



**CENTRO UNIVERSITÁRIO DE BRASÍLIA – UnICEUB**  
**PROGRAMA DE INICIAÇÃO CIENTÍFICA**

**RAILANDER OLIVEIRA A. FARIA**  
**PEDRO PINHEIRO GARCIA**

**PRÓTESE MIOELÉTRICA APLICADA EM MEMBROS SUPERIORES COM**  
**MODELAGEM MECÂNICA EM IMPRESSORA 3D**

**BRASÍLIA**

**2019**



**RAILANDER OLIVEIRA A. FARIA**

**PEDRO PINHEIRO GARCIA**

**PRÓTESE MIOELÉTRICA APLICADA EM MEMBROS SUPERIORES COM  
MODELAGEM MECÂNICA EM IMPRESSORA 3D**

Relatório final de pesquisa de Iniciação Científica  
apresentado à Assessoria de Pós-Graduação e  
Pesquisa.

Orientação: MSc. Luciano Henrique Duque

**BRASÍLIA**

**2019**

## **PRÓTESE MIOELÉTRICA APLICADA EM MEMBROS SUPERIORES COM MODELAGEM MECÂNICA EM IMPRESSORA 3D**

**Railander Oliveira A. Faria – UniCEUB, PIC Institucional, aluno bolsista**  
*railander.faria@sempreceub.com*

**Pedro Pinheiro Garcia – UniCEUB, PIC Institucional, aluno voluntário**  
*pedro.garcia@sempreceub.com*

**Prof. MSc. Luciano Henrique Duque – UniCEUB, professor orientador**  
*luciano.duque@uniceub.br*

As próteses mioelétricas utilizam eletrodos de superfície para interpretar sinais mioelétricos. Estes sinais são comandos que percorrem o sistema nervoso capazes de realizar funções motoras, como contração muscular. A partir da leitura e processamento digital desse sinal, é possível controlar um sistema servomotor e reproduzir a intenção do usuário. Para otimizar o procedimento, se faz necessário amplificar o sinal e filtrar possíveis ruídos na leitura provenientes da rede elétrica e outros equipamentos nas proximidades. Um modelo matemático eficiente é importante para realizar o tratamento do sinal sacrificando o mínimo de processamento que for possível. Com o auxílio de uma impressora 3D, buscou-se realizar testes quanto a montagem de próteses já desenvolvidas e compreender melhor como operar o equipamento. Para trabalhos futuros, espera-se desenvolver um modelo adequado de prótese que comporte os componentes eletrônicos necessários de forma discreta, adquirir mini/micro servomotores capazes de simular os movimentos de uma mão real.

**Palavras-Chave: prótese, prótese 3D, sinais mioelétricos**

## Sumário

<b>1. Introdução</b> .....	<b>1</b>
<b>2. Fundamentação teórica</b> .....	<b>3</b>
2.1 Microcontrolador .....	3
2.2 Processador digital de sinal.....	8
2.3 DSPIC30F4013.....	10
2.4 Linguagem de programação C.....	11
2.5 Amplificadores operacionais .....	12
2.5.1 Amplificador inversor .....	13
2.5.2 Amplificador não-Inversor .....	14
2.5.3 Amplificador diferencial .....	15
2.6 Filtros analógicos .....	15
2.6.1 Filtros passivos.....	16
2.6.2 Filtros ativos .....	19
2.7 Filtros digitais.....	21
2.7.1 Transformada de Fourier.....	22
2.7.2 Transformada Discreta de Fourier .....	23
2.7.3 Transformada Rápida de Fourier .....	24
2.8 Métodos de controle.....	26
2.8.1 Malha de controle .....	28
2.8.2 Controle ON-OFF.....	29
2.8.3 Controle proporcional.....	30
2.9 Motores .....	30
2.9.1 Motores de corrente contínua .....	31
2.9.2 Servomotor .....	32
2.10 Órteses e Próteses .....	34
2.10.1 Órtese .....	35
2.10.2 Prótese mioelétrica .....	36
2.11 Eletromiografia.....	37
2.11.1 Estrutura do sinal mioelétrico .....	39
2.11.2 Coleta do sinal muscular .....	41
<b>3 Metodologia</b> .....	<b>43</b>
3.1 Diagrama de blocos do sistema .....	43
3.2 Etapa 1 – Coleta do sinal Eletromiográfico .....	43
3.3 Etapa 2 – Tratamento primário do sinal .....	47
3.4 Etapa 3 – Processamento .....	55

3.4.1	Hardware.....	56
3.4.2	Firmware.....	66
3.5	Etapa 4 – Estrutura mecânica.....	74
3.5.1	Primeira estrutura mecânica (Estudo aplicado).....	75
3.5.2	Segunda estrutura mecânica (Início do Estudo – Modelagem/impressão 3D) .....	77
3.6	Etapa 5 – Recepção dos sinais sem fio .....	81
3.7	Etapa 6 – Software.....	86
3.7.1	Interface gráfica .....	87
3.7.2	Código fonte .....	88
<b>4</b>	<b>Resultados e discussões.....</b>	<b>90</b>
4.1	Testes da coleta do sinal eletromiográfico .....	91
4.2	Testes do tratamento primário do sinal .....	92
4.3	Testes da placa principal (Processamento).....	94
4.4	Testes da estrutura mecânica .....	97
4.5	Testes da recepção dos sinais sem fio.....	99
4.6	Testes do <i>software</i> .....	100
<b>5</b>	<b>Considerações finais.....</b>	<b>104</b>
	<b>Referências .....</b>	<b>106</b>

## 1. Introdução

Este projeto é uma continuação da pesquisa desenvolvida no UniCEUB desde 2012. Seu desenvolvimento foi dado pelo Grupo de Pesquisa Engenharia de Reabilitação cadastrado no CNPQ e liderado pelo professor MSc. Luciano Henrique Duque.

A evolução das próteses tem sido longa e bem catalogada pela história. Seu início primitivo, com materiais pesados como misturas metálicas e madeiras, é marcado por próteses sem muita flexibilidade de movimento, pesadas e com principal objetivo de esconder deformidades. Desde então, novas próteses foram desenvolvidas durante a história para atender cada vez mais a carência de pessoas necessitadas, tornando-se um objeto cada vez mais complexo e refinado. (NORTON, 2007)

A complexidade e o refino das próteses, causam o encarecimento delas. Com o alto custo, que pode chegar à até 100 mil dólares (Peña, 2016), há uma parcela significativa da sociedade que se encontra desamparada no caso de remoção, completa ou parcial, do membro superior, visto que uma parte razoável da população brasileira se encontra em algum grau de pobreza (ONUBR, 2017)

Com o advento da impressão 3D, se tornou possível modelar uma prótese em um programa de computador, e tê-la completamente funcional em questão de dias. Diversos estudos, indivíduos, famílias e grupos sociais estão desenvolvendo próteses mecânicas de forma muito mais acessível do que o que se tinha antes (Mohammed, 2016). A modelagem e a impressão 3D permitiram que próteses complexas fossem espalhadas globalmente por arquivos e sites, tornando possível que diversas pessoas tivessem acesso a profissionais capacitados em robótica assistiva.

A robótica assistiva é uma linha de pesquisa que engloba conceitos de engenharia, medicina, matemática e computação, sendo capaz de auxiliar pessoas com algum tipo de deficiência, idade avançada ou necessidades especiais (Galhardo, 2012). Diariamente as pessoas têm lesões musculares que são originadas por acidentes, esforços repetitivos ou por problemas vasculares, para a recuperação do paciente os exercícios propostos na reabilitação são repetitivos e algumas vezes são considerados exaustivo ou doloroso. Com isso têm-se tecnologias de baixo custo que se pode observar uma melhora nessa prática. Estratégias que incluam o uso de tecnologias para o tratamento de pessoas com deficiência motora têm sido um dos principais tópicos de discussão nas áreas de ciências da saúde (CHANG, 2011).

Quando se refere a tratamentos fisioterapêuticos, o uso de tecnologias aplicadas é altamente demandado, que são em sua maioria desenvolvidos pelo setor da engenharia biomédica, porém os estudos na área vêm se disseminando por outros ramos de pesquisa (Soares, 2013) (Pazeto, 2014) e, como ocorre neste trabalho, chegou ao departamento de Engenharia Elétrica. Sendo por hardware ou software há diversas formas de auxiliar a reabilitação do paciente em inúmeros aspectos, seja pela criação de ferramentas de auxílio ao fisioterapeuta ou por suporte direto, completo ou parcial, na mobilidade do paciente, como é caso aqui tratado. Nos casos de amputação, o tratamento por fisioterapia deve começar o quanto antes, pois logo após a cirurgia ou trauma o acompanhamento do paciente deve ser feito desde o correto posicionamento na cama ao uso de aparelhos de suporte, e quando aplicável, o uso de prótese é recomendado para uma melhor resposta em menor espaço de tempo (Cordeiro, 2010).

O trabalho em questão se propõe a apresentar uma proposta alternativa que se enquadre na usabilidade, bem como na acessibilidade, visto que o mesmo se refere a uma prótese de baixo custo que além de devolver parte da mobilidade perdida, auxilia diretamente na reabilitação muscular do paciente.

O biofeedback eletromiográfico (EMG), amplamente utilizado na reabilitação musculoesquelética, consiste na medida da atividade elétrica associada à contração muscular e retorno de informações por meio de sinais auditivos e visuais. Os sinais mioelétricos, como são chamados os sinais medidos por EMG, podem ser aplicados em clínica para reeducação muscular, em especial para relaxamento da proteção muscular com redução da dor pela interrupção do ciclo “dor-proteção-dor”. (Prentice, 2002)

Outra aplicação da medida dos sinais mioelétricos, é o controle do grupo de próteses denominado próteses mioelétricas. As próteses mioelétricas utilizam os sinais captados por eletrodos de superfície sobre o ponto motor de determinado músculo para ativar motores do dispositivo terminal (Winkler, 2009). Todo ser vivo de estrutura complexa produz sinais mioelétricos, sinais esses que são transportados pelos nervos e músculos até o membro que se quer realizar algum tipo de trabalho.

Esse trabalho propõe o desenvolvimento de um sistema de controle inteligente de prótese mioelétrica de membros superiores na qual, tem como finalidade a captação dos sinais musculares através de eletrodos de superfície, que transportam os sinais para um comparador

analógico, eliminando interferências e ruídos externos, amplificando e processando apenas o sinal eletromiográfico, a fim de possibilitar o controle de uma prótese para membro superior.

Junto ao uso da prótese, o sistema é contemplado com comunicação sem fio, desta maneira é possível que o profissional (fisioterapeuta ou responsável pela reabilitação) possa acompanhar os dados referentes à região muscular do paciente que está sendo trabalhada. De posse de tais informações será possível acompanhar a qualidade do sinal eletromiográfico (frequência e amplitude) que será definida pela razão sinal/ruído (Duarte, 2006), bem como a sua evolução ao longo do tratamento até que o usuário esteja apto a usar a prótese de forma natural.

O objetivo geral deste trabalho é desenvolver uma prótese mioelétrica para membro superior que seja capaz de coletar, tratar e executar a ação de comando correspondente ao sinal mioelétrico captado. Paralelo a isso o microcontrolador deverá enviar os dados coletados de forma sem fio para o computador do profissional responsável.

## **2. Fundamentação teórica**

### **2.1 Microcontrolador**

Pode-se entender o microcontrolador como um único chip que comporta as ferramentas necessárias para se ter as funções de um computador e desempenhar tarefas e monitoramento e controle. Todas as componentes envolvidas no sistema respeitam a sua devida escala e capacidade de processamento. (Malvino, 1985)

Em geral, a estrutura do microcontrolador pode ser dividida em duas partes, sendo elas os Itens Básico e Itens Avançados. Em breve descrição, deve-se considerar que a arquitetura básica contempla apenas os itens responsáveis pelo funcionamento da máquina, sendo assim, eles têm como função garantir que o processamento seja executado. Ao que tange à estrutura avançada do chip, deve ser entendida como itens de hardware adicionais que podem ser integrados na pastilha do circuito integrado, sendo assim, estes desempenham funções adicionais que podem ser executadas de forma paralela ou independente do processamento principal. (Microchip, 2003)

Para melhor exemplificação, será fornecido como base o microcontrolador PIC12F675 fabricado pela Microchip. O mesmo conta com Itens Básico e alguns avançados, sendo assim se mostra intuitivo para que se tenha o primeiro contato com as definições aqui levantadas.

Junto à duas imagens e blocos nelas devidamente identificados, será feito um descritivo de algumas partes do PIC, as quais foram definidas como fundamentais para o desenvolvimento da pesquisa proposta neste documento. A seguir, a Figura 1 explora alguns componentes básicos para mínimo funcionamento do componente:

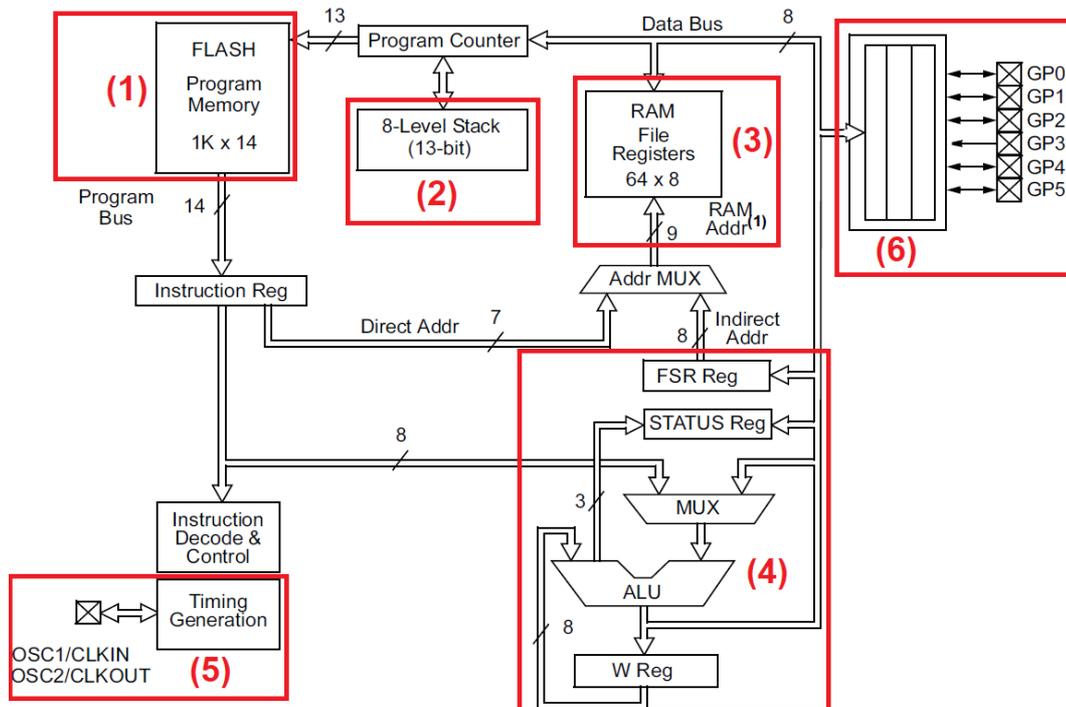


Figura 1: Estrutura básica do PIC12F675 (<http://www.microchip.com/>)

1) Estrutura responsável por manter a programação, a mesma pode ser do tipo FLASH (múltiplas gravações possíveis) ou EPROM (suporta apenas uma gravação) (Codá, 2013). Essa memória é do tipo não-volátil, logo é capaz de manter os dados gravados mesmo com a falta de alimentação, porém apresenta elevado tempo de escrita e baixo ciclo de gravação (aproximadamente 100 mil). (Microchip, 2003)

2) Para que o microcontrolador saiba o que executar e em qual ordem, o mesmo conta com uma estrutura chamada Pilha (stack em inglês), que, como a própria semântica sugere, refere-se à uma série de itens dispostos um em cima do outro, sendo assim o primeiro a ser adicionado será o último a sair. Com esse controle a etapa de processamento é capaz de definir a hierarquia das etapas que devem ser executadas. (Microchip, 2003)

3) Após estar definido o que executar e em qual ordem, necessita-se de um local para armazenar os dados de forma temporária, bem como ter meios para manipulá-los sem

restrição de escrita e com alta cadência de acesso, requisitos estes que a FLASH não atende. A fim de sanar este problema, usa-se a RAM (random memory access – memória de acesso randômico). Esta trata-se de uma memória do tipo volátil que, contrário à FLASH, perde os dados nela gravados assim que a alimentação é suspensa. Seu benefício fica por conta da alta velocidade de leitura e escrita, o que a torna excelente para execução de ações em tempo real, e longo ciclo de vida. (Morimoto, 2007)

4) A etapa seguinte a ser seguida pela máquina é de executar o cálculo proposto na atual instrução para que o valor seja armazenado na RAM e posteriormente usado pelas instruções futuras. Esse conjunto de gerenciamento dos dados e manipulação algébrica é responsabilidade da ALU (arithmetic logic unit – unidade lógica aritmética) e as respectivas estruturas auxiliares. (Camara, 2013)

5) Quando se trata de eletrônica digital, uma das principais componentes do sistema é o clock (relógio interno da máquina). Essa etapa é responsável por gerar uma sequência de pulsos com comprimento e amplitude constantes a fim de informar para a ALU quando deve iniciar e finalizar cada ciclo de cálculo. A estabilidade do sistema está, em grande parte, amparada neste atributo da arquitetura do circuito integrado. De forma simplificada, pode-se dizer que a velocidade com que o sistema irá executar as ações é diretamente proporcional ao valor do clock. (Felipe Gabardo Gonçalves, 2015)

6) Para que o microcontrolador tenha interação com o meio ao qual está incluso, o mesmo precisa de uma estrutura que crie a área de contato entre os dados de entrada, processamento e saída. Assim, apoia-se no uso de portas lógicas (as quais só assumem valores binários) para que o mesmo possa receber e enviar informações. Estas portas podem operar como entrada, ou seja, o estado dela irá depender de uma fonte externa ou como saída (recepção de dados), assim o próprio microcontrolador estará apto a gerenciar seu estado lógico (transmissão de dados). (Silveira, 2011)

A seguir, a Figura 2 explora alguns componentes avançados que fornecem funções adicionais ao microcontrolador:

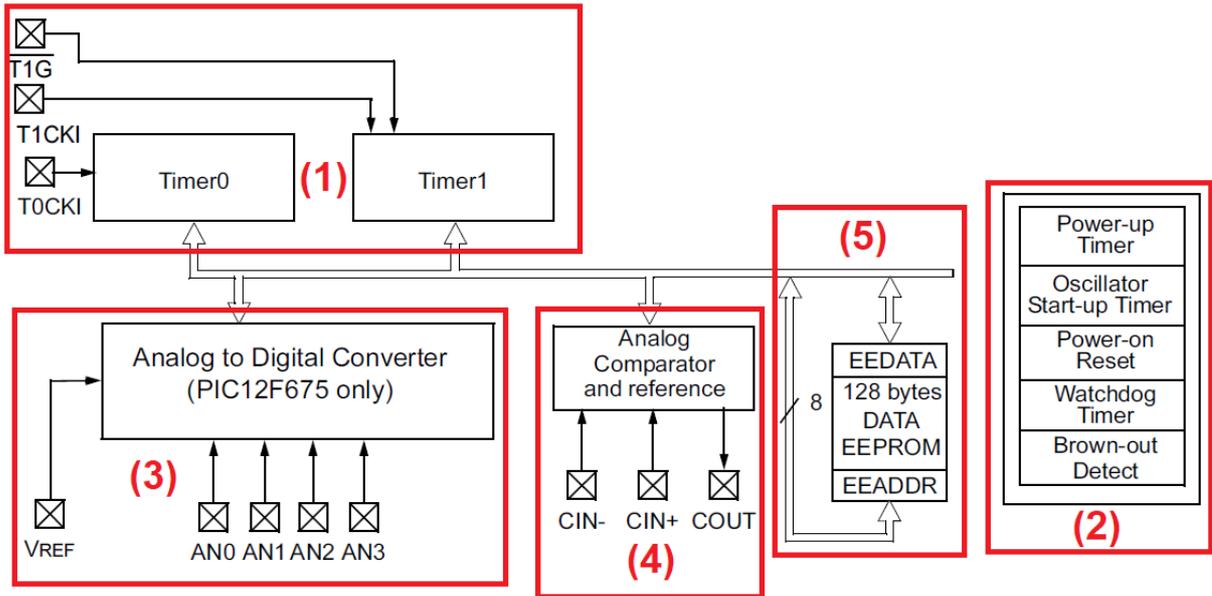


Figura 2: Estrutura avançada do PIC12F675 (<http://www.microchip.com/>)

1) Muitos são os cenários em que a aplicação a nível de hardware/firmware necessita de fazer o controle periódico de uma ou mais etapas do processamento. Com o intuito de evitar o uso de temporizadores externos, é comum que os microcontroladores tragam em sua arquitetura o bloco de Timer, o mesmo pode ser simplificado em um contador com o clock derivado da fonte principal de oscilação do microcontrolador, sendo assim, a cada pulso o contador do Timer incrementa-se em uma unidade. Desta forma é possível definir um controlador temporal que execute de forma paralela ao processamento da ALU. (Malvino, 1985)

2) Na estrutura nomeada na Figura 3 como item 2, há um conjunto de cinco periféricos à arquitetura principal do microcontrolador. Como os mesmos apresentam pouca ou nenhuma utilidade ao trabalho em questão, eles não serão individualmente conceituados neste texto. Em suma, esse grupo de módulos executa, entre outras, funções como esperar um tempo genérico para o oscilador externo estabilizar ao ser energizado (Power-up Timer), evitar que o microcontrolador opere em estado de constante instabilidade do sinal de clock (Oscillator Start-up Timer), garantir que o microcontrolador inicie sempre nos mesmos parâmetros elétricos ao ser energizado (Power-on Reset), evitar o que a estrutura de processo fique presa em uma determinada parte do código (Watchdog Timer) e, por fim, fornecer ao dispositivo a capacidade de monitorar a tensão que lhe é fornecida e gerar a parada dos ciclos

de máquina caso uma alimentação insuficiente seja detectada (Brown-out Detect). (Microchip, 2003)

3) Amplamente utilizado no trabalho aqui desenvolvido, o ADC (Analog to Digital Converter – Conversor de Analógico para Digital) é responsável por discretizar e quantizar os valores provenientes do ambiente contínuo e traduzi-los de forma digital que o mesmo possa ser processado pela máquina binária. Esta ferramenta é fundamental para aumentar a gama de cenários em que se pode aplicar o uso de microcontroladores. As características técnicas podem variar e tendem a seguir a necessidade da aplicação. Mais à frente nesse trabalho, este componente será melhor tratado. (Microchip, 2003)

4) A estrutura responsável pelo módulo comparador analógico consiste em um conjunto de dois comparadores analógicos internos, esses tendem a ter suas entradas e saídas associadas aos pinos do microcontrolador para fazer comparações de tensões analógicas externas entre si ou com a fonte de referência interna derivada da alimentação elétrica do microcontrolador. O módulo comparador permite a implementação de diversos circuitos, dentre eles os conversores ADC, bem como os conversores do tipo Delta-Sigma. (Oliveira, 2010)

5) A estrutura de armazenamento de dados denominada EEPROM (Electrically-Erasable Programmable Read-Only Memory – Memória somente leitura programável eletronicamente apagável) assemelha-se a memória do tipo FLASH, apresentando capacidade de retenção de dados não volátil, porém também sofre com a baixa frequência e ciclos de escrita. A grande vantagem de trabalhar com esse tipo de memória é ter uma série de bytes a disposição para serem gravados e alterados em tempo de execução e que podem ser retidos mesmo na ausência de alimentação sem correr o risco de afetar o código fonte do microcontrolador. (Alecrim, 2005)

## 2.2 Processador digital de sinal

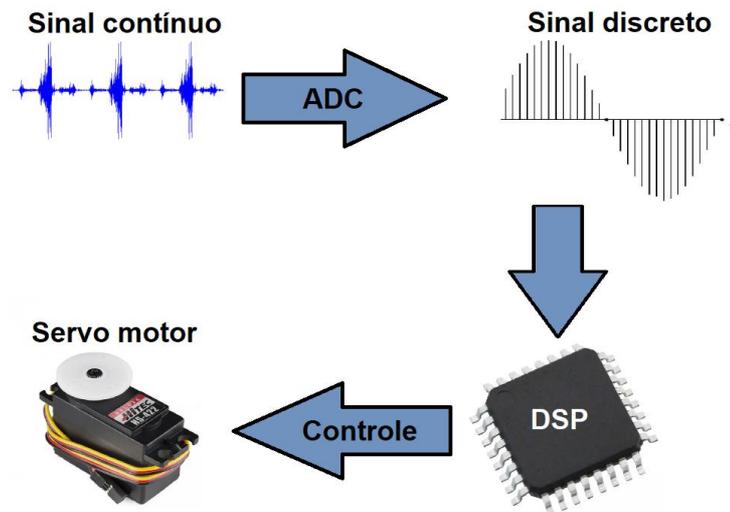


Figura 3: Processamento digital de sinais (imagem elaborada pelo autor)

Ao que tange o processamento digital de sinais, tem-se como definição a ciência que estuda as regras que governam o comportamento dos sinais discretos, bem como os dispositivos que os processam. Trabalhando de forma concorrente ou cooperante ao trabalho analógico de sinais, a manipulação digital do sinal tende a entregar uma resposta similar, porém em espectro digitalizado. (Torres R. C., 2004)

Há três tipos de processamento mais comuns, os quais são: offline, online e tempo real. Para o primeiro modelo tem-se um ambiente em que o sistema primeiro coleta os dados e depois os processa sem vínculo temporal entre processamento e saída. Neste caso é necessário um grupo de memória para armazenar os dados a serem trabalhados, bem como o requisito de processamento é menor, visto que o tempo de processamento não é uma variável crítica. Também é correto afirmar que este tipo de mecanismo é mais comumente utilizado para processamento de imagem, onde primeiramente elas são coletadas e somente depois os dados são trabalhados (Vega, 2017). Já o que define o processamento online, leva-se em consideração o cenário em que os dados são recebidos, processados, executa-se a ação e só então recebe-se o novo pacote de dados a ser trabalhados. O sistema pode ser considerado temporal, porém por mais que os dados sejam tratados a medida em que são recebidos há um certo atraso e possível perda de dados, visto que a recepção de um novo pacote depende da saída do anterior (Torres R. C., 2004). Normalmente, acha-se esse método de controle em sistemas que dependem da resposta a curto prazo, porém há uma certa aceitabilidade quando se trata de retardo em ação/reação. Em ambientes que requerem um

controle contínuo, confiável e com o menor espaço de tempo entre entrada e saída, busca-se os modelos de processamento digital de sinais em tempo real. Assume-se que no mecanismo em questão os dados serão recebidos e ao passo em que são tratados, um novo pacote chega para ser trabalhado. Desta forma, tem-se um sistema mais robusto e assertivo quando se trata de ambientes que precisão de resposta rápida, como seria o caso do tratamento de dados necessários para a navegação de um avião, por exemplo. Intuitivamente, espera-se que para toda a agilidade de um sistema em tempo real seja entregue e se comporte de forma eficaz, a taxa de processamento será razoavelmente maior que os demais modelos, assim o maior impacto fica por conta do valor elevado deste tipo de tecnologia (Torres R. C., 2004).

