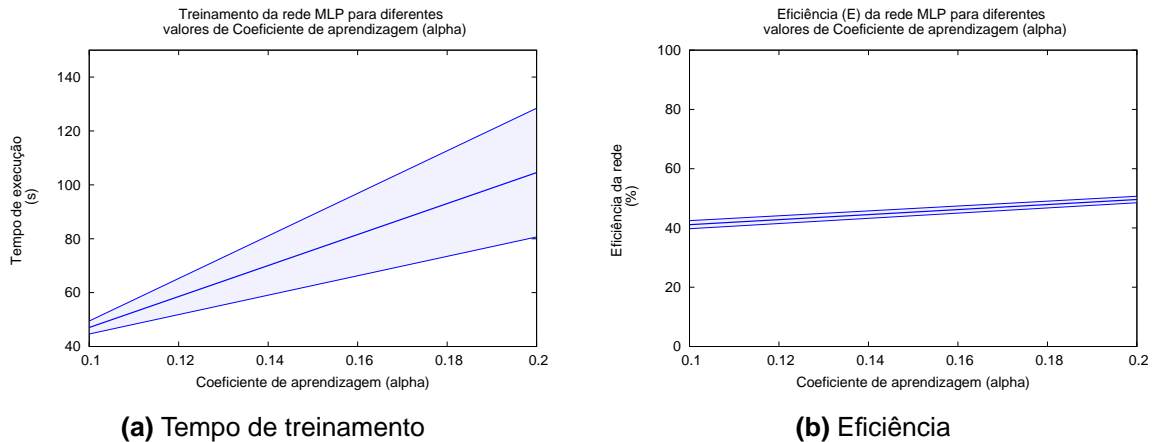


Figura A.9: Gráficos relacionados à taxa de aprendizagem

A.2.1.2 Tolerância

Pode-se observar na Figura A.10a que o tempo de treinamento tende a ser elevado para valores pequenos de tolerância, devido ao aumento no número de ciclos de treinamento, que diretamente influencia o tempo de treinamento. À medida que este valor é aumentado, gradativamente o tempo é reduzido, tendo em vista o critério de parada baseado no erro quadrático total máximo.

É possível verificar na Figura A.3b que, para pequenos valores de tolerância a rede apresenta uma boa eficiência, porém, à medida que este valor é aumentado, a eficiência da rede reduz gradativamente, pois, quando o somatório erro quadrático total máximo na saída de cada neurônio é menor que a tolerância o treinamento é interrompido, por já ter alcançado uma margem de erro aceitável o que determina a busca por uma melhor eficiência.

A.2.1.3 Número de neurônios na camada oculta

Analisando a variação no número de neurônios na camada de oculta na Figura A.11a pode-se perceber que o tempo de treinamento aumenta à medida que o número é elevado. Baseado na arquitetura da rede MLP quando maior é número de neurônios na camada oculta, maior é o número de *broadcast* de informações da camada de entrada da rede para saída, e maior é a retroalimentação dos erros entre as camadas. Isto aumenta consideravelmente o volume de cálculos a serem realizados, aumen-

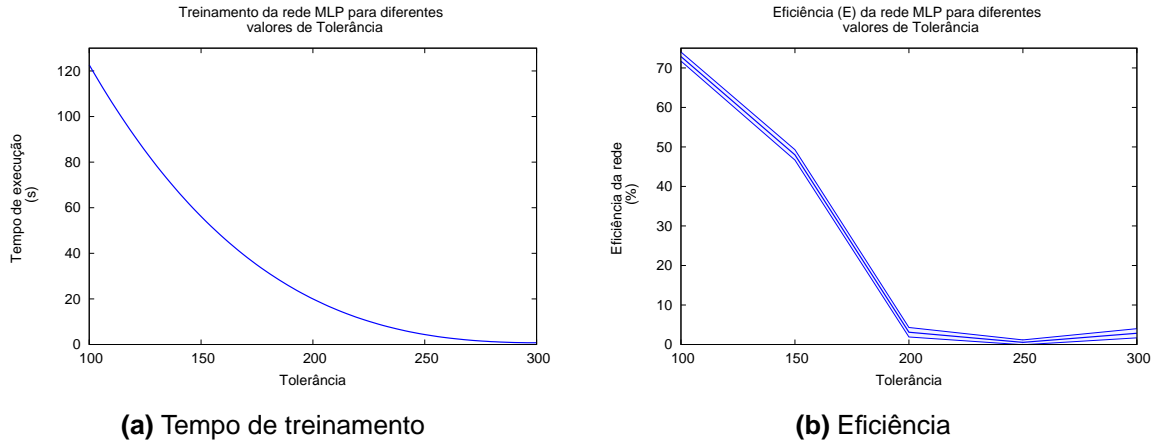


Figura A.10: Gráficos relacionados à tolerância

tando consequentemente o tempo de treinamento.

