



TRABALHO DE GRADUAÇÃO

**METODOLOGIA PARA CONCEPÇÃO
DE PRÓTESE ATIVA DE MÃO
UTILIZANDO IMPRESSORA 3D**

Jeferson Andris Lima Lopes

Lucas Coelho Almeida

Brasília, dezembro de 2013

UNIVERSIDADE DE BRASÍLIA



UNIVERSIDADE DE BRASÍLIA
Faculdade de Engenharia do Gama

TRABALHO DE GRADUAÇÃO

**METODOLOGIA PARA CONCEPÇÃO
DE PRÓTESE ATIVA DE MÃO
UTILIZANDO IMPRESSORA 3D**

Jeferson Andris Lima Lopes

Lucas Coelho Almeida

*Monografia submetida ao curso de Engenharia Eletrônica
da Universidade de Brasília como requisito parcial
para obtenção do Título de Bacharel em
Engenharia Eletônica*

Banca Examinadora

Prof. Marcelino Monteiro de Andrade, FGA/UnB
Orientador

Prof. Fabiano Araujo Soares, FGA/UnB
Examinador interno

Prof. Cristiano Jacques Miosso Rodrigues Mendes,
FGA/UNB
Examinador interno

RESUMO

No presente trabalho de Conclusão de Curso, os autores apresentam uma prótese ativa de mão com um grau de liberdade e impressa com materiais rígido e flexível. Para isso, utilizou-se uma impressora 3D de baixo custo do modelo Prusa Mendel. Desenvolveu-se toda a metodologia de prototipagem da prótese, desde sua modelagem, os parâmetros utilizados nos softwares de impressão, até sua instrumentação e o controle.

A modelagem da prótese foi feita utilizando os *softwares* CATIA e Sketch Up. Levou-se em conta os limites de impressão de 200x200x65mm impostos pela tecnologia da impressora 3D utilizada. A prótese foi impressa em duas etapas, sendo uma delas para impressão da mão robóticas e outra para o suporte e as polias. A mão robótica foi desenvolvida de modo a permitir o movimento de uma pinça que foi possível graças à combinação entre o plástico rígido e o flexível, o suporte foi projetado para o encaixe da mão robótica e os servomotores, já as polias, para serem acopladas aos braços dos servos e prender os fios que, quando tracionados, fecham a mão.

A prótese é controlada pelo sinal eletromiográfico de superfície (EMG-S) proveniente da contração muscular. Utilizou-se para a captação de tal sinal um eletrodo ativo revestido por uma capa impressa na impressora 3D e com conectores de prata no contato com a pele para diminuição do ruído. O circuito do eletrodo consiste em um amplificador de instrumentação (INA 118) e um filtro passa baixas para eliminar as componentes acima de 500 Hz. Externamente ao eletrodo, desenvolveu-se um circuito de conformação do sinal com mais um estágio de ganho, para ajuste em diferentes usuários, e um retificador inversor com detecção de envoltórias. O sinal de saída é conectado diretamente ao conversor A/D do microcontrolador, uma vez que ele não apresentará tensões negativas.

O microcontrolador utilizado foi o Arduino, com o objetivo de amostrar o sinal de EMG resultante do circuito de instrumentação e transformá-lo em um sinal de controle. Essa transformação consiste em gerar uma onda quadrada, onde cada contração é representada por uma janela retangular com a mesma duração. Na borda de subida desse sinal, o microcontrolador gera um sinal modulado por largura de pulso (PWM) para abrir ou fechar a prótese.

Foi proposto um protocolo para captação do sinal eletromiográfico, que vai desde a assepsia da pele até o posicionamento do eletrodo no músculo. A aquisição foi realizada com três voluntários, cada um com diferentes características e que efetuaram ou não o protocolo. Com o protocolo de aquisição do sinal observou-se um aumento da relação sinal ruído dos dados capturados. Contudo, o controle pôde ser realizado mesmo para sinais com significativa contaminação por ruído.

Por fim, os resultados mostraram que é viável a concepção da prótese, tanto na modelagem com materiais rígido e flexível, quanto ao seu controle utilizando o método de comparação de limiar fixo à envoltória do sinal de EMG-S capturado.

Palavras chaves: Próteses Mioelétricas, Sinal de Eletromiografia, Impressora 3D, Prusa Mendel, Prótese de Mão, Mão Robótica, Impressão Multimateriais;

ABSTRACT

In the present coursework conclusion, the authors present an active hand prosthesis with one degree of freedom that is printed with rigid and flexible materials. For this purpose, it was used a low cost 3D printer model named Prusa Mendel. It is presented all the prototyping methodology for the prosthesis, from its modeling, the parameters used in the printing software, to the instrumentation and control .

The modelling of the prosthesis was performed using the software CATIA and Sketch Up, taking into account the size limits of 200x200x65mm, imposed by the technology of the 3D printer model. The prosthesis was printed in two steps, one of them to print the robotic hand and other for the support and pulleys. The robotic hand was developed to provide the movement of a claw, which was possible by the combination of rigid and flexible plastic. The support was designed to fit the robotic hand and servo motors, pulleys were projected as a coupling to the servo arms and to hold the wires, that closes the hand when pulled.

The prosthesis is controlled by surface electromyographic signal (EMG - S), generated from muscle contraction. To capture that signal, an active electrode coated with a cover printed on 3D printer, and witch connectors in contact with the skin are made of silver for noise reduction. The electrode circuit consists of an instrumentation amplifier (INA 118) and a low-pass filter to remove components above 500 Hz. Externally to the electrode, it was developed another circuit with another gain stage, for adaptation to different users, and a inverting rectifier for signal envelope detection. The output signal is connected directly to the A/D converter of the micro-controller, since it does not present negative voltages.

The Arduino microcontroller was used in order to sample the EMG signal resulting from instrumentation circuit and turn it into a control signal. This transformation is to generate a square wave , where each contraction is represented by a rectangular window of the same duration . On the rising edge of this signal , the microcontroller generates a pulse width modulation (PWM) to open or close the prosthesis.

It was proposed a protocol to capture the electromyographic signal, ranging from skin asepsis to the placement of the electrode in the muscle. The acquisition was performed with three subjects , each one with different characteristics and performing or not the protocol. It was noted that following the proposed acquisition protocol induced an increase in the signal to noise ratio of the captured data. However , the control could be performed even for signals with significant noise contamination.

Finally, the results showed that it is feasible to design the prosthesis, both in modeling with rigid and flexible materials , as its control from the comparison between a fixed threshold and the signal envelope captured EMG-S.

Key words: Myoelectric Prostheses, EMG signal, 3D Printer, Prusa Mendel, Prosthetic Hand, Robotic Hand, Multimaterial printing;

SUMÁRIO

1	INTRODUÇÃO	1
1.1	CONTEXTUALIZAÇÃO	1
1.2	OBJETIVOS DO PROJETO	2
1.3	APRESENTAÇÃO DO MANUSCRITO	2
2	FUNDAMENTAÇÃO TEÓRICA	3
2.1	PRÓTESES	3
2.1.1	HISTÓRICO DAS PRÓTESES PASSIVAS E ATIVAS	3
2.1.2	PRÓTESES NEURAIS	4
2.1.3	PRÓTESES MIOELÉTRICAS	5
2.2	ELETROMIOGRAFIA	5
2.2.1	O POTENCIAL DE AÇÃO DO NEURÔNIO MOTOR	6
2.2.2	CONTRAÇÃO MUSCULAR	7
2.2.3	AQUISIÇÃO DO SINAL ELETROMIOGRÁFICO	7
2.3	IMPRESSÃO 3D DE BAIXO CUSTO - REPRAP	8
2.3.1	PARTE MECÂNICA - ESTRUTURAL	9
2.3.2	PARTE ELETRÔNICA	11
2.3.3	<i>Firmware</i>	12
2.3.4	CONTROLE NUMÉRICO	13
3	MATERIAIS E MÉTODOS	14
3.1	APARATO INSTRUMENTAL	14
3.1.1	EXPERIMENTO PILOTO	14
3.1.2	MODELAGEM DA PRÓTESE	16
3.1.3	PROGRAMAS UTILIZADOS PARA IMPRESSÃO	18
3.1.4	CIRCUITOS DE INSTRUMENTAÇÃO DA PRÓTESE	20
3.1.5	PROGRAMAS PARA CONTROLE DA PRÓTESE	25
3.1.6	EQUIPAMENTOS UTILIZADOS	27
3.2	PROTOCOLO EXPERIMENTAL	28
3.2.1	PROCEDIMENTOS DE IMPRESSÃO E MONTAGEM DA GARRA	28
3.2.2	PROTOCOLO PILOTO EXPERIMENTAL COM SINAL DE EMG SIMULADO	28
3.2.3	PROCEDIMENTOS DE IMPRESSÃO E MONTAGEM DA PRÓTESE	29
3.2.4	SIMULAÇÃO NUMÉRICA DO DETECTOR DE ENVOLTÓRIA	31
3.2.5	INSTALAÇÃO E APLICAÇÃO DO APARATO INSTRUMENTAL COM SINAL REAL	32
4	RESULTADOS E DISCUSSÕES	35
4.1	AVALIAÇÃO DOS SINAIS DE EMG-S SIMULADOS E REAL	35
4.1.1	ANÁLISE NO TEMPO E FREQUÊNCIA	35

4.1.2	RELAÇÃO SINAL RUÍDO	38
4.2	EXPERIMENTOS COM SINAL SIMULADO	41
4.3	EXPERIMENTOS COM SINAL REAL	41
5	CONCLUSÕES	44
	REFERÊNCIAS BIBLIOGRÁFICAS	45
	ANEXOS	49
I	FIGURAS E TABELA DE CUSTOS	50
I	DESCRIÇÃO DO CONTEÚDO DO CD	53
I	CÓDIGOS FONTES COMENTADOS	54
I.1	CÓDIGO PARA GERAR BULLHAS UTILIZADAS NA CRIAÇÃO DO SINAL	54
I.2	CÓDIGO PARA CRIAR SINAL E PLOTAR RESPOSTAS NO TEMPO E FREQUÊNCIA ..	55
I.3	CÓDIGO PILOTO PARA CONTROLE DA PINÇA COM SINAL SIMULADO E DETEÇÃO DE ENVOLTÓRIA DIGITAL UTILIZANDO O MICROCONTROLADOR ARDUINO	58
I.4	CÓDIGO DE IMPLEMENTAÇÃO DO CONTROLE DA PRÓTESE COM ENVOLTÓRIA DO SINAL REAL UTILIZANDO MICROCONTROLADOR ARDUINO	59
I.5	CÓDIGO PARA CAPTURA DO SINAL DE EMG PELO DISPOSITIVO NI USB-6009 ATRAVÉS DO MATLAB	62
I.6	ALGORITMO EM LINGUAGEM C PARA ENVIO DO SINAL DE EMG SIMULADO AO ARDUINO	62
I.7	ALGORITMO EM LINGUAGEM C <i>Feedback</i> VISUAL DE FORÇA	65

LISTA DE FIGURAS

2.1	Mão de ferro de Gotz von Berlichingen. Retirada de [4]	3
2.2	a) Prótese que se encontra no museu Kaiserliches e b) Prótese localizada no Museu de Neu-Ruppin. Retiradas de [4]	4
2.3	Próteses mioelétricas da empresa <i>TouchBionics</i> . Retirado de [15]	5
2.4	Unidade Motora. Adaptada de [16].	6
2.5	Potenciais de ação. Adaptado de [16].	7
2.6	Representação da geração do potencial de ação da unidade motora. Retirada de [23]	9
2.7	Impressora 3D do modelo Prusa Mendel. Retirado de [1]	10
2.8	Impressoras 3D que dispõem de maior altura de impressão. Retirado de [1]	10
2.9	Impressoras 3D portáteis. Retirado de [1]	10
2.10	Arduino Mega 2560 - Utilizada como controlador principal da impressora 3D. Retirado de [26]	11
2.11	Ramps - Placa de extensão para controle de impressora 3D com Arduino . Retirado de [27].	11
2.12	Driver de corrente utilizado para controle dos servomotores. Fabricado pela empresa Pololu Electronics. Retirado de [28]	12
3.1	O primeiro gráfico representa módulo (em Volts) de um sinal de EMG extraído do bíceps em contração isométrica voluntária com frequência de amostragem de 2Khz. O segundo gráfico representa a resposta em frequência do sinal	15
3.2	Bulhas retiradas do sinal de EMG da Figura 3.1 utilizando janela de Hamming. Possuem durações entre 1 e 4 segundos. Essas bulhas são utilizadas para compor o sinal de EMG utilizado no controle da garra. Cada bulha representa uma curta contração	15
3.3	Módulo do sinal de EMG simulado. Sinal gerado a partir das bulhas mostradas na Figura 3.2 concatenadas com ruído gaussiano.	16
3.4	amostras do sinal quantizado em 8 bits retirado do arquivo gerado para o controle da garra ..	16
3.5	Vista em perspectiva do modelo 3D da mão	17
3.6	Modelo do dedo utilizado para confecção da prótese. O material preto é o plástico ABS e o branco é o plástico flexível	17
3.7	Vista em perspectiva do modelo 3D da mão robótica	18
3.8	Vista em perspectiva do modelo 3D da mão	19
3.9	Interface gráfica do <i>Software</i> utilizado para conversão do modelo 3D no formato STL para G-code	19
3.10	Interface gráfica do <i>software</i> utilizado para gerenciamento da impressão	20
3.11	Eletrodo ativo utilizado para a captação do sinal eletromiográfico de superfície para controle da prótese	20
3.12	Esquemático do Amplificador de Instrumentação INA 118, retirado de [38]. Seu ganho é ajustável através do resistor pela relação $G = 1 + \frac{50k\Omega}{R_G}$	21
3.13	Amplificador de ganho 5 cascadeado com um filtro passa baixas passivo com frequência de corte de aproximadamente 500 Hz, retirado de [39]	22

3.14	Diagrama de BODE e diagrama de fase do circuito passa baixas da Figura 3.13, obtida através da equação (3.8)	23
3.15	Circuito retificador de meia onda com detecção de envoltória: Circuito A: Amplificador não inversor. Circuito B: Retificador de Meia Onda concatenado com filtro passa baixas para detecção de envoltória	24
3.16	Comportamento do circuito retificador de meia onda: (a) durante o semiciclo positivo da entrada V_{in_2} (b) durante o semiciclo negativo da entrada V_{in_2}	24
3.17	Detector de envoltórias digital quantizado em 1 bit: (a) segmento extraído do sinal simulado; , (b) detecção dos valores onde o módulo da amostra é maior ou igual ao valor de limiar e ; (c) envoltória do pulso de contração obtido pelo detector de envoltórias digitais de 1 bit	26
3.18	(a) Impressão de partes da garra (Fig. 3.18) com a impressora LEIzinha 3D (b) Primeira Garra impressa para validação experimental obtida do site <i>Thingiverse</i> [34]	28
3.19	Prótese de mão impressa na impressora 3D. A parte branca é o plástico flexível e a parte preta é o plástico rígido (ABS)	30
3.20	Suporte para a mão. Possui encaixe para dois servomotores e encaixe para a mão. Impresso em plástico ABS	30
3.21	Prótese de mão com a parte mecânica concluída. Utilizou-se plástico ABS para impressão do suporte e da mão, sendo esta última combinada com plástico flexível. O servo a esquerda possui menor torque e tem função de flexionar o polegar. O servo a direita tem como função flexionar os demais dedos.	31
3.22	Senóide com frequência de 250Hz e amplitude de 250mV introduzida em um circuito retificador de meia onda com detecção de envoltória. A onda verde é a entrada, a onda vermelha é a entrada amplificada e o sinal azul é a envoltória do sinal amplificado.	32
3.23	Sinal de EMG simulado introduzido em um circuito retificador de meia onda com detecção de envoltória. O sinal azul corresponde ao EMG simulado e o sinal vermelho representa sua envoltória	32
3.24	Montagem final do circuito retificador de meia onda com detecção de envoltória. Destacado de vermelho é o amplificador com ganho de ajustado para 5.6. Em azul, é a parte do circuito responsável por ceifar a parte positiva do sinal. Em preto tem-se o capacitor responsável por retirar a envoltória. Por fim, em verde, tem-se os fios de conexão do eletrodo. .	33
3.25	Procedimento para captura do sinal eletromiográfico e controle da prótese	34
4.1	Resposta em frequência do sinal de EMG simulado.	35
4.2	(a) Sinal de EMG extraído do bíceps do voluntário 1 e (b) sua envoltória	36
4.3	Gráfico da resposta em frequência do sinal de EMG extraído do bíceps do voluntário 1	36
4.4	(a) Sinal de EMG extraído do bíceps do voluntário 2 e (b) sua envoltória	37
4.5	Gráfico da resposta em frequência do sinal de EMG extraído do bíceps do voluntário 2	37
4.6	(a) Sinal de EMG extraído do bíceps do voluntário 3 em contração livre e (b) sua envoltória .	37
4.7	Gráfico da resposta em frequência do sinal de EMG extraído do bíceps do voluntário 3 em contração livre	38

4.8	(a)Gráfico referente ao ruído do voluntário 1, capturado com o músculo em repouso e (b) sua envoltória	39
4.9	(a)Sinal de EMG-S extraído do bíceps do voluntário 1 em contração livre máxima e (b) sua envoltória	39
4.10	(a)Gráfico referente ao ruído do voluntário 2, capturado com o músculo em repouso e (b) sua envoltória	40
4.11	(a)Sinal de EMG extraído do bíceps do voluntário 2 em contração livre máxima e (b) sua envoltória	40
4.12	Resultado final da garra controlada por sinal eletromiográfico simulado	41
4.13	Resultado final da prótese ativa de mão controlada por sinal eletromiográfico de superfície ..	43
I.1	Vistas superior, frontal e esquerda do suporte modelado para confecção da prótese	50
I.2	Vistas superior, frontal, isométrica e esquerda da polia modelada para confecção da prótese .	51
I.3	Sinal de EMG-S capturado com eletrodo passivo. O sinal apresentou alta contaminação com ruído conforme pode ser visto na resposta do sinal no tempo e na frequência.	51

LISTA DE TABELAS

4.1	Características dos voluntários	36
4.2	Relação sinal ruído para os diferentes voluntários	41
4.3	Porcentagem do nível de contração para ativar a prótese (0.73) em relação ao nível máximo aplicado pelos voluntários	42
I.1	Custos para confecção e montagem da prótese apresentada. adquiridos no Brasil	52

LISTA DE SÍMBOLOS

Símbolos Latinos

T_s	Período de Amostragem	[s]
F_s	Frequência de Amostragem	[Hz]

Siglas

ABS	<i>acrylonitrile butadiene styrene</i>
A/D	<i>Analógico-Digital</i>
CAD	<i>Computer Assisted Design</i>
CAM	<i>Computer Aided Manufacturing</i>
DARPA	<i>Defense Advanced Research Projects Agency</i>
ECoG	Eletrocorticografia
EMG	Eletromiografia
EMG-S	Eletromiografia de Superfície
FFT	<i>Fast Fourier Transform</i>
FTDI	<i>Future Technology Devices International</i>
GND	<i>Ground</i>
ICSP	<i>In-circuit serial programming</i>
LEI	Laboratório de Engenharia e Inovação
PWM	<i>Pulse-width modulation</i>
RMS	<i>Root Mean Square</i>
STL	<i>STereoLithography</i>
SNR	Signal Noise Ratio
SUS	Sistema Único de Saúde
TCC	Trabalho de Conclusão de Curso
UART	<i>Universal asynchronous receiver/transmitter</i>
USB	<i>Universal Serial Bus</i>
CN	Código Numérico

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

É notória a popularização das impressoras 3D nos últimos anos, e as iniciativas *open hardware* representam a maioria dessas máquinas atualmente em atividade [1]. Os projetos com a impressora 3D são extremamente customizáveis, tendo a capacidade de modelagem do usuário e as características físicas do hardware como limitantes. O modelo Prusa Mendel utilizado neste projeto se limita as dimensões de 20x20x6cm e camadas de 0.3mm. As possibilidades de aplicações são inúmeras e o presente trabalho visa sua utilização no âmbito de reabilitação, mais especificamente na concepção de prótese de mão.

Próteses são substitutos artificiais utilizados para suprir a ausência de membros e suas funções [2], ausências, essas, em sua maioria causadas por lesões, tumores, diabetes, fornecimento insuficiente de sangue ou, em caso de pessoas que já nascem sem o membro, amputações congênitas [3].

Segundo PUTTI [4], a utilização de próteses se tornou um complemento em amputações cirúrgicas a partir do momento em que os pacientes puderam sobreviver ao procedimento sem sangrar até a morte, permitindo a obtenção de um toco útil. Isso se tornou possível quando os médicos aprenderam a estancar o sangramento com o uso de cauterização. Antes disso, somente era possível amputar pedaços de extremidades gangrenadas.

Qualquer comprometimento da mão, que é uma das partes mais expostas do corpo humano, pode causar ao indivíduo uma série de problemas psicológicos, sejam eles motivados pela deformação estética ou pela limitação funcional [5] [6]. Os membros superiores são caracterizados por sua extrema mobilidade e, por esses motivos, as próteses, como são hoje, não substituem verdadeiramente o órgão amputado, servindo apenas como uma ferramenta de auxílio [7].

Segundo o Ministério da Saúde, foram realizadas em 2011 cerca de 50.000 amputações pelo Sistema Único de Saúde (SUS), 6 % de membros superiores. Estima-se ainda que 15 % de todas as amputações realizadas no Brasil seja de membros superiores, porém, não há uma informação precisa [8].

Visto a grande necessidade e o alto custo das próteses ativas de mão, com seus modelos mais avançados variando de \$ 4.000 a \$75.000 dólares [9], os autores se motivaram a desenvolver uma metodologia de concepção de uma prótese ativa de baixo custo. Ademais, a prototipagem utilizando a impressora 3D permite um alto nível de customização não restringindo os projetos a modelos tradicionais de próteses, que podem ser desenvolvidas para desempenhar papel de ferramentas, recipientes, entre outros. Combinadas sua modelagem em *software* CAD, a impressão 3D e instrumentação eletrônica, é possível a atuação da prótese por meio do sinal eletromiográfico (EMG). Nesse sentido, uma unidade de aquisição de sinais para captura do sinal EMG e uma unidade de processamento para convertê-lo em sinal de controle é proposta.

1.2 OBJETIVOS DO PROJETO

O objetivo deste projeto é iniciar um estudo de aplicabilidade da impressora 3D no contexto de reabilitação, desenvolvendo uma prótese de baixo custo com um grau de liberdade, que possa ser adquirida por pessoas que sofrem das enfermidades supracitadas mas não possuem os recursos financeiros para adquirir uma prótese existente hoje no mercado.

O projeto será feito desde a modelagem até a instrumentação da prótese utilizando sinal eletromiográfico (EMG). Ambas serão modeladas em *software* CAD. Serão produzidas em uma impressora 3D de baixo custo com a utilização de plástico ABS(*acrylonitrile butadiene styrene*) e plástico flexível da marca filaflex [10]. O controle será feito com o microcontrolador arduino baseando-se em sinais provenientes da contração muscular captados por um circuito de instrumentação e conformação do sinal. Por fim o *software* deve gerar um sinal modulado por largura de pulso (PWM) para controle do motor que abre e fecha a prótese.

Ao concluir o projeto, a prótese produzida deverá realizar o movimento de uma pinça para auxiliar nas funções mais fundamentais da mão, como ato segurar um objeto ou dar um aperto de mão.

1.3 APRESENTAÇÃO DO MANUSCRITO

No capítulo 2 é feita uma revisão bibliográfica sobre próteses, eletromiografia e impressão 3D de baixo custo. Em seguida, o capítulo 3 descreve o aparato instrumental e o protocolo experimental utilizados para a prototipagem e testes da prótese. Então, no capítulo 4 são apresentados e analisados os resultados obtidos. Por fim vêm as conclusões no capítulo 5. Os anexos contêm desenhos técnicos, uma tabela de custos do projeto e os códigos desenvolvidos.

2 FUNDAMENTAÇÃO TEÓRICA

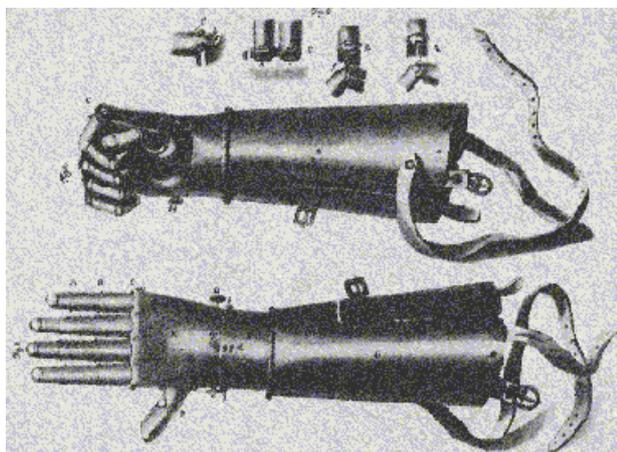
2.1 PRÓTESES

2.1.1 Histórico das Próteses Passivas e Ativas

Membros artificiais aparecem em ilustrações e registros históricos muito antigos, como desenhos em vasos e mosaicos [4]. O primeiro relato documentado da utilização de próteses foi o registrado por Heródoto que diz respeito ao soldado Hegistratos que, após ser preso e condenado a morte em 484 a.C., cortou seu próprio pé para se livrar das correntes e conseguir escapar [4][7].