Em geral, quando se recorre ao uso de processamento digital de sinais, a aplicação requer uma implementação intensa de cálculos matemáticos, fluxo controlado entre sinais de entrada e saída e sistemas que sejam flexíveis e adaptativos (Vega, 2017). Boa parte do trabalho feito pelo DSP pode também ser executado por estruturas eletrônicas analógicas, porém dentre as vantagens da digitalização tem-se a alta capacidade de ser reprogramado de acordo com a necessidade, sendo assim pode-se, por exemplo, utilizar um filtro anteriormente passa-baixa como passa-alta sem a necessidade de alterações a nível de hardware. Outro ponto positivo leva em consideração a alta estabilidade e robustez que sistemas de DSP tendem a entregar, isso se mostra de grande valia quando o dispositivo deve operar e realizar atividades críticas e/ou sensíveis. (Torres R. C., 2004)

Descrito todos os processos acima, deve-se estabelecer conceitos para categorizar um microcontrolador ou microprocessador como DSP ou não. Para que o dispositivo possa ser corretamente intitulado como um processador digital de sinais o mesmo deve apresentar uma relação direta entre o software e hardware interno a fim de otimizar o trabalho do sinais com instruções de máquina já estabelecidas, deve agrupar em seu encapsulamento todos os recursos necessários para capturar, processar e executar ações sobre o sinal a ser trabalhado, realizar operações especiais como o acúmulo de somas e produtos, alta capacidade de processamento e uma unidade aritmética preparada para receber as constantes e repetidas funções de processamento digital de sinais. (Torres R. C., 2004)

### 2.3 DSPIC30F4013

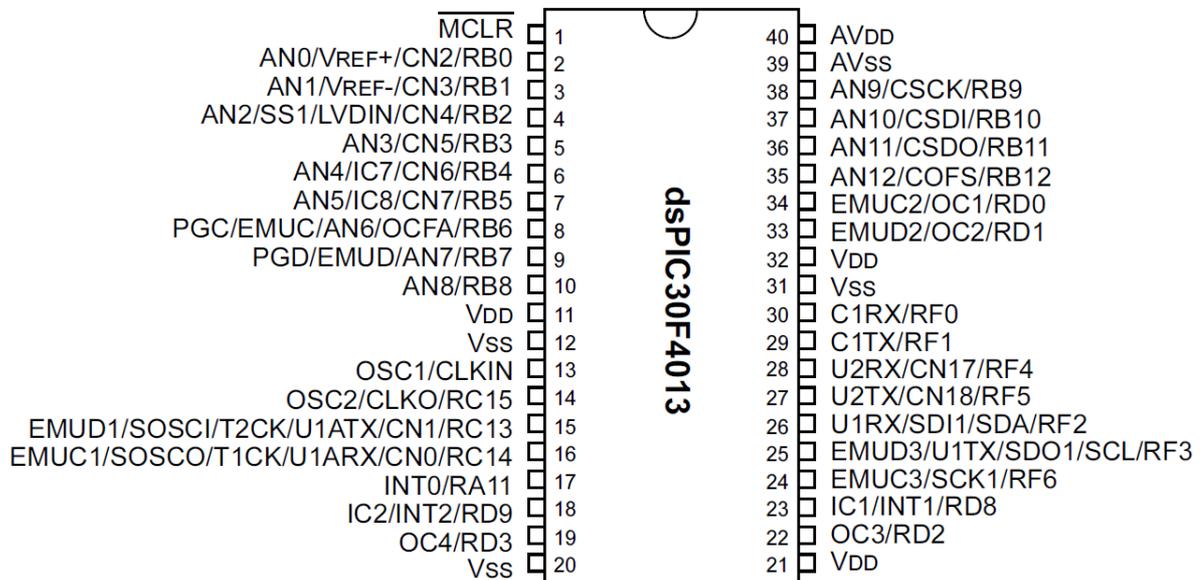


Figura 4: Representação do dsPIC30F4013 (<http://www.microchip.com/>)

A fim de manter uma boa relação entre tempo de desenvolvimento e pesquisa, optou-se pelo uso de um DSP eficaz e com amplo material didático disponível, assumindo essas premissas o dsPIC30F4013 mostrou-se atrativo e capaz de desempenhar o processamento esperado. Para a escolha deste componente, levantou-se as características do mesmo (Microchip, 2004). Valores estes que podem ser vistos abaixo:

- Arquitetura de 16 bits;
- 84 instruções base voltadas para o processamento digital de sinais;
- 48kB de Flash;
- Dois bancos de RAM com acesso duplo de 1KB cada;
- 30MIPS (million instructions per second – milhões de instruções por segundo);
- Busca dupla de instruções;
- Dois registradores de operação de 40 bits;
- Todas as instruções de DSP são de ciclo único;
- Faz operação de deslocamento binário de até 16 posições em um único ciclo;

As organizações dos recursos acima descritos podem ser corretamente visualizadas no diagrama lógico na Figura 5 abaixo:

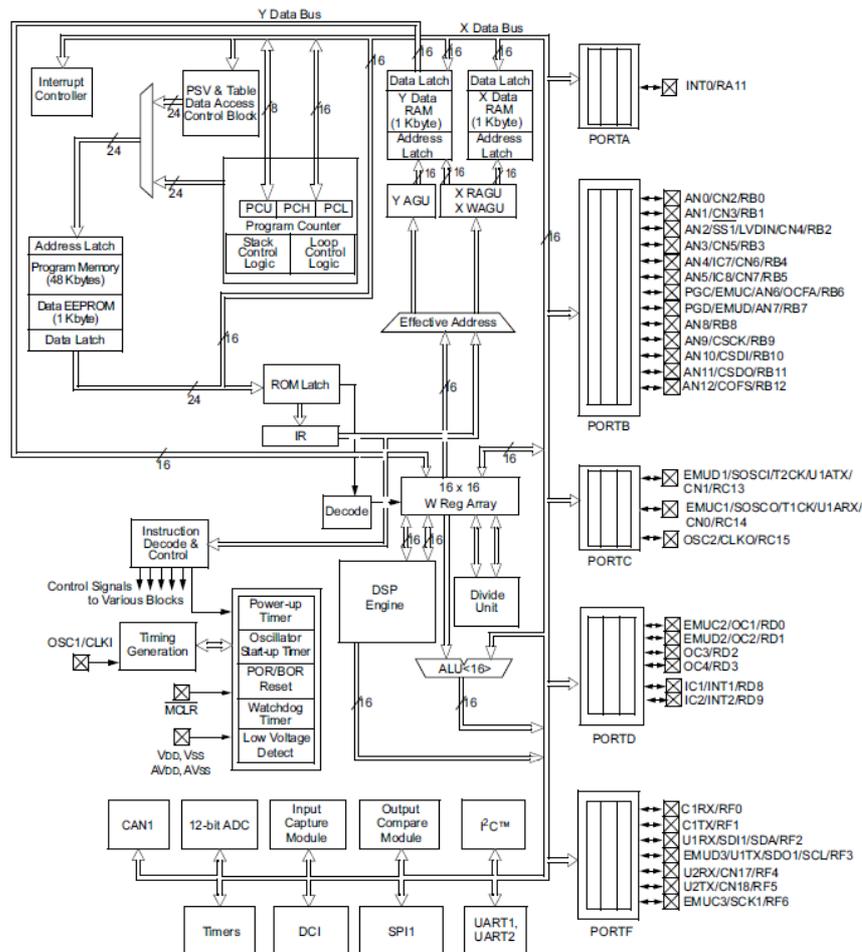


Figura 5: Estrutura lógica do dsPIC30F4013 (<http://www.microchip.com/>)

## 2.4 Linguagem de programação C

Usada primeiramente em um computador DEC PDP-11 por Denis Ritchie nos Laboratórios Bell, a linguagem chamada C executou pela primeira vez em um sistema operacional UNIX. Essa linguagem de programação é a evolução de outra linguagem, até então muito usada, denominada BCPL, a qual foi desenvolvida pelo programador Martin Richards. Sendo assim, desde o início, C foi desenvolvido por programadores para programadores, onde o intuito era otimizar o processo de desenvolvimento de programas de computador, conseqüentemente a ferramenta desenvolvida popularizou-se e está presente em boa parte dos ambientes computacionais. (Pereira, 2009)

Demonstrando grande flexibilidade, eficiência e portabilidade, a linguagem C acabou se tornando a opção de inúmeros desenvolvedores e muitas empresas a adotaram como padrão para o desenvolvimento de seus sistemas. Boa parte da sua eficiência, se dá por estar

preparada para trabalhar em todos os níveis da computação, desde aplicação direta em hardware à serviços web e de vários núcleos e processos. Sendo assim, seu grau de otimização é grande, o que a permite ser executada em boa parte dos dispositivos digitais e com alta velocidade. (Pereira, 2009)

## **2.5 Amplificadores operacionais**

Os AOPs (Amplificadores operacionais) foram desenvolvidos na década de 40 e eram construídos com válvulas. Evidentemente as características desses primitivos amplificadores eram bastante precárias. Com o advento do transistor no final da década de 40, foi possível a construção de amplificadores operacionais com características razoáveis. (Monteiro, 2016)

O AOP é um amplificador de corrente contínua multi estágio com entrada diferencial cujas características se aproximam de um amplificador ideal. Seu nome, deriva do conceito de um amplificador DC (amplificador acoplado em corrente contínua) com uma entrada diferencial e ganho extremamente alto, cujas características de operação estavam determinadas pelos elementos de realimentação utilizados. Mudando os tipos e a disposição dos elementos de realimentação, podiam ser implementados diferentes operações analógicas. Em grande parte, as características globais do circuito estavam determinadas só por estes elementos de realimentação. Dessa forma, o mesmo amplificador era capaz de realizar diversas operações, onde o desenvolvimento gradual dos amplificadores operacionais deu local ao surgimento de uma nova era nos conceitos de desenhos de circuitos. (JOHNSON, 1991)

Um AOP possui duas entradas e uma saída, em que esta obtém valor múltiplo da diferença entre as duas entradas. O fator  $A$  é o ganho de tensão do amplificador operacional, ou seja, a relação entre a tensão de entrada diferencial e a de saída do dispositivo. (Monteiro, 2016)

A simbologia básica de um AOP pode ser vista na Figura 6 a seguir.

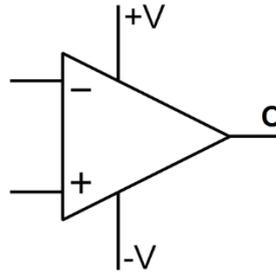


Figura 6: Simbologia do amplificador operacional (Wendling, 2010)

Na Figura 6 percebe-se que existem 5 pinos no AOP, onde + é a entrada não inversora, - a entrada inversora, O a saída do sinal, +V e -V são a alimentação do AOP. (Wendling, 2010)

Todo AOP possui uma tensão de offset de saída, sendo está a amplificação da diferença entre os sinais de sua entrada. Em um amplificador ideal, caso suas entradas fossem iguais, a saída deveria ser igual à 0, porém em um amplificador real essa saída não apresenta esse comportamento, gerando assim uma tensão de offset em seu resultado.

A alimentação do AOP é realizada com fontes de corrente contínua para sua polarização e normalmente se utiliza fontes simétricas. (Monteiro, 2016)

Existem várias formas de se utilizar os amplificadores operacionais, podendo ser sem realimentação do sinal, com realimentação positiva e com realimentação negativa. A Figura 7 mostra, de forma básica, um sistema com realimentação do sinal:

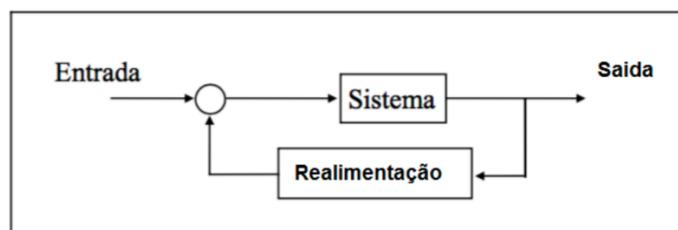


Figura 7: Sistema com realimentação do sinal (Monteiro, 2016)

### 2.5.1 Amplificador inversor

A configuração inversora do amplificador tem como finalidade inverter o sinal do valor de tensão que está sendo aplicada na entrada no AOP. Para que o componente compreenda os valores de sinal negativo tanto quando de sinal positivo, o mesmo precisa ser alimentação com uma fonte simétrica onde os seus limites de tensão devem ser maiores que os valores

esperados na entrada e menor ou igual ao valor máximo admitido pelo dispositivo. A configuração ser dá por dois resistores adicionador em sua configuração, o R1 é o componente de entrada e o Rf o resistor de realimentação. O ganho também pode ser calculado atrás da relação entre eles (Wendling, 2010). A representação do circuito pode ser vista na Figura 8:

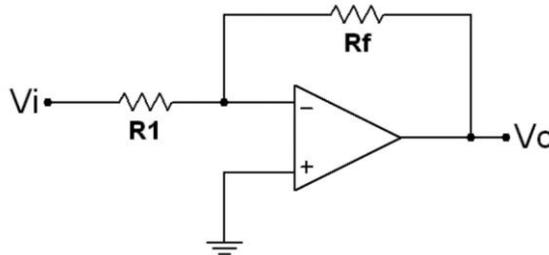


Figura 8: Amplificador operacional inversor (Wendling, 2010)

$$V_o = -\frac{R_f}{R_1} V_i$$

*Equação 1 Equação do ganho do amplificador operacional inversor (Wendling, 2010)*

### 2.5.2 Amplificador não-Inversor

Em cenários onde se precisa amplificar sinais de tensão de forma direta, ou seja, sem inverter o sentido, opta-se pelo da configuração não-inversora. A mesma opera de forma similar ao arranjo inversor, mas conserva a polaridade do sinal aplicado a entrada do componente. Neste caso, o resultado final deve ser similar quando se usar a fonte simétrica e não simétrica, desde que os ajustes de offset tenham sido corretamente feitos. A relação dada entre o sinal de entrada e saída é dada pela Equação 2, a qual depende dos valores de R1 e Rf (Wendling, 2010). Assim, o diagrama elétrico fica como está representado na Figura 9:

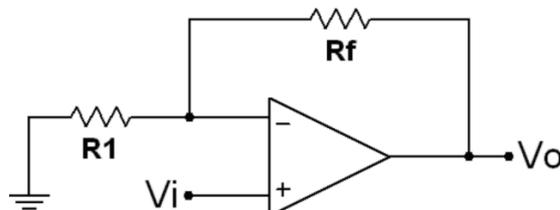


Figura 9: Amplificador operacional não-inversor (Wendling, 2010)

$$V_o = \left(1 + \frac{R_f}{R_1}\right) V_i$$

*Equação 2 Equação do ganho do amplificador operacional não-inversor (Wendling, 2010)*

### 2.5.3 Amplificador diferencial

Também chamado de subtrator, o amplificador operacional disposto de forma a obter um sinal diferencial, irá coletar duas amostras simultânea de entradas distintas e o valor final da sua saída será igual a diferença entre o pino + e – (responsáveis pelas entradas). A configuração em questão mostra-se bem robusta, pois ao passo que realiza a subtração entre os sinais, pode-se também aplicar a operação de ganho, ao fim do circuito, o valor de  $V_o$  poderá ser multiplicado por alguma constante a fim de obter o valor desejado (Wendling, 2010). A figura representativa e seu respectivo equacionamento pode ser observado abaixo:

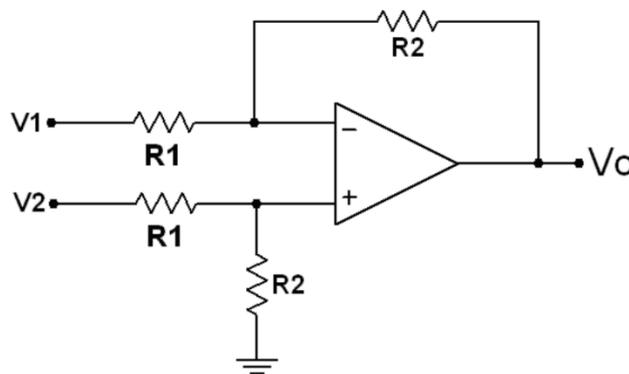


Figura 10: Amplificador operacional diferencial (Wendling, 2010)

$$V_o = \frac{R_2}{R_1}(V_2 - V_1)$$

Equação 3 Equação do ganho do amplificador operacional diferencial (Wendling, 2010)

### 2.6 Filtros analógicos

Ao se tratar do cenário real de aplicação, há uma infinidade de sinais dispostos ao redor do dispositivo e, em muitas vezes, ruídos produzidos por ele mesmo que podem interferir no correto aferimento do sinal de trabalho. A fim de manter constância e precisão nas medições utiliza-se de filtros. Este elemento pode ser primordialmente dividido entre analógico e digital (este último será tratado com maior minuciosidade em tópicos mais a frente), em um refinamento mais profundo há ainda categorias referentes ao modo de operação, seja se ele utilizará de componentes passivos e/ou ativos ou qual tipo de sinal será filtrado (Arndt, 2016).

### 2.6.1 Filtros passivos

A próxima definição de filtros se dá pela atividade elétrica dos componentes, pois aquele que somente tem componentes passivos (resistores, capacitores, indutores, entre outros), ou seja, componentes que apenas recebem energia, tendem a atenuar o sinal. Espera-se como resultado, pela lei da conservação da energia, que o sinal de saída seja sempre inferior ao de entrada, pois haverá dissipação através dos componentes pelo efeito Joule (Silva, 2014). Ao que tange os modos de operação, há quatro configurações básicas, sendo elas:

- Passa-Baixa

Neste modo o circuito se comportará de forma a atenuar sinais que estejam em frequência acima da definida a nível de projeto. Assim, o mesmo tende a atenuar sinais de maior grau dispostos no espectro (Silva, 2014). A configuração da Figura 11 terá um gráfico similar ao exibido na Figura 12:

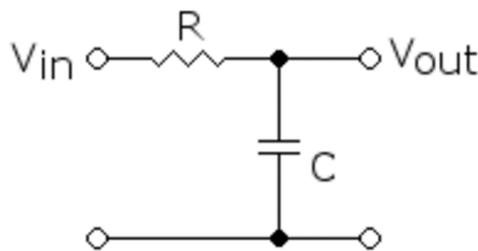


Figura 11: Circuito do filtro passa-baixa (Arndt, 2016)

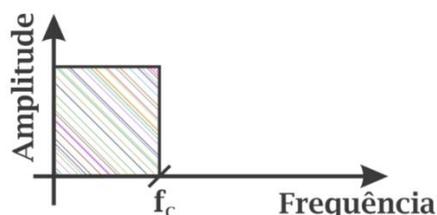


Figura 12: Gráfico do filtro passa-baixa (Arndt, 2016)

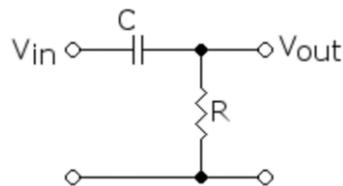
A fim de obter esse resultado, deve-se utilizar a Equação 4, considerando  $f_c$  como sendo a frequência de corte,  $R$  e  $C$  como os respectivos componentes integrantes do circuito:

$$f_c = \frac{1}{2\pi RC}$$

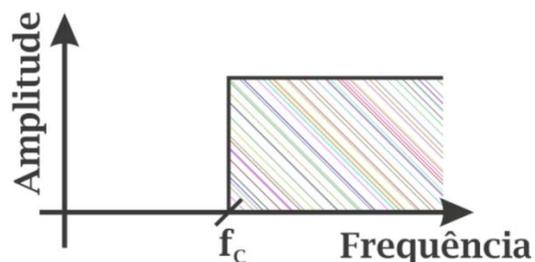
*Equação 4 Equação do circuito passa-baixa passivo (Arndt, 2016)*

- Passa-Alta

Operando de forma contrária ao modelo anterior, esta disposição de componentes fornece um sinal de saída que carece de componentes com baixo valor no espectro, sendo assim, o sinal de trabalho será focado em sinais de maior frequência. Intuitivamente, o arranjo dos componentes se dá de forma inversa ao do Passa-Baixa (Arndt, 2016). O circuito da Figura 13 mostra isso com clareza, bem como a Figura 14 dispõe o comportamento do sinal de saída:



*Figura 13: Circuito do filtro passa-alta (Arndt, 2016)*



*Figura 14: Gráfico do filtro passa-alta (Arndt, 2016)*

Para o equacionamento, deve-se considerar a mesma equação mostrada nos filtros passa-baixa mostrada na Equação 4. Essa reutilização da fórmula matemática, mostra-se verdadeira pelo fato de o cerne do circuito manter-se o mesmo, apenas reposicionando os componentes.

- Passa-Banda

O que pode ser considerado como uma junção das duas outras configurações anteriores, o Passa-Banda permite que apenas uma faixa de frequência não sofra atenuação pelo circuito. O mesmo se dá pela inclusão de um novo componente em seu circuito, o indutor

(Silva, 2014). O arranjo do novo circuito pode ser visto na Figura 15 e seu respectivo gráfico comportamental na Figura 16:

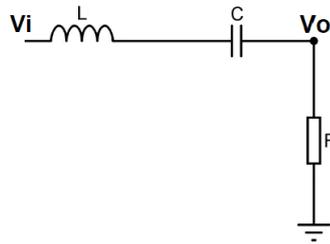


Figura 15: Circuito do filtro passa-banda (imagem elaborada pelo autor)

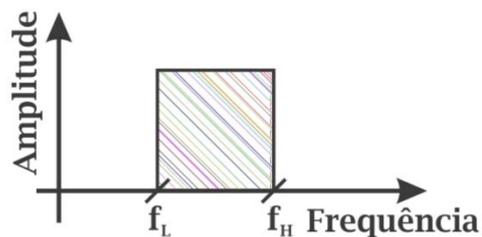


Figura 16: Gráfico do filtro passa-banda (Arndt, 2016)

Para estabelecer a frequência a ser ignorada pelo sistema atenuante presente no filtro, deve-se usar a Equação 5. Observe que a matemática leva em consideração os componentes capacitivos e indutivos, de forma a estabelecer um padrão de ressonância entre eles. Ao atingir esse estado, os componentes citados se comportarão como um curto-circuito, sendo assim a frequência de entrada e de saída seriam iguais. Para valores fora da faixa definida, a relação entre os três componentes definirá o grau de atenuação final (Arndt, 2016).

$$f_c = \frac{1}{2\pi\sqrt{LC}}$$

Equação 5 Equação do circuito passa-faixa passivo (Arndt, 2016)

- Rejeita-Banda

Assim como traçou-se uma relação entre os filtros Passa-Baixa e Passa-Alta, o mesmo pode ser feito quando se trata dos filtros Passa-Banda e Rejeita-Banda. Operando de forma inversa, esse circuito tende a eliminar um único estreito do espectro. Muito utilizado quando o sistema demanda que alguma componente oscilante se posiciona entre duas ou mais

faixas necessárias para a operação do dispositivo. Sua estruturação eletrônica pode ser vista na Figura 17, bem como seu gráfico na Figura 18:

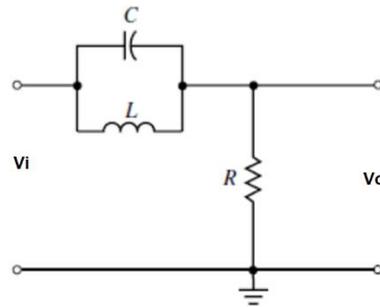


Figura 17: Circuito do filtro rejeita-banda (Arndt, 2016)

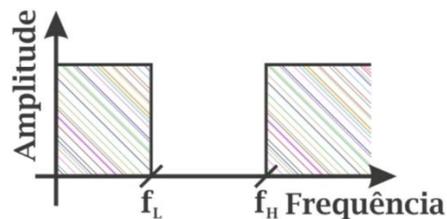


Figura 18: Gráfico do filtro rejeita-banda (Arndt, 2016)

Como a estrutura básica do circuito mantém-se a mesma do filtro anterior, pode-se usar a Equação 5 para fins de dimensionamento.

### 2.6.2 Filtros ativos

Como anteriormente introduzido, um dos modos de operação para filtros analógicos é a arquitetura por componentes ativos. Entende-se componentes ativos como aqueles que tendem a fornecer energia e/ou ganho a sinais de entrada, como o clássico transistor (Bonfim, 2010). Comumente utilizado, o amplificador operacional apresenta boa relação de operação com os filtros. Reutilizando dos modos de operação exibidos no tópico anterior, o funcionamento é deveras similar, porém observa-se a falta de atenuação no sinal de saída e em muitos casos, de acordo com o projeto, o devido ganho fornecido à tensão resultante (Arndt, 2016).

De forma que o cerne do funcionamento dos filtros ativos e passivos mostram-se os mesmos, ao decorrer deste tópico será tratado apenas o diagrama eletrônico, bem como as devidas equações. Maiores informações foram expostas no item 2.3.1.

Os circuitos e equações referentes aos modos de operação podem ser vistos a seguir:

- Passa-Baixa

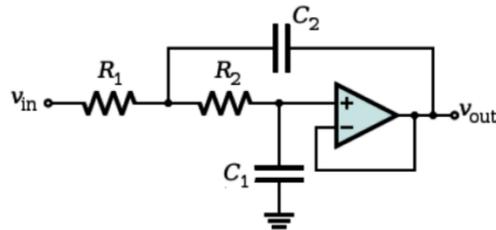


Figura 19: Filtro ativo passa-baixa (Bonfim, 2010)

$$f_c = \frac{1}{2\pi\sqrt{R_1 R_2 C_1 C_2}}$$

Equação 6 Equação do filtro ativo passa-baixa (Bonfim, 2010)

- Passa-Alta

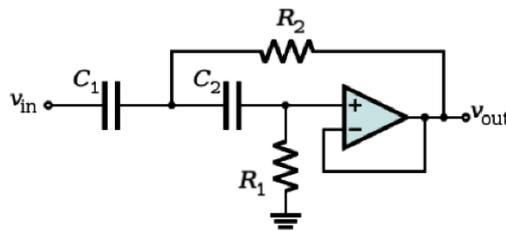


Figura 20: Filtro ativo passa-alta (Bonfim, 2010)

Assim como acontece nos filtros passivos, a equação dos modos de operação Passa-Baixa e Passa-Alta matem-se iguais. Para tanto, a Equação 6 descreve o comportamento deste circuito.

