

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
BIBLIOTECA UNIVERSITÁRIA**

Heron Pereira

**PRÓTESE MIOELÉTRICA PARA MEMBRO SUPERIOR
IMPLEMENTADA EM FPGA**

Araranguá

2016

Heron Pereira

**PRÓTESE MIOELÉTRICA PARA MEMBRO SUPERIOR
IMPLEMENTADA EM FPGA**

Trabalho de Conclusão de curso submetido ao curso de Engenharia de Computação para a obtenção do Grau de Bacharel em Engenharia de Computação.

Orientador: Prof. Anderson Luiz Fernandes Perez, Dr.

Coorientador: Prof. Rafael Inácio Barbosa, Dr.

Araranguá

2016

Heron Pereira

**PRÓTESE MIOELÉTRICA PARA MEMBRO SUPERIOR
IMPLEMENTADA EM FPGA**

Este Trabalho de Conclusão de curso foi julgado aprovado para a obtenção do Título de "Bacharel em Engenharia de Computação", e aprovado em sua forma final pelo curso de Engenharia de Computação

Araranguá, 09 de dezembro 2016.



Coordenador
Prof. Anderson Luiz Fernandes Perez, Dr.

Banca Examinadora:



Prof. Anderson Luiz Fernandes Perez, Dr.
Orientador



Prof. Rafael Ivácio Barbosa, Dr.
Coorientador



Prof. Fábio Rodrigues De La Rocha, Dr.
Universidade Federal de Santa Catarina



Prof. Alexandre Marcio Marcolino, Dr.
Universidade Federal de Santa Catarina

Dedico este trabalho a minha família, os
quais foram fundamentais em todas eta-
pas de minha vida.

AGRADECIMENTOS

Agradeço a todos meus familiares e amigos, os quais sempre depositaram sua fé em mim e que me concederam um suporte imenso que me permitiu superar momentos de fraqueza e de dúvidas. A Universidade Federal de Santa Catarina onde obtive oportunidades únicas de experiências acadêmicas e profissionais. A todos meus colegas do Laboratório de Automação e Robótica Móvel (LARM), por me darem suporte e me fornecerem toda a ajuda disponível a seu alcance. Agradeço a minha amiga e colega Pamela Brunh Cardoso por todo auxílio que me forneceu neste processo e pelo software desenvolvido para a coleta de dados para o conjunto de treinamento. A minha colega e amiga Vanessa Pereira Corrêa por me apresentar a área da saúde, me indicar livros e me ajudar a arquitetar projetos unindo saúde e computação. A toda equipe do Laboratório de Avaliação e Reabilitação do Aparelho Locomotor (LARAL) que disponibilizou seu tempo e colaborou com materiais e na coleta dos dados. Ao professor Dr. Fábio Rodrigues de La Rocha, o qual me disponibilizou muito de seu tempo, conhecimento e esforço, para o desenvolvimento do projeto. Ao meu coorientador professor Dr. Rafael Inácio Barbosa, que me forneceu o conhecimento, tirou dúvidas, estabeleceu critérios e atuou em todas as etapas referentes a saúde. Ao professor Dr. Anderson Luiz Fernandes Perez meu orientador, por todo seu conhecimento na elaboração do trabalho, todo auxílio e ensino relacionado a Redes Neurais Artificiais, e principalmente por sempre me fazer acreditar de que sou capaz de alcançar qualquer objetivo que eu tenha. A todos meus sinceros agradecimentos, não possuo palavras o suficiente para expressar minha gratidão.

RESUMO

As próteses são fundamentais para pessoas que perderam algum membro do corpo, pois elas permitem que os amputados tenham condições de fazer várias atividades cotidianas, ou mesmo para melhor a auto estima. Existem vários tipos de próteses disponíveis, entretanto a mais promissora é a prótese mioelétrica. Uma prótese mioelétrica é um dispositivo mecatrônico, ou seja, composta de partes mecânicas controladas eletronicamente, que é acionado pela leitura do sinal produzido por um ou mais músculos do usuário da prótese. A mão é um órgão fundamental para todas as pessoas, desta forma as próteses de mão devem ser o mais funcional possível. Neste trabalho é proposta uma arquitetura de hardware e software para uma prótese mioelétrica que seja capaz de reproduzir o movimento de preensão palmar, de forma proporcional a ativação do bíceps.

Palavras-chave: Redes Neurais Artificiais. Prótese Mioelétrica. Field Programmable Gate Array. Sistemas Embarcados.

ABSTRACT

The prothesis are fundamental to people who lost some body member, because them allow the amputees have conditions to do daily activities, or even to improve self-esteem. There are a lot of types of prothesis available, but the most promising one is the mioelectric prothesis. The mioelectric prothesis is a mechatronic device, made of mechanical parts electronically controlled, activated by muscle signal read from the prothesis' user. The hand is a fundamental organ for every person, so the hand's prothesis need to be the most functional as possible. This work proporses an architecture of harware and software for a mioelectric prothesis capable of reproduce an palmar grip, proporcional to the byceps activation

Keywords: Artificial Neural Network. Myoelectric Prosthesis. Field Programmable Gate Array. Embedded Systems.

LISTA DE FIGURAS

Figura 1	Exemplo de prótese funcional usada pelos veteranos de guerra.	29
Figura 2	Exemplo de gancho e mão.	31
Figura 3	Exemplo de prótese estética.	31
Figura 4	Exemplo de prótese mecânica.	32
Figura 5	Exemplo de prótese mioelétrica.	32
Figura 6	Exemplo de prótese híbrida.	33
Figura 7	Mão em seu estado funcional, também conhecido como estado de repouso.	34
Figura 8	Exemplo de uma mão operando com uma pinça de precisão.	35
Figura 9	Exemplo de uma mão operando com uma pinça de força ou em preensão palmar.	35
Figura 10	Pinça de ponta, permite a mão manipular objetos pequenos por possuir muita precisão.	36
Figura 11	A pinça lateral também é conhecida como pinça de chave é própria para o manuseio de chaves ou objetos semelhantes.	36
Figura 12	Pinça de palmar permite o uso de força maior sobre o objeto e também possui a maior área de contato entre as três.	36
Figura 13	Os movimentos de flexão e hiperextensão são apresentados na imagem, e o movimento de extensão onde a mão é direcionada a posição central.	37
Figura 14	Os desvios radial e ulnar do punho permitem que a mão se desloque lateralmente.	38
Figura 15	A estrutura óssea da mão é formada por arcos para facilitar o envolvimento do objeto ao tentar pegá-lo.	38
Figura 16	Um segundo facilitador para a captura de objetos é a convergência dos dedos ao mesmo ponto.	39
Figura 17	Estrutura básica de um neurônio biológico.	44
Figura 18	Modelo de um neurônio artificial.	46
Figura 19	Conexão entre um neurônio i e k.	47
Figura 20	Exemplo de uma função degrau.	48
Figura 21	Exemplo de uma função rampa.	49
Figura 22	Exemplo de uma função sigmoide com valor de $a = 1$	50

Figura 23	Exemplo de uma função sigmoide com valor de $a = 2$.	50
Figura 24	Exemplo de uma função sigmoide com valor de $a = 3$.	51
Figura 25	Exemplo de um Perceptron com camada única.	53
Figura 26	Exemplo geral de uma rede neural MLP.....	54
Figura 27	Exemplo de Rede Recorrente.....	55
Figura 28	Sentido de propagação feedforward e feedback	58
Figura 29	Exemplo de um sistema de malha aberta	61
Figura 30	Ilustração de um dos exercícios realizados.....	63
Figura 31	Posicionamento dos eletrodos para a captura dos dados.	64
Figura 32	Posicionamento dos eletrodos para a captura dos dados.	65
Figura 33	Dispositivo Arduino Uno, utilizado para conversão Analógico-Digital.	66
Figura 34	Interface principal do sistema de captura dos dados. ...	66
Figura 35	Dados contidos em um arquivo de log no formato texto.	67
Figura 36	Modelo da RNA proposta	68
Figura 37	Placa de desenvolvimento DE0.....	69
Figura 38	Exemplo da estrutura de uma FPGA	70
Figura 39	Esquemático do projeto da prótese mioelétrica.	71
Figura 40	Servo motor utilizado para atuar no sistema.	71
Figura 41	Brinquedo utilizado para prótese.....	72
Figura 42	Fonte de tensão utilizada.	72
Figura 43	Suporte para a base.....	73
Figura 44	Protótipo da prótese.....	73
Figura 45	Placa desenvolvida no modelo Ilhas.	74
Figura 46	Conjunto de hardware completo do sistema.	75
Figura 47	Conjunto de hardware completo do sistema.	76
Figura 48	Experimento 1 - Prótese movendo frasco de desodorante vazio.	78
Figura 49	Experimento 2 - Prótese movendo copo plástico vazio. .	78
Figura 50	Experimento 3 - Prótese escrevendo.....	79
Figura 51	Experimento 4 - Tentando segurar uma bola de borracha.	80

LISTA DE ABREVIATURAS E SIGLAS

RNA	Redes Neurais Artificiais	23
FPGA	Field Programmable Gate Array	23
EMG	Eletromiografia	40
EMG	Sinal Mioelétrico	40
PID	Proporcional-Integral-Derivativo	52
MLP	Multilayer Perceptron	53
RM	Resistência Máxima	62
VHDL	Very High Speed Integrated Circuits Hardware Description Language	69

LISTA DE SÍMBOLOS

x_m	Entradas da rede	46
w_{km}	Pesos sinápticos	46
b_k	Valor de bias	46
W_{ik}	Peso sináptico da conexão IK	46
Net_k	Potencial de ativação do neurônio	47
\sum	Somatório	47
x_i	Entradas i da rede	47
y_k	Solução da equação de ativação	47
φ	Função de ativação escolhida	47
v	Soma de Net_k e b_k	48
e	Número neperiano	49
a	Inclinação da função sigmoide	49
$t + 1$	Tempo posterior da interação	56
$w(t)$	Peso sináptico atual	56
δ	Ajuste multiplicado ao peso	56
$e(t)$	Erro obtido da interação	56
$d(t)$	Saída desejada	56
$y(t)$	Saída obtida	56
n	Taxa de aprendizado	56
$X(t)$	Valor de entrada obtido no tempo t	56
ε^2	Valor do erro acumulado por épocas	57
p	Número de épocas ocorridas	57
y_d^i	Saída desejada	57
y	Saída real	57

LISTA DE EQUAÇÕES

3.1	Equação do potencial de ativação.	47
3.2	Equação de ativação.	47
3.3	Equação da função degrau.	48
3.4	Equação da função rampa.	48
3.5	Equação da função sigmoide.	49
3.6	Equação de atualização dos pesos sinápticos.	56
3.7	Equação do erro.	56
3.8	Equação da propagação do erro.	56
3.9	Equação do erro acumulado.	57

SUMÁRIO

1	INTRODUÇÃO	23
1.1	MOTIVAÇÃO E JUSTIFICATIVA	24
1.2	OBJETIVOS	24
1.2.1	Objetivo Geral	24
1.2.2	Objetivos Específicos	24
1.3	METODOLOGIA	25
1.4	ORGANIZAÇÃO DO TRABALHO	25
2	PRÓTESES	27
2.1	AMPUTAÇÕES	27
2.2	PROTETIZAÇÃO	28
2.3	TIPOS DE PRÓTESE	30
2.4	BIOMECÂNICA	33
2.5	A PRÓTESE MIOELÉTRICA	40
3	REDES NEURAIS ARTIFICIAIS	43
3.1	REDES NEURAIS BIOLÓGICAS	43
3.2	FUNDAMENTOS DE REDES NEURAIS ARTIFICIAIS .	45
3.2.1	Função de Ativação	47
3.3	VANTAGENS DE UMA RNA	51
3.4	TIPOS DE RNA	52
3.4.1	Redes de Camada Única	52
3.4.2	Redes de Múltiplas Camadas	53
3.4.2.1	Multi-Layer-Perceptron	54
3.4.3	Redes Recorrentes	55
3.5	ALGORÍTIMOS DE TREINAMENTO	55
3.5.1	Algoritmo Backpropagation	57
3.6	APLICAÇÕES	58
4	DESCRIÇÃO DA PRÓTESE MIOELÉTRICA DE- SENVOLVIDA EM FPGA	61
4.1	DESCRIÇÃO GERAL DO PROBLEMA PROPOSTO ...	61
4.2	CRIAÇÃO DO CONJUNTO DE TREINAMENTO DA RNA	62
4.3	ARQUITETURA DA RNA IMPLEMENTADA	68
4.4	PROJETO E DESENVOLVIMENTO DO HARDWARE .	69
5	AVALIAÇÃO DA PRÓTESE MIOELÉTRICA ...	77
5.1	METODOLOGIA EMPREGADA NA AVALIAÇÃO DA PRÓTESE DESENVOLVIDA	77

5.2	EXPERIMENTO 1 - MOVER UM FRASCO DE DESODORANTE VAZIO.....	77
5.3	EXPERIMENTO 2 - MOVER UM COPO PLÁSTICO VAZIO.....	78
5.4	EXPERIMENTO 3 - PRÓTESE UTILIZANDO UM LÁPIS	79
5.5	EXPERIMENTO 4 - CAPTURA DE UMA BOLA DE BORRACHA	79
5.6	CONSIDERAÇÕES SOBRE OS EXPERIMENTOS	80
6	CONSIDERAÇÕES FINAIS.....	81
6.1	PROPOSTA PARA TRABALHOS FUTUROS.....	81
7	APÊNDICE I	83
7.1	CONJUNTO DE TREINAMENTO	84
7.2	PARÂMETROS DE CONFIGURAÇÃO DA RNA	85
8	APÊNDICE II.....	87
8.1	BASE_RNA.VHD	87
8.2	RNA_PROTESE.VHD	96
	REFERÊNCIAS	99

1 INTRODUÇÃO

O uso de próteses é um método conhecido de substituir um membro perdido. Essa substituição pode ter diferentes objetivos, podendo ser uma atividade específica, multifunções ou simplesmente uma questão estética. Todos esses fatores são importantes para a reabilitação de uma pessoa amputada.

Segundo Marcolino et al. (2015) o processo de protetização, que se trata de substituir um membro amputado por uma prótese, tem como objetivo oferecer ao paciente a máxima independência funcional. Para isso, diversos estudos na área da biomecânica e engenharia biomédica são desenvolvidos buscando aprimorar a qualidade funcional das próteses. Em especial, as próteses de membro superior, como mão e antebraço, possuem uma enorme gama para pesquisas de novas tecnologias, devido a complexidade e importância de tal membro.

A prótese mioelétrica entre as outras se destaca por sua captação de sinais elétricos vindos dos músculos do amputado, que são gerados através de sua contração muscular na intenção de realizar algum movimento, são processados em microcontroladores ou microprocessadores que replicam o movimento desejado na prótese através de atuadores. A quantidade e qualidade dos movimentos gerados pela prótese devem-se ao equipamento utilizado e ao sistema de controle da prótese. Entre diversos estudos realizados de determinação para um sistema de controle, as Redes Neurais Artificiais (RNAs) se destacam por sua capacidade de processamento paralelo e de generalização. Uma vez que os sinais captados na superfície da pele variam de indivíduo para indivíduo, apresentam demasiada quantidade de ruídos e podem apresentar resultados não previstos, as características da RNA tornam-se mais que adequados para a solução do problema.