As próteses de membros inferiores começaram a surgir bem antes das superiores, que por sua vez tem seus primeiros registros datados por volta do século XV e XVI [4]. Grandes mentores de próteses foram Ambroise Paré e Leonardo da Vinci, os seus desenhos impressionam pela capacidade inventiva dos autores[7].

Algumas próteses de membros superiores, apesar de serem passivas, possuíam grandes funcionalidades como mecanismos de movimentação dos dedos que permitiam segurar objetos, como no caso da mão de ferro de *Gotz von Berlichingen*, que é a prótese mecânica mais famosa, historicamente falando [4] (Fig. 2.1). Ela foi utilizada pelo próprio após perder a mão em batalha durante o cerco de *Landshut in Bavaria* em 1504. Apesar de ser uma das próteses superiores mais antigas, ela apresenta uma grande quantidade de graus de liberdade, no que diz respeito a articulações, não ficando para trás em relação as encontradas posteriormente. Como é possível ver na Fig. 2.1(b), a mão de ferro se adaptava ao formato de objetos.



(a)



(b) segurando objeto

Figura 2.1: Mão de ferro de Gotz von Berlichingen. Retirada de [4]

Apesar de não ser tão conhecida como a anterior, há também a prótese localizada no museu Kaiserliches em Berlim (Fig. 2.2 (a)) a qual se afirma ter sido feita no século 15. Porém, há controvérsias quanto a essa data. Ao contrário da prótese de *Gotz*, essa tem um polegar que se move na articulação metacarpofalângica

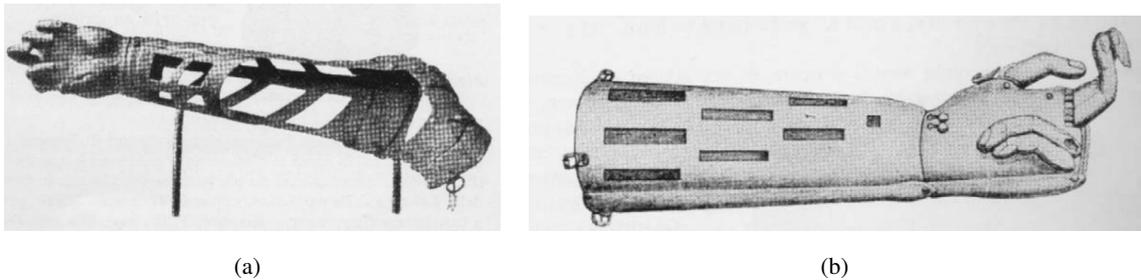


Figura 2.2: a) Prótese que se encontra no museu Kaiserliches e b) Prótese localizada no Museu de Neu-Ruppin. Retiradas de [4]

[4]. Existe ainda outra prótese que se encontra no Museu de Neu-Ruppin (Fig 2.2 (b)) que foi descoberta no canal Reno perto de Alt-Ruppin em 1836 e não se sabe ao certo a data de sua criação, mas apresenta evoluções em relação às citadas anteriormente: além de possuir um polegar que se move na articulação metacarpofalângica, os dedos também podem ser movidos aos pares [4].

Aos poucos as próteses foram evoluindo e se tornando cada vez mais necessárias em virtude das inúmeras guerras no decorrer da história. Ao fim da primeira Guerra Mundial foram registradas cerca de 300.000 amputações na Europa. Durante a segunda Guerra Mundial nos EUA, institutos científicos foram encarregados de constituir uma comissão de estudo de próteses [7] e, a partir daí, a evolução desse campo se tornou cada vez mais acentuada. A próxima sessão apresenta as próteses ativas que se desenvolveram a partir desses estudos.

O design de próteses ativas de mão funcionais é bem mais desafiador que o de mãos robóticas. No primeiro caso, além das dificuldades em se construir os mecanismos físicos que fornecem mobilidade com os graus de liberdade requisitados, existe uma limitação mais severa quanto ao peso, o sistema de alimentação e o controle da prótese, principalmente em se encontrar número suficiente de fontes de controles para as articulações projetadas [5]. Dois tipos de próteses são exibidos nas seções 2.1.2 e 2.1.3

2.1.2 Próteses Neurais

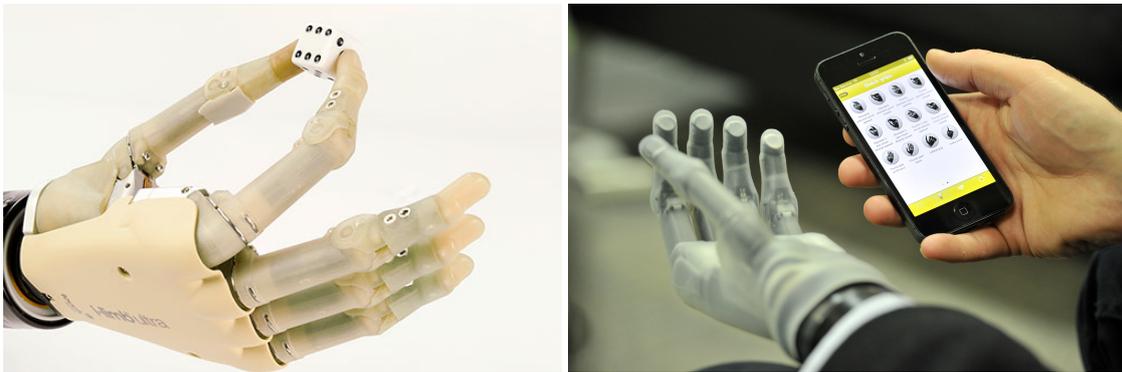
Quando se fala em próteses ativas de mão, uma idéia que pode ser comum é da mesma ser controlada de forma similar à natural. De fato, algumas pesquisas se concentram no desenvolvimento de próteses neurais de extremidades. Em 2006 a agência DARPA¹ criou dois programas inter-relacionados para o desenvolvimento de próteses controladas neuralmente. São eles: *Revolutionizing Prosthetics 2007* e *Revolutionizing Prosthetics 2009*. Com o avanço dos projetos e com o apoio da empresa *DEKA Research and Development*, em 2012 foi desenvolvida uma prótese ativa de braço com 22 graus de liberdade, que utiliza o sinal de eletrocorticografia (ECoG), onde os eletrodos são implantados diretamente na superfície do cérebro [11].

¹Defense Advanced Research Projects Agency

2.1.3 Próteses Mioelétricas

As próteses mioelétricas se baseiam em sistemas de controle que extraem informações dos sinais de Eletromiografia (EMG) [12]. É necessária uma força suficientemente grande para ultrapassar um determinado limiar, que pode ser obtido de forma experimental. Então, baseando-se no limiar e segundo uma lógica de controle realiza-se a ativação do motor, permitindo abertura ou fechamento. Esse tipo de prótese é indicada para amputados transradiais com idade mínima de um ano [13]. Antes disso recomendam-se próteses passivas.

BOCCOLINI afirma que a pesquisa com próteses Mioelétricas vem sendo desenvolvida pela Alemanha desde a primeira guerra mundial. Inicialmente foram construídas próteses para amputados de antebraço e em seguida próteses para amputados de braço. Estas, por sua vez, são mais difíceis de se projetar, devido ao maior número de graus de liberdade (preensão pela mão, pronosupinação do antebraço e flexão e extensão do cotovelo [7]). Diversos países também vêm avançando em suas pesquisas: A *International Business Machine* e a *United States Manufacturing Cy.* nos Estados Unidos estudam desde 1945 o desenvolvimento de próteses. As pesquisas mais atuais em próteses robóticas visam automação com um grande grau de autonomia, funcionalidade, conforto e facilidade de uso [14].



(a) i-Limb Ultra

(b) i-Limb Ultra Revolution

Figura 2.3: Próteses mioelétricas da empresa *TouchBionics*. Retirado de [15]

A empresa *Touch Bionics*, fabricante conceituada em próteses de mão mioelétricas e passivas é responsável, dentre outras, pelas próteses *i-Limb Ultra* (Fig. 2.3[a]) e *i-Limb Ultra Revolution* (Fig. 2.3[b]). Estas se adaptam ao contorno do objeto que se deseja agarrar [15].

2.2 ELETROMIOGRAFIA

Antes de investigar o sinal eletromiográfico é interessante que se entenda como se dá o processo de contração do músculo esquelético, que é onde ocorre o fenômeno bioelétrico responsável por gerar o sinal de EMG [16].

2.2.1 O Potencial de ação do Neurônio Motor

O neurônio motor é responsável por canalizar o impulso elétrico até as fibras musculares, ou seja, é ele quem faz a conexão entre o sistema nervoso e o músculo esquelético. [16] O neurônio motor em conjunto com todas as fibras musculares inervadas a ele compõem as unidades motoras [7] [16] [17] [18] (Fig. 2.4).

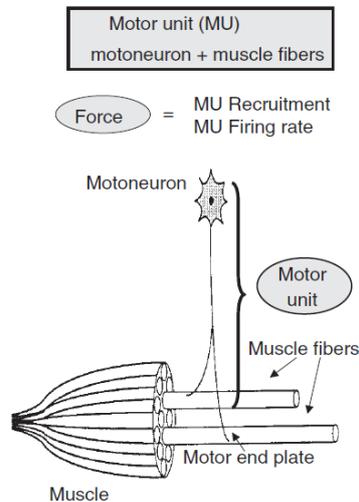


Figura 2.4: Unidade Motora. Adaptada de [16].

O neurônio motor é um nervo eferente² composto por um corpo celular que se encontra na substância cinzenta espinhal e capta o sinal elétrico através de seus dendritos [19], também possui um longo axônio que age como um duto condutor e se estende até a fibra muscular. Nesta se encontra a junção neuromuscular, a qual se inerva ao músculo e propaga o sinal em ambas direções ao longo da fibra muscular [20].

Em equilíbrio, o corpo do axônio permanece negativamente carregado, na faixa de -90 a -70 mV. Isso se deve ao fato da bomba de sódio e potássio lançar 3 íons de sódio para fora da membrana e somente 2 íons de potássio para dentro [18]. Como ambos os íons são positivos, o lado interior da membrana fica negativamente polarizado em relação ao exterior. Este é denominado estado de repouso da membrana.

Ao receber um impulso elétrico proveniente do sistema nervoso central, que chega ao corpo celular através de seus dendritos, são abertos os canais de sódio dos axônios fazendo com que eles adentrem a membrana celular aumentando o potencial no seu interior. Se esse potencial ultrapassar uma certa tensão de limiar, que equivale a uma tensão de 10 a 30 mV acima da tensão de repouso, é dado início ao potencial de ação [17], onde uma grande quantidade de sódio adentra a membrana tornando positivo o seu potencial. Este processo é chamado de despolarização.

Após atingir uma tensão de aproximadamente +35mV os canais de sódio se fecham e os de potássio são abertos, fazendo com que o potássio deixe novamente o interior da membrana, num processo chamado de repolarização, diminuindo o potencial até que ela esteja novamente em repouso [17]. O processo de despolarizar, repolarizar e voltar para o repouso é chamado de potencial de ação. Ele se propaga por todo o axônio até as junções neuromusculares, onde se dá início ao processo da contração muscular.

²carrega informação do sistema nervoso central, cordão espinhal, para a unidade efetora, músculo

2.2.2 Contração Muscular

De acordo com Guyton & HALL [18], a contração muscular começa quando os potenciais de ação se deslocam pelo neurônio motor até suas terminações nas fibras musculares. Em cada terminação, o neurônio libera uma pequena quantidade da substância neurotransmissora acetilcolina, que age em uma área local da membrana da fibra muscular abrindo múltiplos canais regulados pela acetilcolina por meio de moléculas de proteína que flutuam na membrana. A abertura desses canais permite a difusão de grande quantidade de íons de sódio para o lado interno da membrana das fibras musculares, desencadeando o potencial de ação na membrana (Fig. 2.5).

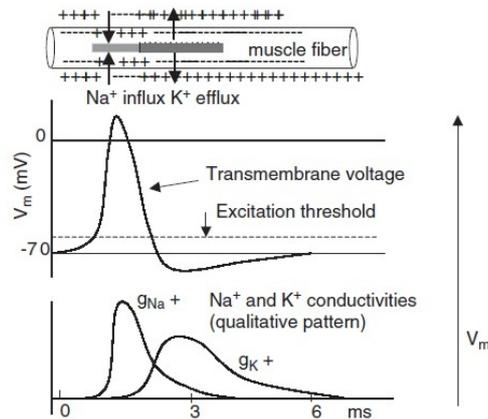


Figura 2.5: Potenciais de ação. Adaptado de [16].

O potencial de ação se propaga por toda a membrana da fibra muscular da forma similar que percorre a membrana das fibras nervosas. O potencial de ação despolariza a membrana muscular, e grande parte de sua eletricidade flui pelo centro da fibra muscular de modo a fazer com que o retículo sarcoplasmático libere uma grande quantidade de íons cálcio que ali estavam armazenados. Estes íons ativam as forças atrativas entre os filamentos de miosina e actina, fazendo com que eles deslizem ao lado um do outro, realizando o processo contrátil. Após uma fração de segundo, os íons de cálcio são bombeados de volta para o retículo sarcoplasmático pela bomba de cálcio da membrana, onde permanecem armazenados até que novo potencial de ação muscular se inicie, finalizando a contração muscular [18].

O sinal de EMG é gerado pela atividade elétrica das fibras musculares durante a contração muscular, é uma representação do campo de potencial elétrico gerado pela despolarização da fibra muscular externa, a sua detecção pode ser feita utilizando eletrodos intramusculares ou de superfície que são colocados a uma certa distância da fonte [16].

2.2.3 Aquisição do Sinal Eletromiográfico

Para a captação do sinal eletromiográfico é necessária a utilização de eletrodos. Estes são responsáveis pelo interfaceamento entre o corpo e o sistema de aquisição. As aquisições podem ser feitas através de eletrodos invasivos ou não invasivos [21]. Para um sinal de melhor qualidade e com maiores informações busca-se reduzir os efeitos causados pelo volume condutor e evita-se zonas de inervações. Além da retirada dos pelos e raspagem da pele.

Eletrodos invasivos (intramusculares) possuem maior precisão que os não-invasivos (de superfície), uma vez que eles podem acessar diretamente a unidade motora, podendo captar de forma mais fidedigna o potencial de ação de suas fibras musculares [17]. Ademais, esses eletrodos em comparação aos não invasivos, captam maiores amplitudes e espectros de potência mais amplos, podendo atingir frequências de até 10KHz [21] [20]. Entretanto, eles apresentam vários inconvenientes, como a necessidade de excelente esterilização, o perigo de quebra dos fios dentro do músculo e, sobretudo, o desconforto do paciente [19] [20].

A proposta dos autores é utilizar eletrodos de superfície. Os quais possuem maiores aplicações em reabilitação, onde a técnica invasiva não é muito aceita, e em casos de exames que são necessárias repetidas avaliações [16]. Com estes eletrodos é possível obter sinais de Eletromiografia, contidos em até 500Hz, com amplitudes variando entre $50\mu\text{V}$ e 5mV dependendo da configuração utilizada e do músculo analisado [21] [20] [19].

Os eletrodos de superfície podem ser classificados em passivos e ativos. Eletrodos passivos geralmente são utilizados com um gel condutor contendo íons de cloro de modo a diminuir a impedância de contato entre o eletrodo e a pele, podendo ainda ser necessário a raspagem dos pelos na região onde o eletrodo é disposto [19].

Eletrodos ativos possuem um amplificador diferencial bem próximo ao eletrodo de captação [20]. Esse amplificador deve apresentar uma alta rejeição de modo comum (CMRR- Common Mode Rejection Rate), a fim de diminuir o ruído do sinal [19]. Pelo fato dos eletrodos ativos possuírem um amplificador diferencial, é necessária a utilização de um eletrodo de referência que deve ter uma área de aproximadamente 4cm^2 e estar localizado em uma região sem músculo para não interferir no resultado obtido [22].

Uma característica dos eletrodos de superfície é a impossibilidade em se detectar seletivamente o sinal do músculo devido ao seu posicionamento a uma certa distância das fibras musculares e da ampla superfície de contato em relação ao invasivos, registrando assim o somatório dos potenciais de ação de várias fibras (Fig. 2.6), cada qual com uma diferente atenuação, pois o volume condutor age como um filtro para o sinal. Entende-se por volume condutor tudo aquilo que esta entre o eletrodo e a fibra muscular que oferece alguma oposição a passagem de corrente como a camada de gordura, pele, sangue, suor, entre outros.

A aquisição do sinal eletromiográfico de superfície pode ser feita pelo método monopolar ou diferencial. No monopolar, é feita a análise isolada de cada canal enquanto no diferencial, cada canal informa o resultado da diferença entre os sinais obtidos por dois eletrodos consecutivos [24].

2.3 IMPRESSÃO 3D DE BAIXO CUSTO - REPRAP

A iniciativa RepRap promoveu soluções que ampliaram a visibilidade, aplicação e popularização das Impressora 3D, de mesa com capacidade de imprimir objetos plásticos. Como parte considerável da impressora é feita desse material, a mesma pode imprimir suas próprias peças, caracterizando-na como uma máquina auto-replicável. A impressora Darwin, da organização RepRap, foi a primeira impressora de baixo custo e iniciou a revolução das impressoras livres [1]. A comunidade de *software* e *hardware* livres empenha grande esforço no seu desenvolvimento. Dessa forma, encontram-se *firmwares*, *hardwares*

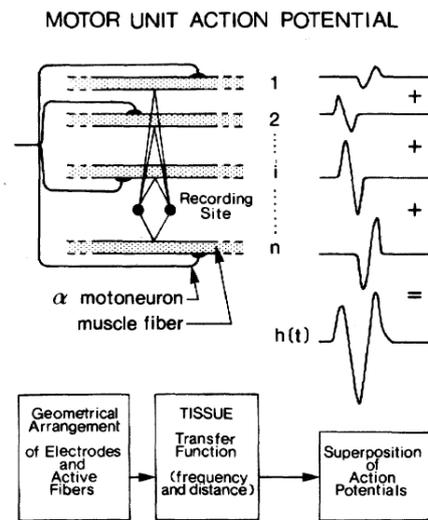


Figura 2.6: Representação da geração do potencial de ação da unidade motora. Retirada de [23]

e *softwares* de controle de impressão e de geração de códigos numéricos de controle (Gcode) gratuitos e aberto.

Exemplos de *firmware* gratuitos são o *Marlin* [25], *Sprinter*, *qrbl*, *Teacup*, dentre outros, quanto aos *hardwares*, alguns deles são: *Darwin*, *Prusa*, *Mendel*, *Printboat*, *Huxury*, *Rostock*. O *software* para controle de impressão mais popular é o *Printrun* [1], que vem sendo atualizado constantemente, outros também atuais são o *ReplicatorG* e o *RepSnaper*. Para a formação do código de máquina (G-code) é necessária uma ferramenta denominada CAM (*Computer Aided Manufacturing*), no qual alguns exemplos de *softwares* atuais são: *Slic3r*, *Skeinforge* e *RepSnapper*.

2.3.1 Parte mecânica - Estrutural

O desenvolvimento de projetos com a impressora 3D apresenta elevada capacidade para customização e prototipagem de soluções. Contudo, alguns fatores limitam projetos implementados em impressoras do tipo Fused Deposition Modeling (FDM), como dimensões, insumo, resolução, velocidade entre outros fatores. No caso do modelo Prusa Mendel (Fig. 2.7) utilizado neste projeto, as dimensões são de 20x20x6cm com camadas a partir de 0.1mm. A velocidade de impressão é ajustável para cada tipo de impressão, podendo ser escolhida no momento da geração do G-code (vide 2.3.4). As velocidades variam de 10 mm/s a 100 mm/s, sendo possível escolher diferentes velocidades em uma mesma impressão com impacto direto na resolução da peça. Pela experiência obtida durante as impressões notou-se que ao imprimir em uma velocidade mais lenta resulta em uma peça mais resistente.

As limitações e performance das impressoras 3D se diferem de acordo com os inúmeros projetos de máquinas livres, onde cada uma atende melhor determinados requisitos. A questão da precisão depende, entre outros fatores, do diâmetro do bico da extrusora, dos passos do motor e da resolução do driver de corrente. Visto que eles são similares para a maioria das impressoras, este requisito não varia muito entre elas. Se o requisito for altura de impressão, existem as impressoras do modelo Morgan (Fig. 2.8a), Rostock (Fig. 2.8b) e Cyberus (Fig. 2.8c), sendo que essa última pode imprimir peças superiores a 50 cm de

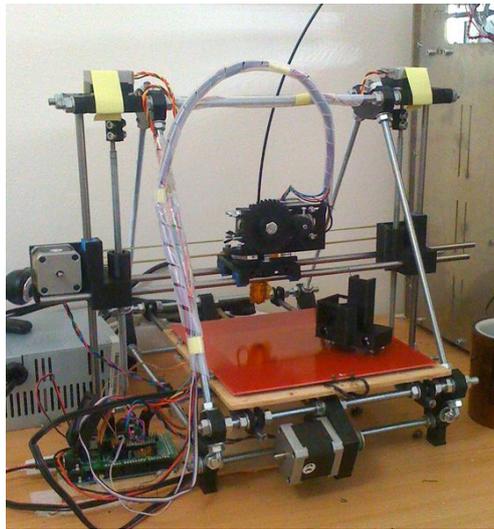
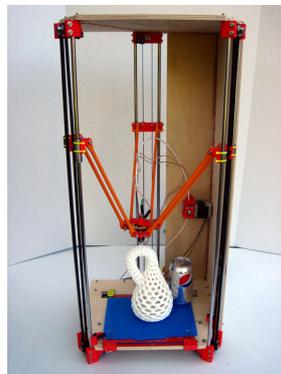


Figura 2.7: Impressora 3D do modelo Prusa Mendel. Retirado de [1]



(a) Morgan



(b) Rostock



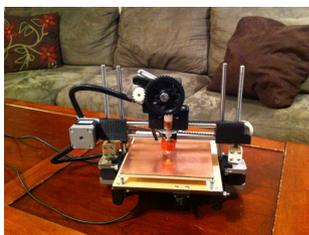
(c) Cerberus

Figura 2.8: Impressoras 3D que dispõem de maior altura de impressão. Retirado de [1]

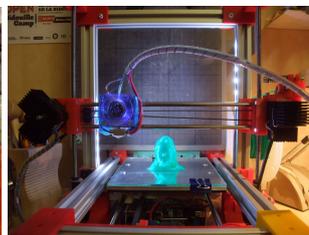
altura. Se for velocidade de impressão, diminui-se o peso da extrusora ou até mesmo pode-se desacoplá-la do bico de extrusão, conduzindo o filamento através de um tubo. Se for portabilidade, existem os modelos Tantillus (Fig.2.9a), Printbot (Fig.2.9b) e FoldaRap (Fig.2.9c). Esta última é dobrável e pode ser carregada em uma mala (Fig. 2.9d). Se for necessário maior suporte técnico, manuais, fóruns, escolhe-se o modelo Prusa Mendel, pois é a impressora com maior grau de popularização das iniciativas RepRap (2013).



(a) Tantillus



(b) PrintRBot



(c) FoldaRap



(d) FoldaRap dobrada

Figura 2.9: Impressoras 3D portáteis. Retirado de [1]

No site da Reprap [1] é possível encontrar todos os componentes e o procedimento necessário para a

confecção e montagem de diferentes modelos de impressoras 3D.

2.3.2 Parte eletrônica

Existem diversos modelos de controladoras de impressão disponíveis. Alguns desses, com microcontrolador externo, são expansões para a plataforma de desenvolvimento Arduino. Este é o caso da Ramps e Ultimaker. Outras possuem todos os componentes embarcados na mesma placa, com exceção dos drivers do motor de passo. Nesta categoria estão a Sanguinololu, Gen7. Na última categoria estão as placas que possuem todos os componentes integrados, exemplos são a Rambo e a Melzi.

A plataforma Arduino é bastante utilizado em projetos *opensource* de impressoras 3D. Isso se deve a sua facilidade de elaboração das soluções e a grande quantidade de documentação referente essa plataforma microcontrolada de desenvolvimento de projetos eletrônicos.

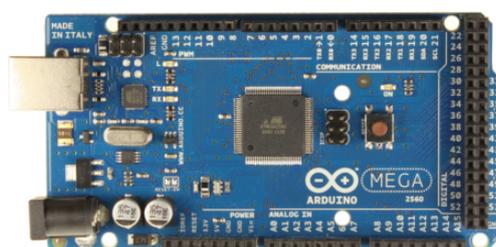


Figura 2.10: Arduino Mega 2560 - Utilizada como controlador principal da impressora 3D. Retirado de [26]

A placa Ramps 1.4(Fig.2.11) foi projetada para encaixe direto sobre a Placa Arduino Mega, e é compatível tanto com o Arduino Mega 1280 quanto ao Arduino Mega 2560. Suas principais características[27] são listadas abaixo



Figura 2.11: Ramps - Placa de extensão para controle de impressora 3D com Arduino . Retirado de [27]

- 3 saídas mosfets controladas com PWM para controle de potência.
- 3 circuitos para termistores
- 5 soquetes para driver de motor de passo Pololu
- 6 conjuntos de pinos digitais em conectores com VCC e GND para sensores de fim de curso

- Usa conector de potencia "plugável" e parafusáveis.
- Pinos extras para expansão: PWM, digital, serial, SPI, I2C e analógico

Com o uso dos pinos de expansão, a Ramps permite que dispositivos externos sejam conectados à impressora. Podem ser conectados display externo e cartão de memória para impressão sem auxílio externo do computador. O próprio controlador Arduino já possui poder de processamento suficiente para tal propósito.