- Passa-Banda

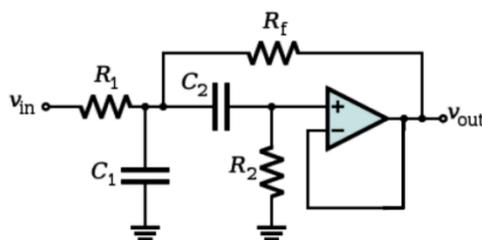


Figura 21: Filtro ativo passa-banda (Bonfim, 2010)

$$f_c = \frac{1}{2\pi} \sqrt{\frac{R_f + R_1}{R_1 R_2 C_1 C_2 R_f}}$$

Equação 7 Equação do filtro ativo passa-banda (Bonfim, 2010)

Vale notar que, diferentemente do filtro analógico Passa-Banda, esse circuito não conta com o componente indutivo, baseando-se apenas em resistores, capacitores e amplificador operacional.

- Rejeita-Banda

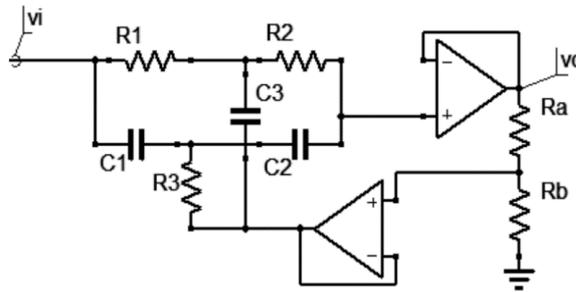


Figura 22: Filtro ativo rejeita-banda (Bonfim, 2010)

$$f_c = \frac{1}{2\pi RC}$$

Equação 8 Equação do filtro ativo rejeita-banda (Bonfim, 2010)

Para que a Equação 8 mostrar-se verdadeira, algumas considerações devem ser feitas. Para tal, R assume o valor de R1 que deve ser igual à R2, bem como R3 deve ser metade do valor de R1. Ao que tange os capacitores, o valor de C será igual ao valor de C1 e C2, já o C3 representa o dobro do valor de C1.

## 2.7 Filtros digitais

Os filtros digitais são sistemas que através de modelos matemáticos, comumente chamados de transformadas, operam sobre um sinal a fim de separar sinais combinados, onde pode-se considerar desde sinais compostos à sinal e ruído, e restaurar um sinal que sofreu algum grau de distorção (Scandelari, 2013). Há uma gama de transformadas que atuam sobre o sistema a fim de fragmentar o sinal em suas componentes no domínio da frequência, mas neste trabalho será abordada apenas a Transformada de Fourier e seus devidas vertentes que somam ao resultado esperado.

### 2.7.1 Transformada de Fourier

Quando se trabalha com um sinal o qual é composta pela soma de diversos outros, é possível que a aplicação requiera a retirada de partes integrantes que em nada somam ou podem até influenciar negativamente ao sinal de estudo. Como ao trabalhar no domínio do tempo as ferramentas de manipulação não se aplicam à essa fragmentação do sinal, dito isso é necessário operar os dados de entrada para que os mesmos sejam convertidos para o domínio da frequência (Cerqueira, 2007). O modelo aqui utilizado será a Transformada de Fourier, a qual pode ser descrita pela Equação 9 abaixo:

$$F(\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t} dt$$

*Equação 9 Equação da Transformada de Fourier (Smith, 2002)*

A transformada de Fourier é uma transformada integral que expressa uma função em termos de funções de base senoidal. Dentro da integral, além da função  $f(t)$ , tem-se a senóide complexa  $e^{-j\omega t}$  que representa a função de base senoidal. A transformada de Fourier expressa o produto interno entre essas duas funções (Cerqueira, 2007). Para melhor entendimento observe a definição de produto interno para o espaço de funções contínuas e complexas para um intervalo  $[a; b]$ :

$$f(t), g(t) = \int_a^b f(t)\overline{g(t)} dt$$

*Equação 10 Produto interno de funções contínuas (Smith, 2002)*

Com base na Equação 10 é possível entender como a definição da Transformada de Fourier se dá pela manipulação de uma senóide complexa em sua forma conjugada.

Pode-se entender que a transformada se trata de uma operação de produto interno que calcula o coeficiente de projeção da função  $f(t)$  sobre uma senóide complexa (Smith, 2002). Deve-se ler  $F(\omega)$  como sendo a medida da amplitude e da fase da senóide complexa que representa o sinal de entrada na frequência  $\omega$ . A condição suficiente para a existência da transformada de Fourier de uma função  $f(x)$  qualquer é que essa função seja integrável e finita (Cerqueira, 2007). Desta forma, deve-se observar a Equação 11 abaixo:

$$\int_{-\infty}^{\infty} |f(x)| dx < \infty$$

*Equação 11 Condição da Transformada de Fourier (Smith, 2002)*

A fim de facilitar a manipulação desta operação, a diversas tabelas que exploram determinados cenários que tendem a se repetir em uma gama de problemas encontrados em tratamento de sinais. Como exemplo, tem-se a Tabela 1:

Propriedade	Domínio do tempo	Domínio da frequência
Linearidade	$\alpha f(t) + \beta g(t)$	$\alpha F(\omega) + \beta G(\omega)$
Atraso no tempo	$f(t - \tau)$	$e^{-j\omega\tau} F(\omega)$
Atraso na frequência	$e^{j\omega_0 t} f(t)$	$F(\omega - \omega_0)$
Convolução	$f(t) * g(t)$	$F(\omega)G(\omega)$
Conjugado simétrico	$\bar{f}(t)$	$\bar{F}(-\omega)$
Simetria par	$f(-t)$	$F(-\omega)$
Simetria ímpar	$-f(t)$	$F(\omega)$

*Tabela 1 Coeficientes de Fourier tabelados (Cerqueira, 2007)*

### 2.7.2 Transformada Discreta de Fourier

Quando se trata do ambiente computacional, há inúmeras barreiras por introduzir cálculos muito distantes da álgebra binária, sendo assim há uma gama de operações matemáticas (integral, derivada, entre outras) que necessitam de algum tipo de aproximação. Seja por processo iterativo ou por uso de aproximações com as de Tustin e Euler, o modelo equacional aplicado ao mundo contínuo necessita de adequação para ser utilizada.

A Transformada Discreta de Fourier (DFT) procura eliminar esses empecilhos através da modelagem do cálculo por uma sequência  $x[n]$  que é definida pela Equação 12:

$$X[\omega_k] = \sum_{n=0}^{N-1} x[n] e^{-\frac{j2\pi nk}{N}}, k = 0, 1, 2, \dots, N - 1$$

*Equação 12 Transformada Discreta de Fourier (Mittra, 1998)*

Para tal, deve-se considerar o  $X[\omega_k]$  como sendo uma sequência finita no domínio da frequência e formou-se por N amostras.

A senóide complexa que aparece na Transformada Discreta de Fourier é chamada de kernel. O kernel consiste em amostras de uma senóide complexa em N frequências discretas  $\omega_k$  uniformemente espaçadas entre 0 e a taxa de amostragem  $\omega_T = 2\pi fT$ , onde T é o espaço de tempo constante entre cada uma das amostras.

Pode-se dizer que a transformada é uma operação de produto interno que calcula o coeficiente de projeção da sequência  $x[n]$  sobre a senóide complexa  $\cos(\omega_k n) + j \sin(\omega_k n)$ . A variável  $X(\omega_k)$  representa a medida da amplitude e da fase da senóide complexa que produz a sequência de entrada na frequência  $\omega_k$ .

### 2.7.3 Transformada Rápida de Fourier

A fim de aplicar a Transformada Discreta de Fourier em processadores de menor capacidade e tornar o seu cálculo menos oneroso para o controlador que o executa, há ainda a vertente da Transformada Rápida de Fourier (FFT).

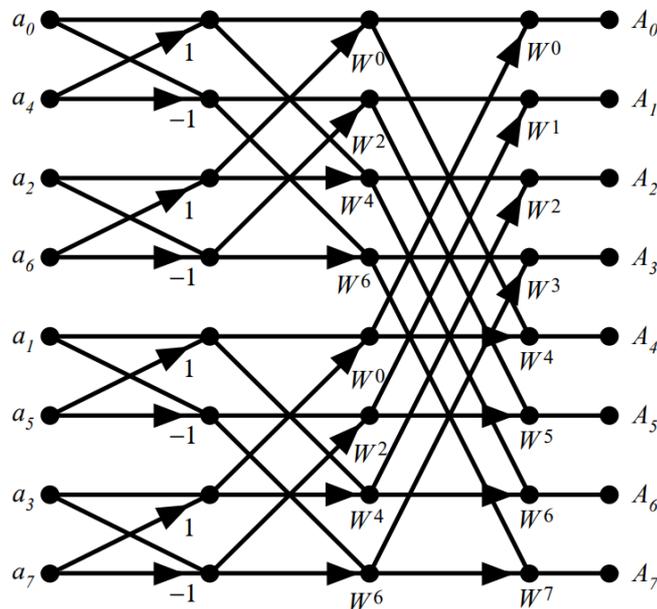
A Transformada Rápida de Fourier é um método razoavelmente eficiente que reordena os cálculos dos coeficientes de uma Transformada Discreta de Fourier. A mesma trata-se de um algoritmo que realiza uma avaliação da DFT priorizando o menor esforço computacional, ao invés de realizar o cálculo da DFT diretamente pela definição. A FFT é uma técnica que possibilita avaliar a DFT de forma mais rápida e econômica, sendo uma das maiores contribuições para a análise numérica já realizada. (Pupin, 2011)

O algoritmo FFT reduz pela metade a DFT de N pontos em duas somas, em termos pares e em termos ímpares, onde cada uma torna-se uma DFT de  $\frac{N}{2}$  pontos. Essa manipulação pode ser vista na Equação 13 onde  $h_{2j}$  representa os componentes pares e  $h_{2j+1}$  os ímpares:

$$H_k = \sum_{j=0}^{\frac{N}{2}-1} h_{2j}(\omega^2)^{jk} + \sum_{j=0}^{\frac{N}{2}-1} h_{2j+1}(\omega^2)^{jk} \omega^k$$

*Equação 13 Equação base da Transformada Rápida de Fourier (Pupin, 2011)*

Ao passo em que a aplicação se aproxima do cenário dos microcontroladores, novas técnicas são estabelecidas para o melhor aproveitamento da capacidade de processamento da máquina. Muito usado, tem-se o algoritmo da borboleta mostrado na Figura 23:



*Figura 23: Método da borboleta para FFT (Heckbert, 1995)*

O algoritmo da FFT decompõe a DFT em estágio de  $\log_2 N$  onde cada um consiste em um cálculo computacional do método da borboleta de tamanho  $\frac{N}{2}$ . Cada ciclo do cálculo utiliza de dois números complexos os quais, para nível de exemplo, serão denominados de  $p$  e  $q$  os quais são definidos seus respectivos conjugados, desta forma tem-se  $p + \alpha q$  e  $p - \alpha q$  onde o  $\alpha$  representa a parte complexa da equação (Heckbert, 1995). Desta forma, por meio da extrapolção, obtém-se o seguinte trecho da manipulação borboleta mostrado na Figura 24:

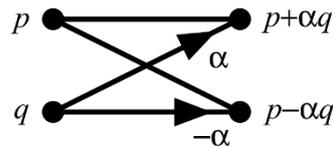


Figura 24: Trecho do algoritmo da borboleta (Heckbert, 1995)

## 2.8 Métodos de controle

Ao se tratar da temática de métodos de controle, deve-se estabelecer o cenário o qual demanda algum tipo de atuação automático (Araújo, 2007). O controle é fundamental em boa parte do campo da engenharia e da ciência, pois tal necessidade mostra-se intrínseca e de relevante importância à sistemas industriais, espaciais, robótica e automação de manufatura em geral. Comumente será levantada obrigatoriedade de alguma variável requerer estabilidade e precisão (vazão, temperatura, pressão, umidade, viscosidade, entre outras inúmeras), sendo assim, essa didática desempenha função fundamental na formação de um engenheiro (Ogata, 2011). Logo, o estudo em questão busca sanar o que se denomina Problema de Controle, para tanto deve-se estabelecer as variáveis atuantes e parâmetros ambientais, os quais são:

- Planta

É uma parte de um equipamento ou instalação industrial, eventualmente um conjunto de itens de uma máquina que funcionam juntos, cuja finalidade é desempenhar uma dada operação. (Araújo, 2007)

- Processo

Pode ser definido como uma operação ou desenvolvimento natural que evolui progressivamente, caracterizado por uma série de mudanças graduais que se sucedem de modo relativamente fixo, conduzindo a um resultado ou finalidade particular. (Ogata, 2011)

- Sistema

É uma disposição, conjunto ou coleção de partes, dentro de um universo, que estão conectadas ou relacionadas de tal maneira a formarem um todo. (Ogata, 2011)

- Sistema Físico

É uma parte do universo que foi delimitada para estudo. (Araújo, 2007)

- Especificações de Desempenho

São descrições do comportamento a ser apresentado pelo sistema físico, conforme solicitação do usuário. (Ogata, 2011)

- Modelo

Consiste na representação de certas características do sistema físico que são relevantes para seu estudo. (Araújo, 2007)

- Controle

É a ação de fazer com que um sistema físico atenda as especificações de desempenho determinadas a priori. (Araújo, 2007)

- Controlador

Dispositivo utilizado para a obtenção do controle de um sistema físico. (Araújo, 2007)

- Sistema de Controle

Conjunto formado pelo sistema a ser controlado e o controlador. (Araújo, 2007)

A Figura 25 expõe um exemplo de método de controle que exemplifica alguns dos conceitos anteriormente citados, assim seu conjunto representa a Planta, o reconhecimento de padrões como sendo o Processo, a disposição e conexão entre os itens representando o Sistema, o bloco do Controlador responsável por atuar sobre a planta e a ação do braço robótico como o Controle.

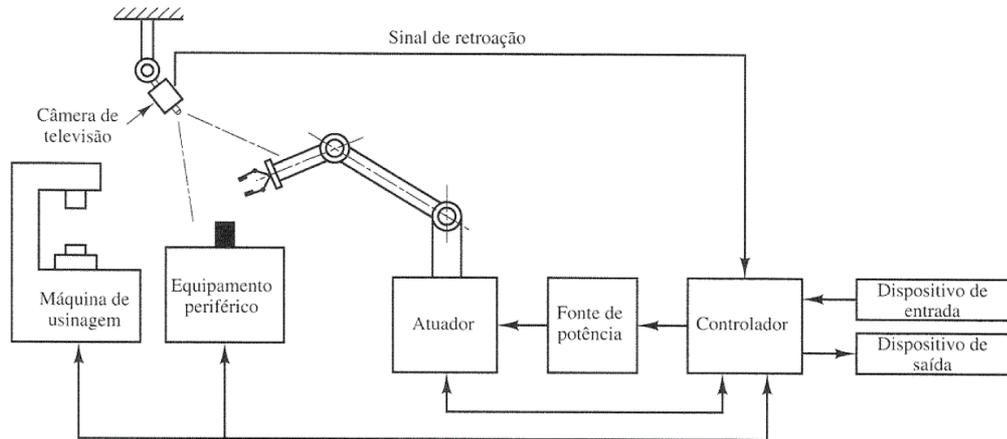


Figura 25: Robô utilizado em processo de reconhecimento de padrões (FERREIRA, 2016)

### 2.8.1 Malha de controle

A malha de controle define a forma como o processo acontecerá, bem como cada etapa de controle irá comunicar com a seguinte. Usando de representação por diagrama de blocos, a mesma apresenta um modelo universal. Há duas formas de estruturar o controle de processo, malha aberta e malha fechada. No primeiro caso tem-se um processo que não conta com realimentação, sendo assim uma vez em que as constantes de proporcionalidade são definidas, o sistema não sofrerá mais ajustes, desta forma a saída não apresenta relação direta com a entrada. É válido notar que, em geral, os sistemas são calibrados em ambientes controlados, ao serem aplicados em campo comumente nota-se diferenças em operação (FERREIRA, 2016). Esse método é facilmente encontrando em fornos domésticos, chuveiros e máquinas de lavar roupa, já um exemplo teórico pode ser visto na Figura 26 abaixo:

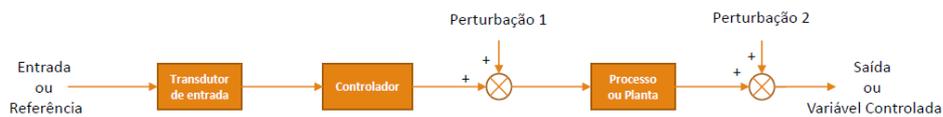


Figura 26: Malha de controle aberta (FERREIRA, 2016)

Para o segundo caso, malha fechada, prioriza-se a precisão e fidelidade da saída em relação a entrada, assim é estabelecida uma relação direta entre entrada e saída. Esse controle é mais confiável e refina que o sistema de malha aberta, pois o mesmo conta com uma linha de realimentação e ajuste de erro, seja ele de estado transitório ou permanente. Usualmente aplicado em sistemas que executam tarefas as quais a saída requer maior nível de confiabilidade, o mesmo se estende desde aplicações simples como o controle de tensão

de uma fonte de banca à até piloto automático de carros e aviões. O sistema de malha fechada tem a seu favor a capacidade de se adequar às características do ambiente, sendo assim, desde que o mesmo esteja corretamente dimensionado e sintonizado, a resposta apresentará valores próximos ao desejado para uma gama de ambientes e distúrbios sobre o sistema (FERREIRA, 2016). A Figura 27 ilustra um modelo teórico de malha fechada:

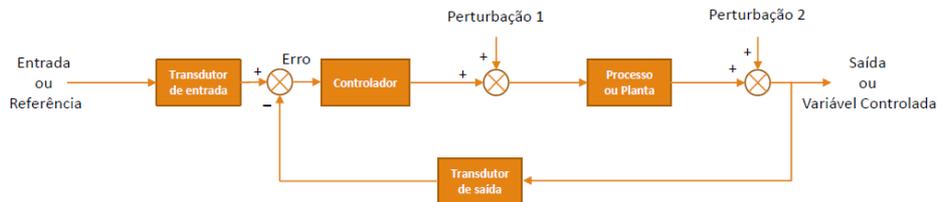


Figura 27: Malha de controle fechado (FERREIRA, 2016)

### 2.8.2 Controle ON-OFF

A estratégia de controle ON-OFF se baseia na ativação do atuador uma vez detectado o sinal de controle. Este atuador é geralmente ativado a uma velocidade constante. Esta abordagem é a que geralmente consome menos energia. Entretanto, sua funcionalidade é limitada e, por operar com velocidades fixas, a sua manipulação pode parecer antinatural. (Lôbo, 2014)

Tendo a baixa demanda de processamento ao seu favor, o controle ON-OFF apresenta certa instabilidade ao se aproximar da região de estabilização. Isso se dá pela participação de ruídos intrínsecos ao sistema e processo de medição. (MONKMAN G. J., 2007) A fim de contornar esse problema, pode ser estabelecida a variável de *Offset* (zona morta), assim o sistema irá ligar ao atingir um valor mínimo e somente suspenderá o fornecimento de potência à planta quando a saída for abaixo de um outro valor definido, desta forma pode-se definir o valor do *Offset* como sendo a diferença entre o valor de acionamento e o de desligamento do sistema a ser controlado, a forma de onde final será de uma histerese (Ogata, 2011). O comportamento deste modelo de controle pode ser mais bem visualizado na Figura 28:

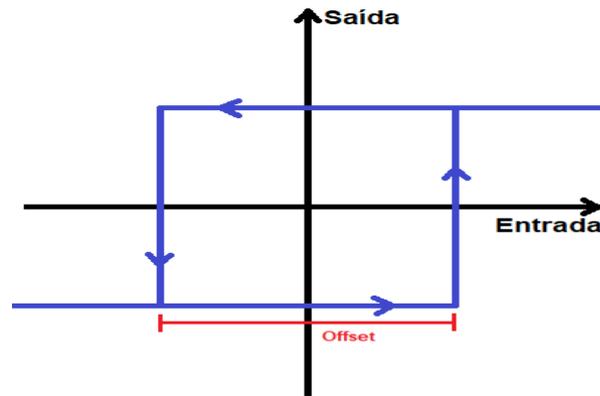


Figura 28: Histerese do controlador ON-OFF (imagem elaborada pelo autor)

### 2.8.3 Controle proporcional

A estratégia de controle proporcional faz a variável de saída ser proporcional à variável de entrada. Diferentemente do controle ON-OFF, que ativa ou desativa a variável de saída, este controle identifica diferentes níveis de ativação. Consome mais energia do que o controle ON-OFF, porém traz maior naturalidade ao movimento, bem como aumenta-se a gama de possibilidades de uso por parte do usuário. Um ponto que merece a devida atenção ao se tratar deste método, é a necessidade de treinamento por parte do usuário. (Lôbo, 2014)

Por mais que este modelo represente uma maior fidelidade ao movimento real, as dificuldades encontradas em estabelecer o modelo matemático que descreva seu comportamento, melhor captação do sinal e problemas diversos em sua implantação, o mesmo não será amplamente abordado nesta dissertação.

## 2.9 Motores

Pode-se definir motores elétricos de forma sucinta como dispositivos capazes de converter energia elétrica em energia mecânica, ou até mesmo comparando-os com conversores eletromecânicos de potência passíveis de gerenciar grandes valores de energia. (PAIXÃO, 2009)

Ao energizar a máquina com corrente alternada, são utilizados os motores de corrente alternada (CA), já quando se trabalha com fornecimento de corrente contínua, são utilizados os motores de corrente contínua (CC). (PAIXÃO, 2009)

### 2.9.1 Motores de corrente contínua

Como o sistema em estudo e desenvolvimento direciona-se à portabilidade, o uso da corrente contínua mostra-se mais atrativo. Dito isso, será levado em consideração apenas os motores de corrente contínua.

Os motores CC podem ser divididos em sistemas de imã permanente e com enrolamento de campo. A principal razão para a utilização deste componente se dá quando precisa de um controle pequeno e de fácil implementação, assim descarta o uso de inversores de frequência, como seria o caso de motores CA. (WEG, 2003)

Podendo ter seu toque estável em diferentes níveis de velocidade, este tipo de máquina pode trabalhar com baixos patamares de tensão e de fácil controle por dispositivos com baixa taxa de processamento. Observa-se que em muitos momentos deste trabalho busca-se otimizar os periféricos a fim de obter o hardware e firmware de menor complexidade possível, logo esse tipo de máquina encaixa-se bem ao propósito.

Em grande maioria das aplicações de pequeno porte, os motores tendem a utilizar a configuração de ímãs permanentes. Para o correto funcionamento a arquitetura da máquina é formada por dispositivos magnéticos fixados em sua armadura e uma bobina enrolada no seu eixo. A condução da corrente é feita através de comutadores por meio das escovas que são simetricamente fixadas em sua carcaça, assim quando o eixo entra em estado de rotação o sentido da corrente em seu enrolamento principal estará continuamente mudando de direção e, conseqüentemente, o sistema se portará como um eletroímã o qual terá o sentido do campo magnético alterado diversas vezes por segundo (Torres A. E., 2004). Com essa relação pode-se definir a taxa de rotações por minuto. A fim de exemplificar melhor o que foi descrito, a Figura 29 exibe um modelo de motor de corrente contínua por ímãs permanentes:

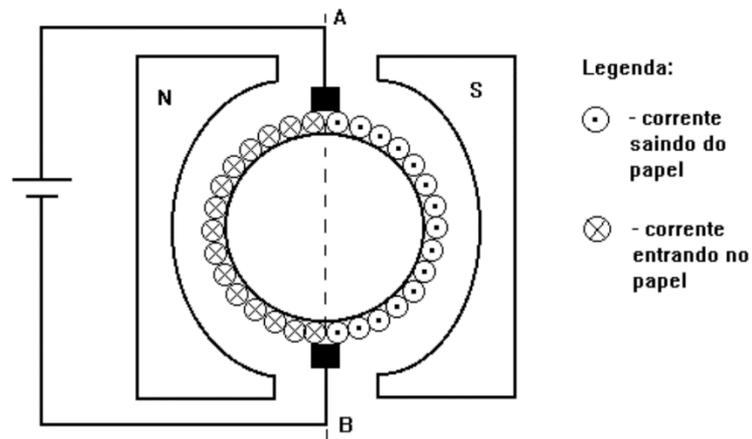


Figura 29: Corte esquemático de um motor DC ímã permanente (Torres A. E., 2004)

### 2.9.2 Servomotor

Os servomotores são recomendados quando se demanda rápido controle de corrente, larga faixa de controle de velocidade, alta precisão de posicionamento, dimensões reduzidas, entre outros fatores (WEG, 2003).

Um servomotor é, em sua totalidade, um conjunto de diversos itens a formar o circuito de controle e potência, desta forma o mesmo precisa de um sistema bem projetado e robusto para desempenhar suas funções. Os três tipos de servomotores mais utilizados são: de corrente contínua, de corrente alternada síncrono e de corrente alternada assíncronos. A utilização de um determinado tipo de servomotor depende da dinâmica do sistema, do torque máximo da carga, da velocidade entre outros (WEG, 2003).

Quando se trata de aplicações em pequeno porte, o circuito do servo motor adiciona um potenciômetro em seu eixo final da caixa de redução. No momento em que o controlador realiza a leitura, o ângulo de rotação do eixo do resistor variável informa a posição em que se encontra o eixo do servomotor. Com esse circuito de realimentação a etapa de controle é capaz de realizar correções até que o valor desejado seja alcançado (Santos, 2007).