Observa-se na Figura A.11b que a partir de 3 neurônios na camada oculta é possível alcançar uma eficiência em torno de 80%. O aumento do número de neurônios na camada oculta não apresentou melhoras na classificação, o que nos leva a concluir que, as características extraídas não são suficientes para que a rede consiga separar uma classe de movimento de outro.

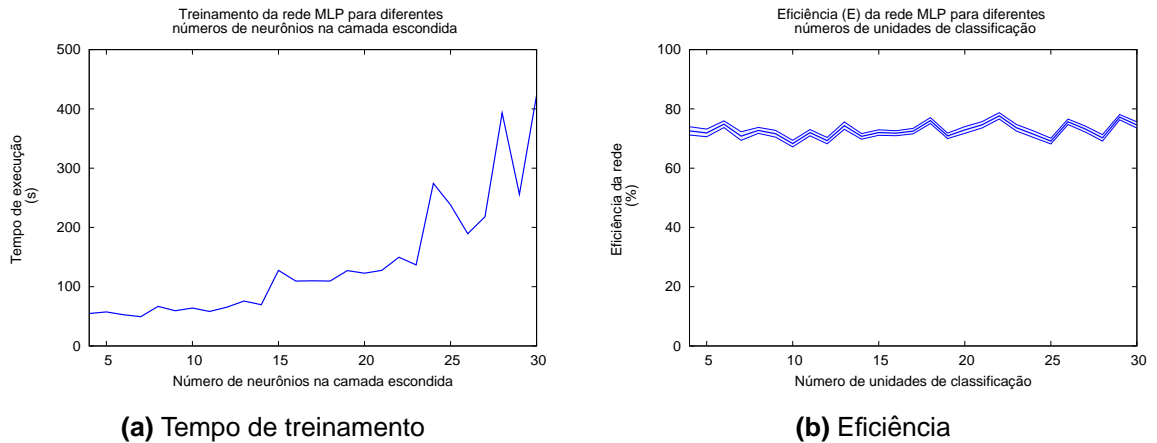


Figura A.11: Gráficos relacionados ao número de neurônios na camada oculta

A.2.2 Movimentos de mão - MAR

A.2.2.1 Taxa de aprendizagem (α)

É possível verificar na Figura A.12a que, para todas as ordens do MAR forma da curva de tempo de treinamento foi diferente para valores de taxa de aprendizado (α) até 0.5. Para um MAR de 3^a ordem o tempo de treinamento foi menor que um segundo para todos os valores de taxa de aprendizado (α).

Pode-se observar na Figura A.12b que, para cada ordem de MAR a curva de eficiência foi diferente e em nenhum momento superou a taxa de eficiência de 30%, porém a rede demonstrou mais eficiência para um MAR de 6^a.