Os *drivers* de motor de passo utilizados são fabricados pela empresa Pololu Electronics[28]. O modelo utilizado é o circuito com o *driver* A4988(Fig. 2.12), que é um *driver* de motor de passo bipolar com micro passos de até um sobre dezesseis. O número de micro passos é um dos fatores que interferem na resolução da impressora, pois o micro passo possibilita um maior controle da rotação do motor de passo. A tensão de operação varia de 8 a 35 V e pode fornecer até dois amperes de corrente por fase do motor.

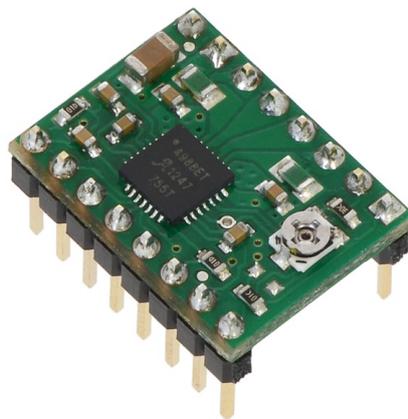


Figura 2.12: Driver de corrente utilizado para controle dos servomotores. Fabricado pela empresa Pololu Electronics. Retirado de [28]

2.3.3 *Firmware*

Para receber os comandos provenientes do computador, além da comunicação serial, um *firmware* instalado no microcontrolador é necessário para fazer a tradução dos dados em movimentos dos motores, comandos de controle de temperatura da mesa aquecida e extrusoras e comandos de regulagem de velocidades das ventoinhas de refrigeração. Os principais *firmwares* são: Teacup, Sprinter, Marlin e Repetier. Destes, o mais simples é o Teacup, ele possui o menor tamanho de instalação quando compilado e é escrito em C e pode ser utilizado em microprocessadores com menor poder de processamento como o Arduino Uno, porém, seus recursos são limitadas. O Sprinter é suportado por grande parte dos controladores de impressão e é de fácil configuração e calibração, por isso, ele é indicado para iniciantes. Os dois outros *firmwares* são baseados no Sprinter, mas possuem funcionalidades adicionais. O Marlin permite controle total da calibração. Ele possui controle de aceleração, permite a impressão de cantos em alta velocidade, suporta controle PID de temperatura, identifica arcos e imprime arcos corretamente, além de opções de segurança, como limitação das temperaturas da extrusora e da mesa aquecida. Por fim, O repetier usa um

protocolo próprio de comunicação com o computador que é mais robusto e confiável, porém para usá-lo é necessário o *software* Repetier [1]. Por fim, o *firmware* escolhido foi o Marlin. Tal escolha se deu devido ao seu conjunto maior de instruções e porque permite ajustar melhor os parâmetros de calibração.

2.3.4 Controle numérico

O Controle numérico (CN) representa um conjunto de comandos para sistemas ou dispositivos eletromecânicos no formato de números, letras, símbolos e palavras. É o principal elemento do processo de manufatura integrada por computador, particularmente para controlar a operação das máquinas. O CN também é essencial para operação dos robôs industriais modernos. Há dois tipos básicos de sistemas de CN: controle ponto-a-ponto e de controle de trajetória [29].

Sistema de CN ponto-a-ponto é o caso onde o dispositivo é programado para realizar uma série de movimentos com ponto de partida e pontos de parada fixos [29]. É normalmente usado em máquinas que realizam operações de fresagem em linha reta e de furos. É relativamente simples de programar e não requer a auxílio de um computador [30].

Sistema de CN de trajetória é o caso em que um dispositivo programado ponto a ponto tem memória suficiente para ser "consciente" de suas ações e seus resultados anteriores, agindo de acordo com essas informações [29]. É normalmente usado em máquinas que realizam operações de contorno, como fresadoras, tornos, máquinas de corte com fogo, máquinas de desenhar e no presente estudo aplicado na impressora LEIzinha 3D. A preparação do programa para máquinas de controle de trajetória é mais complexo e geralmente requer o auxílio de um computador ou microcontrolador com capacidade de processamento suficiente[31].

O CN utilizado pela impressora 3D é o G-code, que se enquadra dentro dos sistemas de controle numérico de trajetória. É necessária a utilização de um *software* para converter o modelo 3D no formato STereoLithography (STL), e outro programa para a conversão STL - G-code. Entre outras ações de controle realimentado, o G-code é utilizado pelo gerenciador de impressão para o comando dos motores de passo e da temperatura de extrusão. Existem diversos *softwares* para realizar essa conversão, dentre eles temos o Skeinforge, que é considerado um dos pioneiros na geradores de G-code. O SFACT é uma versão simplificada do Skeinforge, e o Slic3r o mais novo dentre os citados[1].

Existem diversos *softwares* de auxílio à impressão. Alguns deles são o Pronterface, RedSnapper, ReplicatorG e o Repetier. O Pronterface possui interface intuitiva e é escrito em Python. O RedSnapper já possui um próprio gerador de G-code e é escrito em C++. O ReplicatorG foi originalmente projetado para a impressoras da empresa Makerbot, que é uma fabricante de impressoras 3D de código fechado e por isso pode-se encontrar problemas em conectar com o *firmware* da reprap por questões de compatibilidade. Por fim, o Repetier, que oferece uma interface simples e possui integração com o Slic3r e o Skeiforge no próprio programa[1].

3 MATERIAIS E MÉTODOS

O sistema construído consiste em uma mão robótica impressa em 3D e controlada em tempo real por sinal eletromiográfico de superfície (EMG-S). As principais partes do sistema correspondem ao eletrodo ativo de EMG-S, circuitos eletrônicos, a mão robótica e um pacote de programas de leitura, tratamento e atuação de controle. Nesse capítulo são detalhados aparato instrumental e o protocolo de teste aplicado no estudo.

Atualmente, o laboratório LEI conta com uma impressora 3D, modelo Prusa Mendel v2, obtida de forma colaborativa por integrantes do próprio laboratório, custando por volta de dois mil reais [32]

3.1 APARATO INSTRUMENTAL

3.1.1 Experimento Piloto

Na primeira parte do trabalho de conclusão de curso, uma garra instrumentada com servomecanismo e confeccionada com uma impressora 3D, do modelo Prusa Mendel, foi controlada por meio de um sinal eletromiográfico cujas contrações foram extraídas e adaptadas de uma eletromiografia real do bíceps em contração isométrica voluntária.

Sinal EMG de Controle

O sinal elaborado para controle da garra simula uma captura obtida a partir de um conversor Analógico-Digital (AD) de 8 bits de sinais referentes à contrações rápidas, variando de 1 a 4 segundos. Sinais estes, que seriam a resposta amplificada e somada com valor de *offset*, correspondente à metade da faixa máxima de tensão do conversor AD, do sinal de EMG-S capturado através de um circuito de instrumentação. O simulador do sinal de EMG-S proposto para o controle da garra foi criado com auxílio do software Matlab, o código está no Anexo I.2, e utiliza o seguinte protocolo:

- Carregou-se um sinal de EMG real extraído do bíceps com contração isométrica voluntária e frequência de amostragem de 2kHz [33]. A Figura 3.1 mostra o comportamento no tempo e na frequência do sinal;
- Segmentou-se o sinal de EMG com 4 janelas de Hamming com tempo variando desde 1 a 4 segundos (Anexo I.1). A janela de hamming, assim como as janelas de Blackman, Bohman, Chebyshev, entre outras, não possui mudanças abruptas em suas bordas, sendo assim, sua utilização evita a inserção de altas frequências ao sinal resultante (Fig. 3.2);
- Foram criadas quatro janelas de ruído gaussiano com 10% da amplitude da janela de Hamming e com duração variando de 5 a 10 segundos;
- Concatenou-se os quatro segmentos da Figura 3.2 com as amostras de ruído entre eles. O sinal retificado é mostrado na Figura 3.3;
- Foram criados 50 sinais de EMG conforme os dois passos anteriores e todos foram concatenados em

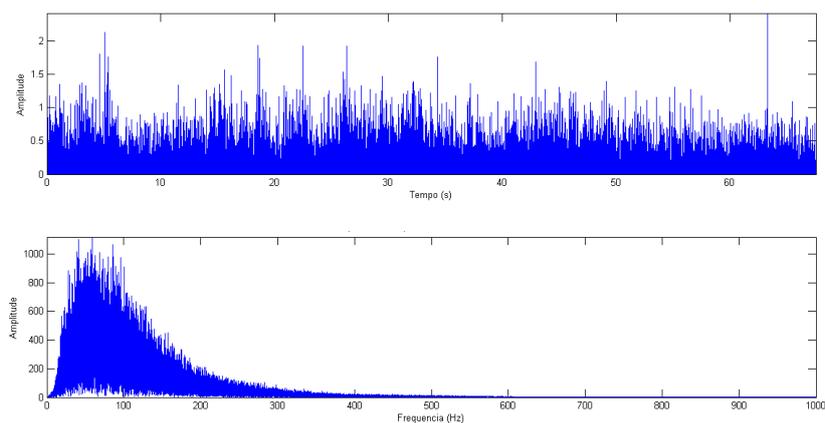


Figura 3.1: O primeiro gráfico representa módulo (em Volts) de um sinal de EMG extraído do bíceps em contração isométrica voluntária com frequência de amostragem de 2Khz. O segundo gráfico representa a resposta em frequência do sinal.

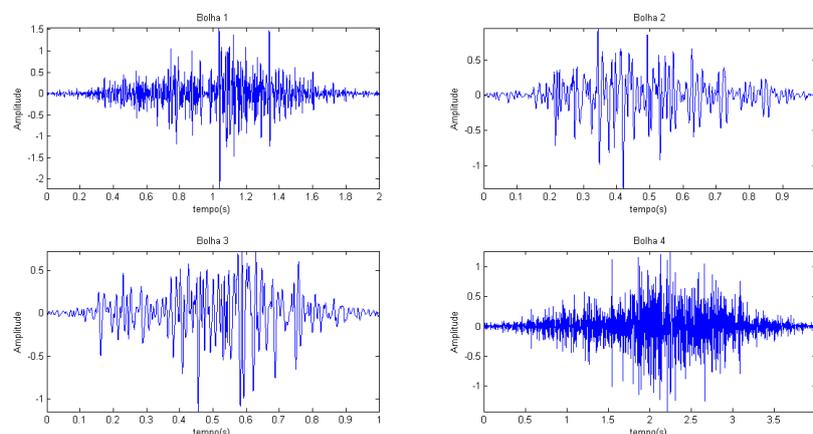


Figura 3.2: Bulhas retiradas do sinal de EMG da Figura 3.1 utilizando janela de Hamming. Possuem durações entre 1 e 4 segundos. Essas bulhas são utilizadas para compor o sinal de EMG utilizado no controle da garra. Cada bulha representa uma curta contração

um único vetor. Criando um sinal de EMG com tempo de aproximadamente 500 segundos.

- Quantizou-se o sinal gerado em 8 bits(fig 3.4), somou-se um valor de *offset* de 127 unidades e armazenou-se em um arquivo. Esse sinal resultante foi utilizado para o controle da garra.

Confecção da Garra

A garra, retirada de [34], foi utilizada com o objetivo de testar o *software* de controle que será utilizado para controlar a prótese. Ela foi escolhida por realizar o movimento de uma pinça, que será similar ao movimento realizado pela prótese. O arquivo STL dessa garra foi retirado do site thingiverse e impressa em plástico ABS. Utilizou-se o Slic3r para gerar o G-code e o Pronterface para fazer o controle de impressão. Por fim, a impressora LEIzinha foi aplicada para a confecção da garra.

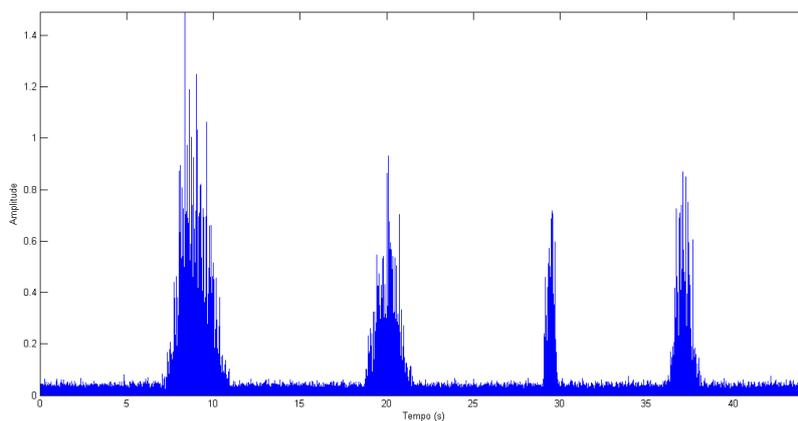


Figura 3.3: Módulo do sinal de EMG simulado. Sinal gerado a partir das bulhas mostradas na Figura 3.2 concatenadas com ruído gaussiano.

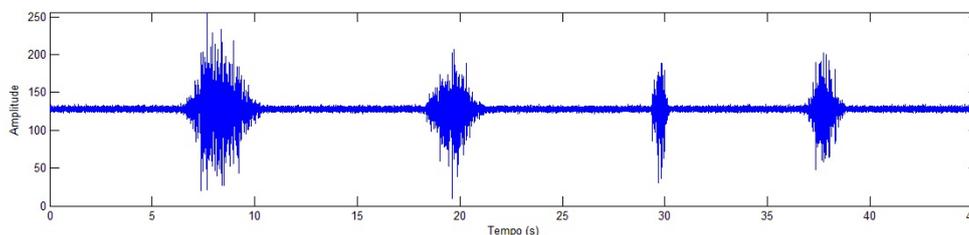


Figura 3.4: amostras do sinal quantizado em 8 bits retirado do arquivo gerado para o controle da garra

3.1.2 Modelagem da Prótese

Para desenho da prótese comprimiu-se todo o projeto no menor número de peças, levando-se em consideração as restrições da impressora LEIzinha : limites de 200 x 200 x 65 mm, apenas uma extrusora e resolução de 0,3mm. Foram necessárias apenas duas impressões: a mão, com material flexível e rígido e as polias junto ao suporte, com material rígido. O tempo total de impressão foi de aproximadamente 6 horas. Com o intuito de facilitar a impressão, todas as peças foram desenhadas com base plana, permitindo maior aderência à mesa e não necessitando a utilização de material de suporte.

Mão Robótica

A mão robótica foi modelada utilizando o *software* Sketch Up. A mão robótica foi desenhada para reproduzir o movimento de pinça, para isso, modelou-se o dedo polegar oposto ao dedo mindinho, permitindo a mão realizar a função de agarrar e segurar objetos. Para sua confecção foi utilizado plástico ABS, que é um material rígido que desempenha o papel dos ossos, e um plástico flexível, que provê o movimento de flexão às articulações da prótese. Uma saliência nas laterais dos dedos permite saber o momento em que se deve realizar a troca do filamento flexível para o rígido. Seu modelo 3D foi disponibilizado em [thingiverse] e pode ser visto na Figura3.5.

O modelo da mão foi inspirado no mecanismo do dedo proposto por massa [35]. O plástico flexível desempenha o papel das molas que provê mobilidade a mão ao mesmo tempo em que desempenha o papel

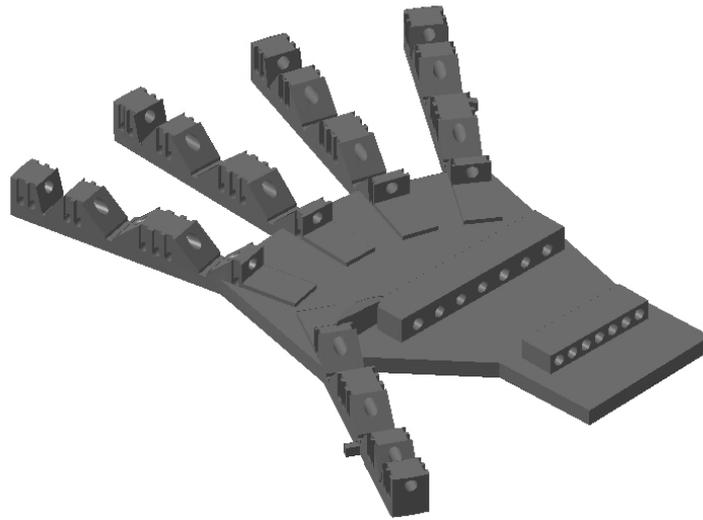


Figura 3.5: Vista em perspectiva do modelo 3D da mão

passivo de retorná-la para a posição aberta. Os dedos foram modelados de modo a permitir que sejam amarrados fios em suas pontas. Eles possuem furos em cada falange que funcionam como guias por onde passam os fios que chegam até os servomecanismos através de outros furos presentes na mão e no pulso. Os Servomotores desempenham o papel de tracionar os fios, flexionando os dedos. Os corpos das falanges proximais, medias e distais foram projetadas em formato trapezoidal. A articulações da base da falange proximal e a da base da falange medial possuem liberdade para gira 90° (Fig 3.6 (b) e (c)). A articulação da base da falange distal possui angulação de 50° (Fig 3.6 (d)).



Figura 3.6: Modelo do dedo utilizado para confecção da prótese. O material preto é o plástico ABS e o branco é o plástico flexível

Dessa forma, a inovação proposta foi o uso de dois materiais durante a impressão. O primeiro maleável e com boa elasticidade, uma espécie de borracha. O segundo possui boa rigidez e resistência mecânica. Unindo ambos os materiais, criou-se uma espécie de compósito que possui as características desejadas, ou seja, flexibilidade nas juntas e rigidez no restante da prótese.

Suporte

O suporte possui um encaixe para a mão da Figura 3.5 e outros dois encaixes para os servo-motores. Ele pode ser visto na Figura 3.7, onde parte hachurado representa o material de suporte, com espessura de 0.3 mm, que deve ser removido após a impressão. Os cantos foram arredondados para evitar possíveis acidentes. O encaixe para a mão possui dimensões de 41.4x3.8mm e os encaixes para os servomotores possuem dimensões de 42x20.5mm. Todas as dimensões do suporte podem ser vistas no desenho técnico em anexo (Fig I.1).

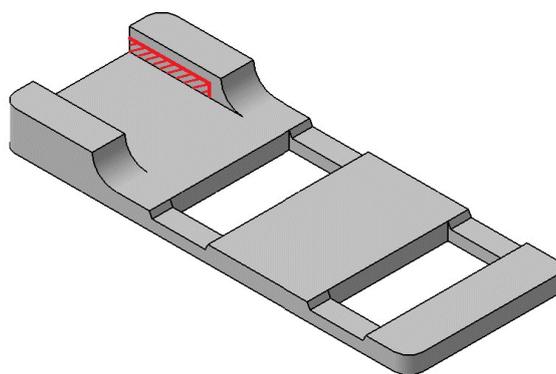


Figura 3.7: Vista em perspectiva do modelo 3D da mão robótica

Polias

A peça ilustrada na Figura 3.8 foi modelada utilizando o *software* CATIA. Sua função é de permitir a conexão dos fios de nylon ao servomotor. Optou-se por fazer o modelo sem encaixe direto ao eixo do servomotor devido às ranhuras presentes no mesmo serem menores que a resolução da impressora, desse modo, ela foi projetada com encaixe para o braço do servomotor. O processo de impressão por deposição de camadas permite a criação da peça já com os furos, sem a necessidade de nenhum processo posterior. Tal furo seria bastante complexo em outros métodos de prototipagem, como fresagem. Todas as dimensões das polias podem ser vistas no desenho técnico em anexo (Fig I.2).

3.1.3 Programas utilizados para Impressão

Gerador de G-code

O gerador de G-code utilizado neste projeto foi o Slic3r (Fig.3.9). Ele é um *software* com código aberto e desenvolvimento livre, e esse é um dos motivos de serem geradas novas versões melhoradas periodicamente. Ele permite a configuração de inúmeras características: como velocidade de impressão, densidade das peças, refrigeração com ventoinhas para impressão de pontes, impressão de uma maior área na primeira camada para melhorar aderência, temperatura da extrusora e da mesa aquecida e uso de

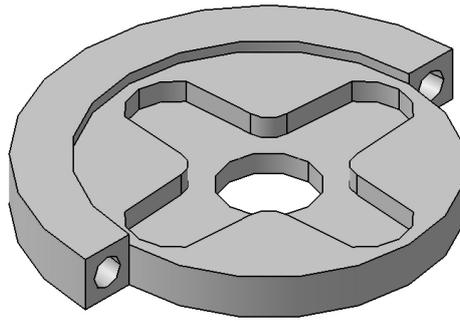


Figura 3.8: Vista em perspectiva do modelo 3D da mão

material de suporte em peças complexas. Também é possível escalonar, rotacionar ou duplicar a peça a ser impressa, ou ainda selecionar outras que serão também impressas ao mesmo tempo, aproveitando, dessa maneira, praticamente toda a área de impressão. Além disso, permite que se determine a quantidade de extrusoras, caso se disponha desse recurso. O Slic3r é um programa baseado na linguagem de programação Perl e utiliza as capacidades de computação paralela dos novos computadores resultando na geração mais rápida do G-code [36].

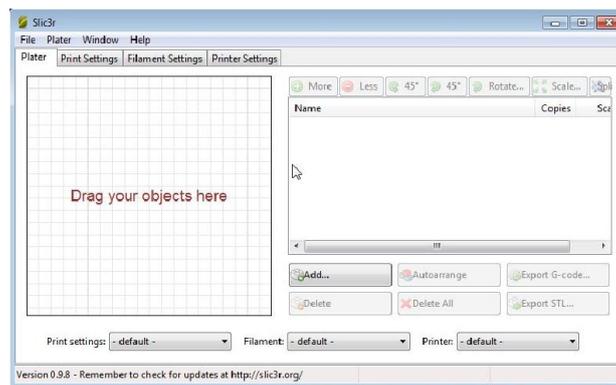


Figura 3.9: Interface gráfica do *Software* utilizado para conversão do modelo 3D no formato STL para G-code

Controlador de Impressão

O *software* escolhido para o controle de impressão foi o Pronterface (Fig. 3.10). Ele se baseia no G-code gerado pelo Slic3r (ou outro *software* similar) para fazer o controle de trajeto da impressão. É necessária a utilização desse *software* de controle uma vez que o sistema de controle numérico é bastante complexo e necessita do auxílio de um computador. Existe a possibilidade dessa impressão ser feita sem esse auxílio caso o Ramps (vide 2.3.2) possua placa de expansão para cartão SD. Como o Arduino não possui memória de armazenamento suficiente, utiliza-se o cartão SD para armazenamento do G-code. Nesse caso o próprio Arduino faz o controle da impressão sem a necessidade de um computador [1].

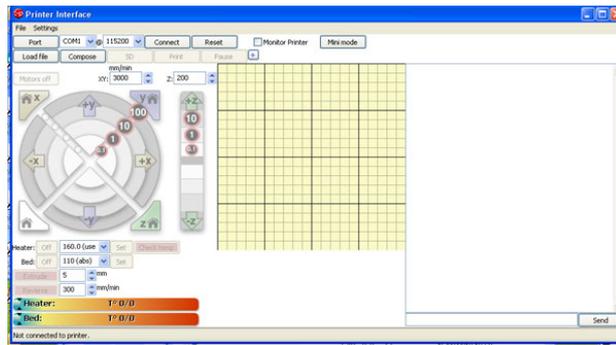


Figura 3.10: Interface gráfica do *software* utilizado para gerenciamento da impressão

3.1.4 Circuitos de Instrumentação da Prótese

Optou-se por fazer a aquisição do sinal de EMG utilizando eletrodos de superfície ativos. Para isso, foi necessária a utilização de um amplificador de instrumentação. Utilizou-se ainda em cascata um filtro passa baixas com uma frequência de corte de 500 Hz, que é a frequência máxima do sinal de EMG captado pelo eletrodo de superfície. O circuito consta ainda com mais um estágio de amplificação e um circuito detector de envoltória, que tem a função de eliminar a parte positiva do sinal e extrair a envoltória do resultado.