Normalmente, os servomotores dispõem de uma angulação de trabalho limitada, a qual costuma ficar em  $0^\circ$  e  $180^\circ$  (ou  $-90^\circ$  e  $90^\circ$ ). A estrutura do servomotor pode ser vista na Figura 30 e Figura 31:



Figura 30: Visão externa do servomotor (Santos, 2007)

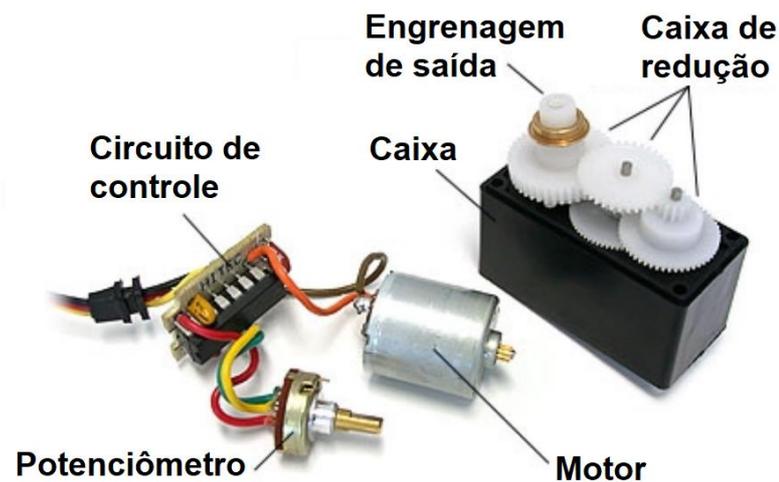


Figura 31: Componentes do servomotor (Santos, 2007)

O controle pode ser feito através de um sinal PWM (*Pulse-Width Modulation* – Modulação por largura de pulso) com valores de frequência e período definidos. O pulso deve ter período total de 20ms (50Hz) o qual é composto por um valor mínimo de 1ms para o nível alto (5 volts, neste caso) e o restante em nível baixo (0 volts). O tempo em que o sinal permanece em nível alto irá determinar a posição em que o controlador deve posicionar o eixo do servomotor. Esse período deve variar entre 1ms (para 0°) e 2ms (para 180°) (Santos, 2007). A fim de exemplificar, a Figura 32 demonstra esse controle:

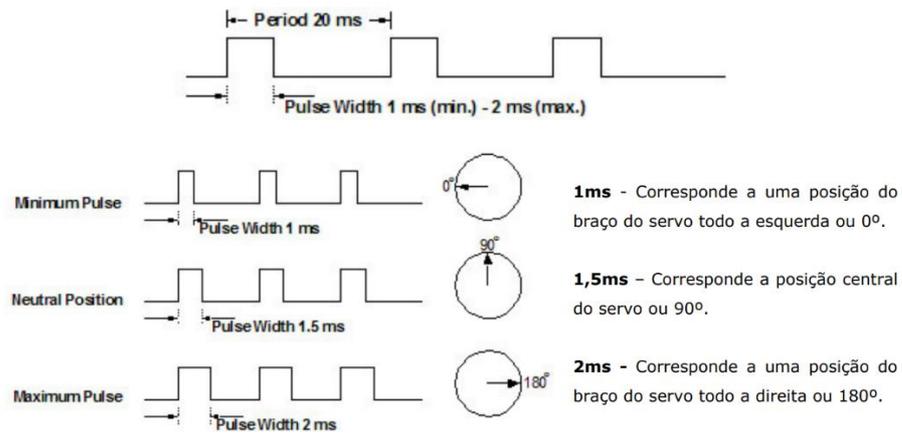


Figura 32: Pulso de controle do servomotor (Santos, 2007)

## 2.10 Órteses e Próteses

Ao contrário do sensor comum, o uso de próteses e órteses tem seu primeiro registro datado do período entre 3500 e 1800 a.C. em um poema hindu. Nesta história cita-se uma rainha guerreira que sofreu amputação de uma de suas pernas, a mesma recorreu à um membro inferior feito de metal para que pudesse retornar ao campo de batalha (Braskem, 2017). Mais à frente em Roma, por volta de 370 a.C. Hipócrates, considerado o precursor da medicina, criou o conceito de talas para corrigir fraturas na região da tíbia. A órtese baseava-se na construção de um arranjo com pedaços de madeira e argolas de couro para que a mantivesse fixa na perna do usuário, a mesma pode ser vista na Figura 33:

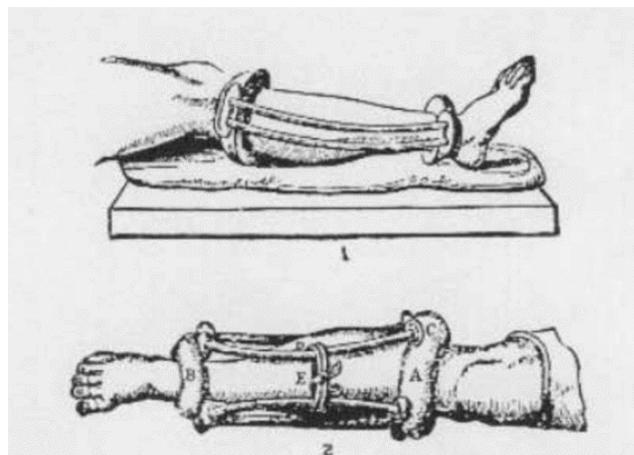


Figura 33: Órtese de Hipócrates (Braskem, 2017)

Após um período de mais de dois mil anos de pouca evolução na área, o cirurgião holandês Peiter Verduyn criou, em 1696, uma prótese que poderia ser usada do joelho para baixo, ela ainda contava com dobradiças que permitia certa movimentação e um tipo de espartilho na altura da coxa para que a mesma pudesse prender-se ao paciente (Braskem, 2017). Após isso o grande número de pesquisas na área alavancou a tecnologia em volta deste campo. Mas, quando se trata de membro superior, por ter uma gama de movimentos mais precisos, ainda ficava a falta de recursos para que o usuário pudesse pegar objetos e realizar atividades corriqueiras do dia-a-dia.

### 2.10.1 Órtese

A órtese pode ser descrita como um equipamento terapêutico de auxílio funcional, utilizados não apenas nos programas de recuperação aplicados sobre os membros superiores e inferiores, como também no tronco na forma de faixas e coletes (Galdino, 2009).

Quando um aparato fisioterapêutico tem como finalidade oferecer apoio ou suporte ao paciente, esse item denomina-se órtese. Sendo assim seja para o auxílio à recuperação à algum membro fraturado e/ou ferido, coletes de postura, talas, ataduras, ou aparatos que tendem a contar movimentos involuntários, a classificação aqui descrita está de acordo (Galdino, 2009). Alguns exemplos podem ser vistos na Figura 34:



Figura 34: Exemplos de órteses (Galdino, 2009)

### 2.10.2 Prótese mioelétrica

Pode-se definir prótese como sendo um artefato que tende a substituir alguma parte do paciente que foi danificada e/ou amputada. Desta forma, a prótese procura devolver integralmente ou em partes algo que foi perdido pelo usuário, geralmente por amputação, e não auxiliar no tratamento de problemas de saúde, como seria o caso das órteses. (Galdino, 2009)

O membro residual de amputação é denominado coto. O coto de amputação, agora considerado como um novo membro, é o responsável pelo controle da prótese durante a apreensão, percepção, comunicação e atividades laborais. Para que isso seja possível, é necessário que o coto tenha um nível adequado, seja estável, presença de um coxim com mioplastia e miodese, um bom estado da pele, com boa sensibilidade, sem úlceras e enxerto cutâneo, ausência de neuromas terminais e espículas ósseas, ter uma boa circulação arterial e venosa, boa cicatrização e ausência de edema. (Tavares, 2000)

Os cotos de amputação de membros superiores são classificados pelo nível de amputação considerando a porcentagem do membro residual. Para tanto, deve ser realizada a medida do comprimento do coto. As amputações proximais ao cotovelo, com exceção da desarticulação do ombro, são também denominadas transumerais, e as distais ao cotovelo, com exceção da desarticulação do punho e parciais de mão, são também chamadas transradiais. (Tavares, 2000)

Ao longo do tempo, as próteses têm servidos mais a fins estéticos do que práticos, sendo assim nos últimos anos o campo de pesquisa em volta de próteses que realizam algum tipo de movimento tem se intensificado em muitas áreas, neste trabalho o tema principal abordado será em próteses de membro superior. Quando o paciente passa pela remoção do membro e parte da musculatura é preservada, a mesma ainda é capaz de receber e conduzir os sinais mioelétricos (este termo será melhor explicado mais a frente). (Carvalho, 2004)

Logo, este aparato que tem se tornado mais popular no munda da tecnologia, presta-se à função de ler e interpretar os sinais elétricos emanados pela musculatura em estudo, processar essa informação e decidir, com base em sinais base previamente medidos, qual função o a parte mecânica deve executar (Carvalho, 2004).

Por dentro da complexa lista de movimentos que o membro superior é capaz de executar, busca-se, entre outros, em um primeiro momento, o movimento de pinça, ou seja,

abrir e fechar a mão mecânica. Por mais que essa meta possa transmitir pouca confiança de início, observa-se que o usuário recebe de bom grado a possibilidade de pegar objetos simples, como um copo com água, e até mesmo utilizar uma caneta, assim recuperando ao menos parte da mobilidade outrora perdida. Nota-se, também, que neste ponto não há somente o impacto físico, como também a direta atuação no psicológico do paciente. (Carvalho, 2004)

Geralmente, esse tipo de sistema requer que o usuário tenha um treinamento prévio para usá-lo. Assim, bem como qualquer outra ferramenta, há um período de adaptação ao uso da tecnologia, prazo esse que varia principalmente com o tempo passado desde a amputação e a saúde muscular da região que será utilizada. (Duque, 2013)

### **2.11 Eletromiografia**

A unidade funcional do músculo é Unidade Motora (UM), a mesma baseia-se em um neurônio alfa e todas as fibras que compõe a musculatura por ele inervadas. As fibras musculares constituem a unidade estrutural contrátil, ou seja, a unidade elementar geradora de força. Uma única UM pode ter entre 3 a 2000 fibras musculares, dependendo do grau de controle e força requerido pelo músculo. (Takahashi, 2006)

Os músculos que controlam movimentos finos, que exigem força precisa e baixa, têm menor número de fibras musculares por UM, geralmente menos que 10 fibras por UM, do que aqueles músculos que controlam grande movimentos e exigem muita força, que podem conter de 100 a 2000 fibras por UM (Takahashi, 2006). A Figura 35 representa o que foi descrito:

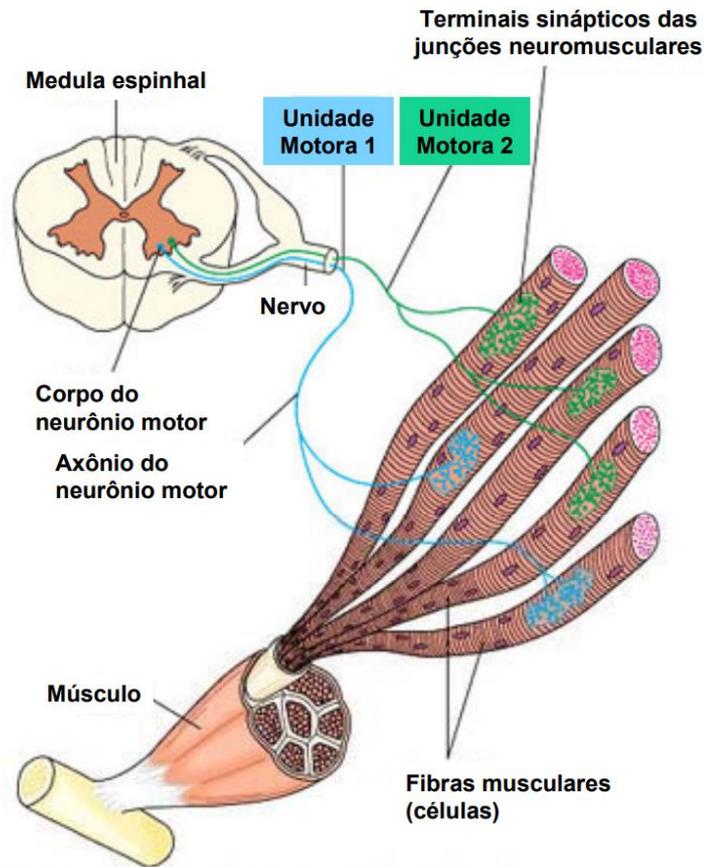


Figura 35: Estrutura da unidade motora (Takahashi, 2006)

A contração das fibras musculares ocorre quando se gera a intenção de movimento no neurônio motor que atende aquela área, assim que o potencial de ação vindo do nervo atinge o terminal do axônio e alcança um limiar de despolarização, a intenção de movimento é difundida por toda a estrutura muscular e assim forma-se o potencial de ação muscular. Propagando-se em ambas as direções da fibra muscular, o potencial de ação, é desencadeado um processo de deslizamento dos filamentos de actina sobre os de miosina, esses últimos são as principais proteínas contráteis das miofibrilas, logo origina-se a contração muscular. A Figura 36 demonstra esse processo:

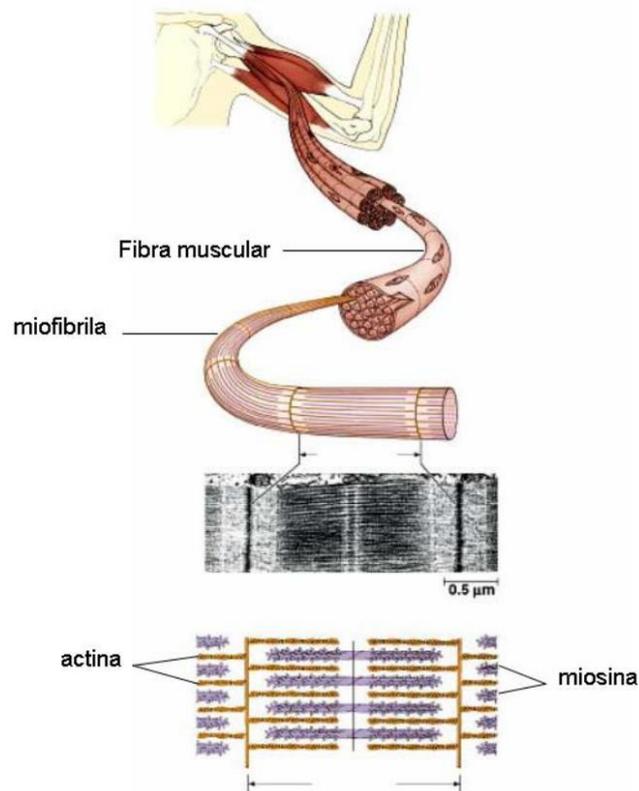


Figura 36: Constituição da fibra muscular (Takahashi, 2006)

Eletromiografia é definida como o registro extracelular da atividade bioelétrica gerada pelas fibras musculares. Pode ser realizada por meio de agulha, que capta a atividade elétrica de poucas unidades motoras ou por meio de eletrodo de superfície, que mensura a atividade elétrica de várias unidades motoras ao mesmo tempo. Apesar de captar a atividade elétrica promovida pelo recrutamento das unidades motoras e não a força muscular, logo há boa correlação entre o número de unidades motoras ativadas e a força muscular. (Resende, 2011)

### 2.11.1 Estrutura do sinal mioelétrico

Ao abrir ou fechar uma mão, o cérebro envia a informação que vai pelo sistema nervoso, passando pelo sistema muscular, até chegar a seu objetivo, que é fazer a mão abrir e fechar. São gerados sinais mioelétricos de níveis muito baixos, sinais esses com amplitude entre 100  $\mu\text{V}$  a 2 mV com uma impedância muito alta e altos índices de ruídos. (Duque, 2013)

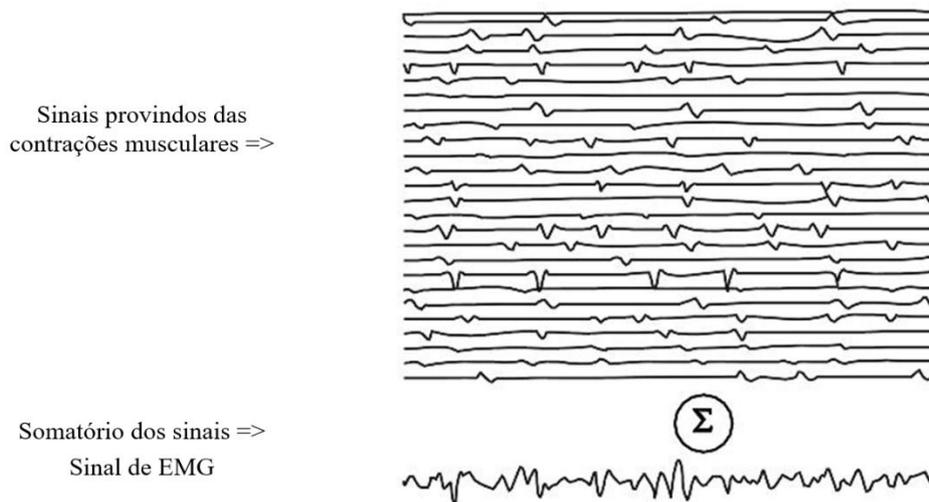
Para tal situação, é necessário o uso de um amplificador diferencial e um processador para processar esses sinais, e torná-los compatíveis com uma prótese mecânica (Duque, 2013). O sinal de Eletromiografia é definido pela equação X apresentada abaixo, onde  $EMG(t)$

é a resultante do sinal eletromiográfico,  $SPAUM_j(T)$  é a série de potenciais de ação da unidade motora,  $n(t)$  representa o ruído presente na leitura e  $t$  denomina o tempo da amostra:

$$EMG(t) = \sum_{j=1}^{Nm} SPAUM_j(t) + n(t)$$

*Equação 14 Definição matemática do sinal eletromiográfico (Duque, 2013)*

Com base na equação acima o final resultante pode ser observado na Figura 37 abaixo:



*Figura 37: Sinal eletromiográfico (Duque, 2013)*

Em resumo, toda o caminho percorrido pelo sinal, desde a intenção de movimento gerada no cérebro, passando por todas as estruturas nervos até chegar ao músculo de destino podem ser vistas na Figura 38:

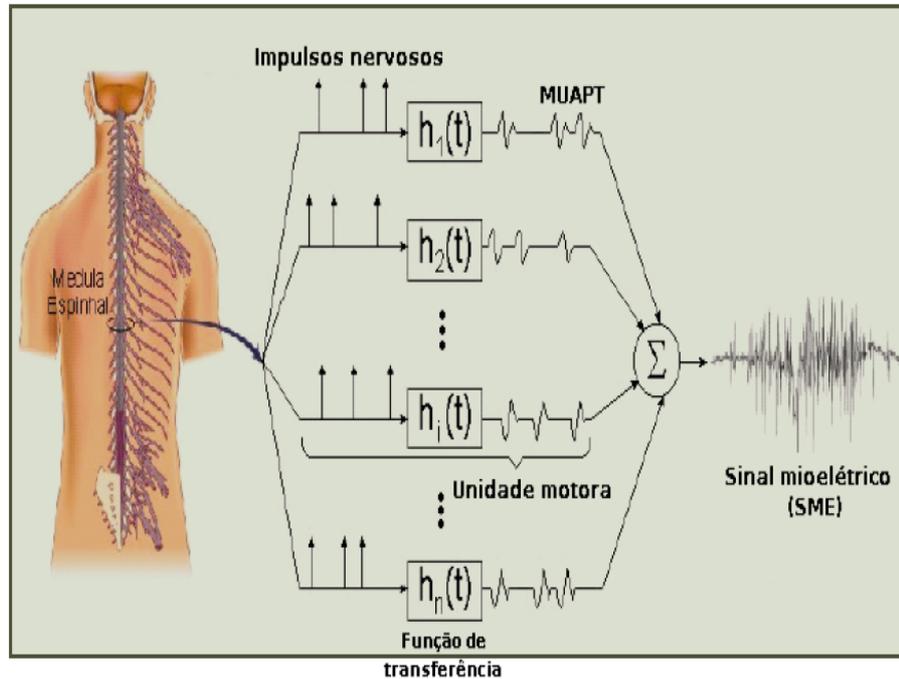


Figura 38: Caminho do sinal eletromiográfico (Duque, 2013)

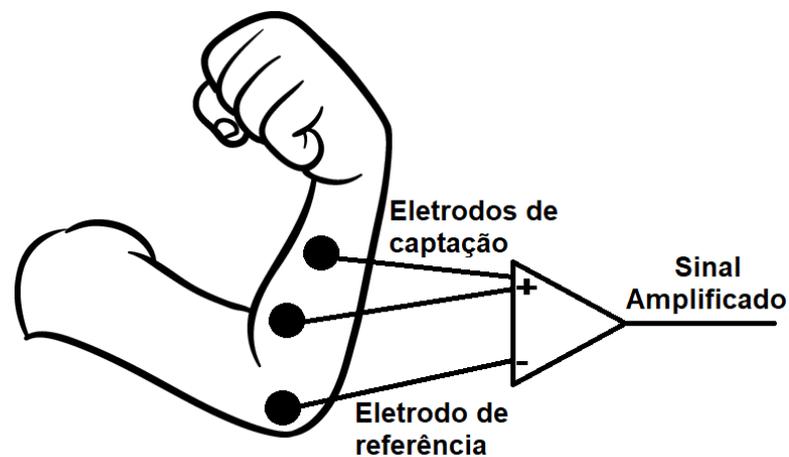
### 2.11.2 Coleta do sinal muscular

Para coletar-se o sinal mioelétrico, deve-se definir o quão invasivo será o sistema. Desta forma, há duas formas de obter a intensão de movimento proveniente da musculatura, e são elas o eletrodo invasivo e de superfície. Para o primeiro método tem-se a utilização de agulhas e/ou microagulhas que perfuram a pele até o músculo ou nervo a fim de captar os sinais eletromiográfico. É uma prática rotineira nas clínicas e causa certo desconforto aos pacientes. Quando se opta pelo uso dos eletrodos de superfície, os quais serão usados neste trabalho, eles baseiam-se na utilização de eletrodos de metais não oxidantes para entrar em contato com a superfície da pele na região do músculo que se quer captar os sinais. Esses sinais são extremamente difíceis de processar para extração de dados de interesse clínico e tem sido alvo de estudos de vários grupos de pesquisa no mundo (Duque, 2013). O tipo de sensor utilizado nessa estratégia pode ser visto na Figura 39 abaixo:



*Figura 39: Eletrodo eletromiográfico de superfície (Duque, 2013)*

Esse tipo de eletrodo traz algumas desvantagens como só ser capaz de detectar a musculatura superficial e, em muitos casos, conta com ampla área de captura, o que pode acarretar a interferência de sinais originados em outras regiões musculares. Por razões técnicas, o eletrodo é tripolar, ou seja, são necessários três pontos de detecção, sendo dois correspondentes aos sinais e um atuando como referência (Takahashi, 2006). Essa configuração pode ser vista na Figura 40 abaixo:



*Figura 40: Posicionamento dos eletrodos de superfície (Imagem elaborada pelo autor)*

### 3 Metodologia

#### 3.1 Diagrama de blocos do sistema

O desenvolvimento do protótipo se deu por etapas. Por diversas razões, entre elas a simplificação, buscou-se trabalhar em módulos. O mesmo inicia com o estudo do correto posicionamento dos eletrodos para um bom resultado na coleta do sinal, depois passa pelo desenvolvimento do condensador e amplificador, a próxima etapa compreende a montagem da central de processamento e transmissão dos dados, em seguida tem-se o trabalho com a prótese de metal juntamente com o servo motor. Paralelo à última etapa descrita desempenha-se esforço no desenvolvimento da placa receptora dos sinais e responsável por traduzir os dados para o computador, já a última parte compreende o desenvolvimento do software que receberá os dados traduzidos e os organizará em forma gráfica. Tal organização pode ser vista na Figura 41 abaixo:

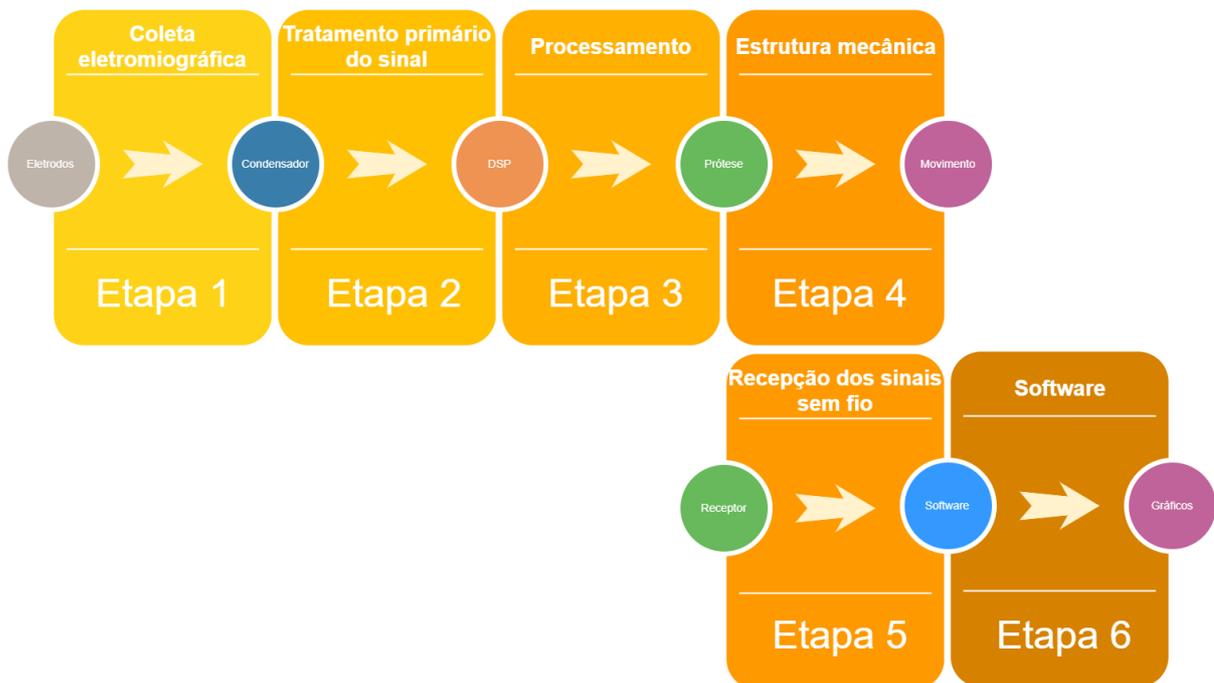


Figura 41: Etapas do desenvolvimento (Imagem elaborada pelo autor)

#### 3.2 Etapa 1 – Coleta do sinal Eletromiográfico

O início do sistema se dá em coletar os sinais eletromiográficos da forma mais eficiente possível. Para um menor grau de complexidade e facilidade de operação, optou-se pelo uso

de sensores se superfície, logo para se usar a prótese não há necessidade de intervenção cirúrgica. Como já é de se esperar, este modo de aquisição de dados implica em uma quantidade maior de ruídos o que requer melhores filtros e ocasiona movimentos possíveis limitados.

Neste momento, trabalha-se com dois componentes, os quais são o cabo de eletromiografia e os sensores de superfície. O primeiro será responsável por conduzir os sinais vindos do sensor até a entrada de sinal do condensador (o qual será melhor explicado mais à frente neste trabalho), sendo assim é crucial que o mesmo apresente uma boa blindagem à ruídos externos. A fim de obter um melhor resultado optou-se pelo uso de um cabo de aplicação profissional de modelo CSI 1073/S, o qual é mostrado na Figura 42, porém o mesmo sofreu alteração em sua conexão com o eletromiógrafo onde a original foi substituída por um conector do tipo DB9, como pode ser visto na Figura 43. Já a estrutura responsável por se conectar com os eletrodos manteve-se a mesma, que é composta por uma pequena trava metálica circular, a qual pode ser vista na Figura 44.



*Figura 42: Cabo de eletromiografia CSI 1073/S (Imagem elaborada pelo autor)*



*Figura 43: Conector modificado para o padrão DB9 (Imagem elaborada pelo autor)*



*Figura 44: Conector para os eletrodos (Imagem elaborada pelo autor)*

O segundo componente, ou sensores que pode ser visto na Figura 45, serão posicionados sobre a pele que recobre a musculatura desejada, assim o mesmo apresenta em suas bordas inferiores uma cola para que possa fixar-se no local, já em seu centro da parte debaixo contém um gel condutivo que fará, de fato, a condução do sinal muscular para o conector metálico localizado em sua parte superior. Para que seja feita a coleta dos sinais necessita-se de três eletrodos conectados ao paciente. Esta última peça pode ser vista na Figura 46.