A característica do processamento paralelo de uma *Field Programmable Gate Array* (FPGA) permite que a mesma seja uma alternativa viável para aplicações que envolvam o uso de RNAs. A FPGA permite efetuar a programação de vários módulos independentes que funcionam de forma paralela, permitindo dessa forma explorar os neurônios artificiais da rede com maior potencial de processamento. O uso da FPGA também permite o desenvolvimento de um microcontrolador especializado para o controle da prótese, que o torna potencialmente mais eficiente que um microcontrolador genérico.

1.1 MOTIVAÇÃO E JUSTIFICATIVA

O projeto de uma prótese mioelétrica de membro superior trata-se de um tema atual, porém com uma ampla margem para novos projetos e avaliações, devido a grande gama de movimentos da mão não replicáveis, além da dificuldade de uma replicação eficiente do movimento conforme a intenção de um amputado. Milhares de amputados passam dificuldade durante o período de reabilitação e adaptação para o uso das próteses, em virtude da falta de replicação dos movimentos reais de uma mão biológica.

O uso de RNAs juntamente com uma FPGA para o controle da prótese é promissor, devido as suas características de processamento paralelo já mencionadas anteriormente, que permitem levar a um outro nível o desempenho de rede, além das próprias características da RNA que possibilitam tratar o sinal mioelétrico de forma mais precisa.

1.2 OBJETIVOS

Nesta seção são descritos o objetivo geral e os objetivos específicos deste Trabalho de Conclusão de Curso.

1.2.1 Objetivo Geral

Desenvolver uma prótese mioelétrica implementada em um hardware do tipo FPGA com controle baseado em Redes Neurais Artificiais.

1.2.2 Objetivos Específicos

1. Fazer um levantamento do estado da arte sobre próteses mioelétricas de mão e antebraço;
2. Estudar as principais características das RNAs;
3. Propor uma prótese mioelétrica com processamento do sinal baseado em RNA embarcada em um hardware do tipo FPGA;
4. Desenvolver uma prótese mioelétrica proposta no item (3);
5. Avaliar a prótese desenvolvida no item (4).

1.3 METODOLOGIA

Este trabalho consiste de uma pesquisa aplicada na área de próteses mioelétricas. Para tanto, a metodologia empregada visou primeiramente fazer um revisão sistemática da literatura para então propor o modelo de prótese mioelétrica que foi desenvolvida neste trabalho.

1.4 ORGANIZAÇÃO DO TRABALHO

Além desta introdução este trabalho está organizado em mais 6 capítulos e dois apêndices.

O Capítulo 2 descreverá brevemente um histórico sobre o processo de amputação e a evolução no desenvolvimento de próteses. Em virtude do objetivo do trabalho este capítulo enfatizará as próteses de mão e antebraço.

O Capítulo 3 apresentará os principais conceitos de Redes Neurais Artificiais e suas principais aplicações. Para um correto entendimento do capítulo também será descrito brevemente o funcionamento de um neurônio biológico.

O Capítulo 4 irá descrever sobre o projeto de implementação da prótese mioelétrica, sobre como irá operar o protótipo, a arquitetura da RNA e o seu uso em FPGA.

O Capítulo 5 apresenta os experimentos realizados para validar o sistema, e as considerações sobre seus resultados.

O Capítulo 6 descreve as considerações finais sobre o trabalho desenvolvido e apresenta algumas propostas para trabalhos futuros.

O Apêndice I apresenta uma descrição sobre os códigos e processos que foram desenvolvidos.

O Apêndice II apresenta os dados necessários dos experimentos, conjuntos de treinamento, dados coletados e alguns dos códigos desenvolvidos.

2 PRÓTESES

Este capítulo descreve brevemente um histórico sobre o processo de amputação e a evolução no desenvolvimento de próteses. Em virtude do objetivo do trabalho este capítulo enfatiza as próteses de mão e antebraço. É apresentado também sobre a prótese mioelétrica, a qual é o foco do trabalho.

2.1 AMPUTAÇÕES

A amputação é uma ressecção cirúrgica realizada em uma extremidade do corpo, podendo ser desde uma desarticulação, onde apenas a articulação é afetada, ou de segmento, onde até o osso é afetado (MARCOLINO et al., 2015). O segmento restante o qual se torna a nova extremidade é denominado coto.

O processo de amputação evoluiu muito desde o seu primeiro registro, a mais de 30000 anos, encontrado em pinturas de cavernas na França e na Espanha (MARCOLINO et al., 2015). Em épocas como essa e mesmo futuramente não existiam os conhecimentos necessários para uma amputação de forma própria. Porém, devido a necessidade, o processo era feito, mesmo de uma maneira arcaica e brutal, onde a extremidade era seccionada de forma rápida e sem nenhum tipo de anestesia, e para estancar o sangramento, o coto era esmagado ou queimado com óleo quente. O procedimento de amputação dessa época era tão severo que poucos eram capazes de sobreviver a ele, e os que sobreviviam não conseguiam um coto funcional para as suas atividades (MARCOLINO et al., 2015).

Com os avanços da área da saúde, a descoberta de anestésias e métodos próprios para amputação foram descobertos, melhorando a expectativa de vida dos amputados. O cuidado para com o paciente e as consequências da cirurgia são consideradas ao fazer o procedimento. Parâmetros foram atribuídos para uma amputação, onde eles determinam que deve-se preservar o maior comprimento funcional possível da extremidade, preservar a sensibilidade funcional do coto, da mesma forma que deve-se também proporcionar um formato esteticamente aceitável do coto para o paciente, evitar amputações precipitadas em crianças, considerando seu potencial de recuperação, entre outros aspectos (MARCOLINO et al., 2015).

Os principais causadores de uma amputação são traumas, in-

feccões, doenças vasculares e deficiências congênitas (KORKMAZ et al., 2012). Os traumas são causados em grande parte das vezes, em acidentes industriais, onde há uso de ferramentas e maquinários de grande porte (PARDINI, 2005). A maior incidência desses acidentes acontecem com homens, geralmente em sua idade produtiva, gerando períodos de afastamento variados, ou até mesmo invalidez, o que compromete diretamente todo o contexto socioeconômico (BRAGA-SILVA, 2001).

Os níveis de amputação do membro superior podem ser separados nos seguintes tipos: amputações do punho, antebraço, braço, quarto superior e também as desarticulações do punho, cotovelo e ombro. Das amputações citadas a que será tratada em específico neste documento será a de antebraço, também chamada de transradial. Nesta amputação é necessário ter o cuidado para que o coto se mantenha após o cotovelo, mesmo que muito curto, pois desta forma a articulação do cotovelo é preservada e é facilitado a protetização (MARCOLINO et al., 2015).

2.2 PROTETIZAÇÃO

A protetização é um termo na área da saúde que é utilizado para a reposição de um membro do corpo por um substituto sintético (prótese) no lugar deste membro que foi amputado. O uso das próteses foi atribuído para compensar a funcionalidade do membro perdido, não na totalidade, mas permitindo que o amputado possa executar suas atividades e não se tornar totalmente incapacitado. Entretanto, as próteses no passado ainda apresentavam muitos problemas, como encaixe ou irritação nos amputados. A Figura 1 exemplifica uma prótese funcional.

Figura 1 – Exemplo de prótese funcional usada pelos veteranos de guerra.



Foi no período de 1940 após as reclamações dos veteranos de guerra sobre a má adaptação com seus membros artificiais que levou Norman T. Kirk, cirurgião do exército, a pedir auxílio à Academia Nacional de Ciências (*National Academy of Sciences* - NAS), formando um Comitê Consultor sobre Membros Artificiais, que futuramente se tornou o Comitê de Pesquisa e Desenvolvimento de Próteses (*Committee on Prosthetics Research and Development* - CPRD), onde realizaram muitos avanços relacionados a biomecânica, materiais apropriados para as próteses, o que permitiu uma grande evolução nos modelos de próteses (MARCOLINO et al., 2015).

Protetizar um paciente não se trata de um procedimento simples. Diversos fatores devem ser considerados antes mesmo da amputação. Fatores como o nível da amputação, o formato e condições do coto, capacidade cognitiva, a habilidade de controle motor da musculatura remanescente e até mesmo aspectos de condições econômicas devem ser considerados. Dessa forma é de extrema importância que o paciente tenha uma avaliação interdisciplinar, com uma equipe de diversos profissionais que possam dar o suporte e atenção à todos os aspectos que influenciam na protetização do paciente (MARCOLINO et al., 2015).

O processo de protetização inicia desde a amputação até o término da reabilitação conta com um treinamento para a adaptação do paciente. Esse treinamento é dividido em dois períodos, pré-protético e o protético. Após a amputação efetuada o paciente irá para o período pré-protético, onde deverá se preparar psicologicamente e se adaptar para o uso da prótese. É um período que é necessário um acompanhamento do paciente, o guiando a aceitar sua nova condição e a ideia de

compensar seu membro perdido com uma prótese, pois diversas complicações podem ocorrer nesse período, tanto no estado físico ou mental do paciente. Uma das complicações que podem ocorrer é a sensação fantasma, onde o paciente consegue sentir o membro que já foi amputado (PARDINI, 2005). Com a conclusão do período pré-protético, o paciente entrará então na fase do período protético, onde neste momento ele deverá ser treinado para o uso da prótese. Durante esse treinamento o paciente terá que lidar com o uso da prótese, a aceitação da mesma, as limitações e restrições que ela possui comparado ao seu membro perdido. Todas as dúvidas que o paciente obter devem ser sanadas assim como ele deve ser ouvido e orientado sobre suas angústias e frustrações relacionadas a prótese. O retorno dado pelo paciente tem grande influência em ambos períodos para a sua reabilitação (PARDINI, 2005).

2.3 TIPOS DE PRÓTESE

Os tipos de prótese são distinguidos conforme sua funcionalidade. Devido ao foco deste documento, os aspectos tratados serão referentes a mão e antebraço. A mão é um membro de extrema importância, totalmente voltado para função. São incontáveis as posições que podem assumir os dedos e o punho (PARDINI, 2005). Essa inúmera quantidade de funcionalidades que a mão possui faz com que seja extremamente difícil replicar a todos com perfeição a uma prótese substituta. Portanto, o melhor que há a se fazer é replicar os movimentos mais importantes e funcionais, os quais serão descritos na próxima seção, e para alguns casos em específico o máximo de funcionalidades possível.

A funcionalidade necessária da prótese irá variar conforme as necessidades do próprio paciente. Segundo PARDINI (2005) quando uma prótese é indicada ela deve cumprir três principais requisitos que são:

- Chamar a menor atenção possível e parecer com o membro que substituem;
- Não causar nenhum desconforto, ou limitar ainda mais os movimentos;
- Oferecer opções funcionais que sejam convenientes para o paciente.

Os tipos de próteses são divididos conforme o nível da amputação que pode variar entre um dispositivo terminal, uma unidade de punho, antebraço, unidade de cotovelo, encaixe de braço superior, ombro ou dobradiça (LUSARDI et al., 2007).

O dispositivo terminal pode ser uma mão ou um gancho, onde o gancho possui uma pinça mais precisa enquanto o dispositivo tem uma força de prensão maior além de ter uma estética mais favorável (MARCOLINO et al., 2015). A Figura 2 exemplifica a diferença entre uma prótese do tipo gancho e mão.

Figura 2 – Exemplo de gancho e mão.



Fonte: Elaborada pelo autor.

As próteses também são divididas em relação ao tipo de sua fonte de energia, dentre elas é possível separá-las em quatro modelos principais: estética, mecânica, mioelétrica e híbrida (MARCOLINO et al., 2015). A prótese estética, (Figura 3), como seu nome sugere, é apenas para questão de aparência, não possui uma finalidade funcional, porém ajuda com a auto estima do paciente o que é de extrema importância para sua reabilitação.

Figura 3 – Exemplo de prótese estética.

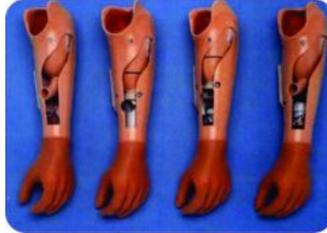


Fonte: Adaptada de

http://www.ortopediamathias.com.br/site/produtos/proteses/membro_superior.

A prótese mecânica possui movimentos mecânicos e é aconselhável para determinadas atividades. A dificuldade que ela apresenta é a necessidade de movimentação dos seus mecanismos para que possa mudar seu estado para executar alguma funcionalidade. A Figura 4 ilustra uma prótese do tipo mecânica.

Figura 4 – Exemplo de prótese mecânica.



Fonte: Adaptada de

<http://www.mundodareabilitacao.com.br/produto/produto-id/169/categoria-id/4/subcategoria-id/19/produto/protese-transradial-mecanica-protese>.

A prótese mioelétrica, diferente das outras, possui bastante opções funcionais. É controlada por pulsos elétricos gerados pelos músculos. Um microcontrolador os processa e os trata de forma quase instantânea. Contudo a prótese mioelétrica necessita de alguma fonte de energia elétrica para mantê-la funcionando. Na Figura 5 é ilustrado um exemplo de prótese mioelétrica.

Figura 5 – Exemplo de prótese mioelétrica.



Fonte: Adaptada de

<http://ortolifeortopedica.com.br/membrossuperiores.html>.

A protese híbrida é uma junção da mecânica com a mioelétrica. Para aumentar suas funcionalidades e ainda assim torna-lá viável em questão de controle e manuseio, alguns movimentos desta prótese são

realizados mecanicamente. Em questão de funcionalidades ela é mais propensa a executar mais funções, porém também apresenta desvantagem de ambas, ainda precisa de uma fonte externa para carga e também necessita movimento em seus mecanismos para determinada funcionalidade. A Figura 6 ilustra um exemplo de prótese híbrida.

Figura 6 – Exemplo de prótese híbrida.



Fonte: Adaptada de
<http://ethnos.com.br/proteses-hibridas/proteses/p/21>.

O avanço na tecnologia das próteses têm evoluído em muitos sentidos. Dentre todos os modelos mencionados a prótese mioelétrica possui o maior destaque, com movimentos de maior precisão e devido ao seu tempo de resposta que a torna praticamente instantânea em alguns modelos. O tempo de resposta menor possibilita maior adaptação para o paciente, tornando o processo de reabilitação mais ágil e eficiente (ZLOTOW; KOZIN, 2012).

Os atributos de uma prótese porém não são dados apenas através do seu modo de operação. Uma prótese necessita de alguns parâmetros que devem ser seguidos para que ela possa ser considerada funcional. A próxima sessão apresentará os conceitos de biomecânica, área que estuda os movimentos do corpo.

2.4 BIOMECÂNICA

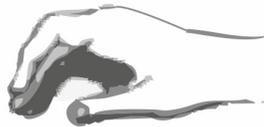
A biomecânica é o estudo dos movimentos efetuados pelo corpo. O estudo inclui músculos, os nervos, a estrutura óssea, todos os componentes que permitem a mão exercer as funcionalidades que ela possui. O estudo da biomecânica é de extrema importância para o desenvolvi-

mento de uma prótese funcional. Como mencionado nas seções anteriores, a mão possui um número enorme de funcionalidades das quais nem todas podem ser replicadas. Trata-se de um membro muito complexo o qual é necessário um mínimo de estudos para poder ter uma noção de seu funcionamento e poder replicar algumas de suas funcionalidades.