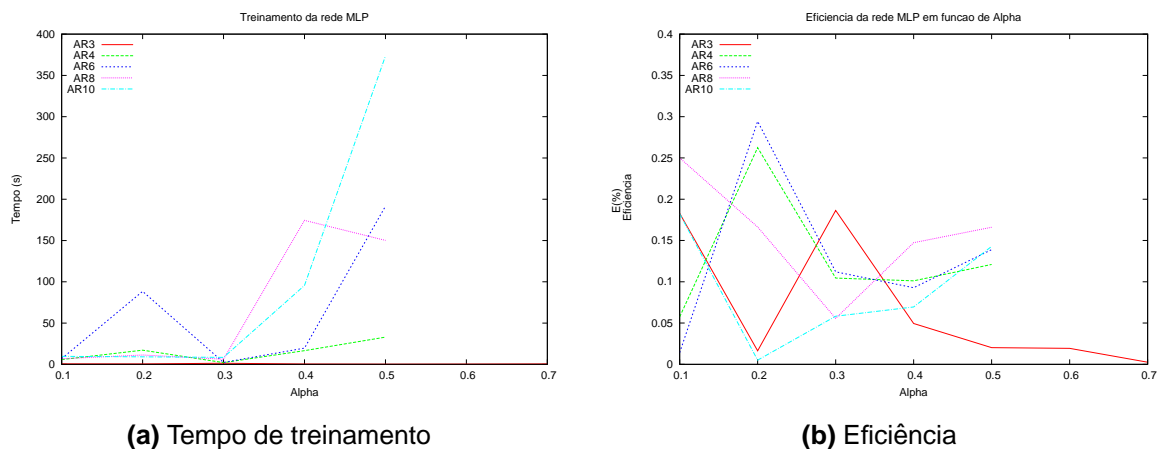


Figura A.12: Gráficos relacionados à taxa de aprendizagem

Com exceção do MAR de 3^a ordem a rede não convergiu por ter exercido o número máximo de ciclos de treinamento igual a 10000 entrando em uma das condições de parada pré-estabelecidas anteriormente.

A.2.2.2 Tolerância

Observa-se na Figura A.13a que o tempo de treinamento reduz gradativamente até um valor de tolerância igual a 200 para todas as ordens de MAR com exceção do MAR de 4^a. Após este valor o tempo de treinamento volta a aumentar, por consumir um número considerável de ciclos de treinamento para ajustar os pesos da rede para todas as ordens de MAR, de forma que o erro quadrático reduza ao invés de aumentar

após sucessivos ciclos, levando a rede a convergir.

É possível verificar na Figura A.13b que a eficiência foi crescente para os MAR de 4ª e 6ª ordem, à medida que o valor de tolerância crescia. Para as demais ordem de MAR a eficiência oscilou.

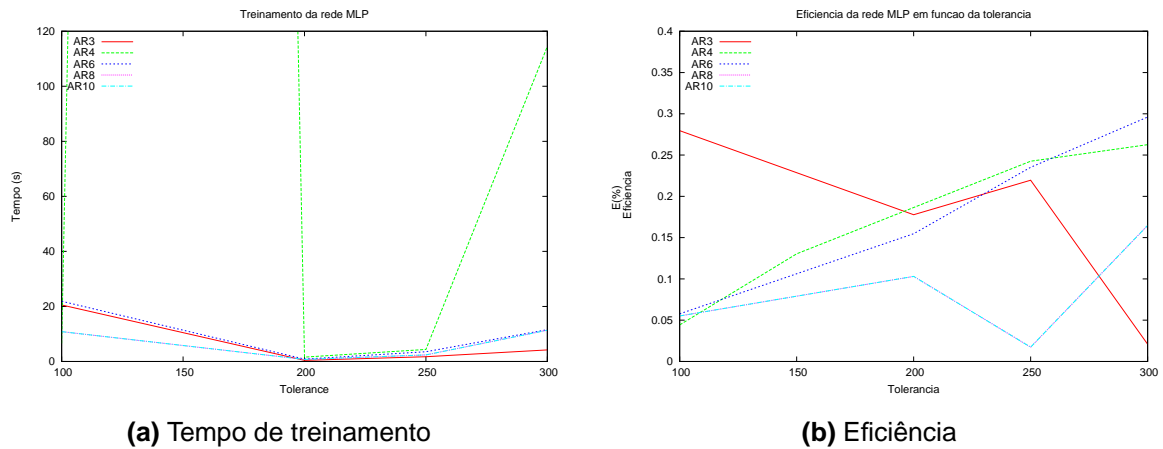


Figura A.13: Gráficos relacionados à tolerância

A.2.2.3 Número de neurônios na camada oculta

Analisando os valores apresentados na Figura A.14a pode-se observar que à medida que se aumenta a ordem do MAR o tempo de treinamento aumenta proporcionalmente. Com até 18 neurônios na camada oculta o tempo de treinamento não excedeu a 1 segundo.

A eficiência da rede não foi satisfatória para nenhuma das diferentes ordens do MAR, não ultrapassando a 27%, conforme exibido na Figura A.14b.