*Figura 45: Eletrodos eletromiográficos de superfície (Imagem elaborada pelo autor)*



*Figura 46: Parte inferior do eletrodo de superfície (Imagem elaborada pelo autor)*

A última parte integrante desta etapa, compreende o posicionamento dos eletrodos. Neste momento algumas considerações serão feitas. Como a prótese pretende executar a função de pinça para que possa simular uma mão e manusear objetos, os eletrodos serão posicionados na parte anterior no antebraço. Sendo assim, assume-se que o usuário tem a estrutura e saúde muscular correspondente ao braquiorradial suficientemente preservados ao ponto de gerar sinais eletromiográficos passíveis de serem captados por eletrodos de superfície.

Como o sistema condensador de sinais trabalha com o sinal muscular diferencial, será necessário posicionar dois eletrodos na musculatura e um terceiro que servirá de referencial para o calcula da diferença. Sendo assim, os eletrodos de sinal serão posicionados suficientemente espaçados entre si a ponto de caber o eletrodo de referência entre eles.

Desta forma será possível captar os sinais que antecedem e procedem o eletrodo central e ao subtrair um do outro, obtém-se o valor eletromiográfico gerado naquela região. Esse posicionamento pode ser observado na Figura 47 abaixo:



*Figura 47: Posicionamento dos eletrodos de superfície (Imagem elaborada pelo autor)*

### **3.3 Etapa 2 – Tratamento primário do sinal**

Após ter definido a forma como o sinal será coletado, é de suma importância aplicar filtros e amplificadores a fim de tornar o sinal inteligível ao processador digital de sinais. Sendo assim o circuito está dividido em três partes, as quais dividem-se em filtro de rejeição de sinais em comum (CMRR – *Common Mode Reject Ratio*), filtros passa-alta e, por último, o circuito responsável por amplificar o sinal.

Os sinais vindos dos eletrodos são direcionados para duas entradas do amplificador operacional de alta impedância, como é o caso do TLC274. Devido à proximidade dos eletrodos, espera-se que os ruídos que afetam uma nos terminais deve impactar no outro, dessa forma é correto afirmar que os sinais que forem iguais em ambas as entradas podem ser considerados ruídos. Logo, mostra-se necessário o desenvolvimento de um circuito capaz de comprar as duas entradas, identificar os sinais iguais e excluir as componentes que se repetirem. Para isso precisa-se do circuito CMRR, para isso o mesmo precisa ser montado com três amplificadores operacionais. Como o TLC274 inclui em seu encapsulamento quatro circuitos de amplificação, o mesmo mostra-se atrativo para solucionar este problema. Assim que os sinais chegam à placa condensadora ambos os sinais são distribuídos para dois AOP (U1:A e U1:B) onde cada um deles tem como função reduzir a impedância do respectivo sinal recebido, logo após saem dois novos sinais que serão encaminhados à um terceiro AOP (U1:C) o qual executará a função do CMRR. Para tanto, a entrada correspondente ao sinal não-

invertido do AOP será conectada à terra por um resistor variável (RV1). O ajuste desse resistor fará com que o CMRR tenha o resultado esperado. Tais detalhes podem ser vistos na Figura 48 abaixo:

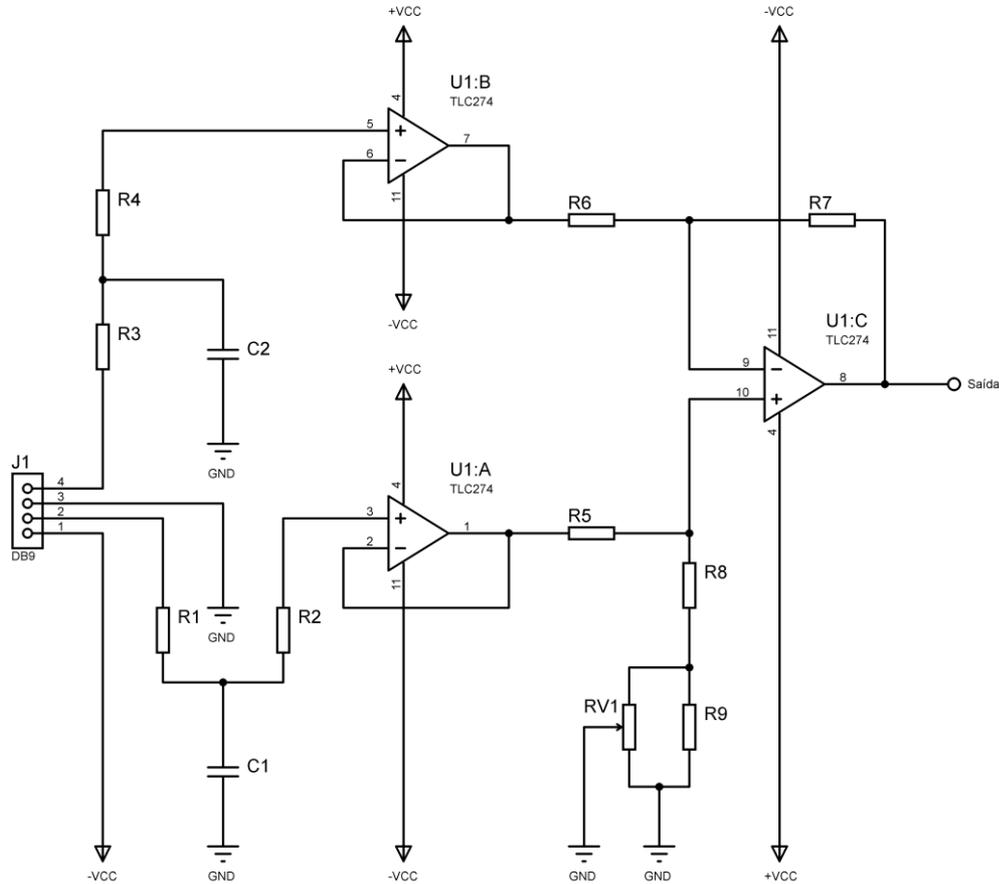


Figura 48: Entrada do sinal no condicionador (Imagem elaborada pelo autor)

Para realizar o ajuste do CMRR deve-se injetar no sistema o mesmo sinal vindo de um gerador de funções (frequência, fase e amplitude) nas duas entradas e ajustar o RV1 até que a saída do sinal pelo U1:C medida em um osciloscópio seja uma linha reta, logo isso significará que quando as duas entradas forem iguais o circuito irá rejeitar este sinal, assim eliminando o ruído que se repete em ambos os eletrodos.

Configurou-se o gerador de função a fim de entregar uma onda senoidal simétrica com 1Vpp, como pode ser visto na Figura 49 abaixo:

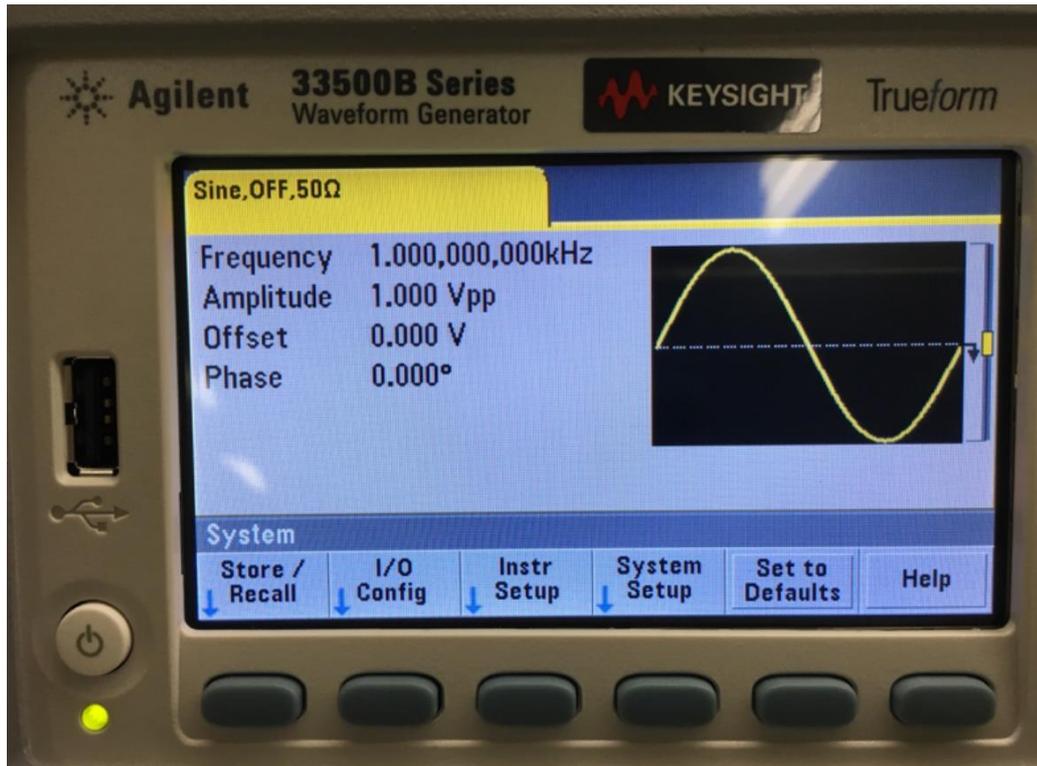


Figura 49: Gerador de função (Monteiro, 2016)

Em seguida o gerador de função terá seu terminal positivo conectado nas pontas branca e vermelha do cabo de EMG e o terra ligado junto à terra da placa condensadora, neste momento já se tem o sinal simulado na entrada do condensador. O próximo passo será conectar a saída do AOP U1:C no terminal de entrada do osciloscópio, e, novamente, o aterramento da placa condensadora junto ao terra do aparelho. Caso o ajuste do resistor variável RV1 esteja mal dimensionado, deve-se ver a forma de onda do sinal de entrada fornecido pelo gerador de função. Para melhor exemplificar, a Figura 50 exibi a tela do osciloscópio:

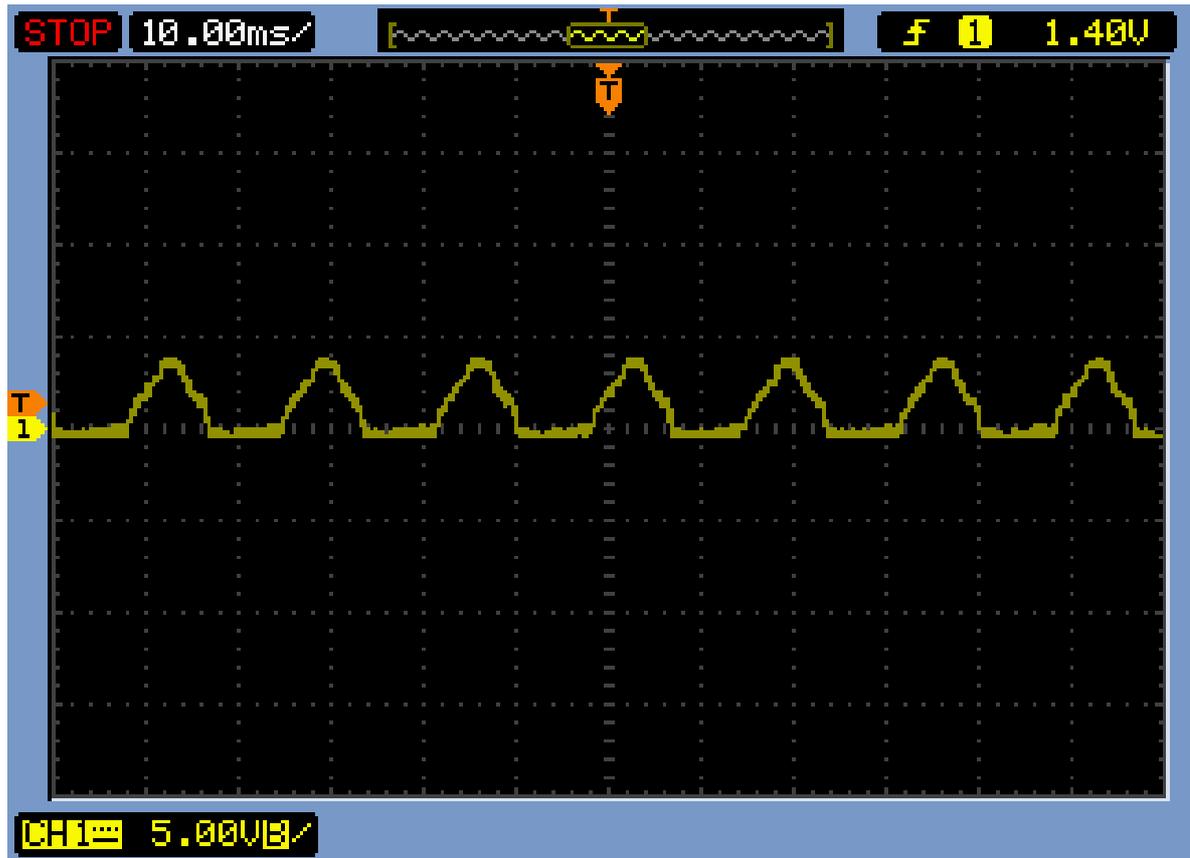


Figura 50: Forma de onda do CMRR desajustado (Imagem elaborada pelo autor)

Tendo como base o sinal mostrado pelo osciloscópio na Figura 51, deve-se ajustar o valor de RV1 de forma que o sinal senoidal diminua sua amplitude gradualmente até que o mesmo assuma uma forma retilínea com um *offset* (deslocamento do valor de tensão em relação à referência) de aproximadamente o valor de entrada do sinal vindo do gerador de função. Para realizar a configuração do RV1 deve-se começar a rotacionar o eixo do componente em um sentido arbitrário enquanto observa-se o resultado na tela do osciloscópio, caso o sinal comece a diminuir, o sentido de rotação está correto, se o contrário acontecer, deve-se inverter o sentido de ajuste do componente. Após ter feito a configuração correta do CMRR, deve-se obter uma forma de onda similar à mostrada na Figura 51 abaixo:

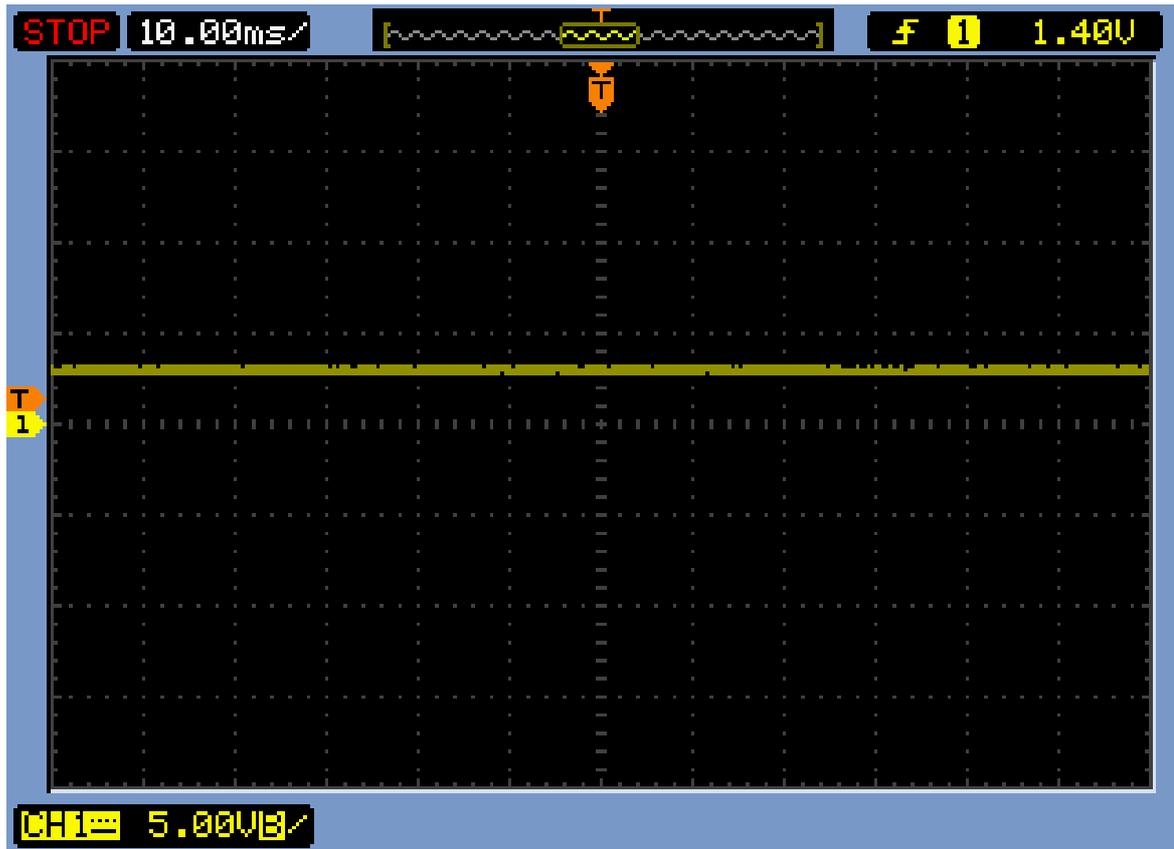


Figura 51: Forma de onda do CMRR ajustado (Imagem elaborada pelo autor)

Em seguida, o sinal que já está previamente filtrado, precisa receber um maior refinamento. Como a musculatura em estudo produz um sinal na ordem de 160Hz, é plausível assumir que as frequências abaixo disso possam ser desprezadas sem acarretar perdas significativas para o sistema. Sendo assim, desenvolveu-se um filtro ativo passa-alta utilizando a topologia exibida na Figura 20, onde usou-se o amplificador operacional LM324. Visto que não há pontos críticos nesta fase do circuito, pode-se usar um AOP genérico, ou seja, sem características especiais, ao contrário do caso do TLC274.

Para montar o circuito, ligou-se os resistores R10 e R11 juntamente aos capacitores C3 e C4 em conjunto com o circuito integrado U2:A de forma a atender o circuito para essa aplicação, o qual foi anteriormente levantado. Logo o circuito desta etapa ficou como é mostrado na Figura 52 abaixo:

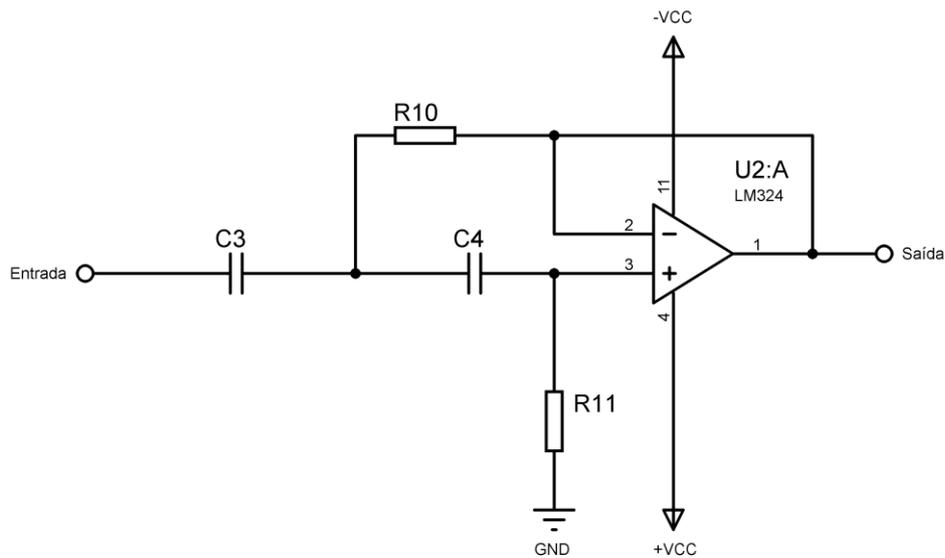


Figura 52: Filtro passa-alta do condensador (Imagem elaborada pelo autor)

Para o correto funcionamento, deve-se calcular os valores dos capacitores e resistores, assim, assume-se que a frequência de corte será em 160Hz, logo, utilizando a Equação 6 tem-se o seguinte cenário:

$$f_c = \frac{1}{2\pi\sqrt{R10R11C1C2}}$$

$$f_c = \frac{1}{2 * \pi * \sqrt{10x10^3 * 10x10^3 * 100x10^{-9} * 100x10^{-9}}}$$

$$f_c \cong 160Hz$$

Feito isso, sabe-se que os valores dos resistores R10 e R11 deverão ser ambos de 10kΩ e os capacitores C3 e C4 serão iguais à 100nF. Desta forma, a filtragem do circuito deve eliminar os ruídos abaixo de 160Hz, porém ainda assim nem todos os ruídos serão devidamente atenuados. A fim de diminuir este problema optou-se pelo cascadeamento de um novo filtro em série com o primeiro, portando as mesmas características. Sendo assim o mesmo será constituído pelo amplificador operacional LM324 que está representado como U2:B, ligado em conjunto com os resistores R12 e R13 que assumem o valor de 10kΩ e os capacitores C5 e C6 que equivalem, individualmente, à 100nF. Assim, obtém-se um filtro

passa-alta de duas etapas e, conseqüentemente, um sinal com menos ruídos. O circuito descrito pode ser visto na Figura 53 abaixo:

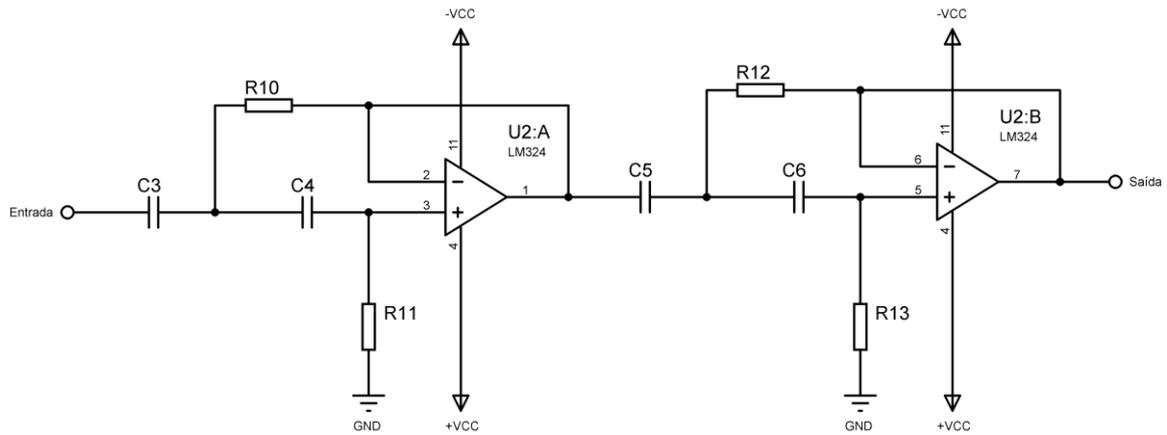


Figura 53: Filtros passa-alta do condensador (Imagem elaborada pelo autor)

Após esses estágios de tratamento, o sinal encontra-se com um nível de ruído razoavelmente baixo, porém ainda não está pronto para ser enviado ao processador digital de sinais, pois falta uma última etapa de tratamento.

O sinal que sai do amplificador operacional U2:B ainda apresenta uma baixa amplitude, então o próximo passo será criar um circuito capaz de manter as componentes de fase e frequência do sinal, porém aumenta seu nível de tensão. Para isso será usado o terceiro conjunto de AOP que integra o encapsulamento do LM324, assim a montagem do circuito se dá por ligar a entrada não-invertida do U2:C no terra (visto que se trata de um sinal de amplitude simétrica), a entrada invertida sendo ligada diretamente à saída do U2:B e um resistor variável (RV2) de precisão sendo ligado entre a entrada invertida do U2:C e a saída do mesmo AOP. Desta forma o circuito será capaz de variar desde o ganho unitário, ou seja, aquele em que o sinal que sai é igual ao que entra, à até o ganho máximo, que representa o valor máximo da fonte de alimentação.

A forma como monta-se o circuito proposto, pode ser visto na Figura 54 abaixo:

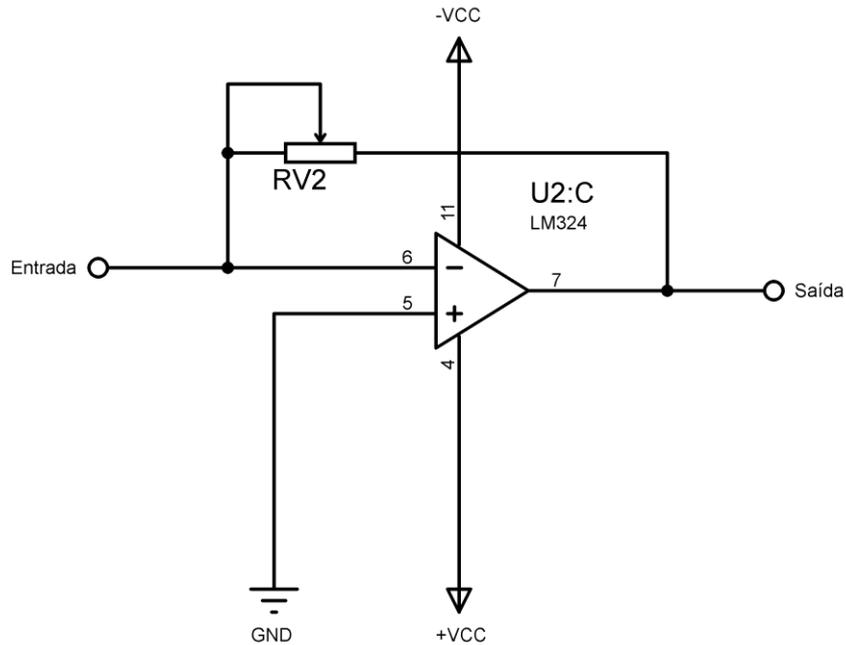


Figura 54: Ajuste de ganho do sinal de saída (Imagem elaborada pelo autor)

Até este ponto, o circuito condensador já foi teorizado, calculado e esquematizado. Para que o sistema se torne aplicável foi necessário realizar a montagem dele em uma placa de circuito impresso. Sendo assim, este ponto trata o posicionamento de todos os componentes ativos e passivos em suas devidas configurações e soldados na placa. A forma final do circuito pode ser vista na Figura 55 abaixo:

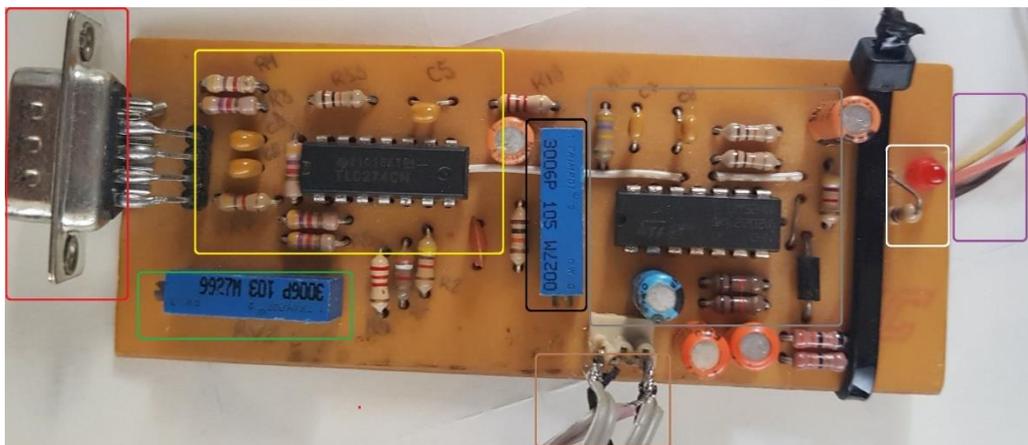


Figura 55: Placa do condensador pronta (Imagem elaborada pelo autor)

Na Figura 55 pode-se observar que a placa está dividida em caixas coloridas, pois bem, desta forma é possível compreender com maior facilidade cada um dos blocos do circuito que foram explicados em tópicos anteriores.