O eletrodo ativo é baseado no modelo bipolar da Delsys [37], foi projetado e confeccionado pelos Engenheiros Filipe Tomé e Igor Luiz Bernardes de Moura da Universidade de Brasília. Os contatos elétricos foram projetados com liga Ag/AgCl e dimensões de 1 x 10 mm. O case do eletrodo foi impresso na impressora 3D e suas dimensões são de 25 x 30 mm. O eletrodo pode ser visualizado na Figura 3.11



Figura 3.11: Eletrodo ativo utilizado para a captação do sinal eletromiográfico de superfície para controle da prótese

Amplificador de Instrumentação Presente no Eletrodo

O amplificador de instrumentação utilizado para aquisição do sinal de EMG foi o INA 118. Segundo as informações técnicas do fabricante ,em [38], o INA 118 é um amplificador de instrumentação de propósito

geral e baixo consumo que oferece boa precisão. Seu esquemático composto por três amplificadores operacionais e seu pequeno tamanho o torna ideal para uma enorme gama de aplicações, dentre elas, engenharia biomédica. O circuito de entrada para realimentação de corrente fornece uma extensa largura de banda, até mesmo para configurações com alto ganho (70 KHz com ganho de 1.000). Um único resistor externo permite ajustar qualquer ganho entre 1 e 10.000. O INA possui um circuito de proteção interno o qual permite suportar picos ± 40 V sem danificá-lo. O INA118 é cortado à laser, por isso possui uma tensão de offset muito baixa ($50\mu\text{V}$), com uma variação de $0,5\ \mu\text{V}/^\circ\text{C}$ e uma alta rejeição de modo comum (110dB com ganho de 1.000). Ele opera com fontes de tensão de no mínimo $\pm 1,35\text{V}$, e corrente de repouso de apenas $350\ \mu\text{A}$, o que o torna ideal para sistemas alimentados com bateria. Sua faixa de operação varia de -40°C a $+85^\circ\text{C}$.

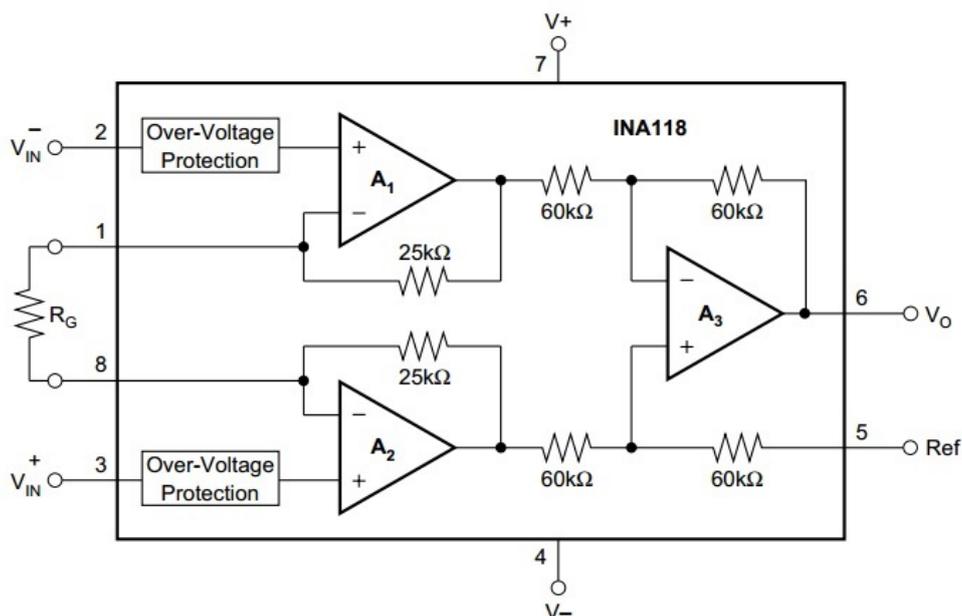


Figura 3.12: Esquemático do Amplificador de Instrumentação INA 118, retirado de [38]. Seu ganho é ajustável através do resistor pela relação $G = 1 + \frac{50k\Omega}{R_G}$.

Filtro passa baixas Presente no Eletrodo

Uma vez que o sinal de EMG captado pelo eletrodo de superfície possui componentes de até 500 Hz, utilizou-se um filtro passa baixas (Fig. 3.13) pra eliminar componentes acima dessa frequência, conservando assim a informação da região onde esta contido o sinal eletromiográfico. O filtro utilizado foi o da Figura abaixo retirado de [39]

Para fazer o cálculo da frequência de corte, considera-se o amplificador ideal, logo as tensões nos terminais 2 e 3 são aproximadamente iguais. Desse modo tem-se que:

$$V_A = V_{in_1} \tag{3.1}$$

Aplicando-se lei dos nós ao nó *A*, obtém-se a seguinte relação:

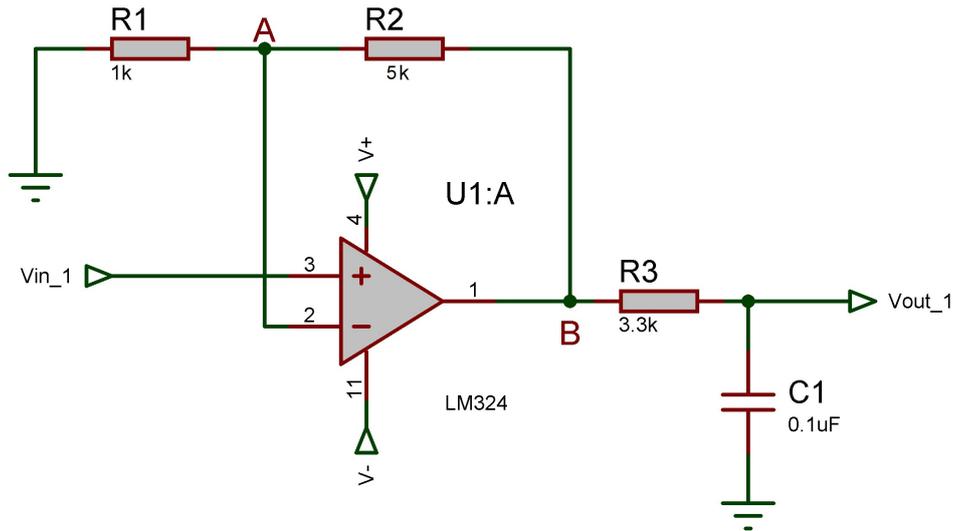


Figura 3.13: Amplificador de ganho 5 cascateado com um filtro passa baixas passivo com frequência de corte de aproximadamente 500 Hz, retirado de [39]

$$\frac{0 - V_A}{R_1} = \frac{V_A - V_B}{R_2} \quad (3.2)$$

Isolando V_B na equação (3.2):

$$V_B = \frac{R_1 + R_2}{R_1} * V_A \quad (3.3)$$

Agora, aplica-se lei dos nós ao nó B e obtém-se a seguinte relação

$$\frac{V_B - V_{out_1}}{R_3} = V_{out_1} * sC \quad (3.4)$$

Novamente isola-se V_B na equação (3.4) para obter a equação (3.5)

$$V_B = V_{out_1}(R_3 * sC + 1) \quad (3.5)$$

Substituindo a equação (3.5) em (3.3) obtém-se uma relação entre a entrada e a saída do circuito:

$$V_{out_1}(R_3 * sC + 1) = \frac{R_1 + R_2}{R_1} * V_A \quad (3.6)$$

Rearranjando os termos da equação (3.6) chega-se a equação (3.7), que é uma relação entre a entrada e a saída do circuito.

$$\frac{V_{out_1}}{V_A} = \frac{R_1 + R_2}{R_1(R_3 * sC + 1)} \quad (3.7)$$

Substituídos os valores das resistências e da capacitância na equação (3.7), a seguinte função de transferência do circuito é obtida:

$$\frac{V_{out_1}}{V_A} = \frac{101}{3,3 * 10^{-4}s + 1} \quad (3.8)$$

Com o auxílio do MatLab foi feito o diagrama de BODE da equação (3.8). Como se pode observar no gráfico da Figura 3.14, em aproximadamente 500 Hz há uma queda de aproximadamente 3dB, o que equivale uma queda por um fator de $\sqrt{2}$, logo, a frequência de corte desse circuito é de aproximadamente 500 Hz.

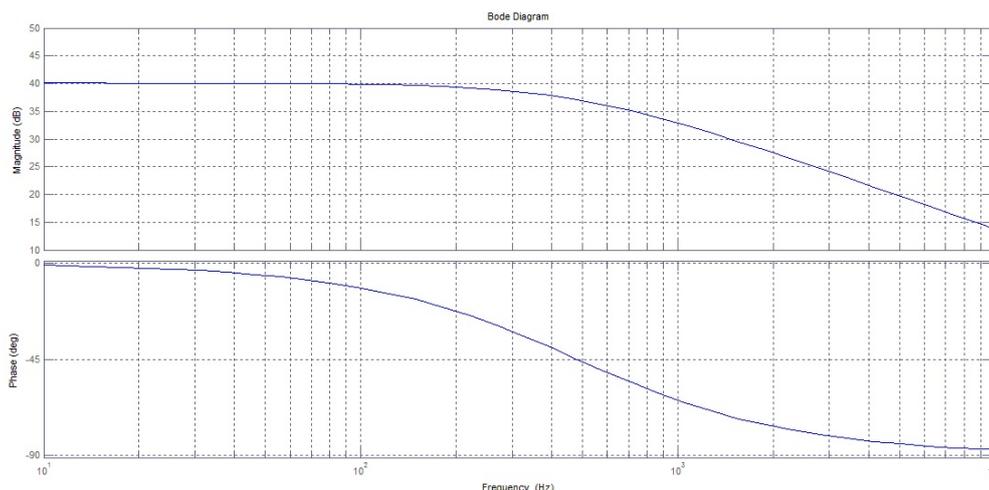


Figura 3.14: Diagrama de BODE e diagrama de fase do circuito passa baixas da Figura 3.13, obtida através da equação (3.8)

Circuito de Envoltória Externo

Utilizou-se mais um estágio de ganho cascadeado com um circuito retificador de meia onda com detecção de envoltória. O circuito em questão ceifa a parte positiva do sinal, porém, como o amplificador esta na configuração inversora (Figura 3.15, amplificador U1:A), a saída do circuito apresentará somente valores positivos, podendo ser ligado diretamente à entrada de um microcontrolador.

O circuito A da Figura 3.15 nada mais é que um amplificador não inversor. Possui um ganho de

$$\frac{R_4 + R_5}{R_4} \quad (3.9)$$

Como a intensidade do sinal de EMG varia de pessoa para pessoa de acordo com o volume condutor. O circuito A é projetado para que se obtenha um o melhor ganho possível para aquele indivíduo.

O circuito B da Figura 3.15, consiste em um retificador de meia onda com detecção de envoltória. Para entender melhor seu funcionamento, considera-se uma senoide de entrada no ponto D. No semiciclo positivo o circuito trabalha de acordo com a Figura 3.16 (a). Como em E tem-se um terra virtual e em F uma tensão negativa, o diodo D₁ encontra-se diretamente polarizado. Porém, como há uma queda de

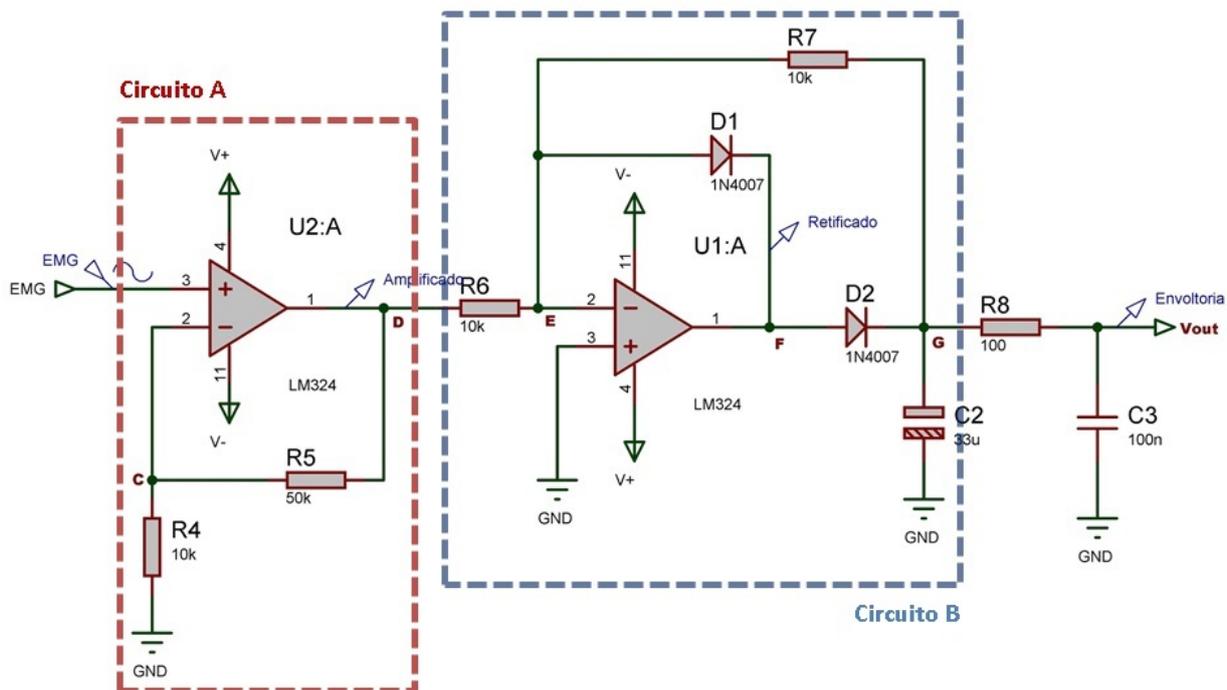


Figura 3.15: Circuito retificador de meia onda com detecção de envoltória: Circuito A: Amplificador não inversor. Circuito B: Retificador de Meia Onda concatenado com filtro passa baixas para detecção de envoltória

tensão equivalente à tensão de ruptura do diodo no loop $E - F$, a tensão em F acaba sendo menor que a tensão em G o que causa o desligamento do Diodo D_2 . Desse modo, o loop $F - G - E$ não é fechado, fazendo com que a tensão em G seja positiva e igual à tensão do capacitor C_2 .

Para o semiciclo negativo, A tensão em F é positiva. Como a tensão em E é zero, desliga-se o diodo D_1 e liga-se o diodo D_2 (Fig. 3.16 (b)), fechando a malha de realimentação negativa $F - G - E$. Desse modo, o circuito passa a operar como um amplificador inversor, com ganho unitário, uma vez que $R_6 = R_7$. Portanto, a tensão em G , será igual a tensão $-V_{in_2}$, carregando o capacitor C_2 com esse valor.

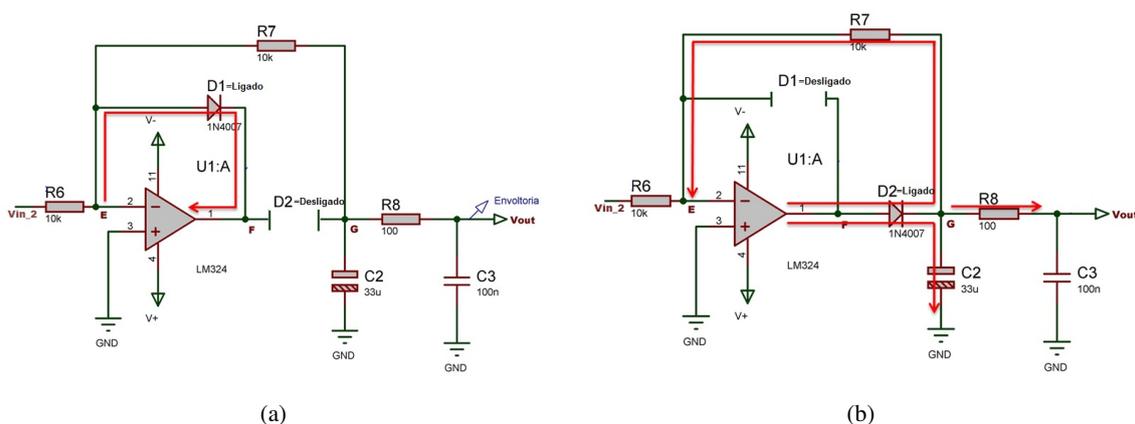


Figura 3.16: Comportamento do circuito retificador de meia onda: (a) durante o semiciclo positivo da entrada V_{in_2} (b) durante o semiciclo negativo da entrada V_{in_2}

subida

3.1.5 Programas para Controle da Prótese

Programas Piloto Para Controle da Garra em Tempo Real com Sinal Simulado

Foram elaborados dois programas para controle da garra com sinal simulado, vide seção 3.1.1, Um para o Computador e outro para o Arduino. O primeiro programa realiza a leitura do sinal de 8 bits (Fig.3.4) e o envia para o microcontrolador Arduino através da porta serial, mostrando em uma interface gráfica o dado que é transmitido. O segundo programa simula a leitura em um conversor A/D de 8 bits através da porta serial e aplica o controle do PWM para rotacionar o servo, que conectado à garra realiza a abertura e fechamento da mesma.

A frequência do sinal simulado é de 2KHz, sendo assim, essa é a frequência de transmissão dos dados entre o computador e o microcontrolador. Neste, uma função programada para execução a cada 500 μ s faz a requisição de um dado de 8 bits pela porta serial. O computador, ao receber a solicitação envia para o microcontrolador uma amostra do sinal. O processo é repetido até o fim do arquivo ou até uma interrupção do programa.

Na abordagem utilizada no teste piloto, a velocidade de abertura e fechamento da garra é fixa e o seu controle é baseado na detecção de um pulso de contração com amplitude significativa. O limiar para abertura e fechamento da garra foi definido como metade do valor RMS da bulha com menor energia do sinal simulado.

Para este primeiro algoritmo, a identificação da contração baseia-se em um detector de envoltória digital quantizado em um bit (Fig. 3.17) e segue o seguinte protocolo:

- Capta-se uma amostra do sinal de EMG simulado a cada 500 μ s;
- Calcula-se o módulo da amostra;
- Compara-se o módulo da amostra com um limiar equivalente à 50 % do menor valor RMS das bulhas utilizadas e obtém-se um valor com nível lógico alto (Fig. 3.17 (b));
- Enquanto o valor acima descrito permanece em nível lógico ativo, um contador redispáravel recebe o valor 400, referente à 0.2 segundos, e se inicia o pulso de contração 3.17 (c)). A cada 500 μ s, se o contador for maior que zero, ele é decrementado em uma unidade. Encerra-se o pulso assim que o contador chegue a zero 3.17 (c));
- Na borda de subida do pulso gerado, ativa-se uma interrupção que controla a garra com PWM. O seu estado muda toda vez que a função é chamada: na primeira vez que ocorre uma borda de subida no pulso de contração, a garra abre, na segunda, ela fecha e assim por diante. Dessa forma, com apenas uma contração, a garra se mantém aberta ou fechada.

O código comentado deste algoritmo se encontra no Anexo I.3. Já o programa de computador que envia os dados se encontra no Anexo I.6.

Programas Para Controle da Prótese em Tempo Real com Sinal Real

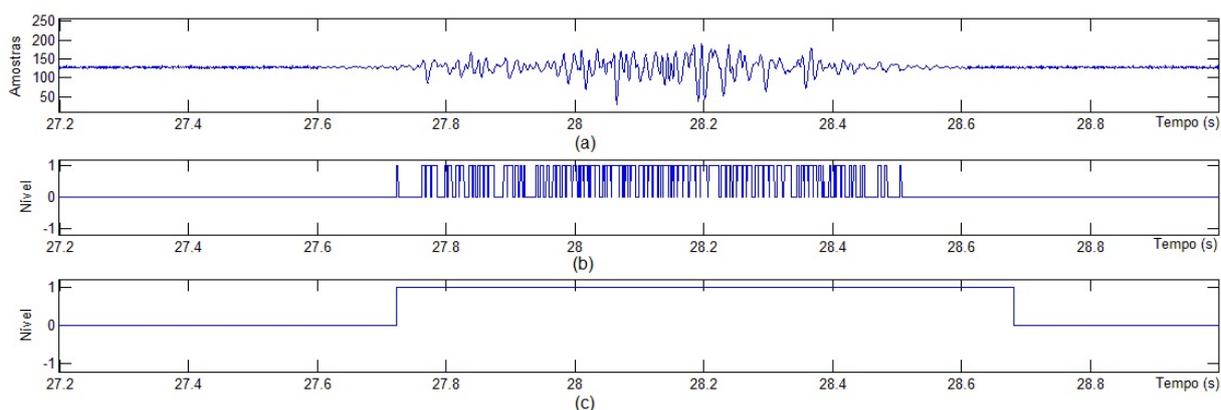


Figura 3.17: Detector de envoltórias digital quantizado em 1 bit: (a) segmento extraído do sinal simulado; (b) detecção dos valores onde o módulo da amostra é maior ou igual ao valor de limiar ϵ ; (c) envoltória do pulso de contração obtido pelo detector de envoltórias digitais de 1 bit

O programa elaborado nesta seção é muito parecido com o descrito para controle com sinal simulado. A principal diferença é que o controle da Prótese é realizado com sinal real tratado com um circuito de detecção de envoltórias e obtido através do conversor A/D do microcontrolador. Dessa forma, para o controle de abertura e fechamento da garra, basta realizar a comparação com um limiar, que foi ajustado de forma empírica para proporcionar conforto ao usuário e ao mesmo tempo em que evita o acionamento indesejado devido ao ruído do instrumento. Outra diferença deste algoritmo é que pulsos com amplitudes acima do limiar e de curtas durações são considerados ruído e descartados do controle de abertura e fechamento da prótese. Dessa forma, realizou-se uma adaptação do algoritmo anteriormente descrito, da seguinte forma:

- Capta-se o sinal de EMG através do conversor A/D a cada $500 \mu s$.
- Compara-se o módulo da amostra com o limiar;
- Caso o motor ainda não tenha sido acionado: se a amostra for maior que o limiar incrementa-se um contador referente a duração do pulso; caso contrário, esse mesmo contador é zerado. Se o contador de duração do pulso chega a um valor de 200, equivalente a 0.1 s, o pulso de controle do servo é disparado e o motor acionado.
- Se a amostra atual permanece acima do limiar, o pulso de controle é mantido em nível lógico alto, senão, o pulso é encerrado. Na borda de subida desse pulso, ativa-se uma interrupção que controla os motores da prótese com PWM. O seu estado muda toda vez que a função é chamada: na primeira vez que ocorre uma borda de subida no pulso de contração, a garra abre, na segunda, ela fecha e assim por diante. Dessa forma, com apenas uma contração com duração suficientemente grande, a garra se mantém aberta ou fechada.

O programa implementado não necessita do controle através de um computador, porém, sua utilização se faz necessária para visualização da contração com intuito de se ajustar o limiar. Por esse motivo, o programa descrito para controle da garra com sinal simulado foi adaptado para receber o valor do sinal de EMG-S captado pelo microcontrolador.

Ao executar o programa, se o microcontrolador estiver conectado na porta serial, ele envia uma so-

licitação para recebimento dos dados pela mesma porta. Com isso, o microcontrolador entra em uma interrupção que habilita o envio de dados para o computador. Além disso, o microcontrolador faz a conversão dos dados de 10 bits para 8 bits com o intuito de permitir uma taxa de transmissão mais alta. Então, o computador recebe os 8 bits de informação a uma frequência de 2 kHz e exibe em tempo real as informações de envoltória do sinal de EMG-S capturadas pelo conversor A/D do microcontrolador. O programa encontra-se disponível no anexo I.7.

3.1.6 Equipamentos Utilizados

Dispositivo de Aquisição de sinais

Para a captação mais precisa do sinal e com maior resolução, utilizou-se o NI USB-6009, que é um instrumento de aquisição de sinais da National Instruments. Esse dispositivo possui 14 bits de resolução do conversor A/D e permite a aquisição de sinais de até 40 KHz. O código desenvolvido para a aquisição de dois canais analógicos a uma taxa de 2 kHz esta disponível no anexo I.5.

Computador

Um computador (marca Samsung, modelo NP300E4C-AD1BR, com processador Intel Core i5 e 2 Gb de memória ram) foi utilizado para realizar a simulação dos sinais, elaboração dos programas e para transmissão e recepção de dados pela porta serial.

Plataforma Arduino

Foi utilizado o microcontrolador Arduino Mega 2560 R3 para realizar o controle da Garra e da Prótese. Essa plataforma é baseado no microcontrolador ATmega2560, que possui 54 pinos de entrada e saída, sendo que 15 podem ser usados como PWM, 16 entradas analógicas, 4 UARTs, um conector de potência, um conector ICSP e um botão de reset. Além de um chip FTDI para a conversão USB/serial, que garante a comunicação do arduino/impressora com o computador.

Fonte de alimentação

Utilizou-se uma fonte de alimentação da Fabricante Minipa e modelo MPL-3303. A fonte possui uma saída fixa de 5 Volts e duas outras fontes variáveis de 0 a 30 Volts e fornecem até 3 Amperes. Tal fonte foi útil para alimentação dos servomotores e os circuitos de instrumentação confeccionados.

Osciloscópio

Com o intuito de avaliar os sinais analógicos gerados pelos circuitos de instrumentação e os sinais Digitais gerados pelo Arduino, recorreu-se a um osciloscópio de 2 canais modelo MSO 2012 da fabricante Tektronix.

Impressora 3D

Para impressão das peças modeladas, foi utilizada a impressora LEIzinha 3D, que foi concebida no Núcleo de Laboratórios de Engenharia e Inovação (LEI). Ela utiliza como sistema de controle o Arduino Mega 2560 R3 (Fig 2.10), com a expansão Ramps 1.4 (Fig 2.11). Essa solução possui localizados todos os circuitos de controle e encaixe para os drivers dos motores de passo em uma única placa.