A mão tem com como principal uso em atividades de manipulação das quais precisam de movimentos finos e precisos, onde um grupo de músculos agem em conjunto sobre ela e o punho, possibilitando suas diversas posições funcionais (HAMILL; KNUTZEN, 2012). Quando a mão for apenas uma estrutura estática na extremidade do braço, ou seja não possuir nenhuma finalidade operante, como por exemplo a mão estática aberta sendo usada para empurrar um objeto ou como a mão fechada de um boxeador utilizada para lutar, este formato não pode ser considerado como uma função especializada (PARDINI, 2005).

A mão possui uma certa quantidade de características relacionadas a biomecânica que são de grande importância para o desenvolvimento de uma prótese, uma delas é a posição funcional, também conhecida como posição de repouso, onde essa formação é o ponto de partida dos movimentos funcionais mais importantes. A Figura 7 ilustra o formato do estado funcional da mão.

Figura 7 – Mão em seu estado funcional, também conhecido como estado de repouso.



Fonte: Adaptada de (HALL, 2013).

Os principais movimentos que a mão executa são as funções de pinça de precisão e pinça de força. Onde a pinça de precisão, em especial, é a função mais importante e a mais especializada da mão, que permite o alcance de objetos com um ajuste fino permitindo uma grande gama no tamanho dos objetos que a mão pode capturar (PARDINI, 2005). Já a pinça de força é mais utilizada para a manipulação de objetos mais robustos e pesados, que requerem menor precisão, mas necessitam de maior força para segurá-los, como por exemplo segurar

uma alça de mala ou um martelo. As Figuras 8 e 9 exemplificam uma funcionalidade da pinça de precisão e prensão, respectivamente.

Figura 8 – Exemplo de uma mão operando com uma pinça de precisão.



Fonte: Adaptada de (HALL, 2013).

Figura 9 – Exemplo de uma mão operando com uma pinça de força ou em prensão palmar.



Fonte: Adaptada de (HALL, 2013).

Os tipos de pinça podem ainda ser divididos em três tipos básicos: de ponta, lateral e palmar. Todos os tipos partem da posição de repouso da mão e cada um possui sua especialidade funcional correspondente. A pinça de ponta é realizada com a ponta dos dedos e é o tipo de pinça mais preciso, onde com ela é possível manipular objetos menores em tamanho e espessura. A pinça lateral é um movimento feito geralmente para segurar objetos como por exemplo uma chave, utilizando a lateral dos dedos, porém não é uma pinça muito utilizada no dia a dia. A pinça palmar é um tipo de pinça que utiliza maior área de contato dos dedos, e é a mais utilizada nas atividades diárias entre as três (PARDINI, 2005). As Figuras 10, 11 e 12, exemplificam, respectivamente, algumas

funcionalidades para pinça de ponta, lateral e palmar.

Figura 10 – Pinça de ponta, permite a mão manipular objetos pequenos por possuir muita precisão.



Fonte: Adaptada de (HALL, 2013).

Figura 11 – A pinça lateral também é conhecida como pinça de chave é própria para o manuseio de chaves ou objetos semelhantes.



Fonte: Adaptada de (HALL, 2013).

Figura 12 – Pinça de palmar permite o uso de força maior sobre o objeto e também possui a maior área de contato entre as três.



Fonte: Adaptada de (HALL, 2013).

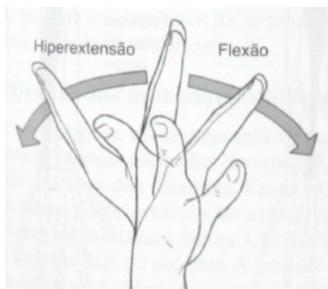
O punho por sua vez possui seus movimentos os quais complementam os movimentos que a mão utiliza, permitindo outra nova variedade de funções especializadas. Segundo Hall (2013) entre os movimentos do punho é válido citar:

- A flexão, onde é direcionado a região da palma da mão em direção ao antebraço;

- A extensão, onde o punho desloca a mão da posição de flexão para a posição original de descanso;
- Hiperextensão, onde a mão é sobre-estendida, além da posição de descanso;
- Desvio radial, o qual a mão é movida lateralmente direcionando o polegar no sentido do antebraço;
- Desvio ulnar, onde a mão também é movida lateralmente porém no sentido inverso do radial;

Além destes movimentos a união deles permite a movimentação de circundução. As Figuras 13 e 14 demonstram os movimentos de flexão e hiperextensão assim como os desvios radial e ulnar, respectivamente.

Figura 13 – Os movimentos de flexão e hiperextensão são apresentados na imagem, e o movimento de extensão onde a mão é direcionada a posição central.



Fonte: Adaptada de (HALL, 2013).

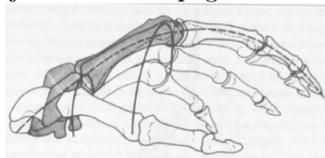
Figura 14 – Os desvios radial e ulnar do punho permitem que a mão se desloque lateralmente.



Fonte: Adaptada de (HALL, 2013).

A movimentação e as funções especializadas da mão e punho são possíveis devido a estrutura óssea e aos diversos músculos em conjunto e concorrentes. A disposição da arquitetura óssea é formada em arcos, o que facilita envolver os objetos quando forem pegados, facilitando os movimentos de pinça e preensão. A estrutura ainda converge para o mesmo ponto no momento em que a mão se fecha fazendo a flexão interfalangeana dos dedos (PARDINI, 2005). As Figuras 15 e 16 ilustram, respectivamente, a os formatos facilitadores na estrutura da mão.

Figura 15 – A estrutura óssea da mão é formada por arcos para facilitar o envolvimento do objeto ao tentar pegá-lo.



Fonte: Adaptada de (HALL, 2013).

Figura 16 – Um segundo facilitador para a captura de objetos é a convergência dos dedos ao mesmo ponto.



Fonte: Adaptada de (HALL, 2013).

O polegar é um dos maiores responsáveis em definir qual será a força e a precisão da pinça. Quanto maior a flexão interfalangeana dos dedos maior será a preensão da pinça, e conseqüentemente menor será sua precisão. No momento em que o punho é cerrado e o polegar cobre dos demais dedos flexionados é considerado uma posição de preensão máxima e por sua vez de precisão mínima ou nula. O mesmo ocorre para o inverso, quando mais fino o ajuste da pinça, maior será a precisão porém menor será a força aplicada (HAMILL; KNUTZEN, 2012).

A importância dos movimentos apresentados nessa seção deve-se ao fato de serem funções especializadas da mão e punho, movimentos funcionais que são utilizados mais comumente no dia-a-dia que outros inúmeros movimentos que a mão e o punho podem gerar em conjunto. Para o desenvolvimento de uma prótese, é fundamental compreender as funcionalidades mais básicas e usuais do membro original, de forma que seja possível replicá-los com qualidade e eficiência, o que torna a prótese mais eficiente e com maior ergonomia e aceitação pelo paciente, contribuindo para o processo de reabilitação.

2.5 A PRÓTESE MIOELÉTRICA

A prótese mioelétrica se destaca entre as demais por sua capacidade de ser controlada diretamente com a intenção do usuário. Isso é possível a partir da leitura do sinal mioelétrico, de onde vem o nome da prótese, gerado pelo músculo. Ela pode ser aplicada em muitos níveis de amputação, desde a secção do punho até mesmo a desarticulação do ombro, com o requisito de seu usuário ser capaz de contrair o grupo muscular medido, gerando um sinal mioelétrico forte o suficiente para comandar a prótese (POLIS et al., 2009).

O sinal mioelétrico também é conhecido como EMG (do inglês *Electromyography*), que consiste em um potencial elétrico gerado de acordo com a contração do músculo, o qual pode ser capturado através de um sensor na superfície da pele (EMG de superfície) ou implantado diretamente no músculo, o que é chamado de sensor invasivo (MATTIOLI et al., 2010). Apesar de sensores invasivos possuem um sinal mais forte e imensamente menos afetado por ruídos, sensores não invasivos são amplamente utilizados, uma vez que não há necessidade da preocupação em perfurar a musculatura e um cuidado extremamente maior com esterilização, tendo sensores os mais simples utilização e até mesmo descartáveis, como os de eletrocardiograma (POLIS et al., 2009). Este potencial elétrico gerado pelo músculo é proporcional a intensidade da contração do mesmo, deste modo a ativação do atuador da prótese, por exemplo o fechar ou abri da mão, é proporcional a uma contração do músculo, operando um movimento mais suave ou brusco, respectivamente, a uma contração mais leve ou forte (BARROS, 2005).

Um circuito integrado é responsável em capturar este sinal através de eletrodos e o transmite para um microcontrolador que irá processá-lo e comandar motores que moverão as articulações da prótese, conforme o movimento previsto do sinal.

Apesar da dificuldade para o desenvolvimento de uma interface humana, a qual deve ser própria para utilização móvel e com tecnologia adequada para o tratamento do sinal EMG (BU et al., 2004), diversos estudos destacam vantagens do uso do sinal de mioelétrico para a reabilitação de pacientes com próteses experimentais (FAVIEIRO; BALBINOT; BARRETO, 2011). Segundo Favieiro, Balbinot e Barreto (2011) estudos mostram que amputados tem a capacidade de gerar uma padronização de sinais mioelétricos gerados repetidamente antes da intenção de gerar um movimento, o que permite ao capturar esse sinal, trata-lo para que a prótese mioelétrica possa replicar um movimento conforme a intenção do amputado.

Uma grande preocupação é a desistência do amputado durante o período de reabilitação, por não conseguir se adaptar a prótese durante o período de treinamento (MATTIOLI *et al.*, 2010). Porém com a prótese mioelétrica, é possível conseguir um ambiente controlado maior, onde o paciente pode ter uma melhor resposta do dispositivo, um maior conjunto de movimentos, o que pode motivá-lo a não desistir do uso da prótese.

O sinal mioelétrico não é capturado de forma trivial, pois possui um nível de tensão muito baixo e ainda conta com inúmeros tipos de ruídos em sua captura, que podem ocasionar a sua perda. Para que seja possível o controle de forma eficiente, é necessário reconhecer esse sinal mesmo que ele possua uma forma incerta, que não seja exatamente igual ao padrão. Isso pode ser realizado com os métodos do capítulo seguinte, a inteligência artificial.

3 REDES NEURAIS ARTIFICIAIS

Este capítulo apresenta os principais conceitos a respeito das Redes Neurais Artificiais e suas principais aplicações. Para facilitar o entendimento dos conceitos apresentados, este capítulo também descreve brevemente o funcionamento de um neurônio biológico.

3.1 REDES NEURAIS BIOLÓGICAS

O cérebro chama a atenção dos pesquisadores devido seu potencial de processamento, realizando em segundos tarefas que são inviáveis para os mais poderosos computadores. Um exemplo da capacidade de processamento do cérebro é de uma pessoa ser capaz de detectar uma imagem e associa-la ao que ela é ou representa, ato que é dificilmente alcançável por computadores com alta capacidade de processamento, o que inclusive gera uma enorme gama de estudos e novos algoritmos para o tratamento desse problema. Segundo Haykin (2001) o cérebro é um computador de alta complexidade, não linear e paralelo, que tem capacidade de organizar seus neurônios de maneira que o permite realizar processamentos muito mais rápidos que o mais rápido computador existente.

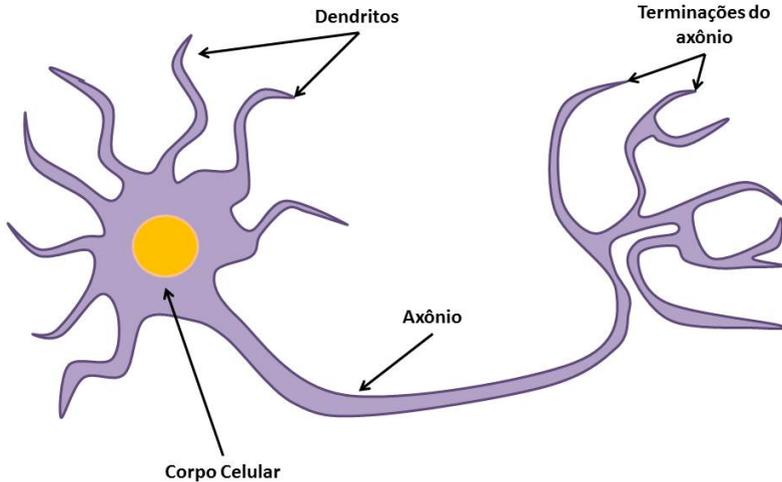
O sistema nervoso humano pode ser observado através de três estágios, a recepção de uma informação, o processamento e a transmissão. O cérebro é o centro do sistema. Ele receberá constantemente informações do corpo que devem ser processadas e seus resultados devem ser transmitidos. A recepção dessas informações é através da conversão de estímulos do próprio corpo humano ou do ambiente externo em impulsos elétricos que serão recebidos pelo cérebro. Os atuadores do corpo devem converter os impulsos elétricos transmitidos pelo cérebro nas respostas esperadas do corpo, como por exemplo movimento (HAYKIN, 2001).

As células nervosas elementares, as quais são chamadas de neurônios, são as responsáveis pelo processamento das informações levadas ao cérebro. O cérebro é composto por cerca de 10^{11} neurônios, onde cada um deles possui aproximadamente 10^3 a 10^4 conexões com outros demais neurônios, estimando-se que o cérebro humano tenha entre 10^{14} a 10^{15} conexões ao todo (JÚNIOR; YONEYAMA, 2002).

Os neurônios possuem uma grande variedade de tipos, com funções e forma próprias, e localização específica. Entretanto qualquer neurônio

e composto basicamente de um corpo celular, o qual pode ser chamado também de soma, dendritos e um axônio (JÚNIOR; YONEYAMA, 2002). A Figura 17 ilustra a estrutura de um neurônio biológico.

Figura 17 – Estrutura básica de um neurônio biológico.



Fonte: Elaborada pelo autor.