Observando a imagem, a caixa vermelha destaca o conector DB9 o qual será responsável pela conexão com o cabo de eletromiografia, sendo assim, os sinais vindos dos eletrodos de superfície serão conduzidos e entrarão no circuito por este componente. Em seguida tem-se o desenho amarelo, o qual compreende o circuito integrado TLC274 (U1:A, U1:B e U1:C) bem como os componentes externos que se fazem necessários para o seu correto funcionamento e, em conjunto, o bloco de cor verde exibe o resistor variável RV1 que, ao ser operado, realiza o ajuste do CMRR.

Seguindo o percurso à direita na placa de circuito impresso, encontra-se a caixa identificada pela cor preta, esta destaca o resistor variável RV2 que executa o ajuste do ganho final do circuito e logo mais à direita tem-se a caixa cinza que exibe o componente LM324 (U2:A, U2:B e U2:C) que realiza a filtragem do sinal por filtros passa-alta em dois estágios e o ganho do sinal em conjunto com os resistores e capacitores adjacentes a ele.

Abaixo do LM324, encontra-se a caixa marrom, a qual mostra o conector responsável por entregar o sinal de saída da placa condicionadora do sinal eletromiográfico, sendo assim, na próxima etapa, o sinal que será encaminhado ao processador digital de sinais virá destes componentes.

Continuando mais à direita, encontra-se a penúltima caixa representativa, que se destaca pela cor branca e circunda o LED que sinaliza que a placa está alimentada e operando. Já na última caixa, a qual é da cor roxa, exibe os fios de entrada de alimentação, desta forma a bateria que irá fornecer energia ao sistema deverá ser conectada neste ponto do circuito.

### **3.4 Etapa 3 – Processamento**

Pontualmente, essa etapa de desenvolvimento recebeu mais atenção que as demais ao longo dos anos que este projeto foi submetido a bancas avaliadoras de projeto de iniciação científica (PIC e PIBIC). Sendo assim, a arquitetura desta placa passou de microcontroladores de 8 bits (PIC12F675) ao então DSP de 16 bits utilizado (dsPIC30F4013), logo diversas técnicas de controle e aquisição do sinal mioelétrico foram utilizadas. Em capítulo anterior deste trabalho, alguns artigos correlatos foram citados, e ao ler alguns deles pode-se ver que o autor

deste projeto usou desde a leitura do conversor de sinais analógicos para digitais (ADC) com amostragem simples à médias, valores de sinais RMS, entre diversas outras técnicas.

O intuito de realizar a transição do MCU básico para um DSP é poder fazer os testes saindo do cenário onde trata os dados do sinal como um todo, e entrar na possibilidade de usar algum modelo matemático que o fragmente em todas as suas componentes de frequência, desta forma o novo microcontrolador seria capaz de trabalhar exclusivamente em cima do sinal proveniente do músculo.

Para a execução desta etapa, foi necessário desenvolver paralelamente o *hardware* que seria capaz de entregar todas as conexões elétricas e o *firmware* respectivo que pudesse analisar todas as informações tomar o devido fim a cada uma. A fim de tonar mais sucinto o descritivo do desenvolvimento, as duas partes já citadas serão explicadas separadamente.

### **3.4.1 Hardware**

Para que tudo funcione, se faz necessário a compatibilização das tensões de operação. Como o sistema tende a portabilidade, conseqüentemente o mesmo dependerá de baterias.

Em um primeiro momento precisa-se de uma chave (CH1) capaz de ligar e desligar o sistema, bem como um conector (J1) para a fácil substituição da bateria. Em seguida o sistema, devido a sensibilidade de alguns componentes, demanda uma etapa de proteção contra alimentação invertida, desta forma dimensionou-se dois diodos (D1 e D2) diretamente polarizados em paralelo para que a fonte seja capaz de entregar até 2ª sem correr risco de danificar o circuito por erros de operação. O próximo passo destina-se à implementação de filtros que eliminem pequenos ruídos, os mesmos são, em geral, aconselhados pelos fabricantes dos reguladores de tensão (bloco que será explicado mais à frente), desta forma os valores são destes componentes dispensam cálculos. O filtro citado, foi feito usando capacitores (C1, C2 e C3) de valores variados e decadenciais. A descrição pode ser vista na Figura 56 abaixo:

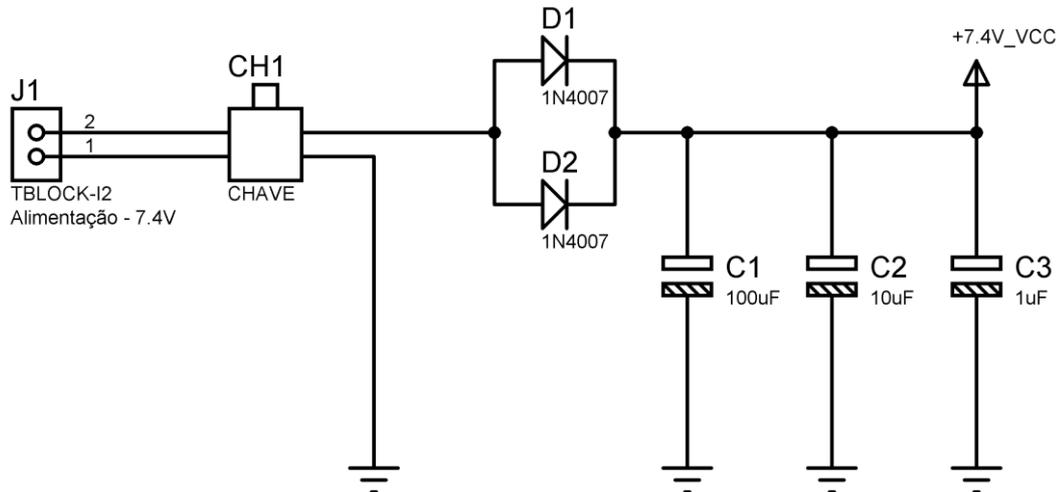


Figura 56: Entrada da alimentação da placa principal (Imagem elaborada pelo autor)

Para os testes, foi usado uma tensão nominal de entrada igual à 7.4V, porém alguns componentes, como o DSP e o servo, trabalham com no máximo 5V. Posta essa questão, buscou-se por reguladores de tensão que fossem capazes de atingir este requisito. O LM7805 atende bem a demanda e fornece até 1A, logo, com o intuito de diminuir os ruídos sobre o DSP, utilizou-se dois deste componente, um para o DSP e outro para o servo.

A parte responsável por estabilizar a alimentação do dsPIC30F4013, bem como demais componentes menores, é formada por um LM7805 (U2) que recebe o valor de tensão acima do aceitável e a estabiliza, de forma linear, em 5V. É válido citar que, devido a característica linear do LM7805, quanto maior for a diferença entre as tensões sobre ele e maior a corrente demandada, maior será o aquecimento do mesmo, desta forma usou-se dissipador de calor quando mostrou-se necessário. Em seguida posicionou-se dois capacitores (C4 e C5) de baixo valor para que filtrem ruídos provenientes da regulação da tensão, um LED (D3) indicativo de funcionamento e seu respectivo resistor (R1) limitador de corrente. A Figura 57 abaixo demonstra o que foi descrito:

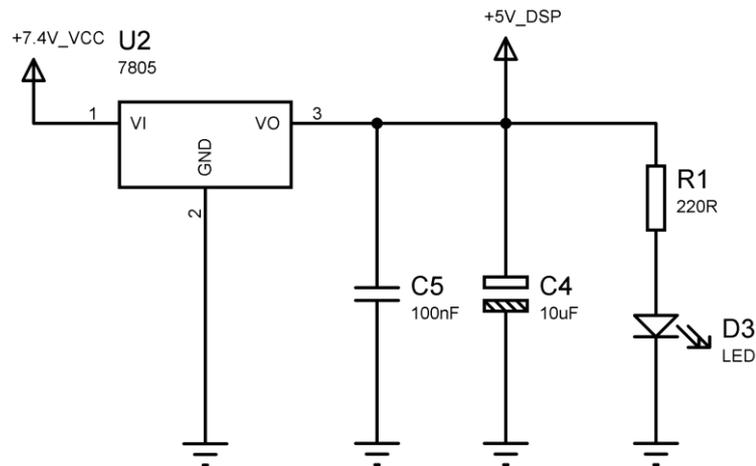


Figura 57: Regulador de tensão do DSP (Imagem elaborada pelo autor)

Até o presente momento o circuito está apto a alimentar boa parte dos componentes integrantes do sistema, exceto o servo. Durante o desenvolvimento observou-se que o servo poderia gerar picos de consumo de corrente e, ao operar junto com o restante do circuito, causava erros na leitura. Com o propósito de findar este problema desenvolveu-se uma fonte independente para o mesmo.

A fonte em questão assemelha-se ao circuito que alimenta o DSP, logo contém um LM7805 (U3), dois capacitores (C6 e C7) como antes, porém com a adição de um capacitor (C8) de alto valor para que suporte os picos de corrente demandados pelo motor ao iniciar o movimento e, enfim, um LED (D4) indicativo e seu respectivo resistor (R2). A estrutura proposta está representada na Figura 58 abaixo:

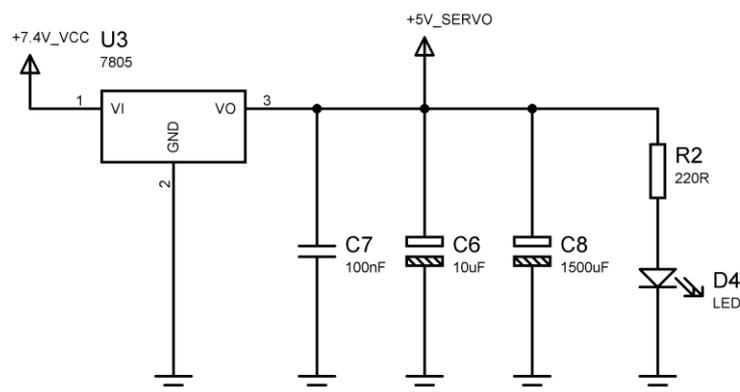
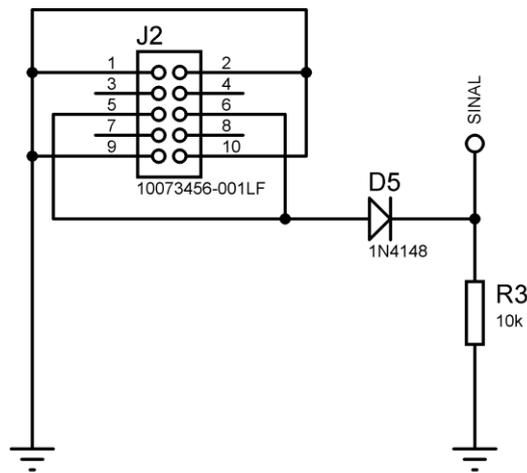


Figura 58: Regulador de tensão do servo (Imagem elaborada pelo autor)

Em seguida, deve-se tratar o sinal mioelétrico vindo do circuito condicionador pelo respectivo conector (J2). Esta parte não requer grandes esforços, pois boa parte do tratamento que o sinal demandava foi realizada junto aos filtros e amplificadores operacionais pertencentes a placa condensadora. Como já foi citado, o sinal é simétrico, sendo assim contém componente positiva e negativa. O DSP trabalha com sinais entre 0V e 5V, logo a entrada de um sinal negativo poderia danificar o processador digital de sinais. Para resolver isso optou-se pelo uso de um diodo (D5) diretamente polarizado, assim somente a parte positiva do sinal seria transferida ao microcontrolador.

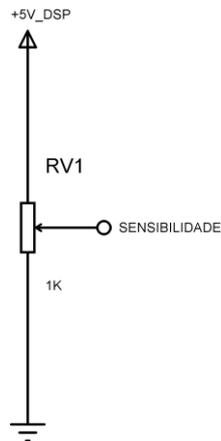
Durante a fase de construção do circuito, notou-se que ao desprezar o sinal negativo não houve grandes impactos no tratamento dos dados, exceto por certas flutuações que foram ocasionadas por falta de aterramento. Para solucionar o último problema citado, usou-se um resistor (R3) que liga o terminal de saída do diodo ao GND. O arranjo dos componentes pode ser visto na Figura 59 abaixo:



*Figura 59: Entrada do sinal mioelétrico (Imagem elaborada pelo autor)*

Operando em conjunto com o sinal lido pelo DSP, foi necessário incluir um ajuste de sensibilidade. Esta ferramenta mostrou-se de grande valia quando se levou em consideração que variáveis externas podem influenciar na decisão do processador digital de sinais para abrir ou fechar a garra mecânica. Logo, observou-se que a tonalidade da pele, oleosidade, qualidade dos sensores de superfície bem como estrutura muscular (que varia entre os indivíduos) geram limites máximos e mínimos distintos. Assim, desenvolveu-se um circuito

simples que consiste de um resistor variável (RV1) tendo como entrada o sinal de 5V proveniente do regulador de tensão do DSP e o GND. Assim, ao ser variado, o terminal central gera um sinal entre 0V e 5V, o qual será lido pelo ADC do DSP e servirá de referência para decidir qual estado a prótese irá assumir para o sinal mioelétrico instantâneo. O circuito está representado na Figura 60 abaixo:



*Figura 60: Ajuste de sensibilidade (Imagem elaborada pelo autor)*

Como trata-se de um circuito digital, é normal deduzir que o mesmo precisa de uma fonte de *clock*, sendo assim é necessário preparar uma parte do *hardware* que seja capaz de produzir o sinal base para a operação do DSP. O circuito é constituído por um cristal de quartzo (X1) que terá seus dois terminais direcionados ao processador, porém, por mais que componente seja posicionado próximo ao DSP, pequenos ruídos podem surgir nesta etapa. O fabricante do cristal sugere faixas de capacitores de cerâmica que podem ser usados para diminuir este efeito e tornar a periodicidade do sinal o mais precisa possível. Sendo assim, seguindo orientações presentes no manual do componente, optou-se por usar capacitores (C9 e C10), um em cada terminal do cristal e ligar a outro terminal de cada um ao GND. O circuito pode ser visto na Figura 61 abaixo:

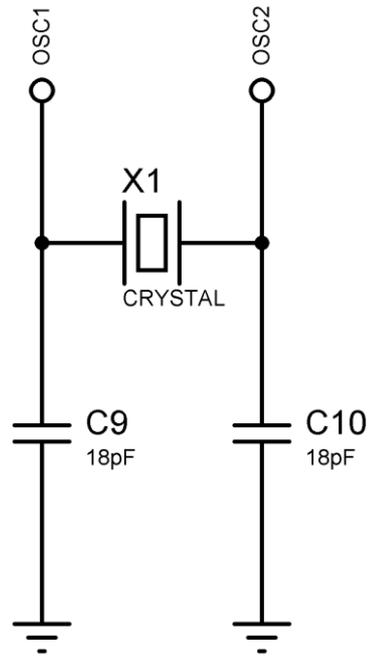


Figura 61: Cristal oscilador de quartzo (Imagem elaborada pelo autor)

Em seguida, foi necessário preparar a parte do circuito responsável pela conexão com o servomotor responsável por manipular a garra eletrônica. Para isso, usou-se apenas um conector (J3) de três terminais e, com a finalidade de facilitar os testes, adicionou-se um LED (D6) indicativo do estado do servo com seu respectivo resistor (R4) limitador de corrente. A Figura 62 abaixo mostra o circuito descrito:

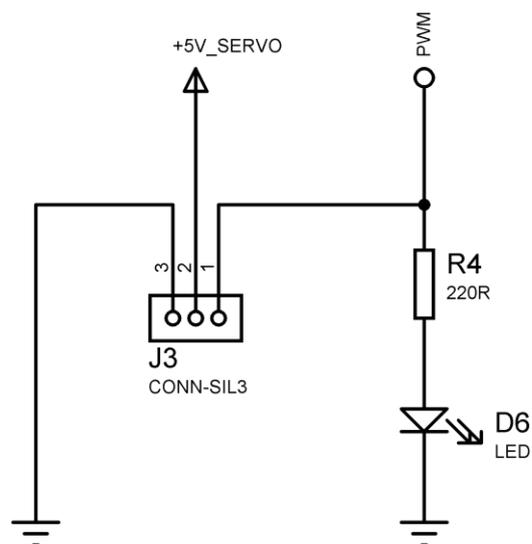
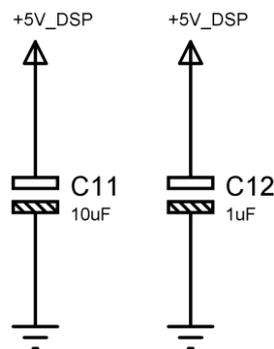


Figura 62: Conector do servomotor (Imagem elaborada pelo autor)

Ao decorrer dos testes, observou-se que em momentos alguns ruídos externos poderiam interferir na operação do DSP. O fabricante do componente sugere, no manual do mesmo, que para resolver este problema pode-se adicionar capacitores de valores tabelados próximos ao processador digital de sinais. Logo, posicionou-se os respectivos capacitores (C11 e C12) nas entradas de alimentação do dsPIC. Abaixo, a Figura 63 demonstra o esquema de ligação:



*Figura 63: Filtros da alimentação do DSP (Imagem elaborada pelo autor)*

A fim de atender a principal proposta deste trabalho em relação aos artigos anteriormente publicados, tem-se a etapa de adição do circuito responsável por transmitir os dados de forma sem fio. Para isso está sendo usado o módulo transmissor da *Texas Instrumets* HC-12, pois o mesmo não requer configuração prévia e funciona de forma *Plug & Play*, ou seja, basta conectá-lo ao circuito em enviar os dados de forma serial. Sua simplicidade mostrou-se atrativo para solucionar um problema, até então, complexo. Logo, o mesmo necessita de um conector (J4) de cinco terminais, onde o primeiro terminal (SET) habilita o modo de transmissão de dados, o segundo (TX) responsável por transmitir os dados recebidos será ligado no RX no dsPIC, o terceiro (RX) desempenhando a função de receber os dados vindos do TX do DSP, o quarto será ligado ao GND e, por último, o quinto vai à alimentação positiva. Em conjunto com o módulo, adicionou-se dois resistores (R5 e R6) conectados ao 5V do DSP e a outra ponta no RX e TX do transmissor, essa estratégia foi usada para evitar flutuações dos terminais e eliminar possíveis dados falsos. Na entrada de alimentação do módulo adicionou-se um diodo (D7) a fim de diminuir o aquecimento do mesmo sobre o regulador de tensão presente na placa do transmissor. O circuito pode ser visto na Figura 64 abaixo:

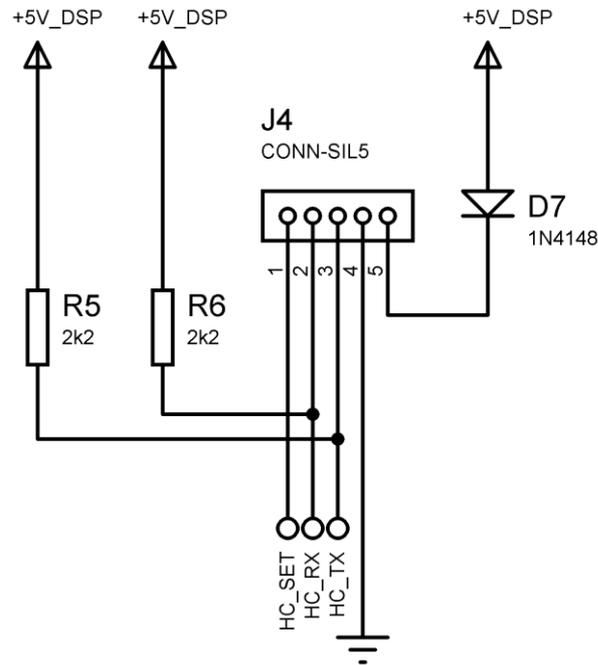


Figura 64: Circuito de suporte ao HC-12 (Imagem elaborada pelo autor)

Por fim, o *hardware* inclui em sua última parte o DSP. Sendo assim, o mesmo irá receber todas as conexões mostradas nas imagens referentes a este tópico e as encaminhará à sua central de processamento, a partir deste ponto todo o restante do trabalho será realizado a nível de *firmware*. Assim, o processador digital de sinais (U1) tem seus terminais de alimentação positiva (VDD – 11, 21 e 32) ligados aos 5V do DSP, os pinos referentes à referência (GND – 12, 20 e 31) ligados ao GND do circuito. Os pinos AVDD (40) e AVSS (39) são as referências de valores máximos e mínimos do ADC e serão ligados, respectivamente, ao 5V do DSP e o GND.

Para que fosse realizada a fácil gravação do microcontrolador, bem como as alterações do código na fase de testes, adicionou-se um conector (J5) com a função de ICSP (*In Circuit Serial Programming – programação serial no circuito*) a fim de poder reprogramar o DSP sem ter que tirá-lo da placa, facilitando o processo de desenvolvimento e testes, assim os pinos RB6 (8) e RB7 (9) serão ligados ao conector, como é informado no manual do componente. Como o ICSP faz uso do terminal MCLR (*Master Clear – pode ser traduzido como o reset do circuito integrado*) é necessário adicionar um resistor (R7) entre esse pino e o 5V do DSP a fim de manter o circuito operando quando não está sendo gravado.

Em seguida, deve-se conectar o DSP aos componentes e dispositivos adjacentes a fim de fazer com que todos funcionem em conjunto. De início, deve-se ligar os pinos OSC1 (13) e OSC2 (14) ao cristal oscilador para que forneça o *clock* ao dsPIC, em seguida conectou-se o pino que sai da etapa de recebimento do sinal mioelétrico à porta analógica AN8 (10), logo conectou-se também o terminal de sensibilidade à porta analógica AN10 (37). Já a comunicação a ser feita com o HC-12 requer terminais específicos também, pois os mesmos devem estar configurados para manter a comunicação serial, sendo assim o RX do HC-12 foi conectado no pino dos DSP U1TX (25) e o TX do transmissor no U1RX (26) do dsPIC.

Diferentemente dos demais pinos, as próximas conexões não fazer obrigatório alguma função específica do processador digital de sinais, sendo assim, basta que a porta do DSP tenha a função de chaveamento digital para que funcionem.

Desta forma, a onda PWM gerada pelo dsPIC que irá comandar a posição do eixo do servomotor, a qual atende os requisitos da Figura 33, será entregue pelo terminal RD2 (22). Já a função SET requerida pelo HC-12 apenas trabalha com valores binários, sendo 1 equivalente à 5V e 0 à 0V, e para transmitir os dados basta configurar a respectiva porta digital em nível alto.

A descrição deste bloco responsável por fazer a interligação dos pertencentes à placa principal de controle podem ser vistos no diagrama elétrico representado pela Figura 65 abaixo:



superior direito e um pouco abaixo pode-se ver dois blocos na cor preta, ambos mostram o LED indicativo e o conector do servo. Os dois blocos da cor marrom, mostram os capacitores que fazem a filtragem fina de possíveis ruídos que se encaminhem ao DSP, já no canto direito próximo à borda, o retângulo cinza demonstra o conector do transmissor sem fio HC-12, bem como os demais componentes necessários para o seu funcionamento. Os dois blocos restante, ambos da cor branca, destacam o DSP dsPIC30F4013 e o conector ICSP.

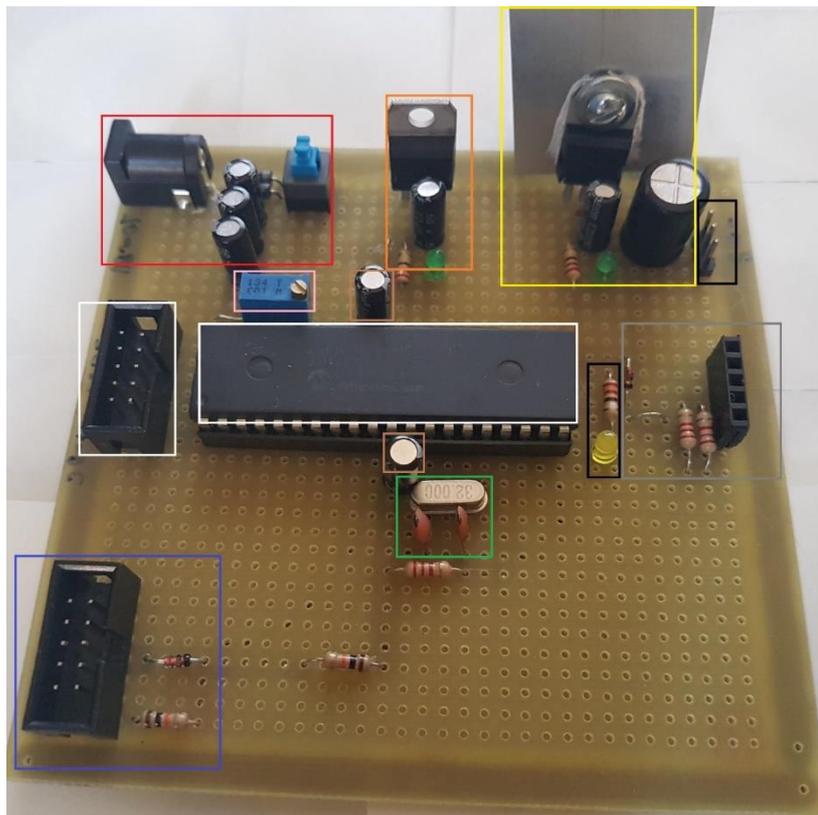


Figura 66: Placa principal de controle (Imagem elaborada pelo autor)

### 3.4.2 Firmware

Tendo o *hardware* pronto, falta apenas integrar as rotinas ao controlador para que ele interaja com os periféricos e execute as ações desejadas, sendo assim, aqui será tratado o desenvolvimento do código fonte.