3.2 PROTOCOLO EXPERIMENTAL

3.2.1 Procedimentos de Impressão e Montagem da Garra

O G-code da Garra foi elaborado com 40 % de preenchimento, camadas de 0.3mm e velocidade média de 30 mm/s. Todas as peças demoraram cerca de três horas e meia para serem impressas. A temperatura de extrusão utilizada foi de 230 ° C sobre a mesa aquecida a 130 ° C. A foto de partes da garra sendo impressa pode ser vista na Figura 3.18 (a).

Para a montagem, confeccionaram-se os pinos com o próprio filamento usado para impressão, que foram cortados e queimados os terminais para fixação das partes móveis. Então, encaixou-se o servomotor, e todas as outras peças plásticas na base impressa e o resultado é apresentado na Figura 3.18 (b).

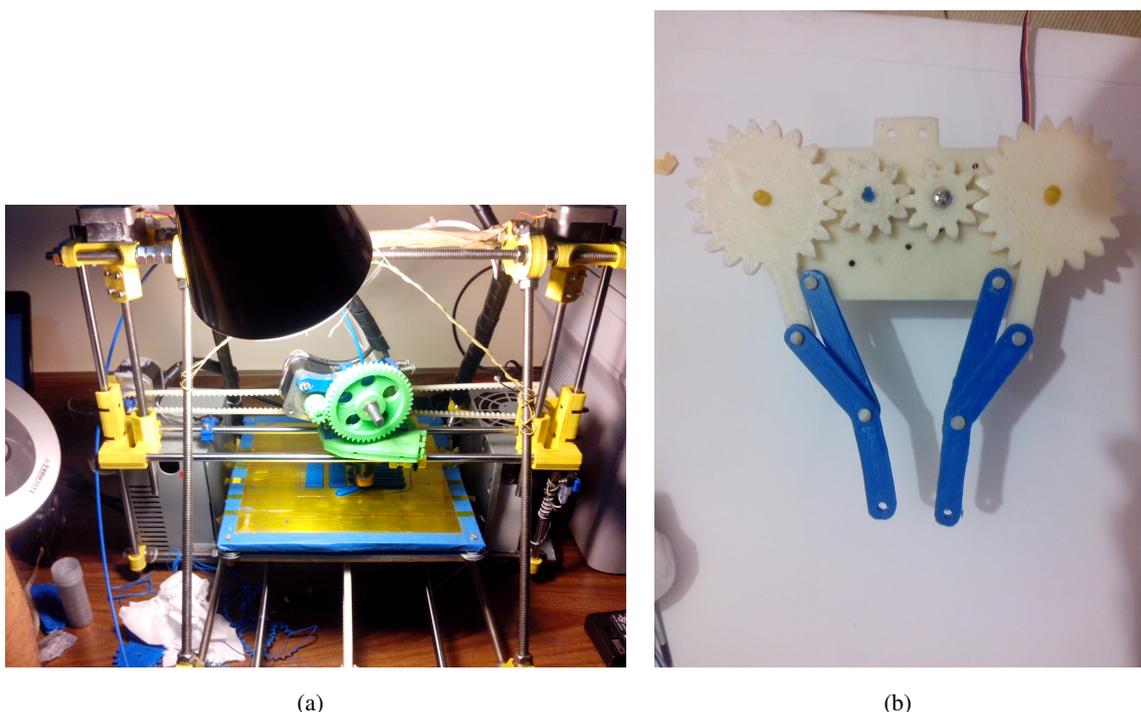


Figura 3.18: (a) Impressão de partes da garra (Fig. 3.18) com a impressora LEIzinha 3D (b) Primeira Garra impressa para validação experimental obtida do site *Thingiverse* [34]

3.2.2 Protocolo Piloto Experimental com Sinal de EMG Simulado

Para realizar o experimento com o sinal simulado conectou-se o pino 9 do Arduino, capaz de implementar PWM, ao servomotor. Este, por solicitar uma corrente maior do que o Arduino pode suprir pela porta USB, foi alimentado com fonte externa de 5V e as referências da fonte e do Arduino foram conectadas. O passo seguinte foi conectar o arduino ao computador pela porta USB.

O software de computador foi desenvolvido para ambiente Linux, além disso, ele utiliza um programa externo chamado *KST*[40] para plotar os dados enquanto é transmitido para o microcontrolador. Dessa forma, para se ter a visualização do gráfico é necessária a instalação do mesmo.

Para iniciar a transmissão de dados do computador para o microcontrolador Arduino, o software desenvolvido para envio dos dados simulados foi executado através do terminal da seguinte maneira:

- Abriu-se o diretório `transmissor_piloto`, onde se encontra o arquivo `transmissor.c`;
- Compila-se o código utilizando o comando `gcc transmissor.c -o transmissor`;
- Conecta-se o Arduino ao computador pela porta USB e verifica-se qual a porta onde o dispositivo foi montado, por exemplo: `/dev/ttyACM0`;
- Executa-se o programa passando-se como argumento a porta obtida no passo anterior. O comando fica então assim: `./transmissor <porta>`.

3.2.3 Procedimentos de Impressão e Montagem da Prótese

As peças utilizadas na confecção da prótese foram impressas utilizando a impressora LEIzinha 3D do LEI. Foram necessárias duas impressões, em uma delas foi impressa a mão robótica, que utiliza materiais flexível e rígido sendo necessária a substituição do plástico durante a impressão, e na outra foram impressos o suporte e as polias, utilizando material rígido.

Impressão da mão robótica

A impressão da mão robótica foi feita com uma temperatura de extrusão de 230°C e com o dorso da mão sobre a mesa aquecida a uma temperatura de 110°C . Todo o processo de impressão durou aproximadamente 2,5 horas.

Como a impressora utilizada possui apenas uma extrusora, o processo de impressão foi dividido em duas etapas. A primeira com o filamento flexível e a segunda com o filamento rígido. Para orientar o momento da troca dos filamentos, modelou-se o dedo com uma pequena saliência na parte lateral, como pode ser observada no dedão da Figura 3.5. Dessa forma, ao fim de sua impressão, pausou-se o processo e substituiu-se o filamento flexível pelo rígido prosseguindo com a impressão.

A união do filamento flexível ao rígido é bastante forte. A deposição do material ABS derretido sobre o flexível apresenta boa ligação, colando uma peça na outra. Além disso, iniciar a impressão com o filamento flexível também diminui o empenamento da peça por variações de temperatura, pois a sua união à mesa de impressão também é melhor que a do ABS, mantendo toda a peça aquecida durante a impressão. A mão impressa pode ser observada na Figura 3.19

A manufatura utilizando a impressora 3D é simplificada permitindo a impressão das peças já com os furos. O filamento flexível possibilita que a mão seja impressa como uma peça única e com juntas flexíveis. Essas características aceleram o processo de montagem, visto que para criar uma prótese utilizando somente material rígido seria necessário um modelo mecânico mais elaborado com diversas peças menores conectadas e utilizando pinos e rolamentos para poder proporcionar o movimento de flexão das juntas. O custo total foi de aproximadamente 264 reais, conforme a tabela I.1 em anexo.

Impressão do suporte e das polias

Como a aderência do plástico ABS à mesa aquecida é menor que a do plástico flexível, foi necessário fazer a limpeza da mesa aquecida utilizando acetona. Como o ABS é solúvel em acetona, ela consegue

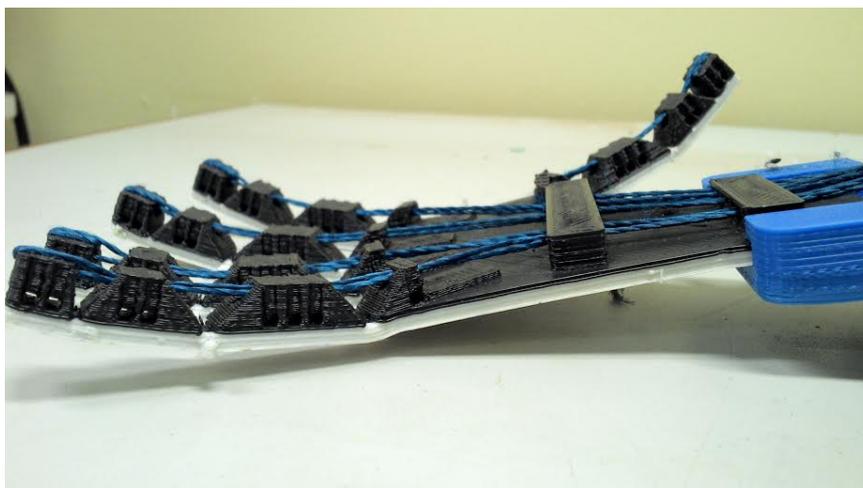


Figura 3.19: Prótese de mão impressa na impressora 3D. A parte branca é o plástico flexível e a parte preta é o plástico rígido (ABS)

remover os restos de plástico provenientes das impressões anteriores. Além disso, limpa também a poeira e a oleosidade que fica na mesa quando se usa a mão para retirar alguma peça. Utilizou-se também uma maior temperatura da mesa aquecida (130°C) com o intuito de aumentar a aderência.

A orientação de impressão foi com a base plana das peças sobre a mesa e com uma temperatura de extrusão de 230°C . Todas foram impressas simultaneamente e o processo durou aproximadamente 2 horas. Assim como na impressão da mão, todos os furos foram elaborados durante o processo de impressão. Na Figura 3.20 pode-se observar o suporte e as polias impressos e já conectados.

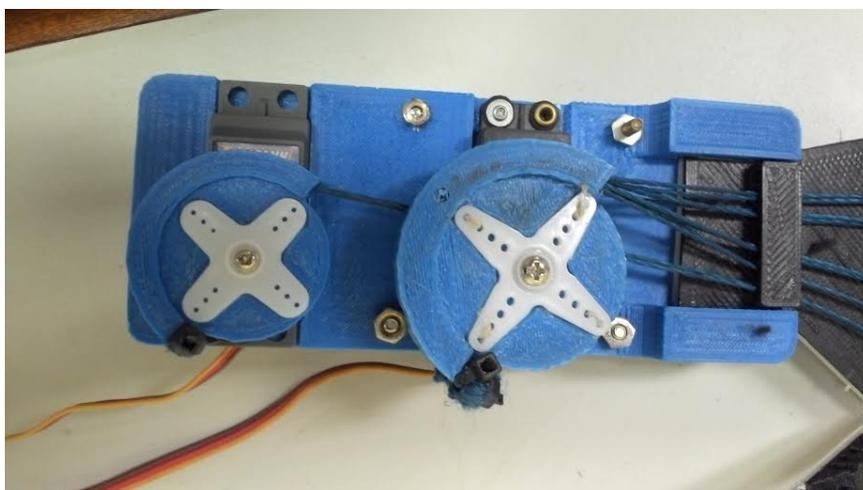


Figura 3.20: Suporte para a mão. Possui encaixe para dois servomotores e encaixe para a mão. Impresso em plástico ABS

Montagem

Depois de concluída a impressão das peças, amarrou-se um fio na ponta de cada dedo. Em seguida, os fios foram passados pelos guias nos dedos e na palma da mão, até chegar aos guias do pulso. Posteriormente, a mão e os servomotores foram encaixados ao suporte. Para parafusar os servos, foram utilizados

parafusos ponta agulha de modo a furar o plástico criando uma rosca. Então as polias foram encaixadas ao eixo dos servomotores, fixando-as com um parafuso. Por fim, conectou-se o fio amarrado à ponta do polegar ao servo que possui menor torque e se encontra na parte posterior do suporte e os outros fios referentes aos outros dedos foram passados pela polia conectada ao servo de maior torque. Na Figura 3.21 pode-se observar a prótese com a montagem da parte mecânica concluída.

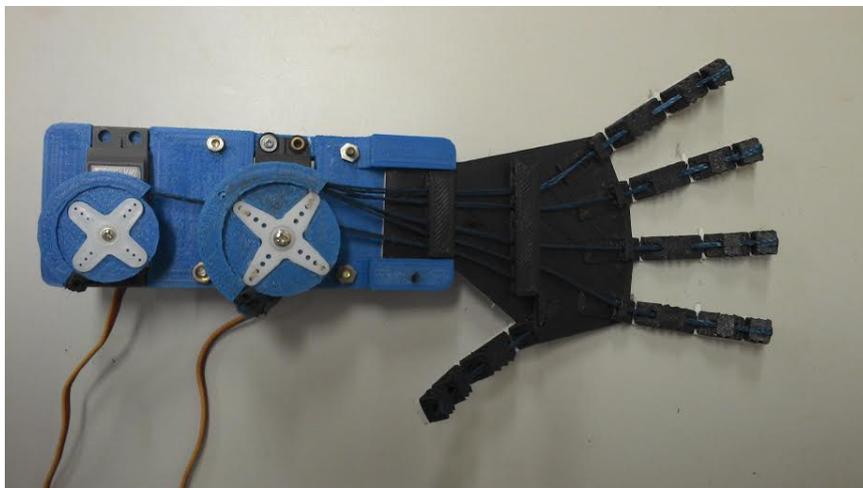


Figura 3.21: Prótese de mão com a parte mecânica concluída. Utilizou-se plástico ABS para impressão do suporte e da mão, sendo esta última combinada com plástico flexível. O servo a esquerda possui menor torque e tem função de flexionar o polegar. O servo a direita tem como função flexionar os demais dedos.

3.2.4 Simulação Numérica do Detector de Envoltória

Foi feita uma simulação do circuito retificador de meia onda com detector de envoltória utilizando o *software* Proteus. Introduziu-se uma senoide de frequência 250 Hz e 250 mV de amplitude na entrada EMG do circuito da Figura 3.15. O resultado pode ser observado na Figura 3.22, onde o sinal verde representa a entrada, o vermelho representa a saída do primeiro amplificador e o azul é a envoltória do sinal. Observa-se que no primeiro semiciclo não há nenhuma alteração na saída do circuito. Isso ocorre porque inicialmente o capacitor está descarregado e nesse ciclo não flui corrente alguma pelo resistor R7. No primeiro semiciclo negativo, fecha-se o laço de realimentação com o resistor R7(Fig 3.16b). Desse modo, o valor da tensão no ponto G é igual ao inverso da tensão de entrada, carregando o capacitor com esse valor. Isso ocorre nos demais ciclos negativos até o circuito atingir a envoltória do sinal. Nota-se ainda um ripple na saída do circuito, ou seja, a cada semiciclo positivo ocorre uma redução da tensão de saída. Esse fato ocorre porque quando o diodo D2 está desligado e não há a realimentação, a corrente gerada pelo capacitor flui pelo resistor R7 que esta conectado ao terra virtual. Resumindo, quando a tensão de entrada é negativa o capacitor é carregado e quando a entrada é positiva, o capacitor é descarregado, porém, em uma velocidade muito menor, fazendo com que a tensão de saída seja aproximadamente constante.

Foi feita uma nova simulação utilizando o sinal de EMG simulado (vide 3.1.1) e seu resultado pode ser observado na Figura 3.23. O sinal da cor vermelha representa a envoltória do sinal de EMG simulado. Deve-se ressaltar que a envoltória obtida diz respeito a parte negativa do sinal de EMG que foi passada por

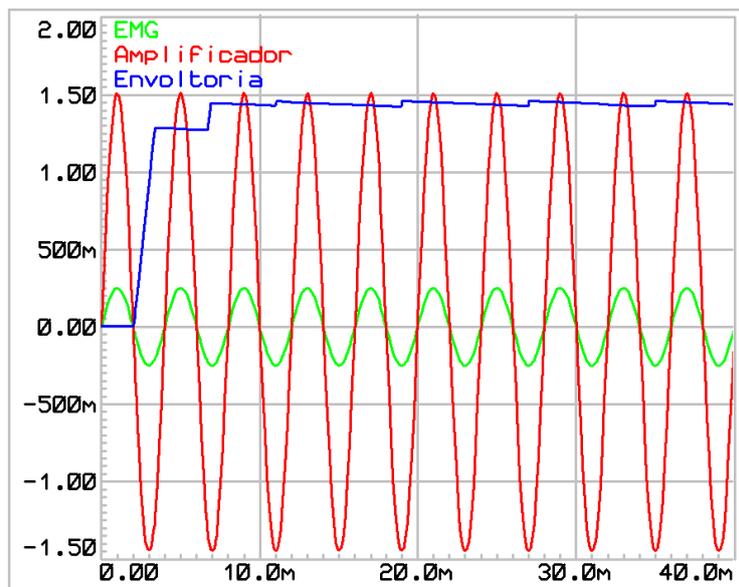


Figura 3.22: Senóide com frequência de 250Hz e amplitude de 250mV introduzida em um circuito retificador de meia onda com detecção de envoltória. A onda verde é a entrada, a onda vermelha é a entrada amplificada e o sinal azul é a envoltória do sinal amplificado.

um circuito inversor, tornando-a positivo.

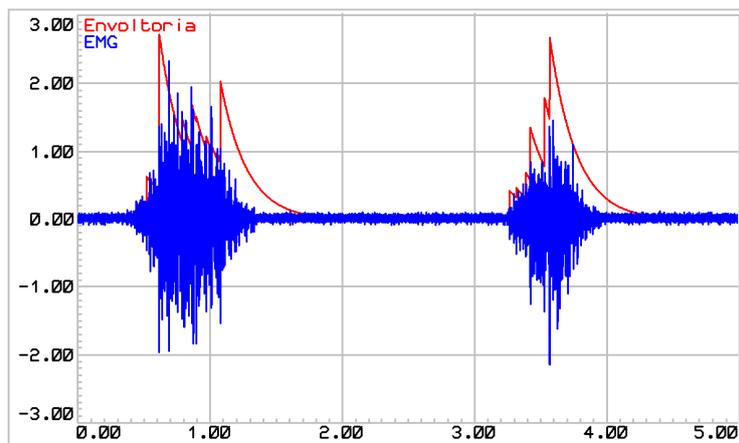


Figura 3.23: Sinal de EMG simulado introduzido em um circuito retificador de meia onda com detecção de envoltória. O sinal azul corresponde ao EMG simulado e o sinal vermelho representa sua envoltória

3.2.5 Instalação e Aplicação do Aparato Instrumental com Sinal Real

Instalação do Aparato Instrumental

Primeiramente, o circuito de detecção de envoltória descrito na seção foi montado na *protobord* (Fig. 3.24). O circuito amplificador permitiu que o ganho fosse ajustado para cada pessoa, como mostrado na Figura, o ganho é de 5.6, porém, caso se perceba saturação no sinal pode-se reduzir o ganho alterando os valores das resistências. Conectou-se também os terminais do eletrodo à *protoboard*. O eletrodo e o

detector de envoltória utilizam alimentação simétrica de +6 e -6 Volts. O circuito montado pode ser visto na Figura 3.1.4.

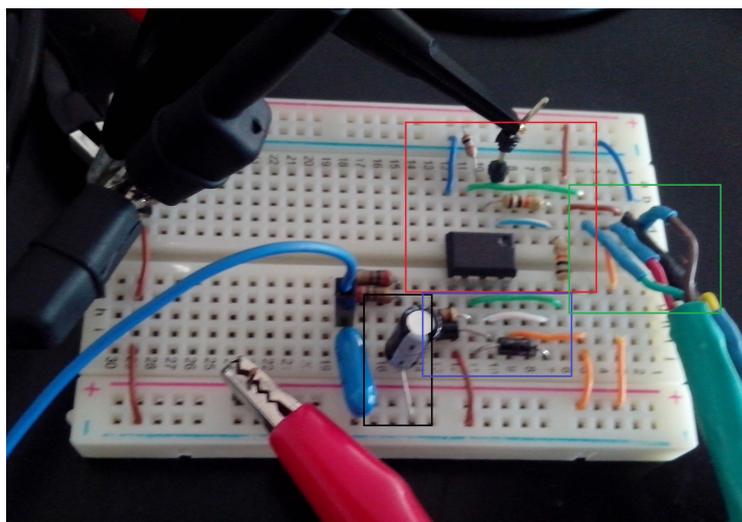


Figura 3.24: Montagem final do circuito retificador de meia onda com detecção de envoltória. Destacado de vermelho é o amplificador com ganho de ajustado para 5.6. Em azul, é a parte do circuito responsável por ceifar a parte positiva do sinal. Em preto tem-se o capacitor responsável por retirar a envoltória. Por fim, em verde, tem-os fios de conexão do eletrodo.

O passo seguinte foi conectar todos os componentes ao Arduino: Os pinos 5 e 6, que fornecem os sinais de PWM, ligam-se aos servomotores, sendo que o segundo pino controla o servo responsável pela contração do polegar e o primeiro controla o outro servo, que contrai os outros dedos; O pino A0 é referente ao conversor A/D e é conectado ao sinal de saída do circuito de detecção de envoltória. Os servos foram alimentados uma fonte de 5V capaz de fornecer 2.5 Amperes.

Aplicação do Eletrodo no Usuário

O eletrodo deve ser aplicado em uma região sem inervações das fibras musculares. A forma utilizada para identificar uma boa posição foi conectando o microcontrolador ao computador para visualizar a envoltória do sinal de EMG-S e capturando contrações em diferentes partes do Bíceps. A posição que apresentou um sinal com melhores características durante os testes foi à uma distância do cotovelo de aproximadamente um terço do comprimento do braço, Fig. 3.25 (h).

Para diminuir o ruído do sinal, é recomendado realizar o protocolo mostrado na Figura 3.25. O uso de fita dupla face para fixação do eletrodo evita que o contato deslize sobre a pele gerando ruído de artefato. A raspagem dos pelos e a limpeza da pele com álcool na região onde se aplica os eletrodos diminui a impedância, resultando na captação de um sinal menos atenuado.



(a) corta-se um pedaço de papel dupla-face do tamanho do eletrodo



(b) cola-se um dos lados da fita no eletrodo, deixando os conectores a mostra



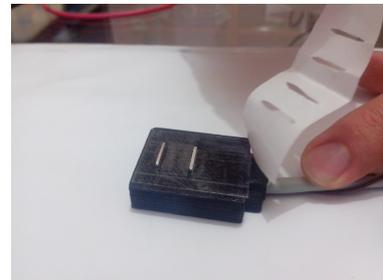
(c) limpa-se a região da pele onde é fixado o eletrodo de referência



(d) fixa-se o eletrodo de referência em região com pouco volume condutor



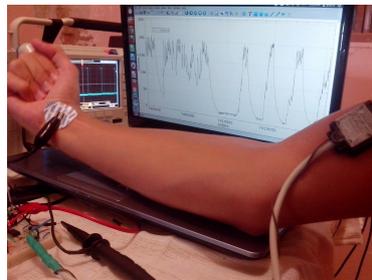
(e) limpa-se o braço com álcool na superfície do músculo onde será posicionado o eletrodo de aquisição



(f) remove-se o outro lado da fita dupla face



(g) fixa-se o eletrodo na região onde será feita a aquisição



(h) o sistema está pronto para realizar a aquisição e o controle da prótese



(i) retira-se o eletrodo da pele no final do processo

Figura 3.25: Procedimento para captura do sinal eletromiográfico e controle da prótese

4 RESULTADOS E DISCUSSÕES

Nesta seção descrevem-se e analisam-se os resultados obtidos. Todos os sinais foram adquiridos à uma frequência de 2 KHz utilizando o software Matlab e um dispositivo de aquisição de sinais de 14 bits de resolução, o NI USB-6009. São realizadas as análises no tempo e na frequência dos sinais obtidos. Além disso, são calculadas as relações Sinal Ruido para 3 voluntários com características distintas e são estimadas os percentuais de energia da contração necessária para acionamento da prótese baseada na contração máxima de cada voluntário.

4.1 AVALIAÇÃO DOS SINAIS DE EMG-S SIMULADOS E REAL

4.1.1 Análise no Tempo e Frequência

Sinal de EMG simulado

A Figura 4.1 ilustra a resposta temporal e na frequência do sinal de EMG simulado utilizado para o controle da garra. Percebe-se que ele está conforme as características de um sinal eletromiográfico captado por um eletrodo de superfície, que possui maior parte de sua energia localizado em frequências de até 500Hz. O valor máximo do eixo y foi ajustado para 0.3, visto que a componente de 0 Hz possui amplitude de 128 devido ao *offset* aplicado ao sinal, e sem esse ajuste não seria possível a visualização das demais componentes de menor amplitude.

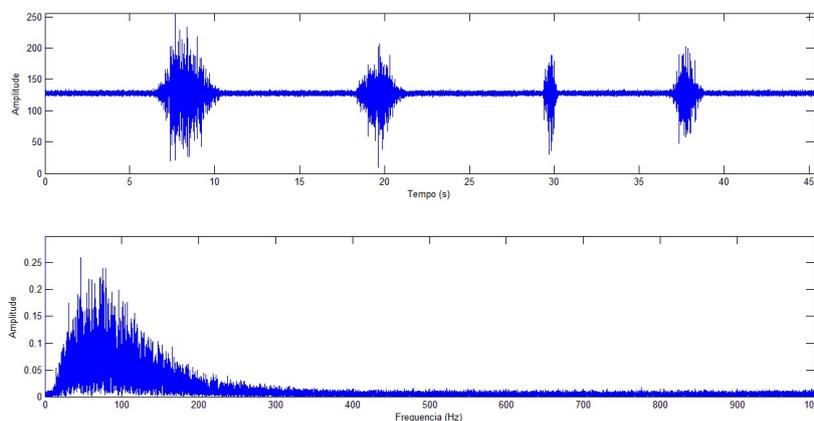


Figura 4.1: Resposta em frequência do sinal de EMG simulado.