A.2.3 Movimentos de braço - TDF

Para observar qual a influência da modificação dos parâmetros (taxa de aprendizagem, tolerância e número de neurônios na camada oculta) de configuração da rede neural tanto na performance quanto no tempo de treinamento foi adotado a seguinte abordagem:

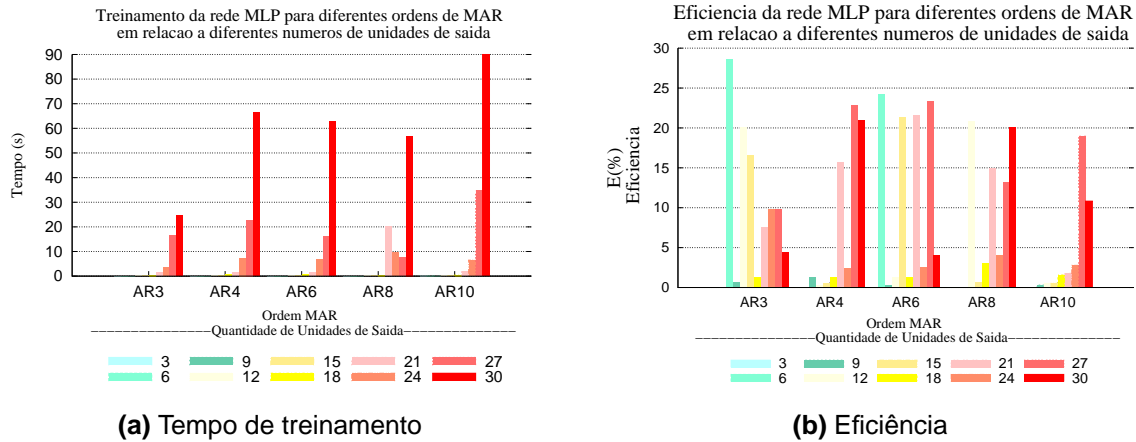


Figura A.14: Gráficos relacionados ao número de neurônios na camada oculta

1. Mantendo-se os demais parâmetros fixos nos valores padrões, o parâmetro em análise é alterado, dentro da faixa de valores estabelecidos abaixo:

- Os valores de α (taxa de aprendizado) serão de 0.1 a 0.7;
- Os valores de ϵ (tolerância) serão de 100 a 300;
- Os valores de neurônios na camada oculta serão de 1 a 30;

2. Número de padrões de treinamento:

- Para os padrões de movimentos de braço será 90% do total de padrões usando TDF e 2% do total de padrões, usando MAR;

A.2.3.1 Taxa de aprendizagem (α)

Pode-se observar na Figura A.15a que para valores de taxa de aprendizagem (α) maiores que 0.1, não há convergência da rede.

Isto ocorre por exceder o número máximo de ciclos de treinamento, pré-definidos. Mas para o valor de taxa de aprendizagem (α) igual a 0.1 o tempo de treinamento da rede 3240 segundos. Apesar do tempo de treinamento ser elevado, a eficiência alcançada foi de 87%, conforme visto na Figura A.15b, para uma taxa de aprendizado igual a 0.1.

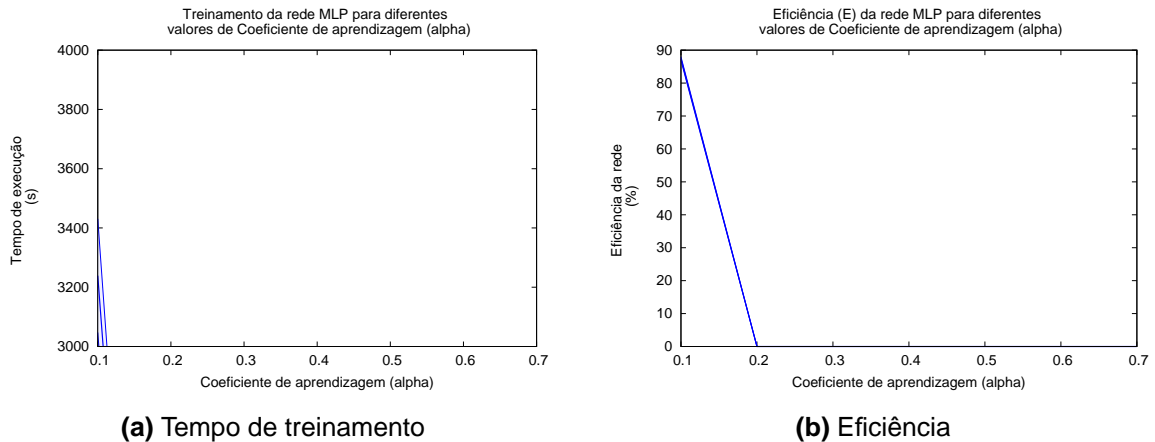


Figura A.15: Gráficos relacionados à taxa de aprendizagem

A.2.3.2 Tolerância

É possível verificar na Figura A.16a que o tempo de treinamento tende a ser elevado para valores pequenos de tolerância, devido ao aumento no número de ciclos de treinamento, que diretamente influencia o tempo de treinamento. À medida que este valor é aumentado, gradativamente o tempo é reduzido, tendo em vista o critério de parada baseado no erro quadrático total máximo.