A fim de manter a boa organização do código, o mesmo é dividido em blocos. Sendo assim, a explicação de como o *firmware* foi feito irá seguir a ordem de como os dados estão dispostos no arquivo. De início, tem-se a seção responsável pelas definições do código, neste

cenário foi necessário apenas definir o valor correspondente ao tempo necessário para fazer a atualização do estado da garra mecânica, bem como atualizar o gráfico. Esta área pode ser vista na Figura 67 abaixo, já a justificativa do valor definido a ela será explicada mais a frente:

```
//-----//
//=====//
//=====//
//=====//
//-----//
// Definições
#define Atualizar 7143
```

*Figura 67: Definições do código (Imagem elaborada pelo autor)*

O próximo passo, foi informar ao controlador em quais pinos estão ligados os periféricos. Como a maior parte dos dispositivos conectados ao DSP usam portas específicas, apenas dois terminais precisaram ser definidos, os quais são o servo e o pino SET do HC-12. Desta forma, a especificação das portas digitais pode ser vista na Figura 68 abaixo:

```
//-----//
//=====//
//=====//
//=====//
//-----//
// Pinagem
sbit _Servo at RD2_bit;
sbit _Servo_Dir at TRISD2_bit;
//-----//
sbit _HC_SET at RD3_bit;
sbit _HC_SET_Dir at TRISD3_bit;
```

*Figura 68: Definição da pinagem do DSP (Imagem elaborada pelo autor)*

Em seguida, deve-se definir as variáveis. Estas exercem a função de manter os dados temporários do processamento na RAM, logo foram usados três tamanhos de dados diferentes e quatro tipos de variáveis distintos. A variável *Servo\_Posição* responsável por guardar a posição em que o servomotor se encontra (garra aberta ou fechada) apenas receberá os valores 0 ou 1, sendo 0 aberto e 1 fechado, logo a mesma pode ser dimensionada com o menor tamanho de variável, que neste caso é a *char* que tem um *byte* de tamanho (assume valores de 0 a 255).

Para que o servo funcione, é necessário um PWM com frequência de 50Hz (período de 20 ms), para isso foi necessário fazer uma rotina que faça esse controle. Para que o sistema saiba qual valor digital deverá ser imprimido da porta de controle do servo, será necessário um contador, o qual é a variável `_Servo_I` que, como pode receber valores maiores que 255, será definida como `int`, assim terá um tamanho de dois `bytes` (assume valores de 0 a 65.535). Da mesma forma que o servo precisa de um contador para configurar o PWM do mesmo, será necessária uma variável responsável por fazer a contagem do tempo para que se envie uma nova informação ao gráfico ou então execute uma nova ação na prótese, essa variável foi denominada de `Enviar` e como o tempo poderia sofrer alterações, definiu-se o maior valor possível à ela, que seria `long`, ou seja, 4 `bytes` (assume valores de 0 a 4.294.967.295). Por último, declarou-se o vetor que receberá os dados para que se execute os cálculos sobre o sinal mioelétrico. Como espera-se fazer testes com a FFT, usou-se a declaração do tipo `ydata` que diz ao DSP que esse dado deve ser alocado em um espaço da RAM próprio para o processamento digital de sinais. Sendo assim, a variável `Amostras` recebeu a definição do tipo `unsigned (2 bytes)` e um tamanho total de 512 posições, sendo assim totaliza um tamanho na memória de 1024 `bytes`. É válido citar que este tamanho permite tratar transformadas de até 256 pontos, pois para cada número real coletado será necessário um complementar imaginário. A descrição dada nos últimos dois parágrafos pode ser visualizada na Figura 69 que se refere ao trecho de código em questão:

```
//-----//
//=====//
//=====//
//=====//
//-----//
// Variáveis
char Servo_Posicao = 0;
unsigned int _Servo_I = 0;
unsigned long Enviar = 0;

ydata unsigned Amostras[512];
```

Figura 69: Declaração das variáveis (Imagem elaborada pelo autor)

O próximo trecho do código inicia parte do algoritmo do sistema. Nele, há as funções `InitTimer1()` e `Timer1Interrupt()`. A primeira é a responsável por configurar o Timer 1 do DSP, sendo assim, o contador é definido para ter um disparo (interrupção) a cada 14 microssegundos. Já a segunda função trata o controle os dados sempre que ocorre uma

interrupção (a cada 14 microssegundos). Como a principal função do Timer 1 é gerar o sinal de PWM para o servo, esse valor foi levando em consideração as informações da Figura 33, logo para se obter o período de 20 milissegundos a variável `_Servo_I` deverá obter um valor máximo de 1429 ( $1429 \times 14\mu s = 20.006\text{ms}$ ). Na mesma rotina, a porta digital responsável por controlar o servo será comutada de forma a gerar o sinal correspondente à posição que a garra mecânica deva assumir. Quando for para fechar (Servo Posição igual à 1) o pulso gerado irá fazer o servo se posicionar à uma angulação de  $160^\circ$ , mas quando for para abrir a mão mecânica (Servo Posição igual à 0) o eixo do motor será manobrado até atingir um valor de  $90^\circ$ .

Ainda na mesma rotina, pode-se observar que é feita a contagem que se responsabiliza por enviar as informações do sinal lido pelo processador digital de sinais, bem como informar ao controlador quando deve atualizar a posição da prótese. Como o valor anteriormente definido foi de 7143, o DSP irá atualizar as informações a cada 100 milissegundo ( $7143 \times 14\mu s = 100.002\text{ms}$ ). Todo esse descritivo pode ser visto na Figura 70 logo abaixo:

```
//-----//
//-----//
//-----//
//-----//
//-----//
// Timer

void Timer1Interrupt() iv IVT_ADDR_T1INTERRUPT{ // 14us
    T1IF_bit = 0;

    _Servo_I++;
    if (_Servo_I > 1429) _Servo_I = 0; // Valor para 20ms

    if (Servo_Posicao == 1){
        if (_Servo_I < 120) _Servo = 1;
        else _Servo = 0;
    }
    else if (Servo_Posicao == 0){
        if (_Servo_I < 60) _Servo = 1;
        else _Servo = 0;
    }
}

if (Enviar < Atualizar) Enviar++;
}

//-----//

void InitTimer1(){
    T1CON = 0x8000;
    T1IE_bit = 1;
    T1IF_bit = 0;
    IPC0 = IPC0 | 0x1000;
    PR1 = 448;
}
}
```

Figura 70: Tratamento e configuração do Timer 1 (Imagem elaborada pelo autor)

A próxima rotina, é deveras simples, porém necessária para o funcionamento do sistema. A mesma tem como função definir o estado da porta digital responsável por operar o servo, logo ela apenas define o sentido do terminal como saída e atribui o valor lógico inicial como 0. Essa configuração pode ser vista na Figura 71 abaixo:

```
//-----//
//-----//
//-----//
//-----//
//-----//
// Servo

void Servo_Init(){
    _Servo_Dir = 0;
    _Servo = 0;
}
```

Figura 71: Inicialização do servomotor (Imagem elaborada pelo autor)

Na sequência, inicia-se a rotina principal do sistema, a qual se denomina *void main()*. De início, são declaradas as variáveis locais, ou seja, só podem ser acessadas de dentro desta rotina. A primeira declaração é um vetor denominado *TXT[]* que receberá os dados a serem transmitidos de forma sem fio, o tratamento desta variável será melhor explicada mais à frente. Em seguida, a variável *Val* é definida para que mais à frente venha receber o valor do módulo do sinal da FFT e depois a variável *I* que será usada como contador para a coleta das amostras do sinal. As próximas variáveis, *Real*, *Imaginario*, *RealF*, *ImaginarioF* e *TempF*, são todas de uso exclusivo da FFT, sendo assim serão usadas para receber os valores reais, imaginários e o módulo da frequência desejada de acordo com a manipulação previamente realizada.

Após a declaração das variáveis, vem a configuração do registrador interno do dsPIC30F4013 denominado *ADPCFG*, o mesmo serve para dizer ao controlador quais portas devem ser usadas como entrada analógica ou entrada e saída digital. No caso em questão, as portas 8 e 10 foram definidas como analógicas, as demais receberam função digital.

As próximas duas linhas fazem a configuração do pino SET do transmissor sem fio HC-12. Para tanto, define-se a porta como sendo saída de sinal digital e, posteriormente, a atribui o nível lógico 1, o que neste caso significa 5V na saída no pino conectado ao terminal SET do módulo.

Nos próximos passos inicializou-se as demais rotinas, sendo assim seguiu-se iniciando a comunicação serial com a taxa de transmissão de 9600 bits por segundo e em seguida definindo um tempo de espera de 100 milissegundos para o oscilador do módulo serial possa

estabilizar. Abaixo, chamou-se as rotinas *InitTimer1()* e *Servo\_Init()* que já foram explicadas neste capítulo.

O descrito neste tópico pode ser visualizado na Figura 72 abaixo:

```
//-----//
//=====//
//=====//
//=====//
//-----//

// Void Main

void main() {
char TXT[] = "A000";
unsigned long Val = 0;
unsigned int I = 0;
unsigned int Real, Imaginario;
float RealF, ImaginarioF, TempR;

ADPCFG = 0xFDF;

_HC_SET_Dir = 0;
_HC_SET = 1;

UART1_Init(9600);
Delay_ms(100);

InitTimer1();
Servo_Init();
}
```

*Figura 72: Rotinha de inicialização do sistema (Imagem elaborada pelo autor)*

Na sequência estruturou-se a rotina responsável por coletar as amostras a serem trabalhadas. Como posteriormente a FFT será de 256 pontos, o laço de repetição contará de 0 a 512, porém incrementando a variável contadora *I* de duas em duas unidades. Sendo assim, para cada ciclo a posição de valor par (componente real) da amostra irá receber o valor instantâneo lido pelo canal 8 do ADC e a amostra ímpar (componente imaginário) será igual a 0, como manda a rotina da FFT posteriormente utilizada. Após coletar a amostra, será dado o comando ao DSP que espera 781 microssegundos, sendo assim serão coletadas 256 amostras em um intervalo de aproximadamente 200 milissegundos. A estrutura desta rotina pode ser visualizada na Figura 73 abaixo:

```
for (I = 0; I < 512; I += 2){  
    Amostras[I] = ADC1_Read(8);  
    Amostras[I + 1] = 0;  
  
    Delay_us(781);  
}
```

Figura 73: Coleta das amostras do sinal (Imagem elaborada pelo autor)

Tendo as amostras a serem trabalhadas, a próxima fase do código é realizar a FFT. Como não é a intenção deste trabalho descrever o funcionamento desta transformada e desenvolver o próprio algoritmo, optou-se por usar a biblioteca interna disponibilizada pelo compilador MikroC Pro for dsPIC, logo após ter coletado o sinal e, seguindo o procedimento descrito no manual do compilador, as mesmas foram submetidas ao modelo matemático.

Para que se use a rotina interna da FFT do compilador, deve-se definir três valores. O primeiro é a potência de dois que corresponde ao número de amostras que estão sendo coletadas, como neste caso são 256, o valor desta entrada será igual à 8 ( $2^8 = 256$ ), o próximo termo é chamado de *TwiddleCoeff\_256*, que é, basicamente, uma tabela de valores constantes que representam a raiz quadrada da componente imaginária para determinados valores, esta tabela pode ser vista na Figura 75. Já o terceiro e último campo recebe o vetor que contém as amostras coletadas. A estrutura final da rotina pode ser vista na Figura 74 abaixo:

```
FFT(8, TwiddleCoeff_256, Amostras);
```

Figura 74: Execução da FFT (Imagem elaborada pelo autor)

```

const unsigned TwiddleCoeff_256[256] = {
0x7FFF, 0x0000, 0x7FF6, 0xFCDC, 0x7FD9, 0xF9B8, 0x7FA7, 0xF695,
0x7F62, 0xF374, 0x7FOA, 0xF055, 0x7E9D, 0xED38, 0x7E1E, 0xEA1E,
0x7DBA, 0xE707, 0x7CE4, 0xE3F4, 0x7C2A, 0xE0E6, 0x7B5D, 0xDDDC,
0x7A7D, 0xDAD8, 0x798A, 0xD7D9, 0x7885, 0xD4E1, 0x776C, 0xD1EF,
0x7642, 0xCF04, 0x7505, 0xCC21, 0x73B6, 0xC946, 0x7255, 0xC673,
0x70E3, 0xC3A9, 0x6F5F, 0xC0E9, 0x6DCA, 0xBE32, 0x6C24, 0xBB85,
0x6A6E, 0xB8E3, 0x68A7, 0xB64C, 0x66D0, 0xB3C0, 0x64E9, 0xB140,
0x62F2, 0xAECC, 0x60EC, 0xAC65, 0x5ED7, 0xAA0A, 0x5CB4, 0xA7BD,
0x5A82, 0xA57E, 0x5843, 0xA34C, 0x55F6, 0xA129, 0x539B, 0x9F14,
0x5134, 0x9D0E, 0x4EC0, 0x9B17, 0x4C40, 0x9930, 0x49B4, 0x9759,
0x471D, 0x9592, 0x447B, 0x93DC, 0x41CE, 0x9236, 0x3F17, 0x90A1,
0x3C57, 0x8F1D, 0x398D, 0x8DAB, 0x36BA, 0x8C4A, 0x33DF, 0x8AFB,
0x30FC, 0x89BE, 0x2E11, 0x8894, 0x2B1F, 0x877B, 0x2827, 0x8676,
0x2528, 0x8583, 0x2224, 0x84A3, 0x1F1A, 0x83D6, 0x1C0C, 0x831C,
0x18F9, 0x8276, 0x15E2, 0x81E2, 0x12C8, 0x8163, 0x0FAB, 0x80F6,
0x0C8C, 0x809E, 0x096B, 0x8059, 0x0648, 0x8027, 0x0324, 0x800A,
0x0000, 0x8000, 0xFCDC, 0x800A, 0xF9B8, 0x8027, 0xF695, 0x8059,
0xF374, 0x809E, 0xF055, 0x80F6, 0xED38, 0x8163, 0xEA1E, 0x81E2,
0xE707, 0x8276, 0xE3F4, 0x831C, 0xE0E6, 0x83D6, 0xDDDC, 0x84A3,
0xDAD8, 0x8583, 0xD7D9, 0x8676, 0xD4E1, 0x877B, 0xD1EF, 0x8894,
0xCF04, 0x89BE, 0xCC21, 0x8AFB, 0xC946, 0x8C4A, 0xC673, 0x8DAB,
0xC3A9, 0x8F1D, 0xC0E9, 0x90A1, 0xBE32, 0x9236, 0xBB85, 0x93DC,
0xB8E3, 0x9592, 0xB64C, 0x9759, 0xB3C0, 0x9930, 0xB140, 0x9B17,
0xAECC, 0x9D0E, 0xAC65, 0x9F14, 0xAA0A, 0xA129, 0xA7BD, 0xA34C,
0xA57E, 0xA57E, 0xA34C, 0xA7BD, 0xA129, 0xAA0A, 0x9F14, 0xAC65,
0x9D0E, 0xAECC, 0x9B17, 0xB140, 0x9930, 0xB3C0, 0x9759, 0xB64C,
0x9592, 0xB8E3, 0x93DC, 0xBB85, 0x9236, 0xBE32, 0x90A1, 0xC0E9,
0x8F1D, 0xC3A9, 0x8DAB, 0xC673, 0x8C4A, 0xC946, 0x8AFB, 0xCC21,
0x89BE, 0xCF04, 0x8894, 0xD1EF, 0x877B, 0xD4E1, 0x8676, 0xD7D9,
0x8583, 0xDAD8, 0x84A3, 0xDDDC, 0x83D6, 0xE0E6, 0x831C, 0xE3F4,
0x8276, 0xE707, 0x81E2, 0xEA1E, 0x8163, 0xED38, 0x80F6, 0xF055,
0x809E, 0xF374, 0x8059, 0xF695, 0x8027, 0xF9B8, 0x800A, 0xFCDC};

```

Figura 75: Coeficientes tabelados necessários para a FFT (Imagem elaborada pelo autor)

Após ter feito a FFT, deve-se tratar os valores retornado para que se obtenha o módulo da frequência desejada, neste caso 160Hz. Para isso deve-se realizar um simples conjunto de operações matemáticas, começando por elevar ao quadrado individualmente a componente real e imaginária, após isso, deve-se somá-las e, por fim, obter o valor da raiz quadrada da soma. Feito isso, é possível obter a amplitude do sinal referente a frequência de estudo. O procedimento descrito pode ser visualizado na Figura 76 abaixo:

```

TempR = RealF * RealF;
RealF = TempR;
TempR = ImaginarioF * ImaginarioF;
ImaginarioF = TempR;

TempR = sqrt(RealF + ImaginarioF);

```

Figura 76: Obtenção do módulo do sinal da FFT (Imagem elaborada pelo autor)

Por fim, o último bloco do código fonte faz o controle da prótese mecânica e o envio das informações de forma sem fio. No início deste bloco, tem-se o condicionamento de só executar as instruções a seguir se o valor da variável *Enviar*, for maior ou igual ao da definição *Atualizar*, ou seja, foi dado o comando pelo *Timer 1* para atualizar as informações.

Em seguida, o valor do módulo da frequência obtido no passo anterior será comparado com o valor do canal analógico 10, o qual foi anteriormente definido como a sensibilidade do sistema. Neste momento é possível observar a estrutura do controlador ON/OFF, pois há duas condições que ponderam se deve abrir ou fechar a garras, além de ser válido citar que após a garra estar fechada o sistema apenas comandará para abrir quando o sinal atingir um valor inferior à 60% do sinal da sensibilidade, logo o *Offset* é dado sobre o valor definido à referência.

Logo abaixo o conteúdo da variável *Val* é compatibilizado através de uma divisão para que possa ser convertido em texto e só então ser enviado. A conversão de valor inteiro para texto se dá pela divisão das centenas, dezenas e unidades do valor e posteriormente somado o valor 48 para se torne uma componente da tabela ASCII que equivalha ao número em forma de texto. Por último o valor armazenado o vetor *TXT* é enviado pela porta serial e, conseqüentemente, enviado de forma sem fio para o *software* de monitoramento. O procedimento aqui descrito pode ser observado na Figura 77 abaixo:

```

if (Enviar >= Atualizar){
    if (Val > ADC1_Read(10)) Servo_Posicao = 1;
    else if (Val <= (ADC1_Read(10) * 0.6)) Servo_Posicao = 0;

    Enviar = 0;

    Val /= 10.2375;

    TXT[1] = ((Val / 100) % 10) + 48;
    TXT[2] = ((Val / 10) % 10) + 48;
    TXT[3] = ((Val / 1) % 10) + 48;
    TXT[4] = '\0';

    UART1_Write_Text(TXT);
}

```

Figura 77: Controle da prótese e envio dos dados (Imagem elaborada pelo autor)

### 3.5 Etapa 4 – Estrutura mecânica

O sistema deverá gerar duas saídas com a resultante do tratamento dos sinais mioelétricos, sendo assim uma delas será a formação do gráfico do sinal, que será melhor

tratada mais à frente, e a segunda saída, que será tratada neste tópico, será a execução do movimento na estrutura mecânica.

### 3.5.1 Primeira estrutura mecânica (Estudo aplicado)

A parte mecânica é composta de dois componentes principais. O primeiro é o servomotor, o modelo escolhido foi o Futaba S3003 que exibe um bom torque, 3.2kg.cm, além de manobrar todo o percurso de 180º em um tempo de aproximadamente um segundo. Sendo assim, o mesmo mostrou uma boa relação entre força e velocidade para que possa executar o movimento de pinça.

As conexões elétricas necessárias para operar corretamente o servo motor são três, sendo assim, o fio preto/marrom deve ser ligado ao GND do circuito, o vermelho na alimentação do servo (neste caso +5V\_SERVO) e o fio branco/amarelo irá ligado ao terminal de saída do PWM do controlador. Ao alterar a largura do pulso o eixo irá girar até a posição desejada e, conseqüentemente, abrir e fechar a garra mecânica. O servo que está sendo utilizado, pode ser visto na Figura 78 abaixo:

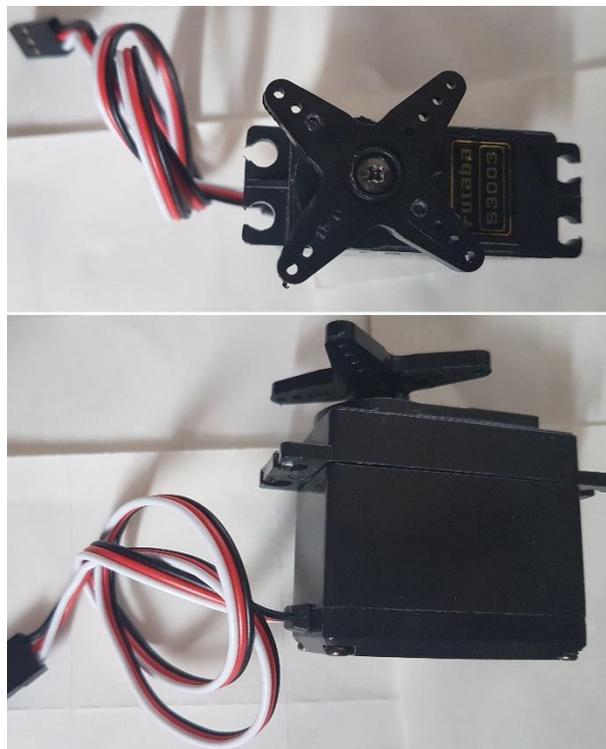


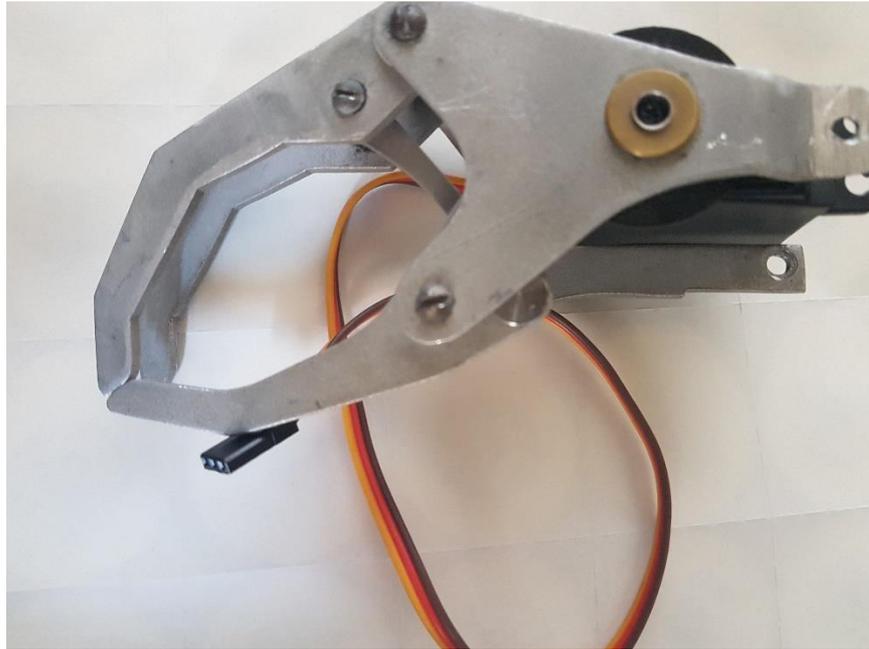
Figura 78: Servomotor Futaba S3003 (Imagem elaborada pelo autor)

A estrutura de metal, que foi aproveitada de trabalhos e artigos anteriores, apresenta boa similaridade com a estrutura óssea da mão, sendo assim, executa bem a função de pegar, segurar e manipular boa parte dos objetos corriqueiros do dia-a-dia. A mesma é capaz de assumir inúmeros ângulos de abertura, porém neste trabalho somente serão usados os pontos de abertura e fechamento, sendo assim o controle é binário e não proporcional.

Quando o servo a manobrar para abertura, ou seja, baixa amplitude da componente do sinal eletromiográfico, a mesma afastará os dedos e se estabilizará da forma como está representado na Figura 79. Por outro lado, quando amplitude da componente do sinal que está sendo analisada atingir o valor de sensibilidade configurado o servomotor irá exercer força sobre a estrutura metálica e, conseqüentemente, manipulará para que feche a garra, logo replicará o momento de pinça. A prótese assumindo a posição de fechada pode ser vista na Figura 80.



*Figura 79: Prótese montada e aberta (Imagem elaborada pelo autor)*



*Figura 80: Prótese montada e fechada (Imagem elaborada pelo autor)*

### **3.5.2 Segunda estrutura mecânica (Início do Estudo – Modelagem/impressão 3D)**

Nos primeiros meses de desenvolvimento desta etapa do projeto, foram realizados alguns estudos sobre próteses em impressões 3D. Neste período conhecemos a ONG “Mao3D”, de onde foram retirados materiais e inspirações para impressões. No entanto, para melhor compreendê-las, foram feitas diversas impressões de próteses mecânicas utilizando a impressora 3D, com o intuito de projetar um modelo que atendesse as principais necessidades de um deficiente físico.

Antes de começarmos a modelar e desenvolver nossas próprias próteses, tivemos que entender o funcionamento da impressora 3D. A impressora que foi adquirida é uma "AiP A3" da empresa "Sethi3D" com as seguintes características:

- Conexão USB
- Módulo de LCD integrado
- Ajuste automático de altura
- Entrada para cartão de memória
- Alimentação 110/220v
- Mesa Aquecida de Alumínio
- Dimensões aprox.: 40cm x 35cm x 40cm

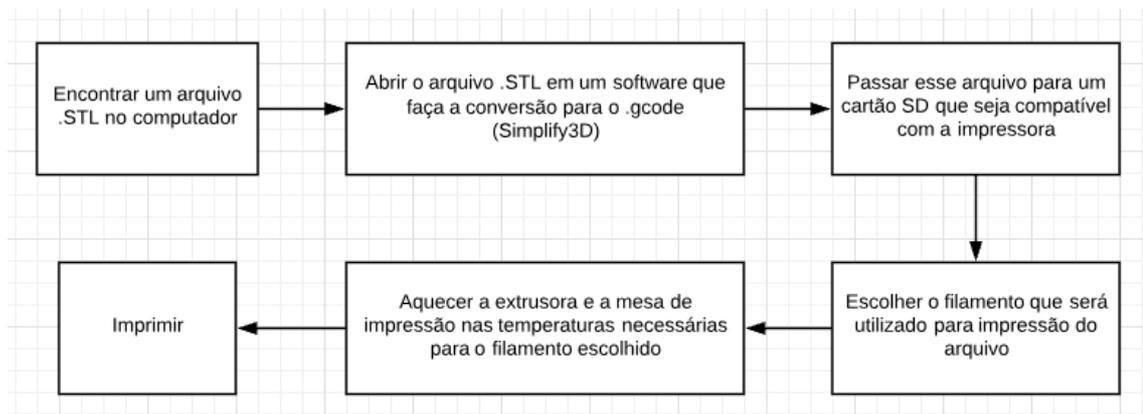
- Peso aprox.: 12Kg
- Suporte para rolo de filamento
- Materiais de impressão: ABS, PLA, Flexível, PETG.