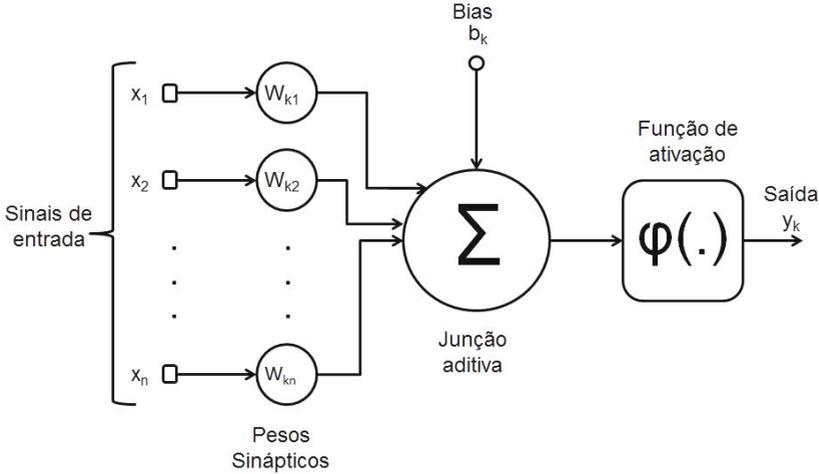
Usualmente um neurônio possui muitos dendritos que normalmente são ramificados, os quais são responsáveis por captar a informação dos demais neurônios, e apenas um axônio o que é responsável por enviar a informação processada no corpo celular. O axônio se ramifica ao seu final fazendo conexões com os dendritos de demais neurônios através de sinapses (ROSA, 2011). A sinapse é uma conexão eletroquímica, e não física, feita entre uma das ramificações do axônio e o dendrito os quais possuem um pequeno espaço entre eles que é chamado de fenda sináptica, que possui um tamanho cerca de 2000^{-10} a 5000^{-10} m (JÚNIOR; YONEYAMA, 2002). A informação transmitida pelo axônio é em forma de pulsos elétricos de tensão os quais são chamados de potencial de ação. Este potencial de ação ao chegar nas sinapses liberam moléculas chamadas de neurotransmissores, os quais irão atravessar a sinapse e modificar o potencial da membrana do dendrito. Conforme a

predominância do tipo de neurotransmissor o potencial da membrana do dendrito será aumentado, o que é chamado de sinapse excitatória, ou diminuído, o que é chamado de sinapse inibidora. Esses sinais recebidos de diversos neurônios no dendritos vão para o corpo celular do neurônio que irá somá-los de forma aproximada. Caso a soma realizada em um pequeno espaço de tempo sobressair um determinado limite, o corpo celular irá gerar um novo potencial de ação que será transmitido para os demais neurônios, repetindo esse ciclo (JÚNIOR; YONEYAMA, 2002). De um modo mais simples, pode-se dizer que o cérebro opera com os neurônios ativando ou inibindo os disparos dos potenciais de ação de outros neurônios. Do modo que esses disparos irão ou não ocorrer conforme as entradas excitatórias ou inibitórias de todos os neurônios conectados a ele (ROSA, 2011).

3.2 FUNDAMENTOS DE REDES NEURAIS ARTIFICIAIS

As RNAs são uma técnica de inteligência artificial, cujo nome é dado devido a ela ser baseada no funcionamento das redes neurais biológicas. Os módulos de uma RNA são baseados nos neurônios biológicos mencionados na seção anterior. Eles também são módulos de processamento e são chamados de neurônios artificiais, onde recebem determinadas entradas de dados que serão processadas e disparam uma saída para outros neurônios ou para um atuador. Entretanto, é importante ressaltar que a RNA replica apenas o conceito da rede neural biológica baseada no conhecimento atual sobre o assunto, mas não tem a responsabilidade em replicar perfeitamente o sistema. Isso porque o foco principal da RNA é replicar as qualidades de uma rede neural biológica que é fornecer um enfoque alternativo a problemas onde um enfoque numérico ou simbólico não seriam muito adequados (JÚNIOR; YONEYAMA, 2002). O primeiro neurônio artificial foi desenvolvido por Warren McCulloch e Walter Pitts no ano de 1943 (CAMARGO, 2008). A Figura 18 ilustra um modelo de um neurônio artificial.

Figura 18 – Modelo de um neurônio artificial.



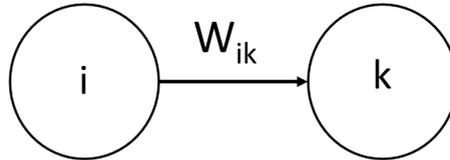
Fonte: Adaptada de (HAYKIN, 2001).

Os valores de x_m são as entradas da rede, os sinais que devem ser processados, podendo ser provenientes do ambiente ou de um neurônio anterior. Os valores de w_{km} são os pesos sinápticos, estes e o valor posterior b_k chamado de bias são nomeados de parâmetros livres, os quais tem a funcionalidade de agregar um peso às entradas e a função somatória, respectivamente. Os valores dos parâmetros livres irão influenciar diretamente no resultado da rede, como será melhor explicado posteriormente. A junção aditiva como seu nome sugere irá aplicar uma soma de todos os valores recebidos agregados com os parâmetros livres. Por fim, a função de ativação será a responsável pela decisão do neurônio, se sua saída irá inferir em um resultado ativo ou não. Esta saída será o final da rede ou a entrada de um novo neurônio.

O objetivo da rede é que a partir de uma ou mais entradas, os dados cumpram as etapas mencionadas acima e definam um resultado ativo ou não, que deve ser propagado até o final da rede, fazendo essas interações e conexões com a saída de outros neurônios até chegar ao final da rede, onde o resultado deve ser interpretado para inferir o resultado real.

A conexão entre dois neurônios pode ser estabelecida como exemplificado na Figura 19, onde o valor W_{ik} é o peso sináptico dessa conexão.

Figura 19 – Conexão entre um neurônio i e k.



Fonte: Elaborada pelo autor.

O resultado do processo de um neurônio é gerido pela chamada regra de propagação onde será definido um potencial de ativação do neurônio chamado de Net_k . O valor de Net_k é dado pela Equação 3.1:

$$Net_k = \sum W_{ik} * x_i \quad (3.1)$$

Onde o valor x_i é a entrada i, que será multiplicada pelo peso sináptico W_{ik} correspondente, e, feito um somatório com os n termos conectados.

3.2.1 Função de Ativação

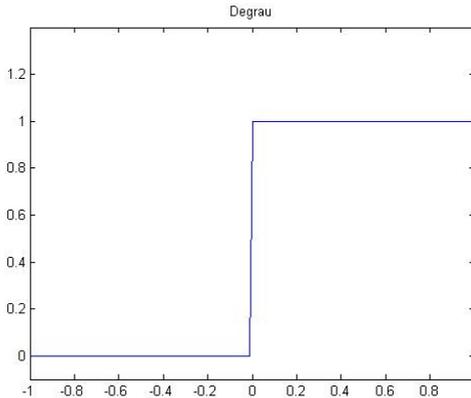
A função de ativação é a responsável por definir o estado do neurônio, se a conexão será ativa ou não. Através de uma função onde a Net_k será aplicada e a ativação será definida a partir dessa função. O formato padrão para a função de ativação é dado através da Equação 3.2:

$$y_k = \varphi(Net_k + b_k); \quad (3.2)$$

Onde y_k é a resposta da função de ativação escolhida φ , e b_k é o valor do bias, que é agregado a Net_k como mencionado anteriormente para agregar a entrada líquida da função de ativação, aumentando ou diminuindo. As funções utilizadas em φ podem adquirir várias formas,

entre elas é importante citar a função degrau e rampa. As Figuras 20 e 21 demonstram as respectivas funções.

Figura 20 – Exemplo de uma função degrau



Fonte: Elaborada pelo autor.

O valor $\varphi(v)$ que trata-se do y_k mencionado anteriormente que é resultado da função de ativação, a posição que é ocupada na função, onde v é a soma de Net_k e b_k .

A função degrau, ou limiar, trata-se de um tipo de função conhecida como tudo ou nada, onde até determinado ponto retorna zero e a partir de seu limiar retorna um. Esta função possui uma variação chamada de degrau bipolar, onde seu princípio é o mesmo, porém seu estágio desativado passa a ser menos um e não zero. A função degrau é definida pela Equação 3.3 .

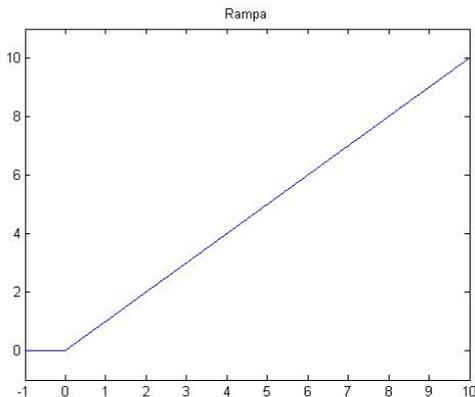
$$\begin{aligned} y_k &= 0 & \text{para } v < 0 \\ y_k &= 1 & \text{para } v \geq 0 \end{aligned} \quad (3.3)$$

A função rampa por sua vez possui dois limiares, onde o resultado será zero ou um e entre eles o valor obtido será o próprio $\varphi(v)$. Ela é definida pela Equação 3.4.

$$\begin{aligned} y_k &= 0 & \text{para } v \leq -\frac{1}{2} \\ y_k &= \varphi(v) & \text{para } -\frac{1}{2} > v > \frac{1}{2} \\ y_k &= 1 & \text{para } v \geq \frac{1}{2} \end{aligned} \quad (3.4)$$

O tipo de função sigmoide possui um equilíbrio adequado entre o comportamento linear e não-linear (SILVA, 2010a). A função é definida

Figura 21 – Exemplo de uma função rampa.

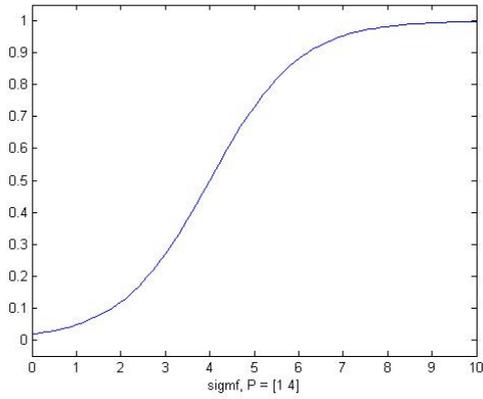


Fonte: Elaborada pelo autor.

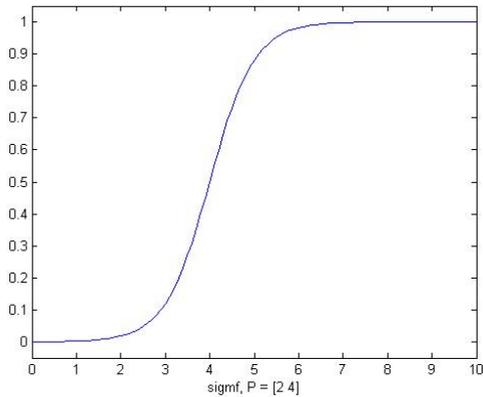
pela Equação 3.5:

$$\varphi(v) = \frac{1}{1 + e^{(-aN\bar{E}T_k + b_k)}} \quad (3.5)$$

Onde o valor de a define a inclinação da função sigmoide, o que é interessante por permitir adaptar a função conforme determinado problema. As Figuras 22, 23 e 24 ilustram, respectivamente, uma função sigmoide e sua variação de conforme a alteração do valor de a .

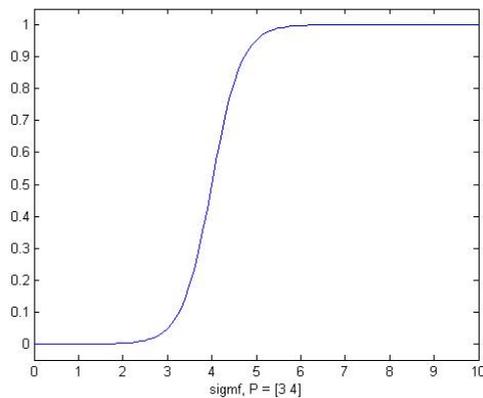
Figura 22 – Exemplo de uma função sigmoide com valor de $a = 1$.

Fonte: Elaborada pelo autor.

Figura 23 – Exemplo de uma função sigmoide com valor de $a = 2$.

Fonte: Elaborada pelo autor.

Figura 24 – Exemplo de uma função sigmoide com valor de $a = 3$.



Fonte: Elaborada pelo autor.

3.3 VANTAGENS DE UMA RNA

O interesse em inteligência computacional obteve uma expansão considerável nos últimos anos, como evidenciado em inúmeras aplicações direcionadas a área de saúde, biomedicina e engenharia biomédica (BALBINOT; FAVIEIRO, 2013).

A capacidade de processamento de uma RNA vem de sua estrutura maciçamente paralela distribuída e sua habilidade de aprendizado e generalização. É dita generalização quando a rede infere um resultado mesmo tendo sido treinada para reconhecer este valor. Essas duas qualidades da RNA permitem que ela solucione problemas massivos e de alta complexidade, alguns intratáveis por outros métodos (HAYKIN, 2001).

A maioria dos computadores são baseados na arquitetura de von Neumann, que consiste em centralizar todas as operações do computador em um único processador potente e complexo (JÚNIOR; YONEYAMA, 2002). Contudo, algoritmos inteligentes como RNAs possuem vantagens com relação a outros métodos, principalmente em problemas de reconhecimento de padrões, como os sinais mioelétricos, pois proporcionam adaptação a cada paciente e um maior antropomorfismo, possibilitando que o amputado tenha um controle mais natural do dispositivo e consequentemente diminuindo a taxa de rejeição (CAMARGO, 2008). Isso é possível devido as características da RNA como, por exemplo, a

capacidade de aprender através de exemplos, tolerância a falhas, capacidade de generalização e processamento paralelo. Isso tudo agregado a uma certa simplicidade na implementação em hardware, tendo uma certa robustez no tratamento de dados com ruído, além de sua aplicabilidade em sistemas de tempo real (CAMARGO, 2008).

É importante ressaltar que as técnicas convencionais de programação são capazes de resolver inúmeros problemas de forma própria e eficiente. Contudo as RNAs podem ser utilizadas para resolver situações onde técnicas de programação mais clássicas seriam menos adequadas (JÚNIOR; YONEYAMA, 2002). A RNA não é um método único para solução de problemas complexos. Outras técnicas como lógica Fuzzy, sistemas de controle Proporcional-Integral-Derivativo (PID) são utilizados para problemas que tenham entrada de dados com ruído ou que possuam a necessidade de um ajuste fino frequente. Entretanto, alguns casos específicos fazem a RNA se sobressair entre os processos mencionados. O controlador PID por exemplo, apesar de ter um excelente desempenho com sistemas lineares, com o tempo de uso do sistema os parâmetros podem mudar de forma que o ganho fixo do PID não funcione de maneira apropriada, deste modo essas variações de parâmetros e distúrbios externos degradam o conhecimento do controlador (JUNG; KIM, 2007). Para um sistema de controle mioelétrico este é um problema grave, uma vez que os sinais mioelétricos recebidos variam conforme a pessoa.

3.4 TIPOS DE RNA

As RNAs podem ser estruturadas de diferentes maneiras. Com base na estrutura de uma RNA escolhe-se o algoritmo para o treinamento da rede. Entre as arquiteturas de rede é necessário mencionar três tipos fundamentais (HAYKIN, 2001):

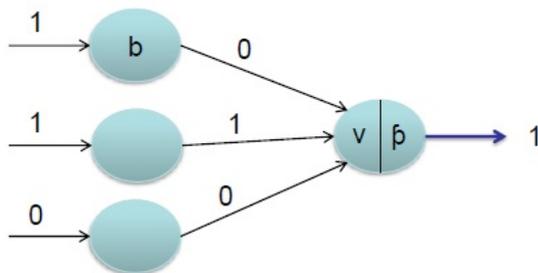
- Redes com camada única;
- Redes com múltiplas camadas;
- Redes recorrentes.

3.4.1 Redes de Camada Única

A rede com camada única é a rede mais básica e simples, onde um conjunto de neurônios referentes a saída recebem um determinado

número de entradas, que são processadas para que então a RNA gere a saída ou saídas. O Perceptron é um exemplo de RNA de camada única e foi considerado um marco histórico das redes neurais. Criado por Frank Rosenblatt, o Perceptron é uma RNA com múltiplos neurônios baseados no modelo de McCulloch, desenvolvido com seu modelo matemático que mimetiza a sinapse humana (CAMARGO, 2008). A Figura 25 apresenta o modelo de um Perceptron simples.

Figura 25 – Exemplo de um Perceptron com camada única.



Fonte: Adaptada de (HAYKIN, 2001).