Analisando os valores de tolerância apresentados na Figura A.16b pode-se observar que para pequenos valores a rede apresenta uma boa eficiência, porém, à medida que este valor é aumentado, a eficiência da rede reduz gradativamente, pois, quando o somatório erro quadrático total máximo na saída de cada neurônio é menor que a tolerância o treinamento é interrompido, por já ter alcançado uma margem de erro aceitável o que determina a busca por uma melhor eficiência.

A.2.3.3 Número de neurônios na camada oculta

Analisando os valores de neurônios na camada oculta na Figura A.17a pode-se perceber que o tempo de treinamento aumenta à medida que o número é elevado. Baseado na arquitetura da rede MLP quando maior é o número de neurônios na camada oculta, maior é o número de *broadcast* de informações da camada de entrada da rede para saída, e maior é a retroalimentação dos erros entre as camadas. Isto aumenta consideravelmente o volume de cálculos a serem realizados, aumentando

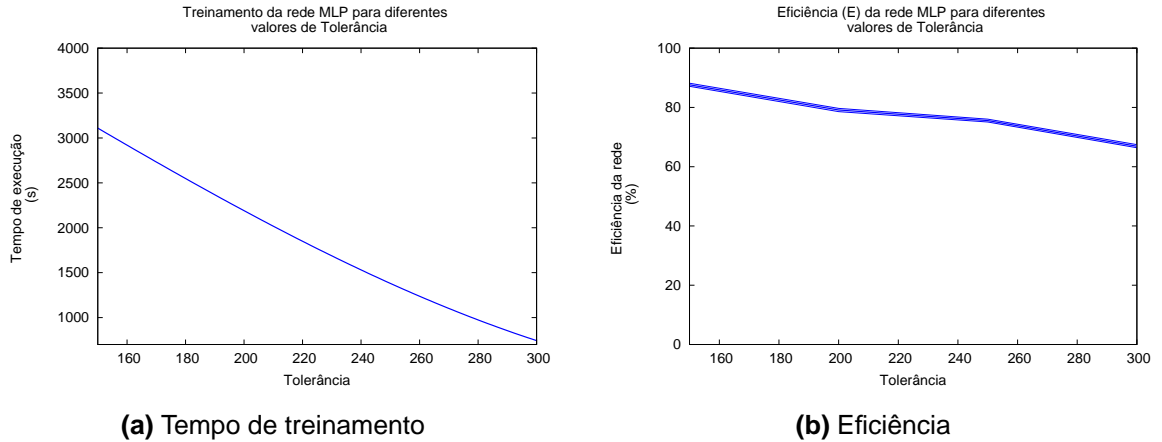


Figura A.16: Gráficos relacionados à tolerância

consequentemente o tempo de treinamento.

É possível verificar na Figura A.17b que a partir de 3 neurônios na camada oculta é possível alcançar uma eficiência em torno de 63%. O aumento do número de neurônios na camada oculta não apresentou melhoras na classificação, o que nos leva a concluir que, as características extraídas não são suficientes para que a rede consiga separar uma classe de movimento de outro.