Sinais de EMG-S reais

Foram captadas amostras de sinais de EMG de três voluntários, com as características presentes na Tabela 4.1, para teste de controle da prótese.

Voluntário 1

Tabela 4.1: Características dos voluntários

Voluntário	Peso (Kg)	Altura (m)	Idade	Sexo
1	73	1.77	21	Masculino
2	64	1.83	23	Masculino
3	76	1.68	52	Masculino

O primeiro voluntário realizou todo o protocolo para a captura do sinal eletromiográfico. Os resultados podem ser observados nas Figuras 4.2 e 4.3.

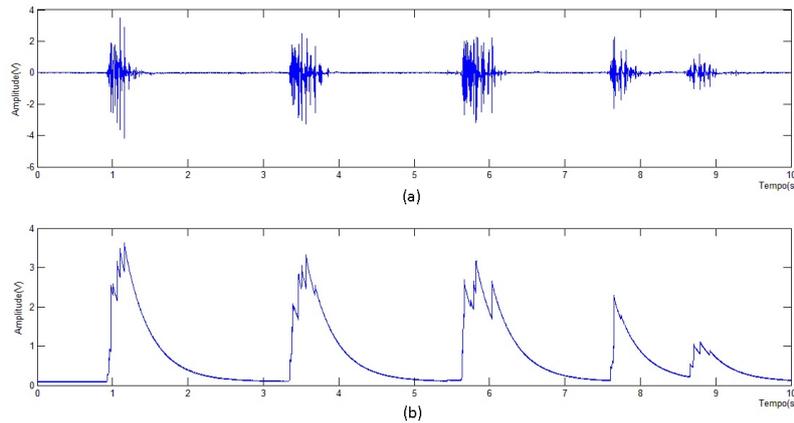


Figura 4.2: (a) Sinal de EMG extraído do bíceps do voluntário 1 e (b) sua envoltória

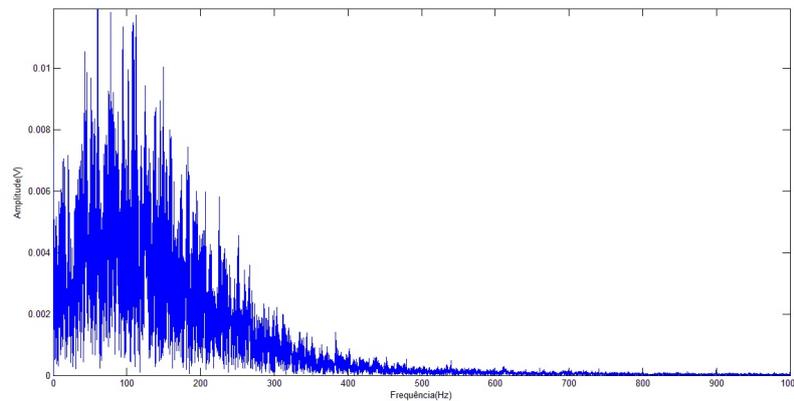


Figura 4.3: Gráfico da resposta em frequência do sinal de EMG extraído do bíceps do voluntário 1

Voluntário 2

O segundo voluntário realizou todos os procedimentos do protocolo para captura do sinal eletromiográfico, exceto a limpeza com álcool da região da pele onde se coloca o eletrodo. Os resultados podem ser observados nas Figuras 4.4 e 4.5.

Voluntário 3

O terceiro voluntário realizou todos os procedimentos do protocolo para captura do sinal eletromiográfico, exceto a limpeza com álcool e a raspagem dos pelos da região da pele onde foi colocado o eletrodo.

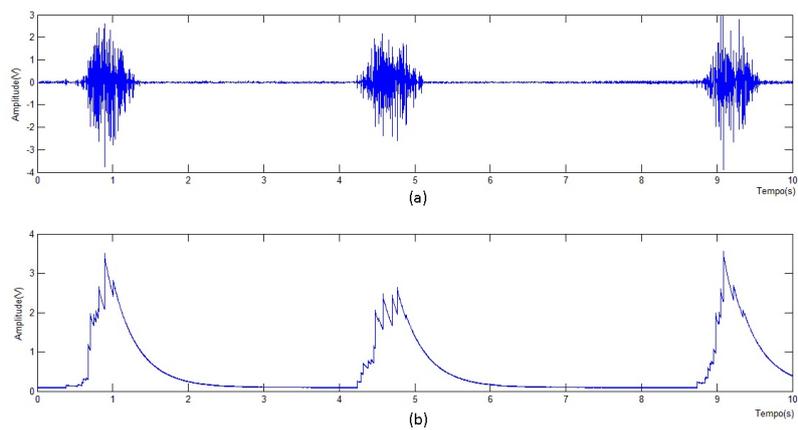


Figura 4.4: (a)Sinal de EMG extraído do bíceps do voluntário 2 e (b) sua envoltória

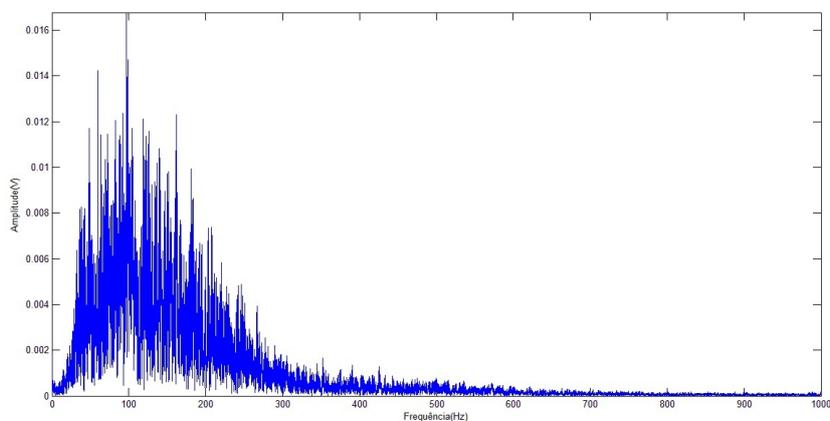


Figura 4.5: Gráfico da resposta em frequência do sinal de EMG extraído do bíceps do voluntário 2

Além disso, este foi o primeiro e único teste realizado com esse voluntário. Os voluntários anteriores já haviam treinado o protocolo de captura dos respectivos sinais. Os resultados podem ser observados nas Figuras 4.6 e 4.7.

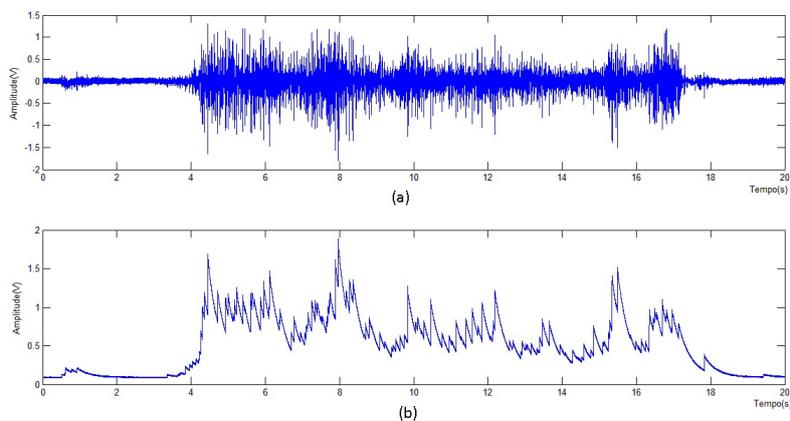


Figura 4.6: (a)Sinal de EMG extraído do bíceps do voluntário 3 em contração livre e (b) sua envoltória

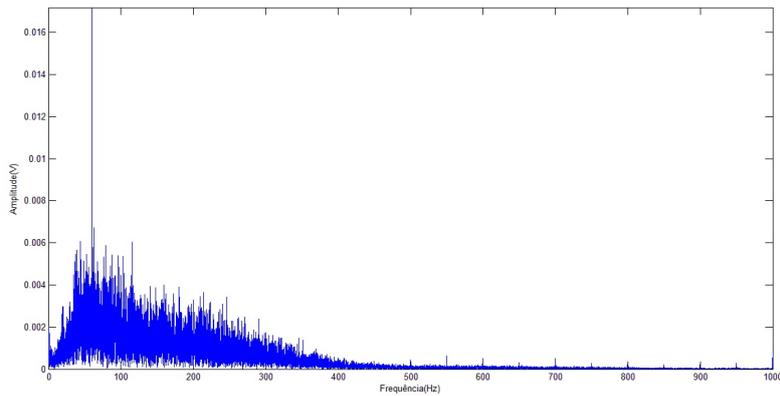


Figura 4.7: Gráfico da resposta em frequência do sinal de EMG extraído do bíceps do voluntário 3 em contração livre

Observa-se uma diferença entre o sinal de EMG do voluntário 1 e do voluntário 2 em relação ao ruído. Nota-se que o voluntário 2, que não fez a limpeza adequada da pele, possui um ruído, ainda que pequeno, mais perceptível que o do voluntário 1. Observando o gráfico da Figura 4.4(a) no tempo de 8 segundos, nota-se componentes de ruído de maior magnitude, o que não é observado no sinal do voluntário 2. Ademais, percebe-se ainda que a resposta em frequência do voluntário 2 possui amostras de maior intensidade nas componentes acima de 500Hz em relação ao voluntário 1.

Analisando o sinal do voluntário 3, que não fez a limpeza adequada da pele, a raspagem dos pelos e nem o treinamento adequado para captação do sinal, observa-se uma grande discrepância, no que diz respeito ao ruído, quando comparado aos demais voluntários. Observando sua resposta em frequência, nota-se que a magnitude da componente de 60Hz, que possui valor bem próximo para todos os voluntários, é bem maior que as demais componentes de frequência, chegando a conclusão que, mesmo em contração máxima, a energia da contração do voluntário 3 é bem menor que a dos demais voluntários. Este fato pode ser justificado por diversos fatores, alguns deles são: posicionamento do eletrodo próximo à região de inervação, pele ressecada, execução errônea do protocolo descrito na Imagem 3.25, sexo, idade, estado psicológico, entre outros [16].

4.1.2 Relação Sinal Ruído

Para o cálculo da relação sinal ruído (SNR), tanto do EMG-S quanto da envoltória, calculou-se o valor RMS de um trecho da contração livre e o valor RMS do ruído, fazendo assim a relação entre os dois. O valor RMS de um sinal com n amostras é definido pela equação (4.1).

$$RMS = \sqrt{\frac{\sum_{i=0}^n (amostra_i)^2}{n}} \quad (4.1)$$

Como no trecho de contração também há ruído, o SNR foi aproximado segundo a equação (4.2). Essa aproximação só pode ser feita se a energia do sinal for muito maior que a energia do ruído.

$$SNR = 20 \log \frac{Energia_do_sinal + Energia_do_ruído}{Energia_do_ruído} dB \quad (4.2)$$

Foram colhidos mais dois sinais de EMG-S referentes aos voluntários 1 e 2, dessa vez com contração livre máxima do bíceps, com intuito de se calcular a máxima relação sinal ruído. Primeiramente, para o voluntário 1, captou-se 10 segundos de ruído, para isso, posicionou-se o eletrodo segundo o protocolo descrito na Figura 3.25, capturando 10 segundos de sinal com o músculo em repouso. O sinal referente ao ruído pode ser visto na Figura 4.8

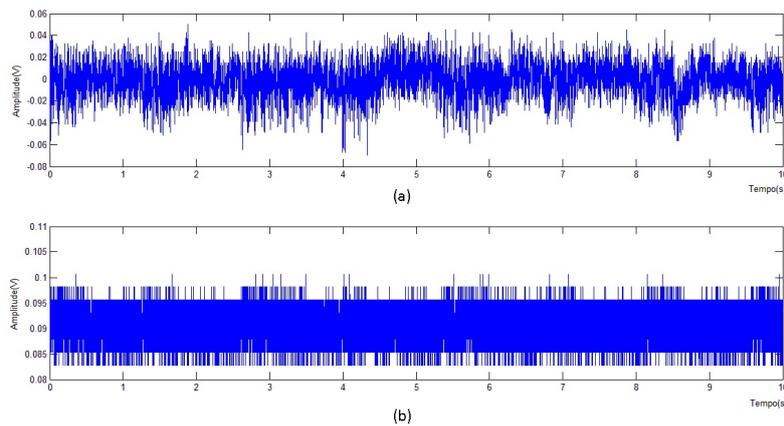


Figura 4.8: (a)Gráfico referente ao ruído do voluntário 1, capturado com o músculo em repouso e (b) sua envoltória

Subsequentemente, capturou-se o sinal de EMG-S do voluntário 1 em contração livre máxima do bíceps durante 20 segundos. O resultado pode ser observado na Figura 4.9. O cálculo do valor RMS do sinal de EMG-S foi feito no intervalo de 5 a 15 segundos e do ruído foi feito para todo o sinal.

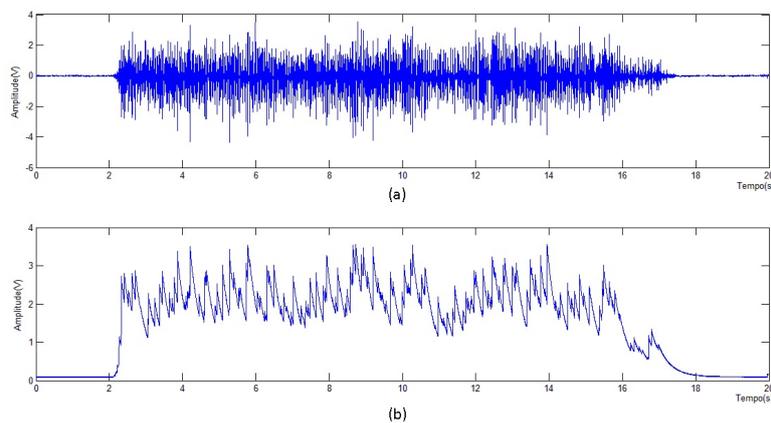


Figura 4.9: (a)Sinal de EMG-S extraído do bíceps do voluntário 1 em contração livre máxima e (b) sua envoltória

O mesmo processo foi repetido para o voluntário 2, que não conseguiu realizar uma contração tão longa quanto o voluntário 1, obtendo um sinal de apenas 15 segundos. O ruído pode ser observado na Figura 4.10 e o EMG-S na Figura 4.11. O valor RMS do sinal EMG-S alusivo ao voluntário 2 foi calculado no

intervalo entre 5 e 10 segundos. Para o cálculo do RMS do ruído, utilizou-se todo o sinal.

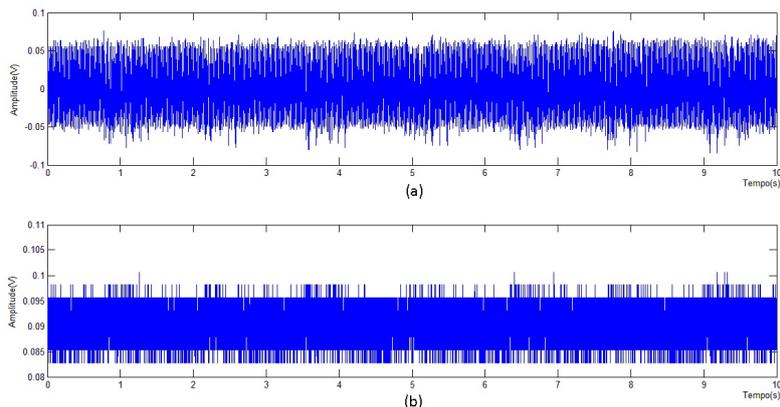


Figura 4.10: (a)Gráfico referente ao ruído do voluntário 2, capturado com o músculo em repouso e (b) sua envoltória

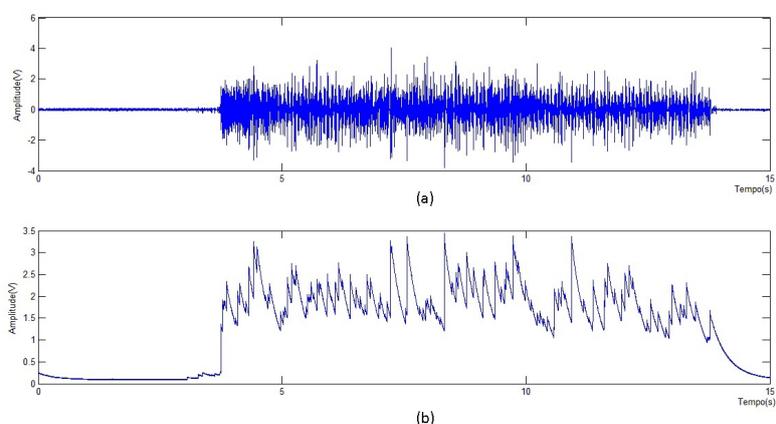


Figura 4.11: (a)Sinal de EMG extraído do bíceps do voluntário 2 em contração livre máxima e (b) sua envoltória

Para o voluntário 3, foram utilizados os mesmos dados do EMG-S da Figura 4.6. O valor RMS do sinal EMG-S foi calculado no intervalo de 5 a 6 segundos e o RMS do ruído foi calculado em uma amostra de 1 segundos deste mesmo sinal. Todos os resultados de SNR podem ser observados na tabela 4.2.

Nota-se que há uma diferença considerável de SNR entre os 3 voluntários. Isso se deve, dentre vários motivos, ao fato dos voluntários 2 e 3 não terem realizado o protocolo adequado para captura do sinal. O voluntário 1 limpou bem a pele com álcool deixando-a um pouco úmida, diminuindo assim impedância da referência de terra, algo que não foi feito pelo voluntário 2, que obteve menos da metade da relação sinal ruído.

Tabela 4.2: Relação sinal ruído para os diferentes voluntários

Dados		Voluntário		
		1	2	3
EMG-S	RMS	0.723 V	0.777 V	0.281 V
	RMS do ruído	0.016 V	0.036 V	0.033 V
	SNR	33.26 dB	26.68 dB	18.55 dB
Envoltória	RMS	2.218 V	2.084 V	0.969 V
	RMS do ruído	0.091 V	0.091 V	0.091 V
	SNR	27.72 dB	27.19 dB	20.50 dB

4.2 EXPERIMENTOS COM SINAL SIMULADO

O experimento com a garra e sinal simulado apresentou o resultado esperado, o microcontrolador foi capaz detectar a contração a partir do sinal simulado e gerar o pulso PWM para controlar o servomotor, abrindo e fechando a garra (Fig. 4.12). Além disso, o experimento facilitou a elaboração do programa para controle com sinal real.

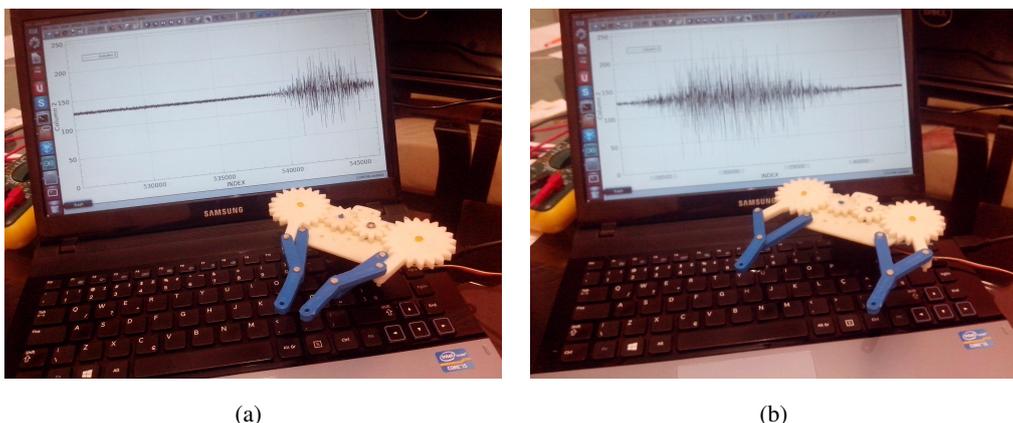


Figura 4.12: Resultado final da garra controlada por sinal eletromiográfico simulado

4.3 EXPERIMENTOS COM SINAL REAL

Comprovou-se o correto funcionamento utilizando o controle em tempo real a partir do sinal proveniente do eletrodo ativo de superfície (Fig. 4.13). A impressão da prótese realizada com juntas flexíveis forneceu os resultados esperados de proporcionar flexibilidade às juntas e retração passiva dos dedos. Por fim, a luva de lã cobrindo a mão forneceu uma aparência mais humanizada ao mesmo tempo em que permitiu maior aderência da prótese ao segurar objetos.

O controle da prótese foi feito por todos os voluntários de modo condizente com o esperado. Até mesmo o voluntário 3, que apresentou a menor relação sinal ruído, realizou o controle da prótese sem problemas. O limiar para abertura e fechamento da prótese foi ajustado para 0.73V para se manter em

Tabela 4.3: Porcentagem do nível de contração para ativar a prótese (0.73) em relação ao nível máximo aplicado pelos voluntários

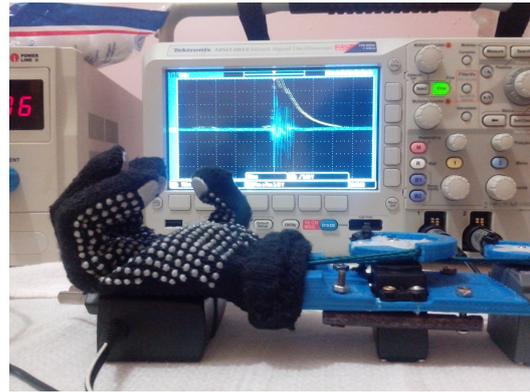
Voluntário	Porcentagem
1	33%
2	35%
3	76%

uma posição intermediária entre o conforto do usuário e a imunidade ao ruído. Visto que para todos os voluntários, o valor RMS referente à envoltória do ruído se manteve em torno de 0.091V, a contração utilizada para o acionamento da prótese foi ajustada de forma empírica em cerca de 7 vezes o nível do ruído. A porcentagem da energia do limiar em relação à energia máxima empregada por cada voluntário para o controle da prótese é apresentada na Tabela 4.3.

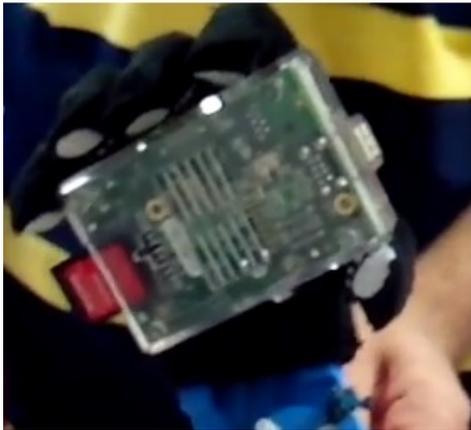
Como pôde ser visto nos resultados obtidos, o usuário 3 precisou de uma contração relativamente forte para o acionamento da prótese, o que geraria desconforto ao usuário. Para solucionar o problema, a recomendação é realizar o procedimento de raspagem e limpeza das regiões onde os eletrodos são posicionados, conforme o protocolo apresentado na Figura 3.25, ou escolher um outro músculo no qual o usuário da prótese tenha maior controle. Além disso, é importante realizar o treinamento do usuário para melhor utilização do músculo no qual o eletrodo de aquisição é conectado.



(a)



(b)



(c)



(d)

Figura 4.13: Resultado final da prótese ativa de mão controlada por sinal eletromiográfico de superfície

5 CONCLUSÕES

No presente Trabalho de Conclusão de Curso foi apresentada uma metodologia para desenvolver uma prótese ativa de mão utilizando impressora 3D. Na prototipagem, o plástico maleável funcionou bem como mecanismo de juntas flexíveis e proporcionou a volta passiva dos dedos à posição aberta. Quanto ao circuito de instrumentação, propôs-se um modelo para tratamento do sinal eletromiográfico com eletrodo ativo e um circuito de detecção de envoltória. Para a captação do sinal, foi proposto um protocolo de limpeza e posicionamento do eletrodo para se obter melhor relação sinal ruído. Utilizou-se o microcontrolador Arduino de modo a transformar a envoltória extraída do sinal de EMG-S em um sinal de controle para abertura e fechamento da prótese. O resultado foi uma prótese funcional controlada por eletromiografia de superfície capaz de realizar funções básicas da mão, como agarrar objetos e permitir um aperto de mão.

A prótese confeccionada utilizando material rígido, que simula os ossos da mão, e o flexível, que simula articulações, atingiu os resultados esperados. O seu baixo custo e a facilidade de reposição das peças com impressora 3D permite que a prótese seja acessível a indivíduos que necessitem desse recurso, mas não possuam meios para aquisição das próteses existentes no mercado. Além disso, a versatilidade na customização dos projetos permite a confecção de próteses adaptadas ao tipo de amputação do indivíduo, tornando a alternativa ainda mais atraente. A metodologia de concepção de próteses com impressão 3D não se restringe à projetos de próteses convencionais e pode explorar melhor as necessidades do usuário, isto é, pode-se projetar próteses que possuam outras ferramentas úteis além da mão em si. Exemplos de projetos podem ser: prótese com compartimentos para guardar pequenos objetos, ou próteses com suportes para ferramentas de trabalho, dentre inúmeros outros projetos.