Um dos problemas do Perceptron foi sua inaptidão em resolver situações não lineares como por exemplo o OU-Exclusivo. Esta problemática inviabilizou a solução de problemas com redes de camadas únicas. Para resolver essa situação foi desenvolvida a segunda arquitetura de redes neurais, a rede de múltiplas camadas.

3.4.2 Redes de Múltiplas Camadas

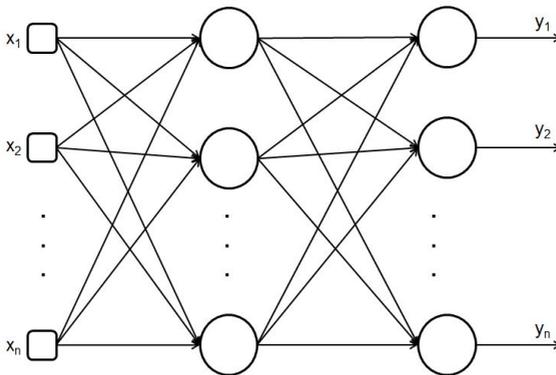
As redes de múltiplas camadas se diferem das redes de camada única por possuírem uma ou mais camadas de neurônios intermediárias entre a camada de entrada e saída. O objetivo do acréscimo dessas camadas intermediárias (também conhecidas como camadas ocultas) é intervir nos dados de entrada externos e os dados de saída, de tal forma que a rede possua capacidade de extrair estatísticas com ordem elevada, o que se trata de uma particularidade valiosa especialmente quando o tamanho da camada de entrada é grande. Com o conjunto extra de conexões sinápticas e interações neurais a rede ganha uma perspectiva global mesmo tendo suas conexões de forma local (HAY-

KIN, 2001). Uma das redes mais famosas que utilizam essa arquitetura é a rede Multilayer-Perceptron (Perceptron de Múltiplas Camadas), também conhecida como rede MLP.

3.4.2.1 Multi-Layer-Perceptron

O Multi-Layer-Perceptron é uma rede neural de múltiplas camadas e é um dos modelos mais utilizados e conhecidos na atualidade (CAMARGO, 2008). Ele consiste em uma camada de entrada que receberá os dados externos, adicionará seus pesos sinápticos e os passará para frente, uma ou mais camadas ocultas que irão tratar os valores conforme o algoritmo de aprendizagem e uma camada de saída que irá inferir o resultado da rede (FAVIEIRO; BALBINOT; BARRETO, 2011). A Figura 26 ilustra a forma geral de uma MLP.

Figura 26 – Exemplo geral de uma rede neural MLP.



Fonte: Adaptada de (HAYKIN, 2001).

A rede MLP é uma sucessora do Perceptron simples, capaz de solucionar problemas não-linearmente separáveis como o OU-Exclusivo, problema impossível de ser resolvido por um Perceptron. Um dos motivos para a rede MLP ser capaz de tal feito é devido a sua função de ativação não-linear, comumente empregada, a sigmoideal ou logística, que trata-se de uma função crescente que possui propriedades homogêneas e assintóticas, sendo diferenciável em seu domínio (CAMARGO, 2008). Agregado a isso o conjunto de sinapses e interações entre as camadas permitem a maior correção dos pesos sinápticos de cada neurônio conforme o algoritmo de aprendizado, por exemplo a *backpropagation* que

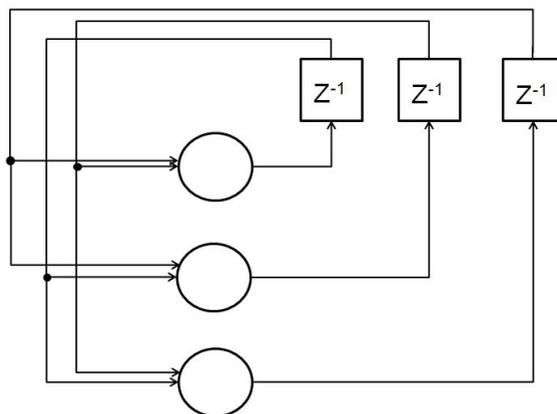
está descrito na sessão 3.5.1.

Segundo Camargo (2008), a MLP já foi implementada com sucesso para o reconhecimento de padrões para uma prótese mioelétrica, que reproduzia movimentos funcionais.

3.4.3 Redes Recorrentes

A arquitetura de redes recorrentes é a mais diferenciada das três. Ela é caracterizada por possuir sempre um laço de realimentação da rede, podendo ter desde apenas uma camada ou múltiplas camadas com neurônios ocultos. Uma rede recorrente pode possuir ou não auto-realimentação, que se trata de um neurônio alimentar sua entrada com a própria saída (HAYKIN, 2001). A Figura 27 ilustra um exemplo de uma rede recorrente de uma única camada sem auto-realimentação.

Figura 27 – Exemplo de Rede Recorrente.



Fonte: Adaptada de (HAYKIN, 2001).

3.5 ALGORÍTIMOS DE TREINAMENTO

Os algoritmos de treinamento, também chamados algoritmos de aprendizagem, são os responsáveis por preparar a RNA para que ela opere como o desejado. Para isso, é efetuado um processo iterativo que ajustará os pesos sinápticos e níveis de bias da rede. O processo se dará como concluído no momento em que a rede adquire uma solução gene-

realizada para o problema, conforme o sugerido no treinamento (SILVA, 2010a).

A atualização dos pesos sinápticos é realizada conforme o algoritmo de treinamento selecionado, e é atualizado de acordo com a Equação 3.6:

$$w(t+1) = w(t) + \delta w(t) \quad (3.6)$$

Onde $t+1$ é o tempo posterior a interação, $w(t)$ é o peso sináptico atual e $\delta w(t)$ é o ajuste aplicado aos pesos, que possui o seu método de atualização diferente conforme o algoritmo de treinamento.

O formato básico do treinamento de uma rede se dá a partir da entrada fornecida como parâmetro, que no momento de atuação a rede irá alterando seus parâmetros livres, os pesos sinápticos e o bias, durante cada interação. Após a atualização dos parâmetros a rede irá responder de uma nova maneira, baseada nos parâmetros atualizados. O treinamento é concluído com base em critérios determinados pelo próprio programador, que pode ser o número das épocas de treinamento (quantidade de vezes que o processo de treinamento é feito), ou relacionado ao valor de erro mínimo, ou seja, uma taxa de valores errôneos aceitáveis pelo sistema.

A RNA deve ser treinada antes da aplicação, não podendo jamais operar sobre a aplicação antes de receber seus pesos sinápticos ajustados. Contudo no momento que os seus parâmetros livres forem atualizados eles não devem mais ser alterados durante o funcionamento da aplicação.

Os algoritmos de aprendizagem são classificados em três métodos principais, o algoritmo supervisionado, não-supervisionado e de reforço.

O algoritmo supervisionado é caracterizado por possuir a figura de um supervisor, o que em termos práticos trata-se da apresentação das saídas desejadas ao conjunto dos vetores de entrada, enquanto o algoritmo não-supervisionado não há apresentação das saídas desejadas (LUCKS, 2009).

Os algoritmos supervisionados trabalham com base nas Equações 3.7 e 3.8:

$$e(t) = d(t) - y(t) \quad (3.7)$$

$$w(t+1) = w(t) + ne(t)X(t) \quad (3.8)$$

Onde $e(t)$ é o erro obtido da interação através da diferença entre a saída desejada ($d(t)$) e a saída obtida ($y(t)$). O n é a taxa de

aprendizado e o $X(t)$ é o valor de entrada obtido pela rede naquele determinado tempo.

O conjunto de erros por épocas são agregados através da Equação 3.9:

$$\varepsilon^2 = \frac{1}{2} \sum_{i=1}^p (y_d^i - y)^2 \quad (3.9)$$

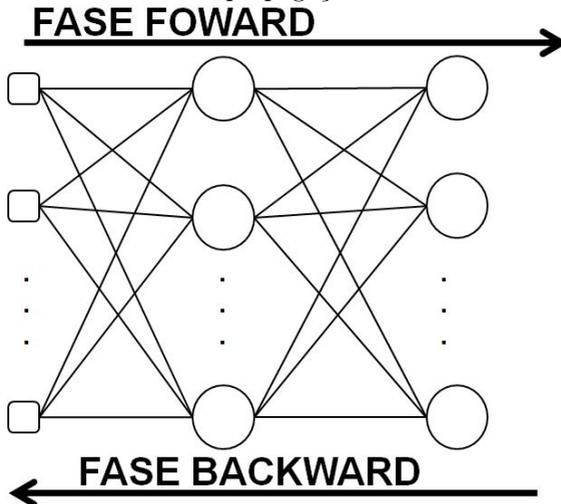
Onde o ε^2 é o valor do erro, p é o número de épocas ocorridas, y_d^i trata-se da saída desejada e y é a saída real. Este valor de erro é importante para verificação do erro da rede durante as épocas, fatores determinantes para o término do treinamento. O algoritmo não supervisionado não calcula o erro, para tanto necessita de bastante regularidade na redundância nas entradas. RNAs baseadas em algoritmos não supervisionados são muito utilizadas para agrupamento de dados.

O treinamento por reforço fortalece as conexões sinápticas que contribuíram com um bom resultado de saída e enfraquece aqueles que não contribuem ou os que contribuíram com uma saída considerada errada.

3.5.1 Algoritmo Backpropagation

O algoritmo backpropagation, ou como também é conhecido retro-propagação, trata-se do método utilizado no treinamento de redes MLP. A maioria das redes utiliza conexões feedforward, ou seja os dados processados em determinado neurônio sempre são passados adiante, até chegar a saída. Contudo, o resultado obtido pela rede pode não ser o esperado, ou seja o resultado não é o desejado. Desta forma, os pesos sinápticos devem ser ajustados. Para tanto, o erro da saída é transmitido de forma inversa, feedback, se propagando na rede da saída até a entrada. A Figura 28 apresenta ambos sentidos de propagação.

Figura 28 – Sentido de propagação feedforward e feedback



Fonte: Adaptada de (HAYKIN, 2001).

O erro propagado pela rede deve gerar novas sinapses e alterar seus pesos sinápticos novamente com a entrada recebida, porém dessa vez incluindo na equação o valor do erro. Caso o resultado final ainda não seja aceitável, o algoritmo deve atualizar o valor do erro e novamente propagá-lo pela rede, de modo que ela possa repetir o processo de conexões e atualização dos pesos sinápticos considerando o novo valor do erro. Esse processo é conhecido como backpropagation do erro, onde a rede é treinada de modo supervisionado a corrigir os erros gerados nos resultados. O algoritmo de treinamento acaba no momento em que a rede atinge um resultado aceitável.

3.6 APLICAÇÕES

As RNAs são aplicadas na solução de problemas devido sua capacidade de aprendizagem com um número conciso de exemplos, e sua capacidade de generalização retornando resultados coerentes com entradas desconhecidas (ROSSALES et al., 2012).

O uso de RNA para soluções com próteses mioelétricas já é a opção utilizada por diversos autores, onde inúmeras funcionalidades são atribuídas para diferentes tipos de soluções, desde a redução do número de eletrodos para leitura dos sinais de entrada utilizando uma MLP para

manter o número de saídas funcionais (FAVIEIRO; BALBINOT; BARRETO, 2011), até para atualização e reestruturação dos sinais de entrada para um sistema PID para eliminar a degradação gradual (JUNG; KIM, 2007), ou mesmo uma solução híbrida utilizando em conjunto a lógica Fuzzy e uma RNA, criando um sistema Neuro-Fuzzy (BALBINOT; FAVIEIRO, 2013).

O uso do algoritmo supervisionado para uma MLP foi escolhido como melhor opção para a prótese mioelétrica que será desenvolvida, uma vez que serão pré-definidos os movimentos funcionais que a prótese opera, ou seja as saídas do sistema. As entradas da rede serão capturadas através de sensores EMG onde o amputado irá gerar o sinal conforme a sua intenção de determinado movimento, definindo um certo padrão no sinal. Através de um algoritmo backpropagation a rede pode ser treinada a inferir determinada saída conforme o padrão do sinal de entrada. Assim, uma vez a rede treinada ela poderá inferir o resultado desejado conforme a intenção do amputado, mesmo com casos de degradação dos sinais, isso devido as características da RNA.

4 DESCRIÇÃO DA PRÓTESE MIOELÉTRICA DESENVOLVIDA EM FPGA

Este capítulo descreve a prótese mioelétrica desenvolvida. No capítulo são abordadas as principais características da prótese, bem como os parâmetros utilizados na Rede Neural Artificial que controla a prótese.

4.1 DESCRIÇÃO GERAL DO PROBLEMA PROPOSTO

A prótese mioelétrica proposta tem como objetivo fornecer um grau de liberdade a prensão palmar, que é descrita na Seção 2.4. O movimento é reproduzido gradualmente, conforme a ativação do bíceps.

O sistema de controle, após treinamento da RNA, irá operar como um sistema de malha aberta, conforme ilustra a Figura 29.

Figura 29 – Exemplo de um sistema de malha aberta



Fonte: Elaborada pelo autor.

A captura dos sinais provenientes do músculo serão a entrada do sistema; a FPGA com a RNA é o de processamento; o resultado obtido é aplicado em um atuador, para este projeto um servo motor é a saída do sistema que saída do sistema deve ser proporcional com sua entrada. Para tanto, é necessário cumprir três etapas:

1. Criação do conjunto de treinamento da RNA;
2. Arquitetura da RNA implementada;
3. Projeto e desenvolvimento do hardware.

As próximas seções irão descrever em detalhes cada uma das etapas listadas acima.

4.2 CRIAÇÃO DO CONJUNTO DE TREINAMENTO DA RNA

A criação do conjunto de treinamento é uma etapa muito importante, pois este conjunto de dados irá determinar como a RNA irá operar. Desta forma foi necessário fazer uma coleta de dados do bíceps para se conhecer a faixa de variação dos seus sinais. Esta coleta foi realizada no Laboratório de Avaliação e Reabilitação do Aparelho Locomotor (LARAL) da UFSC, onde foi aplicado o método Oxford orientado pelos integrantes do laboratório. Foi utilizado apenas um voluntário para o processo do sexo masculino com 25 anos de idade. A opção em utilizar apenas um voluntário foi devido a falta de permissão do Comitê de Ética em Pesquisa com Seres Humanos da UFSC para se coletar dados de outras pessoas.

Para a captação do sinal eletromiográfico foi calculada a resistência máxima (1 RM) do bíceps braquial dominante. Com base no protocolo Oxford, foram realizadas 10 repetições com 100% de 10 RM, seguidos por 10 repetições com 75% de 10 RM e, por fim, 10 repetições com 50% de carga de 10 RM, além de contrações isométricas (10 segundos de sustentação) em 45°, 90° e na flexão máxima. Para o cálculo da RM e o fortalecimento, foram utilizadas anilhas (1,0, 2,0 e 5,0 Kg) adaptadas à barra de exercícios conforme ilustra a Figura 30.

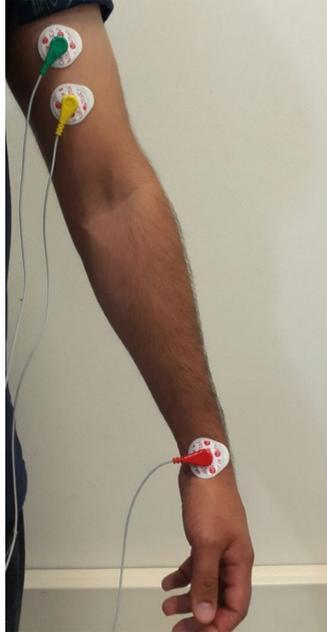
Figura 30 – Ilustração de um dos exercícios realizados.