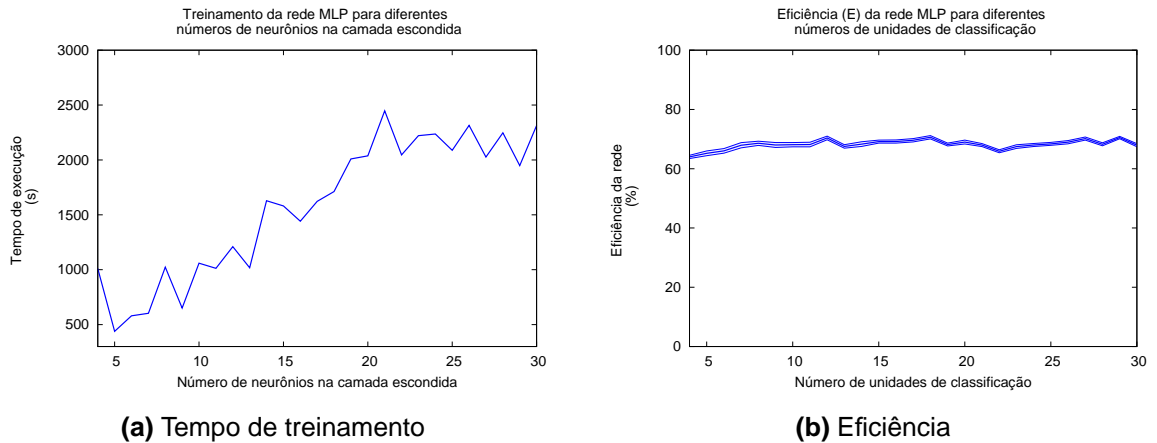


Figura A.17: Gráficos relacionados ao número de neurônios na camada oculta

A.2.4 Movimentos de braço - MAR

A.2.4.1 Taxa de aprendizagem (α)

Pode-se observar na Figura A.18a que, para todas as ordens do MAR a curva de tempo de treinamento elevou-se para valores de taxa de aprendizado (α) a partir 0.5. Apenas para um MAR de 10^a ordem é que o tempo de treinamento elevou anterior a este valor. Mas conforme dito anteriormente em outros testes realizados, à medida que a taxa de aprendizado (α) é aumentada o tempo de treinamento aumenta gradativamente.

É possível verificar na Figura A.18b que para cada ordem de MAR, a curva de eficiência foi diferente e em nenhum momento superou a taxa de eficiência de 80%. Uma outra observação em relação a taxa de aprendizado (α) é que a eficiência da rede não oscilou tanto, mostrando que, para qualquer um destes valores a rede apresentará uma boa eficiência.

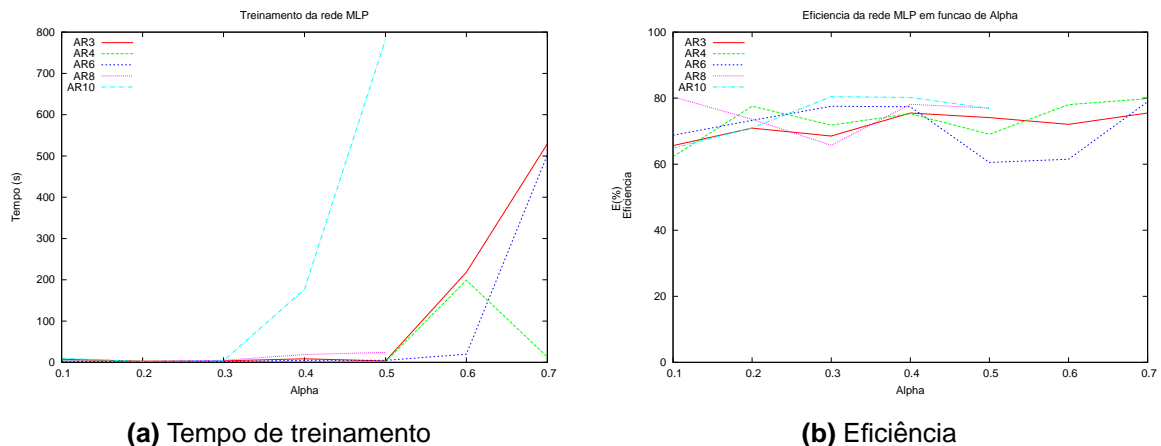


Figura A.18: Gráficos relacionados à taxa de aprendizagem

A.2.4.2 Tolerância

Observa-se na Figura A.19a que o tempo de treinamento aumenta gradativamente para todas as ordens de MAR à medida que a taxa de tolerância é incrementada. Diferentemente dos testes anteriores, o que nos leva a concluir que a rede necessita de mais ciclos de treinamento para fazer com que o erro reduza com elevados valores

de tolerância.

É possível verificar na Figura A.19a que a rede apresenta maior taxa de eficiência para maiores valores de taxa de tolerância, conforme demonstra a Figura A.19b. Isto explica-se facilmente por não ser necessário uma margem de erro muito pequena para alcançar-se uma boa eficiência.