Percebeu-se que a utilização do eletrodo ativo resultou na aquisição de um sinal pouco ruidoso comparado com a captação utilizando eletrodos passivos (Figura I.3 em anexo). Isso se deve ao fato do amplificador de instrumentação INA 118, que possui uma alta rejeição de modo comum e uma baixa tensão de *offset*, estar localizado dentro do eletrodo próximo à pele, amplificando o sinal já na entrada do circuito antes da inserção de ruídos térmico, eletromagnético, entre outros. O eletrodo também possui um filtro passa baixas com frequência de corte de 500Hz, que é a banda onde se localiza maior parte da energia do sinal de EMG-S. Esse tipo de eletrodo é bastante utilizado para diversos tipos de aplicações no âmbito de biomédica, visto que o sinal capturado, mesmo não sendo tão fiel quanto o captado por eletrodos invasivos, possui uma alta relação sinal ruído.

Viu-se essencial seguir um protocolo para aquisição do sinal de EMG-S, realizando a raspagem dos pelos e a limpeza e umidificação da pele, com intuito de se obter um sinal com baixa contaminação por ruído. Entretanto, mesmo com sinais contendo certa contaminação, foi possível realizar o controle da prótese, visto que, para esses casos, pode-se adequar o limiar para seu acionamento de modo que ele seja ultrapassado somente quando houver uma contração de maior intensidade. Porém, nesse caso é gerado certo desconforto ao usuário, pois este precisa fazer um maior esforço físico para ultrapassar o limiar de acionamento. Logo, o ideal é realizar todo o protocolo de posicionamento dos eletrodos para diminuição do ruído (Fig 3.25) e o treinamento do usuário para captação do sinal de EMG-S.

O programa proposto para controle da prótese em tempo real, utilizando o microcontrolador Arduino combinado com um circuito detector de envoltória, foi bastante vantajoso em relação ao proposto no teste piloto, de introduzir um *offset* no sinal para ajuste à faixa de captura do conversor AD. Essa metodologia simplificou consideravelmente o algoritmo de abertura e fechamento da prótese, permitindo que o controle dos servomotores fosse baseado apenas em uma comparação com o limiar estabelecido para acionamento da prótese. Com o intuito de evitar ruídos de artefatos, o programa no microcontrolador despreza sinais com durações muito curtas, 0.1 segundos, contudo, é imperceptível ao usuário. A detecção de uma contração gera uma borda de subida no microcontrolador, cada vez que se detecta tal variação, muda-se o estado atual entre aberto e fechado. Dessa forma, evita-se o desconforto proveniente de se manter a contração para que a garra permaneça fechada. O controle com detector de envoltórias do modo como foi proposto poderia também ser realizado sem o uso de um sistema microcontrolado, bastando apenas um circuito comparador e um gerador de PWM para controlar os servomotores e uma lógica com flip-flops.

Sugestões para trabalhos futuros são a utilização de baterias e reguladores de tensão internos à prótese, com o intuito de fazer o controle sem a utilização de uma fonte externa, e a concepção do eletrodo integrado com o circuito de detecção de envoltórias. Propõe-se também a utilização de um número maior de eletrodos combinados com algoritmos inteligentes para controle da prótese com um maior número de graus de liberdade. Sugere-se ainda uma mudança na parte estrutural da mão introduzindo mecanismos de travas em seu fechamento, de modo que o servomecanismo não precise estar sempre polarizado para manter a mão fechada.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] REPRAP - Main Page. 2013. Acesso em 26 de junho de 2013. Disponível em: <<http://reprap.org/>>.
- [2] BRITANNICA, E. *prosthesis*. 2013. Acesso em 14 de maio de 2013. Disponível em: <<http://www.britannica.com/EBchecked/topic/479532/prosthesis>>.
- [3] BRITANNICA, E. *prosthesis*. 2013. Acesso em 14 de maio de 2013. Disponível em: <<http://www.britannica.com/EBchecked/topic/21786/amputation>>.
- [4] PUTTI, V. Historical prostheses. *Journal of Hand Surgery (British and European Volume)*, v. 30, n. 3, p. 310–325, 2005. Disponível em: <<http://jhs.sagepub.com/content/30/3/310.short>>.
- [5] RICHARD, F. *Standard Handbook of Biomedical Engineering and Design*. [S.l.]: McGraw-Hill, 2004.
- [6] PILLET, J.; DIDIERJEAN-PILLET, A. {AESTHETIC} {HAND} prosthesis: {GADGET} {OR} therapy? {PRESENTATION} {OF} a {NEW} {CLASSIFICATION}. *The Journal of Hand Surgery: British e European Volume*, v. 26, n. 6, p. 523 – 528, 2001. ISSN 0266-7681. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0266768101906588>>.
- [7] BOCCOLINI, F. *Reabilitação, Amputados - Amputações - Próteses*. segunda. [S.l.]: Robe Editorial, 2000.
- [8] SAÚDE, M. da. *Diretrizes de Atenção à pessoa Amputada*. 2013. Acesso em 29 de junho de 2013. Disponível em: <http://bvsmms.saude.gov.br/bvs/publicacoes/diretrizes_atencao_pessoa_amputada.pdf>.
- [9] RESNIK, L. et al. Advanced upper limb prosthetic devices: Implications for upper limb prosthetic rehabilitation. *Archives of Physical Medicine and Rehabilitation*, v. 93, n. 4, p. 710 – 717, 2012. ISSN 0003-9993. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0003999311009750>>.
- [10] FILAFLEX. *WELCOME TO RECREUS -REVOLUTION OF 3D PRINTING MATERIALS*. 2013. Acesso em 02 de dezembro de 2013. Disponível em: <<http://recreus.com/>>.
- [11] THAKOR, N. V. Building brain machine interfaces - neuroprosthetic control with electrocorticographic signals. *Life Science*, v. 181, 2012. Disponível em: <<http://lifesciences.ieee.org/publications/newsletter/april-2012/96-building-brain-machine-interfaces-neuroprosthetic-control-with-electrocorticographic-signals>>.
- [12] MUÑOZ, G. et al. Evaluating different combinations of feature selection algorithms and cost functions applied to ipca tuning in myoelectric control systems. In: *Engineering in Medicine and Biology Society (EMBC), 2012 Annual International Conference of the IEEE*. [S.l.: s.n.], 2012. p. 6508–6513. ISSN 1557-170X.
- [13] PROSTHETIC Electrical and Powered/ Myoelectric Hand Prosthetics. [S.l.].

- [14] POTLURI, C. et al. Optimal tracking of a semg based force model for a prosthetic hand. In: *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*. [S.l.: s.n.], 2011. p. 1604–1607. ISSN 1557-170X.
- [15] I-LIMB ultra. 2013. Acesso em 13 de maio de 2013. Disponível em: <<http://www.touchbionics.com/products/active-prostheses/i-limb-ultra/>>.
- [16] MERLETTI, R.; PARKER, P. A. *ELECTROMYOGRAPHY Physiology, Engineering, and Noninvasive Applications*. [S.l.]: EMB - IEE Press Editorial Board, 2004.
- [17] BEGG, R.; LAI, D. T. H.; PALANISWAMI, M. *Computational Intelligence in biomedical Engineering*. first. [S.l.]: CRC Press, 2008.
- [18] GUYTON, A. C.; HALL, J. E. *Tratado de Fisiologia Médica*. decima primeira. [S.l.]: Elsevier, 2006.
- [19] ANDRADE, N. A. *Desenvolvimento de um Sistema de Aquisição e Processamento de Sinais Eletromiográfico de Superfície para a Utilização no Controle de Próteses Motoras Ativas*. Faculdade de Tecnologia Departamento de Engenharia Elétrica: [s.n.], 2007.
- [20] CUNHA, F. L. da. *Mão de São Carlos - Uma Prótese Multifunção para membros Superiores - Um estudo dos Mecanismos, Atuadores e Sensores*. Escola de Engenharia de São Carlos: [s.n.], 2002.
- [21] SONO, T. S. P. *Projeto de um Sistema de Controle Sub-atuado para uma Pótese de Mão*. Ministério da Defesa-Departamento de Ciência e Tecnologia: [s.n.], 2008.
- [22] LUCA, C. J. D. *Surface Electromyograph - Detection and Recording*. 2013. Acesso em 29 de junho de 2013. Disponível em: <http://www.delsys.com/Attachments_pdf/WP_SEMGintro.pdf>.
- [23] LUCA, C. D. Physiology and mathematics of myoelectric signals. *Biomedical Engineering, IEEE Transactions on*, BME-26, n. 6, p. 313–325, 1979. ISSN 0018-9294.
- [24] SOARES, F. A. *Estudo do Comportamento de Variáveis Eletromiográficas ao Longo do Ciclo Menstrual*. Departamento de Engenharia Elétrica da UnB: [s.n.], 2007.
- [25] MARLIN 3D Printer Firmware. 2013. Acesso em 2 de julho de 2013. Disponível em: <<https://github.com/ErikZalm/Marlin>>.
- [26] ARDUINO Mega 2560. 2013. Acesso em 24 de novembro de 2013. Disponível em: <<http://arduino.cc/en/Main/arduinoBoardMega2560>>.
- [27] RAMPS Pre-Assembled Kit. 2013. Acesso em 24 de novembro de 2013. Disponível em: <<https://ultimachine.com/content/ramps-pre-assembled-kit>>.
- [28] POLULU - A4988 Steper Motor Driver Carrier. 2013. Acesso em 24 de novembro de 2013. Disponível em: <<http://www.pololu.com/product/1182>>.
- [29] BRITANNICA, E. *numerical control*. 2013. Acesso em 24 de novembro de 2013. Disponível em: <<http://global.britannica.com/EBchecked/topic/422398/numerical-control>>.

- [30] BRITANNICA, E. *point-to-point*. 2013. Acesso em 24 de novembro de 2013. Disponível em: <<http://global.britannica.com/EBchecked/topic/466386/point-to-point>>.
- [31] BRITANNICA, E. *continuous-path*. 2013. Acesso em 24 de novembro de 2013. Disponível em: <<http://global.britannica.com/EBchecked/topic/135122/continuous-path>>.
- [32] ESPOSTE, A. de M. D.; SALES, A. B. de; ANDRADE, M. M. de. A aplicação do crowdfunding e da aprendizagem baseada em problemas em projetos acadêmicos colaborativos. 2012. COBEM XL.
- [33] ANDRADE, M. M. *Análise Digital Eletromiográfica para Esforços Isométricos*. Departamento de Engenharia Elétrica: [s.n.], 2000.
- [34] THINGIVERSE, M. *Parametric Robot Claw*. 2013. Acesso em 29 de novembro de 2013. Disponível em: <<http://www.thingiverse.com/thing:27468>>.
- [35] MASSA, B. et al. Design and development of an underactuated prosthetic hand. *Proceedings of the 2002 IEEE International Conference on Robotics & Automation*, p. 3374 – 3379, 2002.
- [36] SLIC3R - G-code generator for 3D printers. 2013. Acesso em 24 de novembro de 2013. Disponível em: <<http://slic3r.org/>>.
- [37] DELSYS. *Surface EMG Sensors*. 2013. Acesso em 02 de dezembro de 2013. Disponível em: <<http://www.delsys.com/Products/EMGSensors.html>>.
- [38] BURR-BROWN. *Precision, Low Power INSTRUMENTATION AMPLIFIER*. 2000. Acesso em 26 de novembro de 2013. Disponível em: <<http://pdf.datasheetcatalog.com/datasheet/texasinstruments/ina118.pdf>>.
- [39] MOURA, I. L. B. de. *Sistema para Aquisição sem Fio dos Sinais de Eletromiografia de Superfície Baseado no Protocolo IEEE 802.15.4*. Departamento de Engenharia Eletrônica: [s.n.], 2013.
- [40] WEBMASTER. *Kst - Visualize your data*. 2011. Acesso em 01 de dezembro de 2013. Disponível em: <<http://kst-plot.kde.org/>>.

I. FIGURAS E TABELA DE CUSTOS

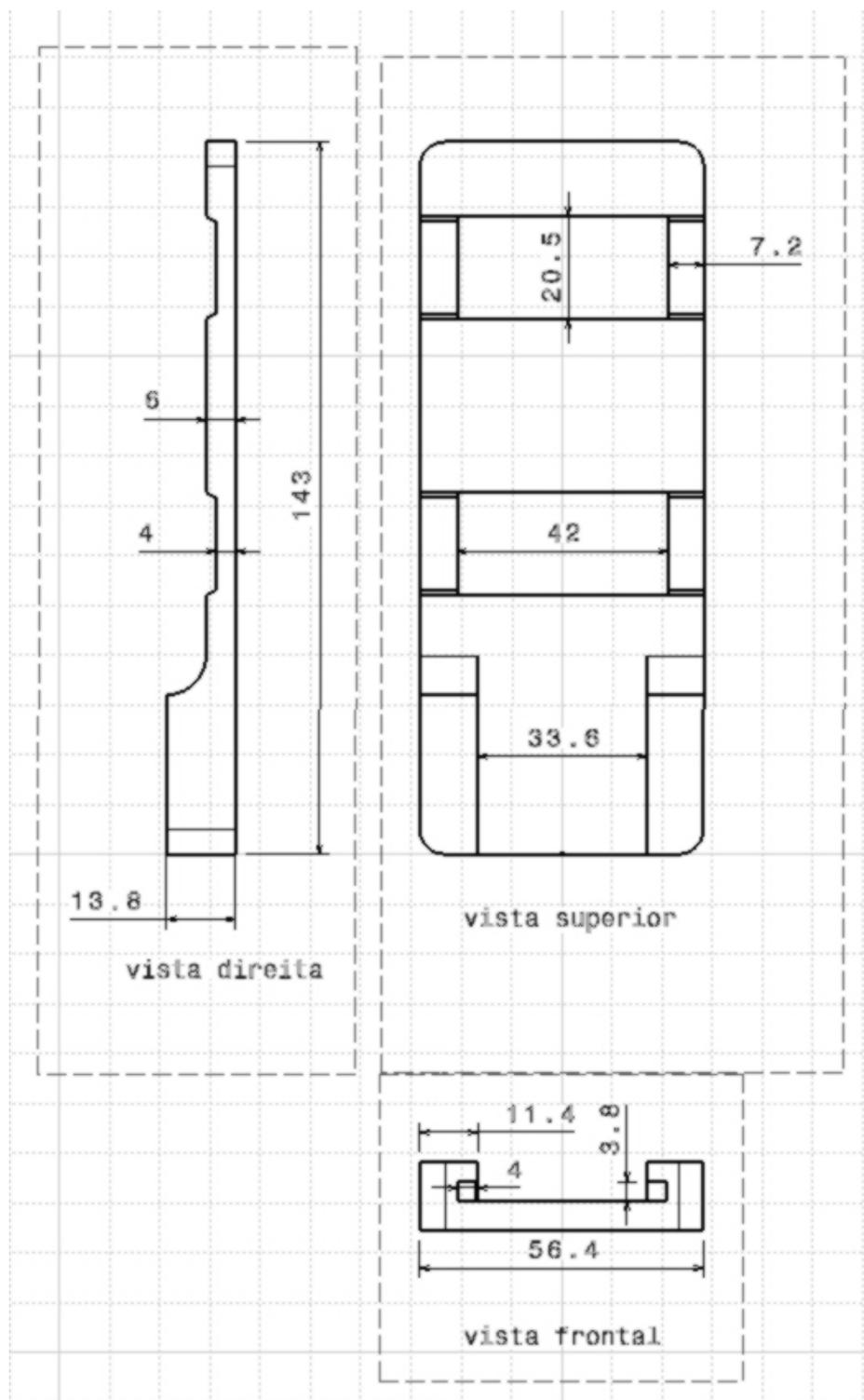


Figura I.1: Vistas superior, frontal e esquerda do suporte modelado para confecção da prótese

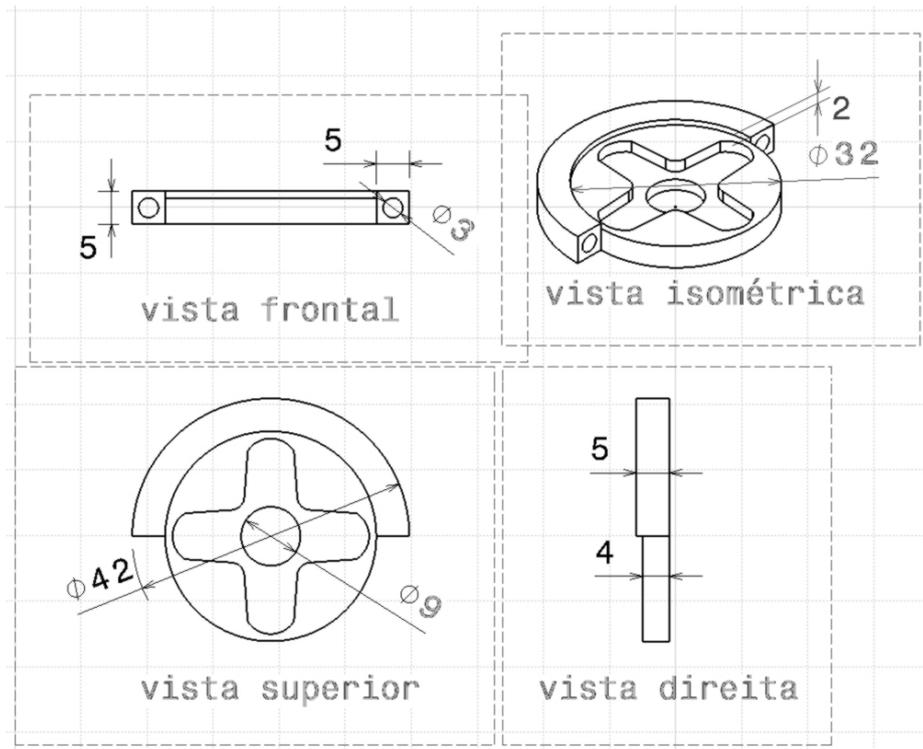


Figura I.2: Vistas superior, frontal, isométrica e esquerda da polia modelada para confecção da prótese

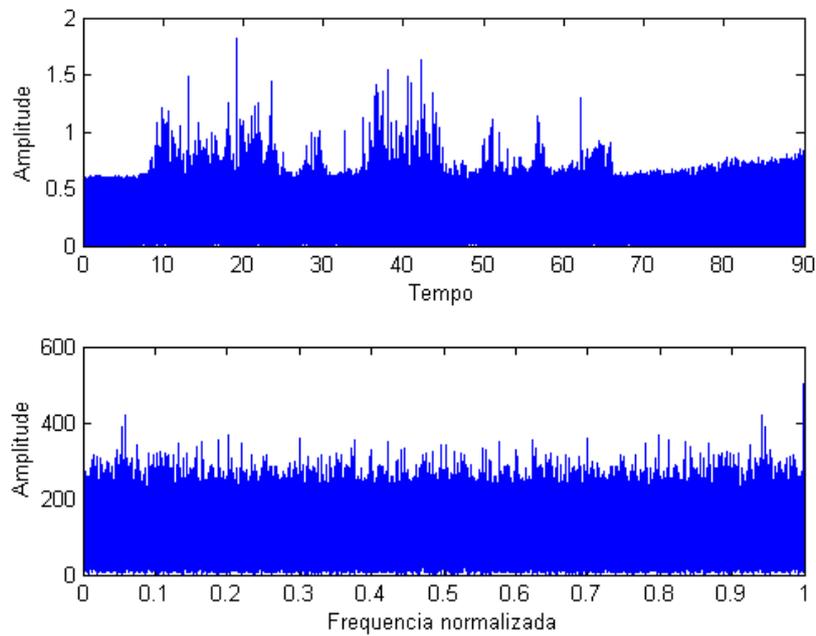


Figura I.3: Sinal de EMG-S capturado com eletrodo passivo. O sinal apresentou alta contaminação com ruído conforme pode ser visto na resposta do sinal no tempo e na frequência.

Tabela I.1: Custos para confecção e montagem da prótese apresentada. adquiridos no Brasil

Item	Modelo	Peso (g)	Custo (R\$)
Servomotores	Tower-Pro MG996R (10 Kg/cm)	52	58,00
	HK15138 (4.3 Kg/cm)	38	24,00
Plásticos	ABS	64	7,70
	Filaflex	22	6,82
Contatos (eletrodo)	liga Ag/AgCl	-	30,00
Componentes eletrônicos	INA 118	-	43,00
	Arduino Mega R3 2560	-	73,00
	Resistores e capacitores	-	2,00
	LM358	-	1,09
	LM324 SMD	-	2,58
	PCB fibra de vidro	-	1,00
Outros	Fios, protoboard 400 furos e cabos	-	15,00
Total	-	180	264,19

I. DESCRIÇÃO DO CONTEÚDO DO CD

- relatório de conclusão de curso (relatorio.pdf);
- imagens utilizadas no TCC (diretório figs);
- arquivos do latex para gerar o relatório (relatorio.tex, resumos.tex, anexo_CD.tex);
- arquivos com referências utilizadas (relatorio.bib);
- modelos utilizados para elaboração do documento (ft2unb.sty, abntex2–alf.bst);
- modelos para mudar o estilo do documento (ft1unb.sty, abntex2–num.bst);
- códigos dos programas elaborados (diretorio codigos);
- layouts e simulações no software Proteus (diretorio Proteus);

I. CÓDIGOS FONTES COMENTADOS

I.1 CÓDIGO PARA GERAR BULLHAS UTILIZADAS NA CRIAÇÃO DO SINAL

```
1 clear all;
2 close all;
3 clc
4
5 fs=2000; %frequencia de amostragem
6 load a1 %carrega o sinal de EMG
7
8 %% Gera as bolhas do sinal de EMG. Cada uma delas representa um pulso.
9 N=50000;
10 t=4;
11 bul1=(hamming(t*2000).*a1(N:(N+t*2000-1)))';
12
13 N=75000;
14 t=3;
15 bul2=(hamming(t*2000).*a1(N:(N+t*2000-1)))';
16
17 N=100000;
18 t=1;
19 bul3=(hamming(t*2000).*a1(N:(N+t*2000-1)))';
20
21 N=25000;
22 t=2;
23 bul4=(hamming(t*2000).*a1(N:(N+t*2000-1)))';
24
25 %% Plotando cada uma das bolhas.
26
27 t1=linspace(0,length(bul1)/fs,length(bul1)); %Ajuste do eixo x para tempo.
28 subplot(2,2,1)
29 plot(t1,bul1)
30 title('Bolha 1')
31 xlabel('tempo(s)')
32 ylabel('Amplitude')
33 axis([0 max(t1) min(bul1) max(bul1)])
34
35 t2=linspace(0,length(bul2)/fs,length(bul2)); %Ajuste do eixo x para tempo.
36 subplot(2,2,2)
37 plot(t2,bul2)
38 title('Bolha 2')
39 xlabel('tempo(s)')
40 ylabel('Amplitude')
41 axis([0 max(t2) min(bul2) max(bul2)])
42
43 t3=linspace(0,length(bul3)/fs,length(bul3)); %Ajuste do eixo x para tempo.
44 subplot(2,2,3)
```

```

45 plot(t3 , bul3)
46 title('Bolha 3')
47 xlabel('tempo(s)')
48 ylabel('Amplitude')
49 axis([0 max(t3) min(bul3) max(bul3)])
50
51 t4=linspace(0, length(bul4)/fs , length(bul4)); %Ajuste do eixo x para tempo.
52 subplot(2,2,4)
53 plot(t4 , bul4)
54 title('Bolha 4')
55 xlabel('tempo(s)')
56 ylabel('Amplitude')
57 axis([0 max(t4) min(bul4) max(bul4)])
58
59 save baseemgham bul1 bul2 bul3 bul4 %salvando as bolhas em baseemgham.