Fonte: Elaborada pelo autor.

Para capturar os sinais mioelétricos foram utilizados um conjunto de eletrodos conectados ao braço conforme ilustrado na Figura 31.

Figura 31 – Posicionamento dos eletrodos para a captura dos dados.



Fonte: Elaborada pelo autor.

Onde um par de eletrodos é conectado no ventre do músculo, posição a qual obtêm-se máxima potência de ativação do músculo (FAVIEIRO, 2009), ambos capturando um sinal semelhante para obter uma média e fornecer um dado mais preciso, enquanto outro eletrodo é conectado em uma região óssea, que não emita sinal elétrico ou o mínimo possível, para que possa servir como referência do circuito. Para a coleta de dados foi escolhida extremidade do rádio distal, porém outras opções poderiam ter sido escolhidas.

Os sinais capturados nos eletrodos são lidos em um dispositivo sensor EMG, que dispõe de filtros e amplificadores operacionais que tratam o sinal e retornam um sinal analógico. O dispositivo pode ser visto na Figura 32.

Figura 32 – Posicionamento dos eletrodos para a captura dos dados.



Fonte: Adaptada do datasheet *Three lead Differential Muscle/Electromyography Sensor for Microcontroller Applications*.

A leitura dos dados provenientes do sensor necessitam ser processadas por algum dispositivo com conversor analógico-digital. Como a FPGA utilizada não dispõe deste conversor, foi necessário utilizar um outro microcontrolador que dispunha de conversor analógico-digital. Para este projeto foi utilizado o Arduino Uno, por possuir um conversor analógico-digital de dez bits, ou seja possui uma variação de 0 a 1023 unidades, e possui uma interface mais simples de ser trabalhada. O Arduino realiza a leitura do sinal analógico, e o envia para sua porta serial, que transmitirá a um computador. A Figura 33 ilustra o dispositivo.

Figura 33 – Dispositivo Arduino Uno, utilizado para conversão Analógico-Digital.



Fonte: Elaborada pelo autor.

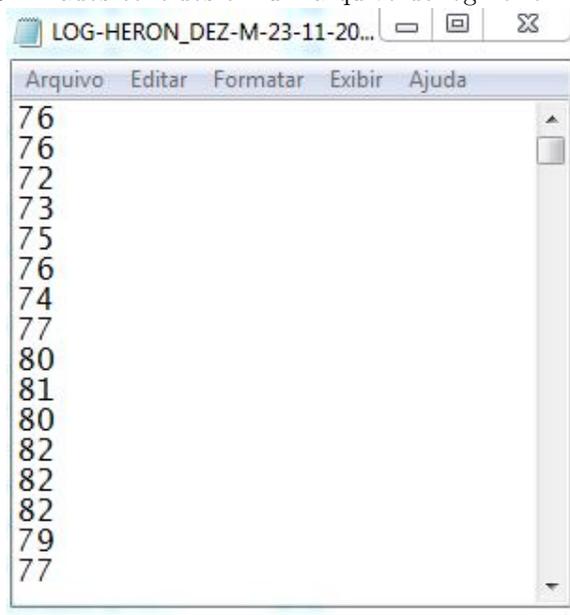
O registro dos dados foi realizado por um programa desenvolvido na linguagem C#, o qual gera um arquivo .txt que armazena os dados coletados. Foi gerado um arquivo de dados para cada teste realizado. As Figuras 34 e 35 ilustram, respectivamente, a tela inicial do programa de captura e um conjunto de dados armazenados em um arquivo no formato .txt.

Figura 34 – Interface principal do sistema de captura dos dados.



Fonte: Elaborada pelo autor.

Figura 35 – Dados contidos em um arquivo de log no formato texto.



Fonte: Elaborada pelo autor.

Com as coletas de dados realizadas, foi então definido o conjunto de dados para se aplicar na RNA. Foi analisado que as faixas de valores mais baixas as quais estavam próximas de 20, e as mais altas de 180. Contudo foi possível observar o estado de relaxamento em valores próximos a 50, e a partir de 80 sendo possível definir como sendo uma contração. Deste modo, foi reconhecido que os dados de entrada possuem uma variação considerável para mesma ativação muscular, então foram atribuídos resultados com valores semelhantes em diferentes leituras da ativação do músculo. Um conjunto de 34 amostras foi definido, onde estas foram divididas em 5 áreas significativas, onde indiscutivelmente ocorre uma transição no estado do músculo. Desta forma, foi definido que áreas muito próximas e abaixo do valor 40 devem ser consideradas como uma desativação, enquanto áreas muito próximas ou superiores a 80 serão consideradas como uma ativação máxima, e a área intermediária possui as demais áreas, com uma variação linear entre as mesmas.

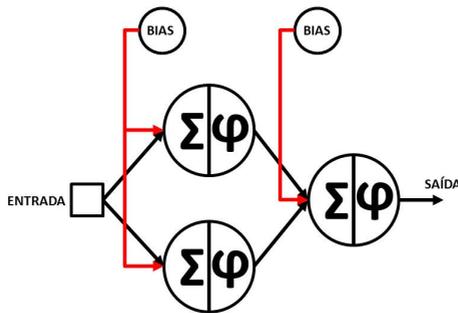
Os valores utilizados para o conjunto de treinamento foram definidos entre zero e um, e que o resultado irá inferir na ativação do servo

motor, tendo 0 para a mínima e 1 para a máxima respectivamente. Valores próximos de 1 são considerados mais próximos da ativação máxima, enquanto valores próximos do 0, são definidos como ativação mínima ou nula. O conjunto de treinamento utilizado na RNA está listado no Apêndice I.

4.3 ARQUITETURA DA RNA IMPLEMENTADA

A RNA proposta é do tipo Multilayer Perceptron com uma entrada, proveniente do sensor EMG, dois neurônios na camada oculta, e um neurônio na camada de saída. Este formato foi definido a partir de um programa desenvolvido na linguagem de programação C, de uma RNA que alcançasse valores satisfatórios conforme o conjunto de treinamento. Com uma quantidade de neurônios maior e mais camadas é possível conseguir o mesmo resultado e até mesmo mais precisão, contudo foi escolhido reduzir o tamanho da RNA, onde dessa forma ela ainda atingiu resultados satisfatórios, mesmo com camadas e neurônios reduzidos. Esta escolha foi tomada para facilitar a transição do código para o dispositivo FPGA, tendo em vista que a diferença dos resultados mais precisos comparado com os do modelo de dois neurônios na camada oculta é ínfima para aplicação desejada, tornando o menor modelo ainda mais atraente. A Figura 36 ilustra o formato da RNA projetada.

Figura 36 – Modelo da RNA proposta



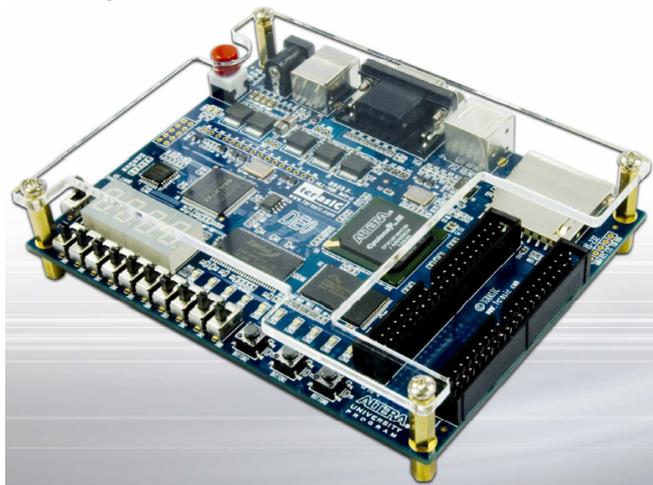
Fonte: Elaborada pelo autor.

O algoritmo de treinamento utilizado foi o *backpropagation*, descrito nas Seções 3.5.1 e 3.6. A função de ativação utilizada para todos os neurônios foi a sigmoide. Os dados necessários para a programação da RNA estão descritos no Apêndice I.

4.4 PROJETO E DESENVOLVIMENTO DO HARDWARE

A FPGA escolhida para a execução do projeto foi uma Cyclone III - EP3C16F484C6N da Altera. A placa do dispositivo é a DE0, da Terasic, onde detalhes sobre ela e seu manual do usuário podem ser encontrados no site: <https://www.altera.com/solutions/partners/partner-profile/terasic-inc-/board/altera-de0-board.html>. A Figura 37 ilustra o dispositivo utilizado.

Figura 37 – Placa de desenvolvimento DE0.



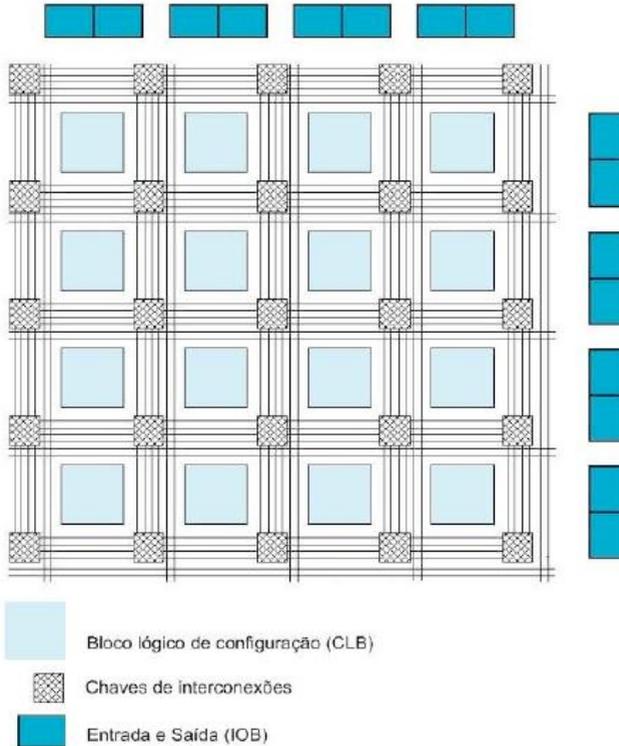
Fonte: Adaptada do manual do usuário da DE0.

A FPGA é um dispositivo de descrição de hardware, o qual permite sintetizar projetos de hardware através da linguagem de descrição de hardware *Very High Speed Integrated Circuits Hardware Description Language*(VHDL) (D'AMORE, 2012). O ambiente de desenvolvimento utilizado foi o Quartus II da Altera, versão 13.1, última versão a qual possui suporte ao modelo de placa utilizado.

O dispositivo FPGA foi escolhido devido aos blocos de processa-

mento programáveis paralelos. Devido a este formato, foi possível desenvolver um procedimento em três processos diferentes, onde cada procedimento é um neurônio atuando. O código desenvolvido em VHDL está disponível no Apêndice II. A Figura 38 ilustra o princípio da FPGA.

Figura 38 – Exemplo da estrutura de uma FPGA

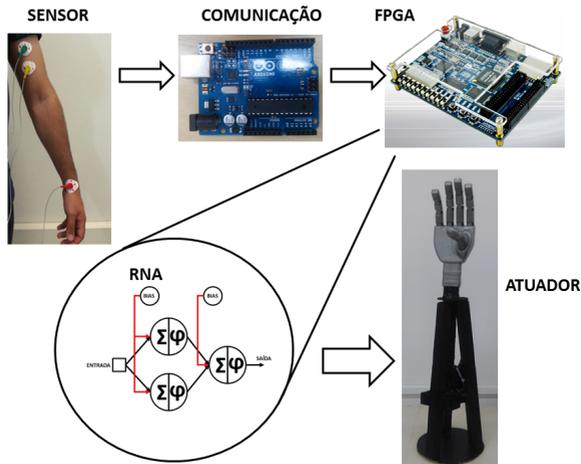


Fonte: Adaptada de (SILVA, 2010b).

Como mencionado na Seção 4.2 foi necessário o uso de um Arduino para a conversão analógico-digital do sinal capturado, pois a FPGA não dispõe de determinado conversor. Deste modo foi necessário desenvolver um protocolo de comunicação entre o Arduino e a FPGA. Como a placa DE0 não comporta tensões acima de 3.3 volts, foi necessário fazer o uso de divisores de tensão para manter a comunicação entre os dois dispositivos compatível.

A Figura 39 ilustra a visão geral da prótese desenvolvida..

Figura 39 – Esquemático do projeto da prótese mioelétrica.



Fonte: Elaborada pelo autor.

A prótese desenvolvida conta com um servo motor, modelo MG995 com 10kgcm de torque (Figura 40), sendo o atuador do sistema, responsável por operar a prótese.

Figura 40 – Servo motor utilizado para atuar no sistema.



Fonte: Adaptada do datasheet do MG995.

O motor é acoplado através de um arame à mão a qual foi retirada de um brinquedo, mostrado na Figura 41.

Figura 41 – Brinquedo utilizado para prótese.



Fonte: Elaborada pelo autor.

Para a alimentação do motor foi considerado utilizar um conjunto de quatro pilhas AA conectadas em série totalizando 6 volts necessários para alimentação do servo motor, porém esta ideia foi descartada por não fornecer a carga necessária para mantê-lo operante. A alternativa encontrada foi a utilização de uma fonte de tensão, modelo FA-3003 com um canal, tensão até 32 volts e corrente de até 3 ampéres, da Instrutherm. Com esta fonte de tensão foi possível garantir que o atuador estava devidamente alimentado, e assim focar na funcionalidade propriamente dita do sistema. A fonte de tensão utilizada está ilustrada na Figura 42.

Figura 42 – Fonte de tensão utilizada.



Fonte: Elaborada pelo autor.

A base e a estrutura da prótese foram construídas com madeira e compensado MDF, de forma a manter uma estrutura rígida, porém leve. Três hastes de madeira foram utilizadas e parafusadas entre as bases. Uma base foi desenvolvida para suportar o servo motor suspenso, o qual seu eixo por sua vez está conectado a um arame que executa o movimento da prótese. Foi parafusado na estrutura três suportes que ficam abaixo da base do motor. A base então é parafusada a estes suportes com o objetivo de reter a posição do motor quando ele atuar

sobre o sistema, para que o mesmo não seja puxado ao invés de puxar conforme ilustrado na Figura 43.

Figura 43 – Suporte para a base.



Fonte: Elaborada pelo autor.

Por fim a prótese possui um cone plástico transparente que permite ver o servo motor operando, porém também serve para proteção do sistema. A Figura 44 ilustra o modelo completo do protótipo.

Figura 44 – Protótipo da prótese.

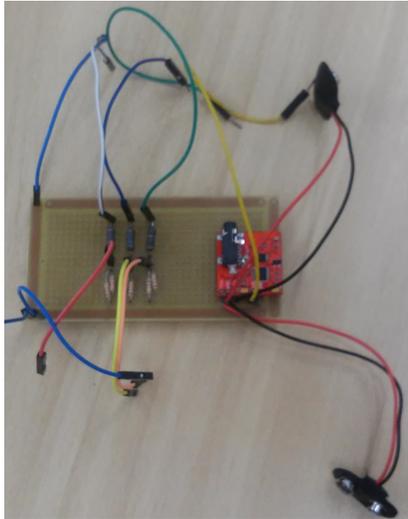


Fonte: Elaborada pelo autor.