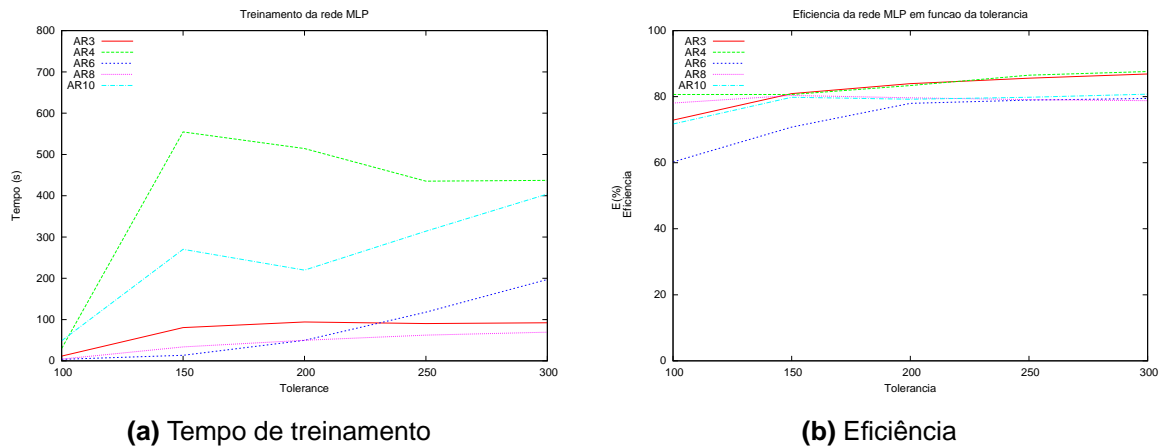


Figura A.19: Gráficos relacionados à tolerância

A.2.4.3 Número de neurônios na camada oculta

É possível verificar na Figura A.20a que o tempo de treinamento não aumentou linearmente à medida que se aumentou a ordem do MAR, pois, como pode ser observado o tempo de treinamento foi maior para um MAR de 4^a ordem, mas o tempo de treinamento aumentou gradativamente à medida que se aumentou o número de neurônios na camada oculta. Importante observar que a rede não convergiu para os MAR de 6^a e 10^a ordem, por sempre exceder o limite máximo de ciclos de treinamento.

Observa-se na Figura A.20b que apesar da rede ter apresentado uma taxa de eficiência média igual a 80%, para diferentes números de neurônios na camada oculta, isto ainda não é satisfatório para um controle de prótese, por deixar uma margem de erro considerável, expondo o paciente a risco.

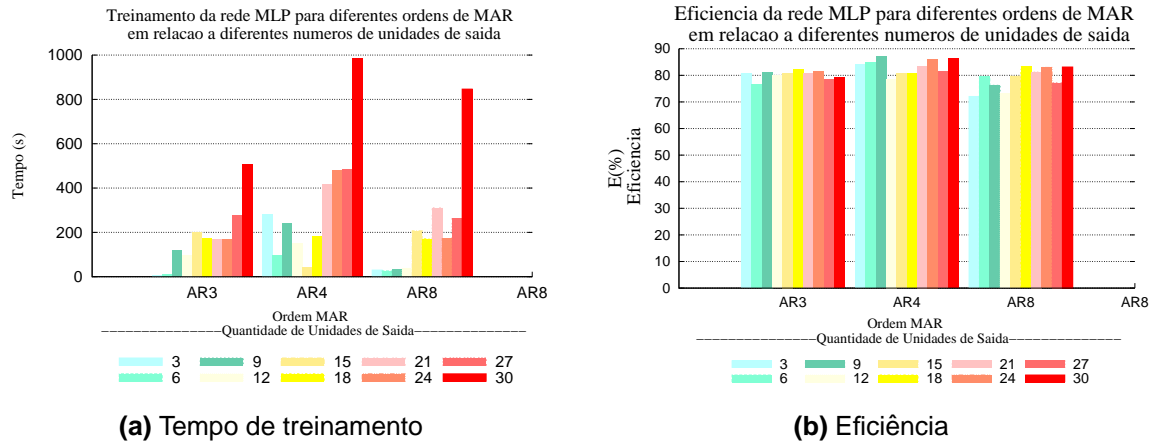


Figura A.20: Gráficos relacionados ao número de neurônios na camada oculta