```

codigos/Gerando_base_para_criar_o_sinal_de_EMG/gerar_base.m

I.2 CÓDIGO PARA CRIAR SINAL E PLOTAR RESPOSTAS NO TEMPO E FREQUÊNCIA

```

1 clear all
2 close all
3 clc
4
5 %carregando as bolhas utilizadas para criar o sinal
6 load baseemgham
7
8 %% Parte 1
9 %construindo o sinal simulado
10
11 N=50; %quantidade do sinais que irão compor o banco de filtros
12 fs=2000; %frequência de amostragem
13
14 %Definição do limiar.
15 lim=[sqrt((bul1*bul1')/length(bul1))];
16 lim=[lim sqrt((bul2*bul2')/length(bul2))];
17 lim=[lim sqrt((bul3*bul3')/length(bul3))];
18 lim=[lim sqrt((bul4*bul4')/length(bul4))];
19
20 limiar=min(lim); %escolhe o menor valor rms das bolhas para servir
21 %de limiar
22
23 bancodesinal2=[];
24
25 for i=1:N
26
27 %coletando amostras de ruido de comprimento aleatorio. (media 3 e
28 %desvio padrão 10)

```

```

29 n=round(rand*10000)+10000;
30 ruido1=0.1*limiar*randn(1, n);
31
32 n=round(rand*10000)+10000;
33 ruido2=0.1*limiar*randn(1, n);
34
35 n=round(rand*10000)+10000;
36 ruido3=0.1*limiar*randn(1, n);
37
38 n=round(rand*10000)+10000;
39 ruido4=0.1*limiar*randn(1, n);
40
41 n=round(rand*10000)+10000;
42 ruido5=0.1*limiar*randn(1, n);
43
44 %gerando um sinal de EMG
45 sinal=[ruido1 bul1 ruido2 bul2 ruido3 bul3 ruido4 bul4 ruido4];
46
47 %concatenando os sinais
48 bancodesinal2=[bancodesinal2 sinal];
49
50 %% Parte 2
51 %Nesta parte é realizada a simulação do tratamento do sinal que
52 %também foi feito no arduino para controle do servomotor. O sinal
53 %tratado para que os bulhas fossem transformadas em pulsos quadrados
54 %para que os movimentos de abrir e fechar a garra seja realizada
55 %na borda de subida do pulso.
56
57 %ajustando o eixo x para o tempo em segundos
58 t=linspace(0, length(sinal)/fs, length(sinal));
59 %t=1/fs:1/fs:length(sinal)/fs;
60
61 %plotando o sinal
62 subplot(3,1,1)
63 plot(t, abs(sinal))
64 axis([0 max(t) 0 max(sinal)])
65
66 %ajustando valores acima do limiar para 1 e abaixo do limiar para 0 e
67 %salvando na variavel sinalcontrole
68 sinalcontrole=zeros(1, length(sinal));
69 sinalrms=abs(sinal); %sinalrms=(sinal.^2).^0.5;
70 pos=find(sinalrms>(limiar*0.5));
71 sinalcontrole(pos)=1;
72
73 %pontando o sinal de controle
74 subplot(3,1,2)
75 plot(t, sinalcontrole)
76 title(['int2str(i) '/' int2str(N)'])
77 axis([0 max(t) -1.2 1.2])
78
79 %ajustando o sinal de controle. Durante a contração ha algumas amostras
80 %menores que o limiar devido a oscilação

```

```

81     sinalcontroleF=zeros(1,length(sinal));
82 for i=1:length(sinalcontrole)
83
84     if sinalcontrole(i)==1
85         %Evita oscilações no decorrer do pulso: Quando detecta uma amostra
86         %acima do limiar, garante que as proximas 350 serão setadas para 1.
87         %Para que haja oscilação no pulso resultante seriam necessarias 350
88         %amostras iguais a 0 no decorer de um pulso da contração.
89         sinalcontroleF(i:i+350)=1;
90     end
91 end
92
93 %plotando o sinal de controle
94 subplot(3,1,3)
95 plot(t,sinalcontroleF)
96 title(int2str(length(sinalcontroleF)))
97 axis([0 max(t) -1.2 1.2])
98 pause(0.1)
99 end
100
101 %% Parte 3
102 %adequando o sinal de controle entre 0 e 5V (0 e 256)para que possa ser
103 %utilizado no arduino
104
105 bancodesinal2= bancodesinal2./max(abs(bancodesinal2));
106 bancodesinal2=bancodesinal2*127;
107 bancodesinal2=bancodesinal2+128;
108 bancodesinal2=round(bancodesinal2);
109
110 %armazena o vetor em um arquivo binario.
111 fid = fopen('bancodesinal.bin','wb');
112 fwrite(fid,bancodesinal2,'uchar');
113 fclose(fid);
114 close all
115
116 % Parte 4
117 %analise dos dinais obtidos.
118 subplot(2,1,1)
119 plot(t,sinal)
120 title('Sinal de EMG simulado')
121 xlabel('Tempo (s)')
122 ylabel('Amplitude')
123 axis([0 max(t) 0 max(sinal)])
124
125 h=fft(sinal);
126 f=linspace(0,1,length(sinal)); %adequando o eixo x para frequencias
127 f=f.*fs;
128 subplot(2,1,2)
129 plot(f,abs(h))
130
131 title('Espectro de frecuencia do sinal de EMG simulado')
132 xlabel('Frecuencia (Hz)')

```

```

133 ylabel('Amplitude')
134 axis([0 1000 0 abs(max(h))])

```

codigos/Criando_EMG/sinalsimulado3pulsos.m

I.3 CÓDIGO PILOTO PARA CONTROLE DA PINÇA COM SINAL SIMULADO E DETEÇÃO DE ENVOLTÓRIA DIGITAL UTILIZANDO O MICROCONTROLADOR ARDUINO

```

1 #include <Servo.h> //biblioteca de PWM para controle do servo motor
2 #include <TimerOne.h> //biblioteca para interrupcao do timer, utilizada para
   solicitar amostra de sinal de EMG
3 #include <math.h> //biblioteca para funcao matematica, utilizada para calcular o
   valor RMS
4
5 Servo motor; // cria variavel servo
6
7 int angulo= 0; //armazena a posicao angular do servo;
8 boolean controle = 0; //para inverter o estado da barra
9 byte dado; //variavel que armazena os dados lidos da porta serial.
10 char RMS; //varivel para alocar o valor RMS
11 const byte media = 128, limiar = 7; //definicao da media e do limiar, que no proximo
   tcc sera automatico
12 byte contador =0; //contador para manter o pulso de sinal
13
14 void setup()
15 {
16   Serial.begin(115200); //define a taxa de comunicacao serial
17   pinMode(2, OUTPUT); //pino 2 como saida
18   digitalWrite(2,0); //atribui ao pino 2 nivel logico 0.
19   attachInterrupt(0, servo_open, RISING); //interrupcao para controlar o servo
20   motor.attach(9); // a saida PWM para controle do motor e definida no pino 9
21   Timer1.initialize(500); // inicia timer1, e seta a frequencia de amostragem em 2
   kHz
22   Timer1.attachInterrupt(captura_amostra); // na interrupcao do timer1, vai para a
   funcao captura_amostra
23 }
24
25 /*funcao de interrupcao para controlar a garra*/
26 void servo_open(){
27   if (controle= controle^0xFF){ //se entra a primeira vez, abre a garra, na segunda a
   feixa.
28     for(angulo= 0; angulo< 180; angulo++){ // vai de 0 a 180 graus
29       motor.write(angulo); //funcao de PWM para definir o angulo no servo.
30       //delay(40); // atraso no servo
31     }
32   }
33   else{
34     for(angulo= 180; angulo>=1; angulo--){ // o servo anda de 180 ate 0 graus

```

```

35     motor.write(angulo); //funcao de PWM para definir o angulo no servo.
36     // delay(40); //atraso no servo
37 }
38 }
39 }
40
41 /*funcao periodica (1Khz) para solicitar ao computador uma amostra de sinal*/
42 void captura_amostra(){
43     Serial.write(1); //pede para o computador mandar o dado
44 }
45
46 /*funcao de interrupcao serial, para receber o dado e o processar em tempo real*/
47 void serialEvent(){
48     dado = Serial.read(); //recebe o dado
49     RMS = sqrt((dado-media)*(dado-media)); //calcula o valor RMS da amostra.
50     if (RMS > limiar){ //define o inicio do pulso se o valor RMS for maior que o
51         limiar definido
52         contador = 2000; //reativa o contador sempre que o sinal RMS passa do limiar
53             conta 0.6 segundos
54         digitalWrite(2,1); //inicia o pulso para a interrupcao do servo, que e
55             sensivel a borda de subida
56     }
57     if((RMS < limiar) & (contador>0)){ //mantem o pulso se o valor do contador for
58         maior que zero
59         contador --; //decrementa o contador
60     }
61     if (contador==0){ //se o contador chegou a zero, finaliza o pulso
62         digitalWrite(2,0); //finaliza o pulso
63     }
64 }
65
66 void loop(){
67     //nao usado
68 }

```

codigos/controlador_teste_piloto/controlador_teste_piloto.ino

I.4 CÓDIGO DE IMPLEMENTAÇÃO DO CONTROLE DA PRÓTESE COM ENVOLVIMENTO DO SINAL REAL UTILIZANDO MICROCONTROLADOR ARDUINO

```

1 #include <Servo.h> //biblioteca de PWM para controle do servo motor
2 #include "TimerOne.h" //biblioteca para interrupcao do timer, utilizada para
3     solicitar amostra de sinal de EMG
4 #include <math.h> //biblioteca para funcao matematica, utilizada para calcular o
5     valor RMS
6
7 Servo motor1, motor2; // cria variavel servo
8 byte motor1_enable = 0;

```

```

7 | byte motor2_enable = 0;
8 |
9 | int angulo= 0; //armazena a posicao angular do servo;
10 | boolean controle = 0; //para inverter o estado da barra
11 | boolean serial_enable = 0; //para habilitar comunicacão serial com o computador
12 | unsigned short envoltoria = 0; //onde a leitura da envoltoria do AD e armazenada;
13 | const unsigned short limiar = 200; //definicao do limiar
14 | int contador = 0; //contador para manter o pulso de sinal
15 | int cont_duracao = 0; //contador para determinar a duracão do pulso
16 | boolean ativo_servo = 0;
17 |
18 | void setup()
19 | {
20 | // initialize timer1
21 | noInterrupts(); // disable all interrupt
22 |
23 | Serial.begin(115200); //define a taxa de comunicacao serial
24 | pinMode(2, OUTPUT); //pino 2 como saida
25 | digitalWrite(2,0); //atribui ao pino 2 nivel logico 0.
26 | attachInterrupt(0, servo_open, RISING); //interrupcao para controlar o servo
27 | motor1.attach(5); // a saida PWM para controle do motor 1 e definida no pino 6
28 | motor2.attach(6); // a saida PWM para controle do motor 2 e definida no pino 7
29 | interrupts(); // enable all interrupts
30 | Timer1.initialize(500); // inicia timer1, e seta a frequencia de amostragem em 2
    kHz
31 | Timer1.attachInterrupt(captura_amostra); // na interrupcao do timer1, vai para a
    funcao captura_amostra
32 | }
33 |
34 | /*funcao de interrupcao para controlar a garra*/
35 | void servo_open(){
36 | long i;
37 | if (controle == 1){//se entra a primeira vez, abre a garra, na segunda a fecha.
38 | for(angulo = 10; angulo <=180; angulo++){ // vai de 0 a 180 graus
39 | if (motor1_enable){
40 | motor1.write(angulo); //funcao de PWM para definir o angulo no servo.
41 | }
42 | if (motor2_enable){
43 | motor2.write(angulo); //funcao de PWM para definir o angulo no servo.
44 | }
45 | }
46 | for (i=0;i<100000;i++);
47 | controle = 0;
48 | }
49 | else{
50 | for(angulo = 180; angulo >= 10; angulo--){ // o servo anda de 180 ate 0 graus
51 |
52 | if (motor1_enable){
53 | motor1.write(angulo); //funcao de PWM para definir o angulo no servo.
54 | }
55 | if (motor2_enable){
56 | motor2.write(angulo); //funcao de PWM para definir o angulo no servo.

```

```

57     }
58     for (i=0;i<100000;i++);
59 }
60 controle = 1;
61 }
62 }
63
64 /*funcao periodica (2Khz) para calcular se abre ou fecha a garra e
65 enviar ao computador uma amostra de sinal caso disponivel*/
66 void captura_amostra(){ // timer compare interrupt service routine
67     byte amostra_feedback; // valor de 1 byte para resposta visual
68     envoltoria = (unsigned short) analogRead(A0);
69     if (serial_enable){
70         amostra_feedback = (byte) ((envoltoria)>>2);
71         Serial.write(amostra_feedback); //envia dado capturado para o computador
72     }
73     if (ativo_servo == 0){
74         if (envoltoria > limiar){ //define o inicio do pulso se o valor RMS for maior
75             que o limiar definido
76             cont_duracao++;
77         }else {
78             cont_duracao = 0;
79         }
80     }
81     if ((cont_duracao > 200) & (envoltoria > limiar)){
82         contador = 500; //reativa o contador sempre que o sinal RMS passa do limiar
83             conta 1 segundo
84         digitalWrite(2,1); //inicia o pulso para a interrupcao do servo , que e sensivel
85             a borda de subida
86         ativo_servo = 1;
87         motor1_enable = 1;
88     }
89     if((envoltoria < limiar) & (contador>0)){ //mantem o pulso se o valor do contador
90         for maior que zero
91         contador --; //decrementa o contador
92     }
93     if (contador==0){ //se o contador chegou a zero , finaliza o pulso
94         digitalWrite(2,0); //finaliza o pulso
95         ativo_servo = 0;
96         motor1_enable = 0;
97     }
98 }
99
100 /*funcao de interrupcao serial , para receber o dado do computador*/
101 void serialEvent(){
102     serial_enable ^= 1;
103 }
104
105 void loop(){
106     //nao usado

```

I.5 CÓDIGO PARA CAPTURA DO SINAL DE EMG PELO DISPOSITIVO NI USB-6009 ATRAVÉS DO MATLAB

```

1 clear all;
2 close all;
3 clc
4
5 s = daq.createSession('ni')
6 s.addAnalogInputChannel('Dev1', 'ai0', 'Voltage');
7 s.addAnalogInputChannel('Dev1', 'ai1', 'Voltage');
8 s.Rate=2000;
9 s.DurationInSeconds=30;
10
11 [data,time] = s.startForeground;
12
13 subplot(2,1,1)
14 plot(time,data(:,1))
15 xlabel('Amplitude(V)')
16 ylabel('Tempo (s)')
17 subplot(2,1,2)
18 plot(time,data(:,2))
19 xlabel('Amplitude(V)')
20 ylabel('Tempo (s)')

```

I.6 ALGORITMO EM LINGUAGEM C PARA ENVIO DO SINAL DE EMG SIMULADO AO ARDUINO

```

1 #include <stdlib.h>
2 #include <stdio.h> // Definições de entrada e saída padrao
3 #include <stdint.h> // Standard types
4 #include <string.h> // Definições de funções de string
5 #include <unistd.h> // UNIX standard function definitions
6 #include <fcntl.h> // File control definitions
7 #include <termios.h> // POSIX terminal control definitions
8 #include <sys/ioctl.h>
9 #include <signal.h> // Definições e funções de sinais
10 #include <math.h> // Definições matemáticas
11 #include <errno.h> // Definições de número de erros
12 #include <pthread.h>
13

```

```

14
15 #define porta    "/dev/ttyACM0" //porta de comunicacao
16 #define baudrate 115200 //taxa de comunicacao com microcontrolador
17
18 //arquivo para leitura do sinal simulado
19 FILE * fd2;
20 FILE * fd3;
21
22 //funcao para encerramento do programa
23 void captura_sinal(){
24     fclose(fd2);
25     printf("saindo/n");
26     exit(0);
27 }
28
29 int main(char argc , char *argv[]){
30
31     pid_t filho;
32
33     filho = fork();
34
35     if (filho!=0){
36         //sleep(10);
37     }
38     else {
39         sleep(2); //espera 2 segundos
40         if (execlp("kst2"," ","plot.kst",NULL)==-1){
41             perror("exec");
42             exit(0);}
43     }
44
45     signal(SIGINT,&captura_sinal); //com o comando Ctrl+C vai para a funcao captura
        sinal e encerra o programa
46
47     int fd = serialport_init(argv[1], baudrate); //inicia a porta serial
48     unsigned char b; //caracter lido do banco de sinal
49     unsigned char buf; //caractere enviado
50     long long total=0; //contador de caracteres enviados
51
52
53     //flush na porta serial
54     sleep(2); //necessario para fazer o flush na porta de comunicacao
55         if (tcflush(fd, TCIOFLUSH)==-1){
56             perror("flush serial port: ");
57         }
58     //abre o banco de sinais gerado
59     fd2 = fopen("bancodesinal.bin","rb"); //em binario
60
61     fd3 = fopen("dados_enviados.txt","w");
62     //loop infinito para adquirir os dados e enviar para o arduino
63
64     for(;;){

```

```

65     fd3 = fopen("dados_enviados.txt", "a");
66     fread(&b, sizeof(unsigned char), 1, fd2); //le um caractere do banco de sinais
67     fprintf(fd3, "%lld\t%d\t%d\n", total, b, b);
68     while(read(fd, &buf, 1) != 1); //enquanto nao consegue ler, fica parado
69         if (write(fd, &b, 1) == -1) { //envia o caractere do banco de sinais
70             perror("escrita: ");
71         }
72         printf("enviado: %u total: %lld\n", b, total++); //imprime na tela
73     fclose(fd3);
74     usleep(10);
75 }
76 fclose(fd2); //fecha o arquivo
77 }
78
79 //funcao adaptada de https://github.com/todbot/arduino-serial/blob/master/arduino-
    serial-lib.c
80 int serialport_init(const char* serialport, int baud){
81     struct termios toptions;
82     int fd;
83
84     fd = open(serialport, O_RDWR | O_NONBLOCK); //abre a porta de comunicacao
85
86     if (fd == -1) {
87         perror("serialport_init: incapaz de abrir a porta 0");
88         exit(0);
89     }
90
91     if (tcgetattr(fd, &toptions) < 0) {
92         perror("serialport_init: Couldn't get term attributes");
93         return -1;
94     }
95     speed_t brate = baud; //permite escolher a taxa de amostragem conforme a funcao
        abaixo
96
97     switch(baud) { //definicoes de taxa de comunicacao
98         case 4800: brate=B4800; break;
99         case 9600: brate=B9600; break;
100        #ifdef B14400
101        case 14400: brate=B14400; break;
102        #endif
103        case 19200: brate=B19200; break;
104        #ifdef B28800
105        case 28800: brate=B28800; break;
106        #endif
107        case 38400: brate=B38400; break;
108        case 57600: brate=B57600; break;
109        case 115200: brate=B115200; break;
110    }
111    cfsetispeed(&toptions, brate); //define a velocidade de recebimento de dados
112    cfsetospeed(&toptions, brate); //define a velocidade de transmissao de dados
113
114    //configuracao das opcoes de porta

```

```

115 // 8N1
116 toptions.c_cflag &= ~PARENB;
117 toptions.c_cflag &= ~CSTOPB;
118 toptions.c_cflag &= ~CSIZE;
119 toptions.c_cflag |= CS8;
120 // sem controle de fluxo
121 toptions.c_cflag &= ~CRTSCTS;
122 //toptions.c_cflag &= ~HUPCL; // desabilita hang-up-on-close para evitar reset.
123 toptions.c_cflag |= CREAD | CLOCAL; // habilita READ e ignora linhas de controle
124 toptions.c_iflag &= ~(IXON | IXOFF | IXANY); // turn off s/w flow ctrl
125 toptions.c_lflag &= ~(ICANON | ECHO | ECHOE | ISIG); // faz arquivo raw
126 toptions.c_oflag &= ~OPOST; // faz arquivo raw
127 // see: http://unixwiz.net/techtips/termios-vmin-vtime.html
128 toptions.c_cc[VMIN] = 0;
129 toptions.c_cc[VTIME] = 0;
130 //toptions.c_cc[VTIME] = 20;
131
132 tcsetattr(fd, TCSANOW, &toptions); //configura a porta com as opcoes atribuidas
133 if( tcsetattr(fd, TCSAFLUSH, &toptions) < 0) {
134     perror("init_serialport: Couldn't set term attributes");
135     return -1;
136 }
137 return fd;
138 }

```

codigos/transmissor_piloto/transmissor.c

I.7 ALGORITMO EM LINGUAGEM C FEEDBACK VISUAL DE FORÇA

```

1 #include <stdlib.h>
2 #include <stdio.h> // Definições de entrada e saída padrao
3 #include <stdint.h> // Standard types
4 #include <string.h> // Definições de funcoes de string
5 #include <unistd.h> // UNIX standard function definitions
6 #include <fcntl.h> // File control definitions
7 #include <termios.h> // POSIX terminal control definitions
8 #include <sys/ioctl.h>
9 #include <signal.h> // Definições e funcoes de sinais
10 #include <math.h> // Definições matematicas
11 #include <errno.h> // Definições de numero de erros
12 #include <pthread.h>
13
14
15 #define porta "/dev/ttyACM0" //porta de comunicacao
16 #define baudrate 115200 //taxa de comunicacao com microcontrolador
17
18 //arquivo para gravação do sinal recebido
19 FILE * fd_recebidos;
20 int fd_serial;

```

```

21
22 //funcao para encerramento do programa
23 void captura_sinal(){
24     unsigned char b = 1; //caracter enviado para terminar a transmissÃo
25     //envia um caracter informando fim da transmissÃo
26     if (write(fd_serial,&b,1)==-1){
27         perror("escrita: ");
28     }
29     printf("saindo/n");
30     exit(0);
31 }
32
33 int main(char argc , char *argv[]){
34
35     pid_t filho;
36
37     filho = fork();
38
39     if (filho!=0){
40         //sleep(10);
41     }
42     else {
43         sleep(2); //espera 2 segundos
44         if (execlp("kst2"," ","plot.kst",NULL)==-1){
45             perror("exec");
46             exit(0);}
47     }
48
49     signal(SIGINT,&captura_sinal); //com o comando Ctrl+C vai para a funcao captura
        sinal e encerra o programa
50
51     fd_serial = serialport_init(argv[1], baudrate); //inicia a porta serial
52     unsigned char b = 1; //caracter enviado para terminar a transmissÃo
53     unsigned char buf; //caractere recebido
54     long long total=0; //contador de caracteres recebidos
55
56
57     //flush na porta serial
58     sleep(2); //necessario para fazer o flush na porta de comunicacao
59     if (tcflush(fd_serial , TCIOFLUSH)==-1){
60         perror("flush serial port: ");
61     }
62
63     //envia um caracter informando inÃcio da transmissÃo
64     if (write(fd_serial,&b,1)==-1){
65         perror("escrita: ");
66     }
67
68
69     //loop infinito para receber os dados enviados pela porta serial
70     for(;;){
71         fd_recebidos = fopen("dados_enviados.txt","a");

```

```

72     while(read(fd_serial,&buf,1)!=1); //enquanto nao consegue ler , fica parado
73     fprintf(fd_recebidos,"%lld\t%d\t%d\n",total,buf,buf);
74     printf("enviado: %u total: %lld\n",buf,total++); //imprime na tela
75     fclose(fd_recebidos);
76     usleep(10);
77 }
78
79 //envia um caracter informando o fim da transmissÃo
80 if (write(fd_serial,&b,1)==-1){ //
81     perror("escrita: "); //
82 } //
83 }
84
85 //funcao adaptada de https://github.com/todbot/arduino-serial/blob/master/arduino-
86 serial-lib.c
87 int serialport_init(const char* serialport, int baud){
88     struct termios toptions;
89     int fd;
90
91     fd = open(serialport, O_RDWR | O_NONBLOCK ); //abre a porta de comunicacao
92
93     if (fd == -1) {
94         perror("serialport_init: incapaz de abrir a porta 0");
95         exit(0);
96     }
97
98     if (tcgetattr(fd, &toptions) < 0) {
99         perror("serialport_init: Couldn't get term attributes");
100        return -1;
101    }
102
103    speed_t brate = baud; //permite escolher a taxa de amostragem conforme a funcao
104    abaixo
105
106    switch(baud) { //definicoes de taxa de comunicacao
107        case 4800: brate=B4800; break;
108        case 9600: brate=B9600; break;
109        #ifdef B14400
110        case 14400: brate=B14400; break;
111        #endif
112        case 19200: brate=B19200; break;
113        #ifdef B28800
114        case 28800: brate=B28800; break;
115        #endif
116        case 38400: brate=B38400; break;
117        case 57600: brate=B57600; break;
118        case 115200: brate=B115200; break;
119    }
120
121    cfsetispeed(&toptions, brate); //define a velocidade de recebimento de dados
122    cfsetospeed(&toptions, brate); //define a velocidade de transmissao de dados
123
124    //configuracao das opcoes de porta
125    // 8N1

```

```

122 toptions.c_cflag &= ~PARENB;
123 toptions.c_cflag &= ~CSTOPB;
124 toptions.c_cflag &= ~CSIZE;
125 toptions.c_cflag |= CS8;
126 // sem controle de fluxo
127 toptions.c_cflag &= ~CRTSCTS;
128 //toptions.c_cflag &= ~HUPCL; // desabilita hang-up-on-close para evitar reset.
129 toptions.c_cflag |= CREAD | CLOCAL; // habilita READ e ignnora linhas de controle
130 toptions.c_iflag &= ~(IXON | IXOFF | IXANY); // turn off s/w flow ctrl
131 toptions.c_lflag &= ~(ICANON | ECHO | ECHOE | ISIG); // faz arquivo raw
132 toptions.c_oflag &= ~OPOST; // faz arquivo raw
133 // see: http://unixwiz.net/techtips/termios-vmin-vtime.html
134 toptions.c_cc[VMIN] = 0;
135 toptions.c_cc[VTIME] = 0;
136 //toptions.c_cc[VTIME] = 20;
137
138 tcsetattr(fd, TCSANOW, &toptions); //configura a porta com as opcoes atribuidas
139 if( tcsetattr(fd, TCSAFLUSH, &toptions) < 0) {
140     perror("init_serialport: Couldn't set term attributes");
141     return -1;
142 }
143 return fd;
144 }

```

codigos/transmissor/transmissor2.c