As conexões entre os dispositivos foram realizadas através de jumpers e protoboard, porém este formato gerava muitos casos de mal-contato e tornavam o processo de conexão do sistema pouco intuitivo. Para isso os componentes foram retirados e soldados em uma placa modelo ilhas onde todas as conexões foram soldadas. Deste modo, foram

eliminados os mal-contatos e tornaram o circuito como um todo esteticamente mais agradável. A Figura 45 apresenta a placa desenvolvida no modelo ilhas.

Figura 45 – Placa desenvolvida no modelo Ilhas.



Fonte: Elaborada pelo autor.

O sistema completo é demonstrado na Figura 46, onde é possível observar a conexão das placas ao computador, porém a mesma é feita com a única finalidade de alimentar os dispositivos, podendo ser substituídos por qualquer fonte de energia apropriada.

Figura 46 – Conjunto de hardware completo do sistema.



Fonte: Elaborada pelo autor.

A madeira da prótese foi pintada com tinta acrílica da cor preta, para fins meramente estéticos. O resultado final é apresentado na Figura 47.

Figura 47 – Conjunto de hardware completo do sistema.



Fonte: Elaborada pelo autor.

5 AVALIAÇÃO DA PRÓTESE MIOELÉTRICA

Este capítulo descreve os experimentos realizados com a prótese mioelétrica desenvolvida.

5.1 METODOLOGIA EMPREGADA NA AVALIAÇÃO DA PRÓTESE DESENVOLVIDA

Foram utilizados quatro tipos de experimentos, um lápis adaptado, um copo plástico pequeno, um frasco de desodorante e uma bola de borracha. Devido ao tamanho da mão do protótipo, os itens foram escolhidos e adaptados para terem dimensões próprias para que a prótese possa agarrá-los com uma preensão palmar.

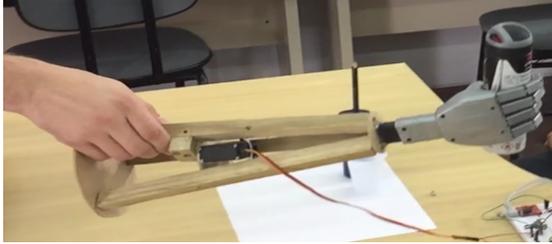
O objetivo dos experimentos foi observar o manuseio da prótese com os itens, analisando se há um controle da força aplicada, consistência na movimentação e verificar os limites e restrições que este modelo possa apresentar. Foram utilizados dois voluntários para realizar o experimento. Um deles responsável por emitir os sinais mioelétricos através da contração do bíceps, controlando a atuação da prótese. O segundo voluntário responsável por manusear a prótese, move-la para capturar e deslocar os itens. Foi optado em utilizar dois voluntários devido a testes realizados anteriormente com um voluntário para realizar ambas tarefas, onde foi possível observar que uma trepidação no sinal, gerando movimentos semelhantes a espasmos, com constante troca em abrir e fechar da mão. O movimento do voluntário causou interferência no sinal mioelétrico capturado, o que resultou em atuações indesejáveis da prótese.

5.2 EXPERIMENTO 1 - MOVER UM FRASCO DE DESODORANTE VAZIO

Neste experimento foi avaliado a capacidade da prótese em segurar um objeto, levantá-lo e deslocá-lo. O objeto utilizado foi um frasco vazio de desodorante. A distância de deslocamento do frasco foi cerca de 50 centímetros, e a altura de suspensão do objeto foi cerca de 20 centímetros. Estas medidas foram utilizadas em virtude do espaço disponível na superfície da mesa utilizada e devido as conexões necessárias entre os dispositivos, mas não são necessariamente os limites de alcance

da prótese. A Figura 48 ilustra o experimento realizado.

Figura 48 – Experimento 1 - Prótese movendo frasco de desodorante vazio.



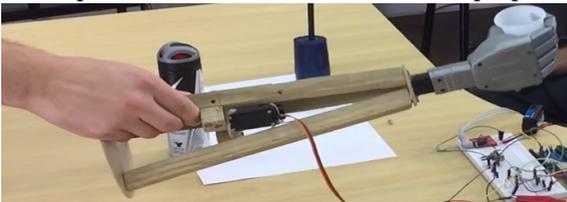
Fonte: Elaborada pelo autor.

O voluntário um operou uma prensão máxíma da prótese para segurar o objeto e liberou-o no momento desejado, e o voluntário dois movimentou o objeto com a prótese da posição de origem ao destino sem dificuldades.

5.3 EXPERIMENTO 2 - MOVER UM COPO PLÁSTICO VAZIO

O segundo experimento foi realizado com um copo plástico vazio, de tamanho médio. O processo de deslocamento foi exatamente o mesmo do primeiro experimento, contudo desta vez o voluntário um gerou uma contração inferior a primeira, para capturar o copo sem amassá-lo. A Figura 49 ilustra o experimento realizado.

Figura 49 – Experimento 2 - Prótese movendo copo plástico vazio.



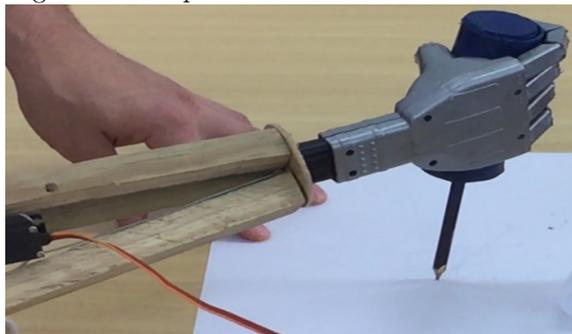
Fonte: Elaborada pelo autor.

O voluntário dois direcionou a prótese e movimentou-a pela trajetória sem dificuldades. O copo foi capturado, deslocado e solto de maneira correta e sem danos por esmagamento.

5.4 EXPERIMENTO 3 - PRÓTESE UTILIZANDO UM LÁPIS

O terceiro experimento foi realizado com o lápis adaptado. Devido ao tamanho do lápis, comparado com o movimento de preensão palmar, foi necessário adapta-lo, o acoplado a um cilindro de madeira, coberto por um tubo de PVC. Neste experimento foi analisado também a quantidade de tempo que ela permaneceu segurando o lápis durante o processo de escrita. A Figura 50 ilustra o experimento realizado.

Figura 50 – Experimento 3 - Prótese escrevendo.



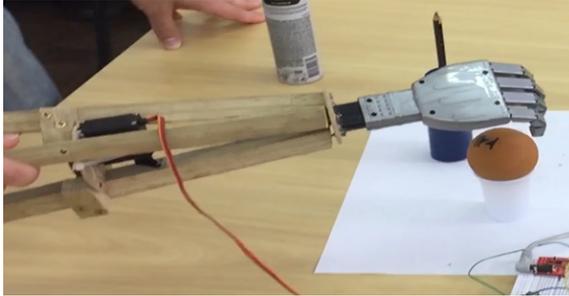
Fonte: Elaborada pelo autor.

O voluntário um capturou-o com uma contração máxima, que permitiu o voluntário dois movimenta-lá e utiliza-lá para rabiscar em uma folha. A prótese segurou o lápis firmemente por um período de 25 segundos, e após este tempo acabou soltando o objeto.

5.5 EXPERIMENTO 4 - CAPTURA DE UMA BOLA DE BORRACHA

A última etapa realizada foi com uma bola de borracha. A captura da bola não foi concluída nas tentativas realizadas, conforme ilustrado na Figura 51.

Figura 51 – Experimento 4 - Tentando segurar uma bola de borracha.



Fonte: Elaborada pelo autor.

5.6 CONSIDERAÇÕES SOBRE OS EXPERIMENTOS

Os experimentos realizados com os quatro diferentes tipos de objetos resultaram em três etapas de sucesso e uma captura não concluída. Julga-se a não conclusão do último experimento devido a prótese ter sido desenvolvida com base no brinquedo o qual não possui um formato adequado para captura da bola, uma vez que seu polegar a empurrava para fora da preensão ao invés de fornecer suporte. Entretanto, esta etapa não necessariamente inválida a operação do sistema, uma vez que os demais objetos com formato mais próprio para a preensão palmar realizada pelo brinquedo operaram corretamente sem dificuldades. O ponto mais interessante de se observar foi a segunda etapa com o copo plástico, o que permite visualizar com mais clareza a atuação da RNA, gerando uma preensão proporcional a intenção do voluntário, permitindo-o manusear um objeto relativamente frágil sem danificá-lo. Pontos onde podem ocorrer melhorias são visíveis após a realização do experimento, como um melhor tratamento dos dados de entrada e melhor projeto mecânico da prótese. Estes e outros detalhes serão expostos com maior afinco na conclusão.

6 CONSIDERAÇÕES FINAIS

A partir dos resultados obtidos foi possível validar o funcionamento do sistema de controle para prótese mioelétrica com controle baseado em uma RNA implementada em um dispositivo de FPGA. A prótese desenvolvida obteve um desempenho satisfatório, cumprindo maior parte das tarefas propostas conforme o esperado. Embora complexo, a FPGA demonstrou ser um dispositivo bastante apropriado para o uso de RNA, operando neurônios de forma paralela, levantando questionamento de qual seria o limiar de sua capacidade de processamento. Um revés no método escolhido foi a necessidade de discretização da função de ativação, que diminui em muito a precisão do sistema. Felizmente, para o sistema proposto, o divergência na precisão não influenciou em muito no resultado final, uma vez que esta diferença gera um desvio ínfimo na posição de fechamento da mão. Foi possível também reconhecer a dificuldade em um certo nível, do processo de pré e pós protetização, onde leves movimentos, mesmo com diferentes grupos musculares, posicionamento, repetições de movimento e fadiga muscular, todos estes fatores influenciam no sinal de ativação do músculo, o que afeta diretamente o funcionamento do sistema.

6.1 PROPOSTA PARA TRABALHOS FUTUROS

Diversas melhorias podem ser acrescentadas ao sistema. Sendo que é possível destacar:

- O uso de um sistema Neuro-Fuzzy, para o melhor tratamento do sinal mioelétrico recebido do músculo, eliminando as oscilações do sinal;
- Desenvolvimento de uma realimentação para o sistema, que permita reconhecer a força sendo aplicada, obtendo maior eficiência no tratamento dos erros;
- Agregar um projeto de prótese mais sofisticado, com um número maior de grau de liberdade, juntamente com ativação de mais grupos musculares;
- Desenvolver novas soluções e comparar a eficiência entre as mesmas.

7 APÊNDICE I

Neste apêndice são listados os dados utilizados para o treinamento da Rede Neural Artificial (RNA) que controla a prótese mioelétrica. O apêndice também descreve os parâmetros utilizados para configurar a RNA.

7.1 CONJUNTO DE TREINAMENTO

Entrada	Saída
0	0.000
7	0.040
14	0.080
33	0.120
36	0.160
39	0.310
41	0.368
43	0.426
45	0.470
46	0.485
49	0.510
52	0.535
57	0.560
64	0.700
66	0.725
68	0.750
70	0.800
72	0.888
75	0.894
78	0.900
85	0.906
92	0.912
100	0.918
115	0.924
125	0.930
130	0.936
138	0.942
146	0.948
155	0.954
160	0.960
168	0.966
172	0.972
179	0.978
186	0.984

7.2 PARÂMETROS DE CONFIGURAÇÃO DA RNA

Os dados necessários para treinamento da RNA.

- Valor de bias : 1;
- Taxa de aprendizado: 1;
- Número de épocas de treinamento: 10000;
- Número de amostras: 34;
- Número de entradas: 1;
- Número de camadas ocultas: 1;
- Número de neurônios da camada oculta: 2;
- Número de saídas: 1.

8 APÊNDICE II

Neste apêndice é listado o código em VHDL para implementação da RNA em FPGA.

8.1 BASE_RNA.VHD

```

library ieee;
library ieee_proposed;

use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use ieee_proposed.fixed_pkg.all;

package base_rna is
----- Procedimento Neuronio -----
procedure neuronio(constant id: in integer;
signal entrada1,entrada2: in sfixed(11 downto -10);
signal saida: out sfixed(10 downto -14)) ;
end base_rna;

package body base_rna is

procedure neuronio(variable id: in integer;
signal entrada1,entrada2: in sfixed(11 downto -10);
  signal saida: out sfixed(11 downto -10)) is

type vetor is array (0 to 6) of sfixed(7 downto -14);
variable net: sfixed(21 downto -24);
variable mult1, mult2: sfixed(19 downto -24);

----- Vetor de Pesos Sinápticos -----

```

```

variable pesos_sinapticos : vetor :=
(to_sfixed(53.625857,7, -14), --w1n1
to_sfixed(-2.392972,7, -14),--bias1
to_sfixed(-18.139395,7, -14),--w1n2
to_sfixed(-0.711901,7, -14),--bias2
to_sfixed(5.494640,7, -14),--w1n3
to_sfixed(-5.918550,7, -14),--w2n3
to_sfixed(-2.120000,7, -14));--bias3

```

----- Realização da Multiplicação e Soma -----

```
begin
```

```
--De acordo com o id é definido o neurônio
```

```
if(id = 0)then
```

```
mult1 := entrada1*pesos_sinapticos(0);
```

```
net := mult1 + pesos_sinapticos(1)+0;
```

```
elsif (id = 1)then
```

```
mult1 := entrada1*pesos_sinapticos(2);
```

```
net := mult1 + pesos_sinapticos(3)+0;
```

```
elsif (id = 2)then
```

```
mult1 := entrada1*pesos_sinapticos(4);
```

```
mult2 := entrada2*pesos_sinapticos(5);
```

```
net := mult1 + mult2 + pesos_sinapticos(6);
```

```
--net := resize(entrada1*pesos_sinapticos(6)
```

```
+entrada2*pesos_sinapticos(7)
```

```
+pesos_sinapticos(8),21, -24);
```

```
end if;
```

----- Aplica Resultado na Função Sigmoide Discretizada -----

```
if (net < to_sfixed(-4.1,21, -24)) then
```

```
saida <= to_sfixed(0.01,11, -10);
```

```
elsif(net >= to_sfixed(-4.1,21, -24) and
```

```
net < to_sfixed(-3.6,21, -24)) then
```

```
saida <= to_sfixed(0.02,11, -10);

elseif(net >= to_sfixed(-3.6,21, -24) and
net < to_sfixed(-3.3,21, -24)) then

saida <= to_sfixed(0.03,11, -10);

elseif(net >= to_sfixed(-3.3,21, -24) and
net < to_sfixed(-3.0,21, -24)) then

saida <= to_sfixed(0.04,11, -10);

elseif(net >= to_sfixed(-3.0,21, -24) and
net < to_sfixed(-2.8,21, -24)) then

saida <= to_sfixed(0.05,11, -10);

elseif(net >= to_sfixed(-2.8,21, -24) and
net < to_sfixed(-2.6,21, -24)) then

saida <= to_sfixed(0.06,11, -10);

elseif(net >= to_sfixed(-2.6,21, -24) and
net < to_sfixed(-2.5,21, -24)) then

saida <= to_sfixed(0.07,11, -10);

elseif(net >= to_sfixed(-2.5,21, -24) and
net < to_sfixed(-2.3,21, -24)) then

saida <= to_sfixed(0.08,11, -10);

elseif(net >= to_sfixed(-2.3,21, -24) and
net < to_sfixed(-2.2,21, -24)) then

saida <= to_sfixed(0.09,11, -10);

elseif(net >= to_sfixed(-2.2,21, -24) and
net < to_sfixed(-2.1,21, -24)) then

saida <= to_sfixed(0.10,11, -10);
```

```
elseif(net >= to_sfixed(-2.1,21, -24) and
net < to_sfixed(-2.0,21, -24)) then

saida <= to_sfixed(0.11,11, -10);

elseif(net >= to_sfixed(-2.0,21, -24) and
net < to_sfixed(-1.9,21, -24)) then

saida <= to_sfixed(0.12,11, -10);

elseif(net >= to_sfixed(-1.9,21, -24) and
net < to_sfixed(-1.8,21, -24)) then

saida <= to_sfixed(0.13,11, -10);

elseif(net >= to_sfixed(-1.8,21, -24) and
net < to_sfixed(-1.7,21, -24)) then

saida <= to_sfixed(0.14,11, -10);

elseif(net >= to_sfixed(-1.7,11, -10) and
net < to_sfixed(-1.6,21, -24)) then

saida <= to_sfixed(0.15,11, -10);

elseif(net >= to_sfixed(-1.6,21, -24) and
net < to_sfixed(-1.5,21, -24)) then

saida <= to_sfixed(0.17,11, -10);

elseif(net >= to_sfixed(-1.5,21, -24) and
net < to_sfixed(-1.4,21, -24)) then

saida <= to_sfixed(0.18,11, -10);

elseif(net >= to_sfixed(-1.4,21, -24) and
net < to_sfixed(-1.3,21, -24)) then

saida <= to_sfixed(0.20,11, -10);
```

```
elseif(net >= to_sfixed(-1.3,21, -24) and
net < to_sfixed(-1.2,21, -24)) then

saida <= to_sfixed(0.21,11, -10);

elseif(net >= to_sfixed(-1.2,21, -24) and
net < to_sfixed(-1.1,21, -24)) then

saida <= to_sfixed(0.23,11, -10);

elseif(net >= to_sfixed(-1.1,21, -24) and
net < to_sfixed(-1.0,21, -24)) then

saida <= to_sfixed(0.25,11, -10);

elseif(net >= to_sfixed(-1.0,21, -24) and
net < to_sfixed(-0.9,21, -24)) then

saida <= to_sfixed(0.27,11, -10);

elseif(net >= to_sfixed(-0.9,21, -24) and
net < to_sfixed(-0.8,21, -24)) then

saida <= to_sfixed(0.29,11, -10);

elseif(net >= to_sfixed(-0.8,21, -24) and
net < to_sfixed(-0.7,21, -24)) then

saida <= to_sfixed(0.31,11, -10);

elseif(net >= to_sfixed(-0.7,21, -24) and
net < to_sfixed(-0.6,21, -24)) then

saida <= to_sfixed(0.33,11, -10);

elseif(net >= to_sfixed(-0.6,21, -24) and
net < to_sfixed(-0.5,21, -24)) then

saida <= to_sfixed(0.35,11, -10);

elseif(net >= to_sfixed(-0.5,21, -24) and
```

```
net < to_sfixed(-0.4,21, -24)) then

saida <= to_sfixed(0.38,11, -10);

elseif(net >= to_sfixed(-0.4,21, -24) and
net < to_sfixed(-0.3,21, -24)) then

saida <= to_sfixed(0.40,11, -10);

elseif(net >= to_sfixed(-0.3,21, -24) and
net < to_sfixed(-0.2,21, -24)) then

saida <= to_sfixed(0.43,11, -10);

elseif(net >= to_sfixed(-0.2,21, -24) and
net < to_sfixed(-0.1,21, -24)) then

saida <= to_sfixed(0.45,11, -10);

elseif(net >= to_sfixed(-0.1,21, -24) and
net < to_sfixed(0.0,21, -24)) then

saida <= to_sfixed(0.48,11, -10);

elseif(net >= to_sfixed(0.0,21, -24) and
net < to_sfixed(0.1,21, -24)) then

saida <= to_sfixed(0.5,11, -10);

elseif(net >= to_sfixed(0.1,21, -24) and
net < to_sfixed(0.2,21, -24)) then

saida <= to_sfixed(0.52,11, -10);

elseif(net >= to_sfixed(0.2,21, -24) and
net < to_sfixed(0.3,21, -24)) then

saida <= to_sfixed(0.55,11, -10);

elseif(net >= to_sfixed(0.3,21, -24) and
net < to_sfixed(0.4,21, -24)) then
```

```
saida <= to_sfixed(0.57,11, -10);

elsif(net >= to_sfixed(0.4,21, -24) and
net < to_sfixed(0.5,21, -24)) then

saida <= to_sfixed(0.60,11, -10);

elsif(net >= to_sfixed(0.5,21, -24) and
net < to_sfixed(0.6,21, -24)) then

saida <= to_sfixed(0.62,11, -10);

elsif(net >= to_sfixed(0.6,21, -24) and
net < to_sfixed(0.7,21, -24)) then

saida <= to_sfixed(0.65,11, -10);

elsif(net >= to_sfixed(0.7,21, -24) and
net < to_sfixed(0.9,21, -24)) then

saida <= to_sfixed(0.70,11, -10);

elsif(net >= to_sfixed(0.9,21, -24) and
net < to_sfixed(1.0,21, -24)) then

saida <= to_sfixed(0.71,11, -10);

elsif(net >= to_sfixed(1.0,21, -24) and
net < to_sfixed(1.1,21, -24)) then

saida <= to_sfixed(0.73,11, -10);

elsif(net >= to_sfixed(1.1,21, -24) and
net < to_sfixed(1.2,21, -24)) then

saida <= to_sfixed(0.75,11, -10);

elsif(net >= to_sfixed(1.2,21, -24) and
net < to_sfixed(1.3,21, -24)) then
```

```
saida <= to_sfixed(0.77,11, -10);

elseif(net >= to_sfixed(1.3,21, -24) and
net < to_sfixed(1.4,21, -24)) then

saida <= to_sfixed(0.78,11, -10);

elseif(net >= to_sfixed(1.4,21, -24) and
net < to_sfixed(1.5,21, -24)) then

saida <= to_sfixed(0.80,11, -10);

elseif(net >= to_sfixed(1.5,21, -24) and
net < to_sfixed(1.6,21, -24)) then

saida <= to_sfixed(0.82,11, -10);

elseif(net >= to_sfixed(1.6,21, -24) and
net < to_sfixed(1.7,21, -24)) then

saida <= to_sfixed(0.83,11, -10);

elseif(net >= to_sfixed(1.7,21, -24) and
net < to_sfixed(1.8,21, -24)) then

saida <= to_sfixed(0.85,11, -10);

elseif(net >= to_sfixed(1.8,21, -24) and
net < to_sfixed(1.9,21, -24)) then

saida <= to_sfixed(0.86,11, -10);

elseif(net >= to_sfixed(1.9,21, -24) and
net < to_sfixed(2.0,21, -24)) then

saida <= to_sfixed(0.87,11, -10);

elseif(net >= to_sfixed(2.0,21, -24) and
net < to_sfixed(2.1,21, -24)) then

saida <= to_sfixed(0.88,11, -10);
```

```
elseif(net >= to_sfixed(2.1,21, -24) and
net < to_sfixed(2.2,21, -24)) then

saida <= to_sfixed(0.89,11, -10);

elseif(net >= to_sfixed(2.2,21, -24) and
net < to_sfixed(2.3,21, -24)) then

saida <= to_sfixed(0.90,11, -10);

elseif(net >= to_sfixed(2.3,21, -24) and
net < to_sfixed(2.4,21, -24)) then

saida <= to_sfixed(0.91,11, -10);

elseif(net >= to_sfixed(2.4,21, -24) and
net < to_sfixed(2.6,21, -24)) then

saida <= to_sfixed(0.92,11, -10);

elseif(net >= to_sfixed(2.6,21, -24) and
net < to_sfixed(2.7,21, -24)) then

saida <= to_sfixed(0.93,11, -10);

elseif(net >= to_sfixed(2.7,21, -24) and
net < to_sfixed(2.9,21, -24)) then

saida <= to_sfixed(0.94,11, -10);

elseif(net >= to_sfixed(2.9,21, -24) and
net < to_sfixed(3.1,21, -24)) then

saida <= to_sfixed(0.95,11, -10);

elseif(net >= to_sfixed(3.1,21, -24) and
net < to_sfixed(3.4,21, -24)) then

saida <= to_sfixed(0.96,11, -10);
```

```
elseif(net >= to_sfixed(3.4,21, -24) and
net < to_sfixed(3.7,21, -24)) then

saida <= to_sfixed(0.97,11, -10);

elseif(net >= to_sfixed(3.7,11, -10) and
net < to_sfixed(4.3,11, -10)) then

saida <= to_sfixed(0.98,11, -10);

elseif(net >= to_sfixed(4.3,11, -10)) then

saida <= to_sfixed(0.99,11, -10);

end if;

end neuronio;
end package body;
```

8.2 RNA_PROTESE.VHD

```
library ieee;
library ieee_proposed;

use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use ieee_proposed.fixed_pkg.all;
use work.base_rna.all;

entity RNA_PROTESE is
port( CLK: in std_logic;
DT: in std_logic;
START: in std_logic;
clkServo: in std_logic;
sigServo: out std_logic);
end RNA_PROTESE;
```

architecture hardware of RNA_PROTESE is

```

signal dado: std_logic_vector (9 downto 0);
signal ii: integer range 0 to 9 :=0;
signal entrada: sfixed(10 downto 0);
signal estouro: integer range 50000 to 100000;
signal entrada_discretizada, sn1, sn2, saida:
  sfixed(11 downto -10);

begin
----- Comunicação com Arduino -----
process(CLK)
begin
if(rising_edge(CLK)) then
if(START = '1')then
ii <= 0;
entrada <= to_sfixed(to_integer(unsigned(dado)),entrada);
entrada_discretizada <= entrada / to_sfixed(1023, 10, 0);
else
dado(ii)<= DT;
ii <= ii + 1;
end if;
end if;
end process;
----- RNA -----
N1: process(entrada_discretizada) -- Neuronio 1 da oculta
begin
neuronio(0,entrada_discretizada,entrada_discretizada,sn1);
end process;
N2: process(entrada_discretizada) -- Neuronio 2 da oculta
begin
neuronio(1,entrada_discretizada,entrada_discretizada,sn2);
end process;
NS: process(sn1, sn2) -- Neuronio da saída
begin
neuronio(2,sn1,sn2,saida);
end process;

----- Controle Servo Motor -----

```

```
AtualizaEstouro: process(saida)

begin
estouro <= to_integer(to_sfixed(50000.0,16,0)*saida +
  to_sfixed(50000.0,16,0));

end process;

Servo: process(clkServo)
variable prescaler : integer range 0 to 1000000;
begin

if(rising_edge(clkServo))then
prescaler := prescaler + 1;
end if;

if(prescaler = 0) then
sigServo <= '1';
end if;

if(prescaler = estouro)then
sigServo <= '0';
end if;
end process;

end hardware;
```

REFERÊNCIAS

- BALBINOT, A.; FAVIEIRO, G. A neuro-fuzzy system for characterization of arm movements. **Sensors**, MDPI AG, v. 13, n. 2, p. 2613–2630, feb 2013. Disponível em: <<http://dx.doi.org/10.3390/s130202613>>.
- BARROS, K. R. d. Metodologia para classificação de sinais emg para controle de próteses com baixo esforço computacional. 2005.
- BRAGA-SILVA, J. [single digit replantations in ambulatory surgery. 85 cases]. In: **Annales de chirurgie plastique et esthetique**. [S.l.: s.n.], 2001. v. 46, n. 2, p. 74–83.
- BU, N. et al. Fpga implementation of a probabilistic neural network for a bioelectric human interface. In: IEEE. **Circuits and Systems, 2004. MWSCAS'04. The 2004 47th Midwest Symposium on**. [S.l.], 2004. v. 3, p. iii–29.
- CAMARGO, D. R. d. **Desenvolvimento do protótipo de uma prótese antropomórfica para membros superiores**. Tese (Doutorado) — Universidade de São Paulo, 2008.
- D'AMORE, R. **VHDL-Descrição e Síntese de Circuitos Digitais, 2 Edição**. [S.l.: s.n.], 2012.
- FAVIEIRO, G. W. Controle de uma prótese experimental do segmento mão-braço por sinais mioelétricos e redes neurais artificiais. 2009.
- FAVIEIRO, G. W.; BALBINOT, A.; BARRETO, M. M. Decoding arm movements by myoelectric signals and artificial neural networks. In: IEEE. **ISSNIP Biosignals and Biorobotics Conference 2011**. [S.l.], 2011. p. 1–6.
- FONSECA, M. C. R. et al. **Órteses e próteses: indicação e tratamento**. [S.l.]: Águia Dourada, 2015.
- HALL, S. J. **Biomecânica Básica**. [S.l.: s.n.], 2013.
- HAMILL, J.; KNUTZEN, K. M. **Bases Biomecânicas do Movimento Humano**. [S.l.: s.n.], 2012.
- HAYKIN, S. **Redes Neurais—Princípios e Prática**. [S.l.]: Porto Alegre: Bookman, 2001.

JUNG, S.; KIM, S. su. Hardware implementation of a real-time neural network controller with a dsp and an fpga for nonlinear systems.

IEEE Transactions on Industrial Electronics, IEEE, v. 54, n. 1, p. 265–271, 2007.

JÚNIOR, C. L. N.; YONEYAMA, T. **Inteligência artificial: em controle e automação**. [S.l.]: Edgard Blucher, 2002.

KORKMAZ, M. et al. Evaluation of functionality in acquired and congenital upper extremity child amputees. **Acta orthopaedica et traumatologica turcica**, v. 46, n. 4, p. 262–268, 2012.

LUCKS, M. B. Proposta de implementação de redes de base radial em tecnologias cmos e bicos. Universidade Estadual Paulista (UNESP), 2009.

LUSARDI, M. M. et al. Orthotics and prosthetics in rehabilitation 2nd ed. 2007.

MARCOLINO, A. M. et al. Amputações e próteses do membro superior. In: **Órteses e próteses: indicação e tratamento**. [S.l.]: Águia Dourada, 2015. cap. 6, p. 103–126.

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

PARDINI, P. F. **Reabilitação da mão**. [S.l.]: Atheneu, 2005.

POLIS, J. E. et al. Projeto e construção de parte estrutural de protese de mão humana com movimentos. Campinas, SP, 2009.

ROSA, J. L. G. **Fundamentos da inteligência artificial**. [S.l.: s.n.], 2011.

ROSSALES, I. R. d. P. et al. Implementação em hardware de uma rede neural wisard. UNIVERSIDADE DE SÃO PAULO, 2012.

SILVA, C. A. d. A. Contribuição para o estudo do embarque de uma rede neural artificial em field programmable gate array (fpga). Universidade Federal do Rio Grande do Norte, 2010.

SILVA, C. A. d. A. Contribuição para o estudo do embarque de uma rede neural artificial em field programmable gate array (fpga). Universidade Federal do Rio Grande do Norte, 2010.

ZLOTOW, D. A.; KOZIN, S. H. Advances in upper extremity prosthetics. **Hand clinics**, Elsevier, v. 28, n. 4, p. 587–593, 2012.