Junto com a impressora, também foi adquirido o software "Simplify3D", que é o responsável por receber os arquivos em 3D e transformá-los em algo que a impressora possa ler.

Com tudo isso em mente, pode-se começar a descrever o processo de como é realizada uma impressão em 3D. Inicialmente precisa-se saber qual arquivo será impresso, ele precisa estar em algum formato: ".STL", ".OBJ" ou ".3MF". Esses são os únicos arquivos que o software, "Simplify3D", consegue ler, sendo o ".STL" o mais comum entre eles.

Ao encontrar, ou criar algum arquivo de formato compatível, foi aberto ele no software para fazer a conversão para um arquivo ".gcode" ou "código G". O código G é uma linguagem de programação criada nos anos 50, que foi criada com o intuito de definir os eixos e as distâncias de máquinas industriais, sendo aproveitado para o uso em Impressoras 3D.

Quando esse arquivo ".gcode" é formado, ele precisa ser compilado pela Impressora de alguma forma.



*Figura 81: Diagrama de blocos da impressão*

A primeira impressão Figura 82, trata-se de um modelo para deficientes que sofreram amputação parcial da mão. Para esta prótese, havia necessidade da existência de parte do membro, para que fosse possível gerar mecanicamente o movimento de uma mão.



*Figura 82: Imagem elaborada pelo Autor*

Percebendo a complexidade do tema amputação, foram procuradas outras formas de se imprimir e desenvolver as próteses mecânicas. Uma das maneiras encontradas foi utilizando a água quente para modelar o material escolhido para impressão, o PLA.

O Quadro 1 apresenta um novo modelo de prótese impressa para deficientes que sofreram desarticulação de punho, e ilustra o processo de modelagem do PLA com água quente. Foi solicitado ao labocien um laboratório, equipamentos de proteção individual - EPI's (Luvas e pinças), e os materiais necessários para aquecimento e medições de temperatura para monitoramento da água.

A água utilizada para moldar o PLA, foi aquecida em até 80°C, logo, foi despejada em um recipiente com área de base superior ao tamanho das peças a serem modeladas. Após a transferência de calor água/material de aproximadamente 25 segundos, foi retirada uma por vez com o auxílio de uma pinça, e utilizado moldes também impressos na impressora 3D para finalizar as peças para montagem da prótese.



*Quadro 1: Modelagem do PLA com água quente*



*Figura 83: Resultado da modelagem*

### **3.6 Etapa 5 – Recepção dos sinais sem fio**

Como anteriormente citado, a segunda saída que o sistema gera é o envio dos dados para que o computador receptor possa gerar o gráfico do sinal mioelétrico medido pelo circuito. Desta forma, será necessário desenvolver um *hardware* que possa receber esses dados, traduzi-los e encaminhá-los ao computador de destino. Como mantem-se o uso do módulo HC-12 que dispensa configuração prévia, essa nova placa dispensa o uso de algum microcontrolador externo para operá-lo.

Assim como foi feito com o circuito da placa principal, está também foi feita por blocos, sendo assim, a metodologia de desenvolvimento e construção desta será semelhante.

Logo de início, se faz necessário a construção da entrada de energia elétrica para a placa, posicionou-se um conector (J1) que será ligado à bateria, sem seguida adicionou-se uma chave (CH1) para ligar e desligar o circuito e, por último, um diodo (D1) que servirá como proteção ao restante dos componentes contra ligação invertida da fonte de alimentação. A Figura 84 abaixo demonstra o bloco aqui descrito:

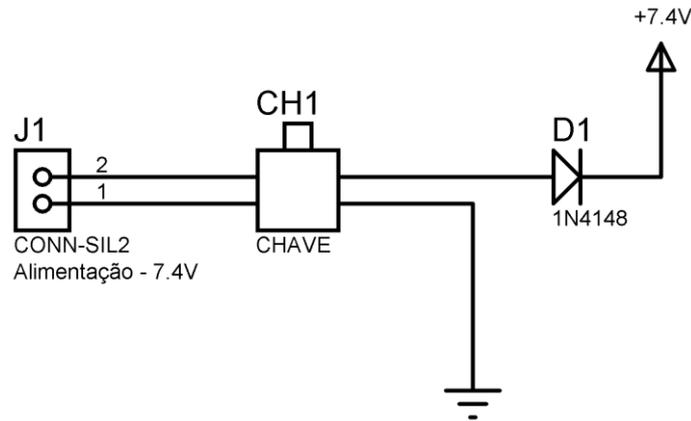


Figura 84: Entrada de alimentação da placa receptora (Imagem elaborada pelo autor)

Continuando a parte de alimentação do circuito, é necessário fazer a regulação da tensão de entrada e abaixá-la de 7.4V para 4.3V, valor esse sugerido pelo fabricante. O nível de tensão que sai do diodo de proteção (D1) será diretamente ligado ao regulador de tensão LM7805 (U1) o qual terá em sua saída um capacitor (C1) para a filtragem de ruídos decorrentes da estabilização da tensão e, em paralelo com este capacitor, será ligado um circuito de um LED (D2) e o devido resistor (R1) que servirá para indicar que a placa se encontra energizada. Ainda após o LM7805 será ligado um diodo (D3) a fim de abaixar o valor de 5V para 4.3V, assim diminuirá o aquecimento do módulo HC-12, e para ajudar a estabilizar a tensão foi ligado após o diodo um capacitor (C2). Abaixo é possível verificar este bloco do circuito na Figura 85:

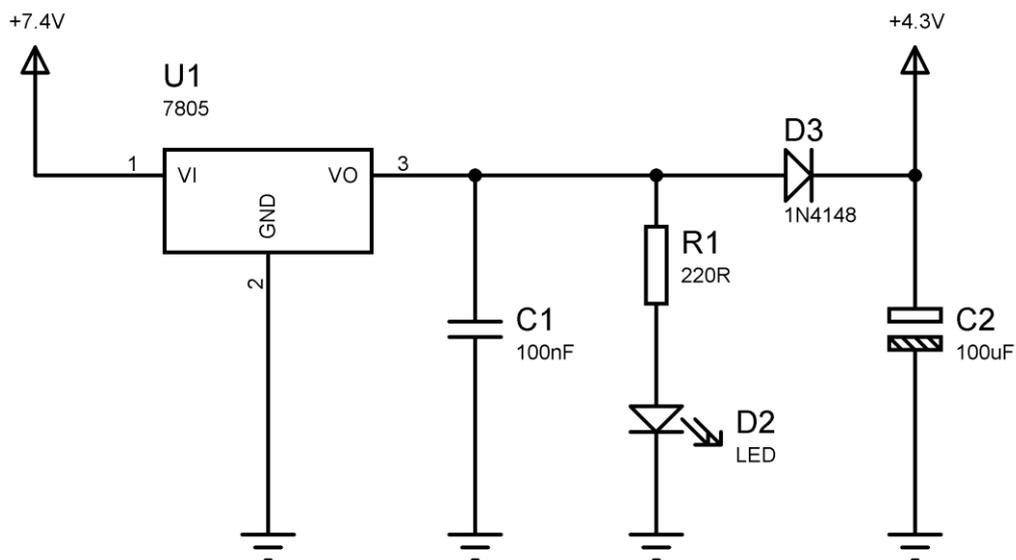
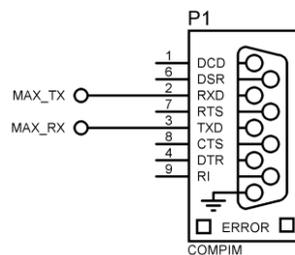


Figura 85: Regulador de tensão do receptor (Imagem elaborada pelo autor)

Para que o módulo se comunique com o computador, optou-se pelo uso da porta serial RS-232, sendo assim o protocolo usado pelo HC-12 é o mesmo, logo descarta-se a necessidade de algum circuito intermediário que iguale as linguagens utilizadas pelos dispositivos. A única barreira apresentada aqui são os níveis de tensão que o computador utiliza e o módulo transmissor, então a ligação não pode ser direta, pois necessita de um equalizador dos níveis de tensão, o qual será descrito no próximo tópico. Logo, o circuito deste bloco é apenas constituído de um conector (P1) do tipo DB9, onde o terminal dois dele será ligado no pino de saída de sinal TX e o pino três será ligado junto ao pino de entrada de dados RX, já o pino cinco será conectado ao GND do circuito da placa receptora a fim de estabelecer o mesmo nível de referência para o computador e o circuito. O circuito pode ser visualizado na Figura 86 abaixo:



*Figura 86: Conector DB9 do receptor (Imagem elaborada pelo autor)*

Como foi anteriormente citado, será necessário utilizar um circuito que crie a equivalência de tensões de níveis lógicos entre o computador e o módulo HC-12. Como se trata de um protocolo específico (RS-232), há o circuito integrado MAX232 (U2) que, seguindo o arranjo de ligação disponível em seu manual, será capaz de converter os níveis de tensão de ambas as partes e fazer com que elas comuniquem entre si. Desta forma, será necessário ligar um capacitor (C3) entre seus terminais um e três, outro capacitor (C4) nos pinos quatro e 5. Para continuar o circuito, que deve gerar valores de tensão menor que zero (negativa), será necessário adicionar outros dois capacitores (C5 e C6) os quais serão ligados, respectivamente, no terminal dois do circuito integrado e no +4.3V e o outro entre o pino seis e o GND.

A última etapa deste bloco do circuito, será fazer as conexões entre o DB9 e o HC-12, logo o pino 11 e 12 serão ligados aos pinos TX e RX do módulo transmissor, respectivamente. E, por fim, o pino 14 será conectado ao dois do DB9 e 13 ao três. O circuito pode ser visto na Figura 87 abaixo:

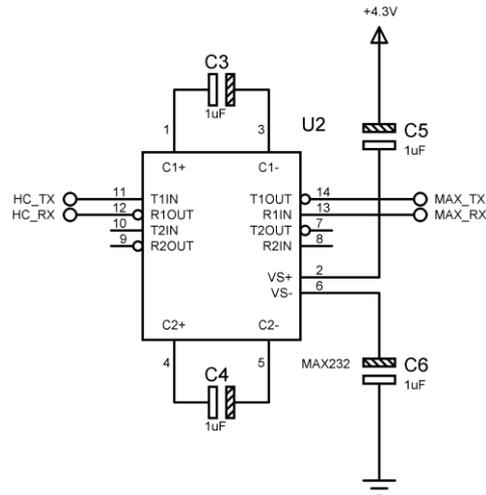


Figura 87: 84 MAX232 (Imagem elaborada pelo autor)

Complementar ao circuito anterior, foi adicionado um conjunto de dois LEDs (D4 e D5) e seus respectivos resistores (R3 e R4) aos pinos 11 e 12 do MAX232 a fim de ter um sinal visual da comunicação entre o computador e o módulo HC-12 do receptor e entre a placa receptora e a placa principal. O arranjo destes componentes pode ser visto na Figura 88 abaixo:

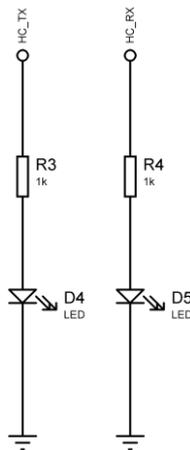
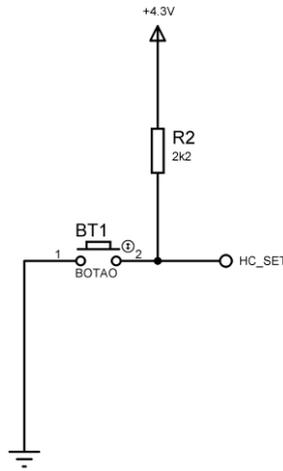


Figura 88: LEDs indicativos (Imagem elaborada pelo autor)

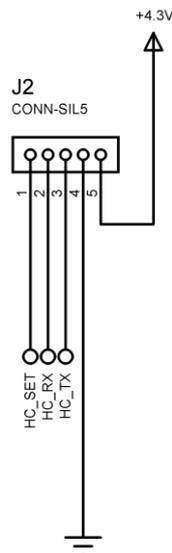
Outro circuito auxiliar, porém, fundamental para os testes, foi a adição de um arranjo com um resistor (R2) e um botão (BT1) para que o módulo transmissor HC-12 possa ser manualmente habilitado e desabilitado a enviar e receber informações da placa principal, logo

a saída do botão foi conectada ao pino SET do transmissor. O circuito pode ser visto na Figura 89 abaixo:



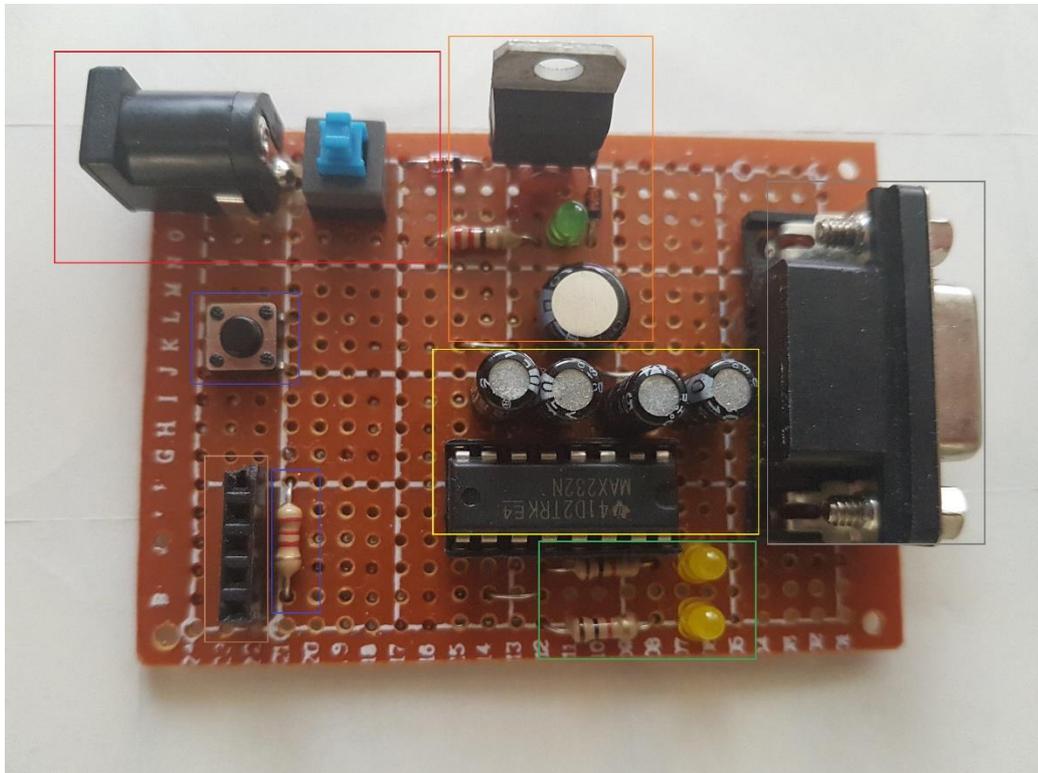
*Figura 89: Botão de controle (Imagem elaborada pelo autor)*

Por último, foi posicionado o conector para ao módulo HC-12. Como a ligação deste módulo já foi explicada em tópico anterior, ela não será detalhadamente abordada aqui. A diferença que deve ser citada é a ausência dos resistores que eliminariam sinais flutuantes, os mesmos podem ser desconsiderados neste caso, pois o circuito integrado MAX232 já os cotem em sua arquitetura interna, logo, basta ligar o HC-12 diretamente ao U2. Abaixo é possível ver este bloco do circuito na Figura 90:



*Figura 90: Conector do HC-12 do transmissor (Imagem elaborada pelo autor)*

Ao fim da montagem, foi possível obter a placa mostrada na Figura 91, onde os blocos explicados acima foram destacados nela. Sendo assim, a estrutura responsável pela entrada da alimentação está rodeada pela região vermelha, já o retângulo de cor laranja está mostrando a parte da placa que é responsável por filtrar o sinal de entrada e estabilizar a tensão em 4.3V. O conector DB9 que encaminhará os dados ao computador está marcado com a cor cinza e o MAX232, e os devidos capacitores, que fará a equalização dos níveis de tensão nas linhas de dados está dentro do retângulo amarelo. O circuito auxiliar com o botão de habilitação do transmissor e os LEDs indicativos estão marcados pelos retângulos de cor azul e verde, respectivamente. Por último, tem-se o conector do transmissor, o qual está na cor marrom.



*Figura 91: Placa receptora (Imagem elaborada pelo autor)*

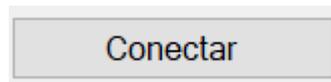
### **3.7 Etapa 6 – Software**

A última parte do sistema, corresponde ao software responsável por exibir o gráfico com os valores mioelétricos medidos pela placa principal. Esta é a principal proposta que difere dos trabalhos anteriores realizados pelo autor deste documento. Esta etapa será dividida em duas partes, interface gráfica e código fonte.

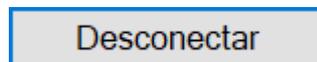
### 3.7.1 Interface gráfica

Como o principal intuito deste sistema é ser funcional, os elementos gráficos presentes correspondem apenas ao essencial, sendo assim, buscou-se tornar a operação mais intuitiva possível.

O sistema conta com um único botão, o qual é responsável por realizar a conexão do software com a placa receptora. O mesmo conta com dois estados, sendo assim, quando deve conectar seu texto é “Conectar” e quando já está conectado, o texto altera, bem como a sua função se inverte, para “Desconectar”. Os estados citados podem ser vistos, respectivamente, na Figura 92 e Figura 93 abaixo:



*Figura 92: Botão conectar (Imagem elaborada pelo autor)*



*Figura 93: Botão desconectar (Imagem elaborada pelo autor)*

Os próximos dois elementos gráficos correspondem à configuração da porta usada pelo computador, sendo assim, é possível definir o parâmetro de velocidade de comunicação (o qual já foi anteriormente citado como sendo 9600 bits por segundo) e o nome da porta que será utilizada. O primeiro parâmetro será ajustado por meio de um campo de texto que apenas aceita números, já o segundo será selecionado por meio de uma lista de nomes possíveis. Essas configurações podem ser vistas na Figura 94 e Figura 95 abaixo:



*Figura 94: Ajuste da velocidade de comunicação (Imagem elaborada pelo autor)*



Figura 95: Ajuste do nome da porta (Imagem elaborada pelo autor)

Por último se tem a interface gráfica completa, onde todos os elementos já citados são posicionados e, na maior parte da tela, o gráfico. O mesmo conta com a escala no eixo da ordenada com valores entre zero e duzentos, o que equivale à 0V a 2mV. Já no eixo da abscissa é dado o tempo em décimos de segundo, logo, como são enviadas dez amostras por segundo, cada valor deste eixo será correspondente à uma amostra coletada e enviada pela placa principal. O resultado pode ser visto na Figura 96 abaixo:

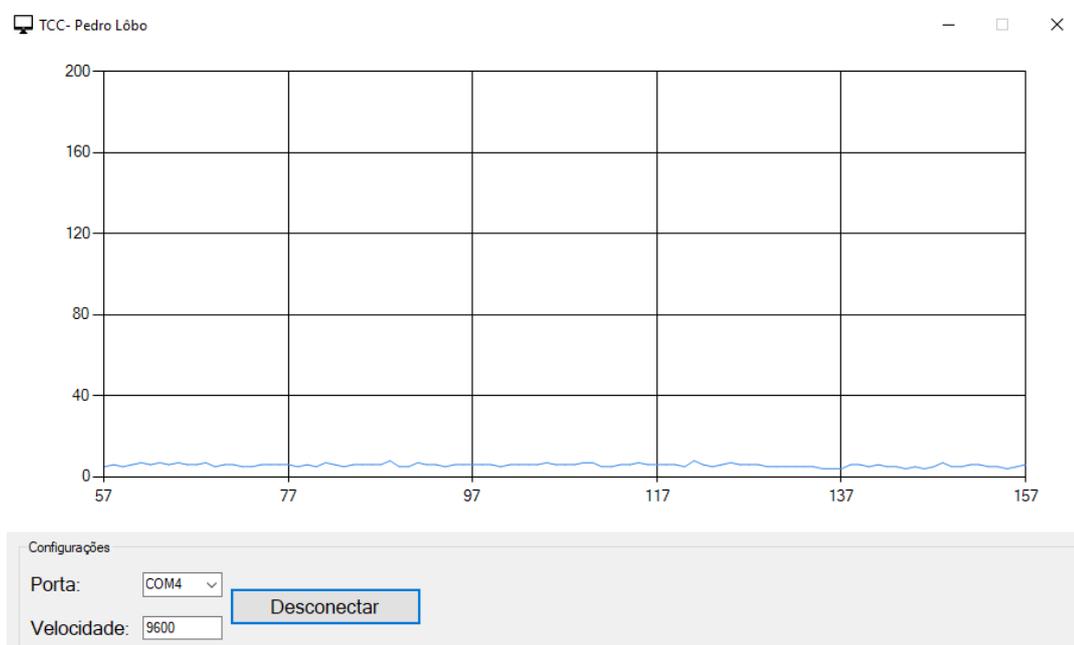


Figura 96: Interface gráfica do software (Imagem elaborada pelo autor)

### 3.7.2 Código fonte

A segunda parte do *software* é o código fonte, ou seja, a sequência de comando lógicos dados em uma linguagem de programação para que o computador execute determinada ação. Logo o código está dividido em três blocos.

O primeiro bloco se dá pela declaração das variáveis. Serão duas principais declarações, sendo a primeira do tipo *string* denominada *Pacote*, ou seja, ela receberá valores do tipo texto vindos da porta de comunicação serial. A segunda é do tipo *UInt64* chamada de

*Val\_X* e será responsável por contar as amostras recebidas e atribuir o valor da abscissa do gráfico e pode conter valores entre 0 e  $9,22 \times 10^{18}$ . Esse trecho do código pode ser visto na Figura 97 abaixo:

```
string Pacote = "";
UInt64 Val_X = 0;
```

Figura 97: Variáveis do software (Imagem elaborada pelo autor)

A segunda parte será criar a rotina responsável por atribuir as configurações para a porta e realizar a conexão. Então, ao clicar no botão “Conectar”, se a porta já estiver desconectada, o sistema irá atribuir o nome selecionada para a porta, depois ajustará a velocidade que foi previamente definida, irá abrir a conexão e, por último, alterar o nome do botão para “Desconectar”. Caso clique novamente nele o *software* irá fechar a conexão e nomear o botão como “Conectar”. A partir daí o ciclo se repetirá todas as vezes em que o mesmo for pressionado. O trecho do código em questão pode ser visto na Figura 98 abaixo:

```
private void Conectar_Click(object sender, EventArgs e)
{
    if (Conectar.Text == "Conectar")
    {
        try {
            Serial.PortName = Porta.Text;
            Serial.BaudRate = Convert.ToInt32(Velocidade.Text);
            Serial.Open();
            Conectar.Text = "Desconectar";
        }
        catch (Exception Erro)
        {
            MessageBox.Show(Erro.Message);
        }
    }
    else
    {
        try
        {
            Serial.Close();
            Conectar.Text = "Conectar";
        }
        catch (Exception Erro)
        {
            MessageBox.Show(Erro.Message);
        }
    }
}
```

Figura 98: Rotina para conectar porta (Imagem elaborada pelo autor)

Por último, o código tem a parte responsável por receber, tratar e exibir os dados no gráfico. A rotina começa com a transferência dos dados do *buffer* da porta serial para a variável “Pacote”, em seguida verifica-se se a variável contém algum dado, caso seja verdade uma série de tratamentos serão feitos para garantir que este dado é válido e pode ser usado.

Em seguida, o dado é convertido, inserido no gráfico e definido para ser exibido no gráfico em forma de linha. A parte final da rotina inclui em deslocar o gráfico horizontalmente para que tenha a continuidade do sinal mostrado e manter o limite de mínimo e máximo da abscissa. Os procedimentos aqui descritos podem ser vistos na Figura 99 abaixo:

```
Pacote = Serial.ReadExisting();
if (Pacote != "")
{
    if (Pacote.Length > 3) Pacote = Pacote.Substring(Pacote.IndexOf("A") + 1, 3);
    else if (Convert.ToInt64(Pacote) > 200) Pacote = "200";

    if (Pacote.Length < 3) return;

    Grafico.Series[0].Points.AddXY(Val_X, Convert.ToInt16(Pacote));
    Grafico.Series[0].ChartType = System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Line;

    if (Val_X > 100)
    {
        Grafico.ChartAreas[0].AxisX.Maximum = Val_X;
        Grafico.ChartAreas[0].AxisX.Minimum = Val_X - 100;
    }
    else
    {
        Grafico.ChartAreas[0].AxisX.Maximum = 100;
        Grafico.ChartAreas[0].AxisX.Minimum = 0;
    }

    Grafico.ChartAreas[0].AxisY.Maximum = 200;
    Grafico.ChartAreas[0].AxisY.Minimum = 0;

    Val_X++;
}
```

Figura 99: Rotina de recebimento dos dados (Imagem elaborada pelo autor)

#### 4 Resultados e discussões

Neste ponto do projeto, assume-se que o sistema está parcialmente desenvolvido e apresentando algum tipo de resultado. Logo, neste capítulo serão feitos os testes e a aferição se os resultados condizem com os resultados esperados.

Da mesma forma como foi feito no capítulo anterior, as rotinas de testes serão divididas nas mesmas etapas e respeitando o funcionamento de cada um dos sistemas. Ao fim, será feita a aferição dos resultados do sistema como um todo.

#### 4.1 Testes da coleta do sinal eletromiográfico

De início levanta-se a necessidade de testar o sinal coletado pelos eletrodos de superfície, porém, nesta etapa foi primeiramente aferido o nível de continuidade de cada um dos terminais do cabo. Por fim, aferiu-se a intensidade do sinal, para isso conectou-se os eletrodos e o cabo como fora anteriormente sugerido e as pontas finais foram ligadas à um osciloscópio do modelo DSO1072B. A intenção deste método é poder visualizar o sinal em natura, ou seja, antes de qualquer tratamento, porém já de antemão levanta-se a possibilidade de, devido à alta taxa de ruído, o sinal ser encoberto e não visível à olho nu.

Como já levantado, é possível observar um sinal altamente poluído e foi possível levantar o valor máximo atingido pela tensão, bem como o valor RMS. É importante citar que este teste foi feito com a musculatura contraída e, como pode ser visto destacado em azul na Figura 98, a tensão RMS foi de 1.96mV, o que condiz com o esperado para essa condição. Destacado em vermelho, é possível ver que a tensão máxima atingiu 6.40mV, o que confirma que o nível de ruído está atingindo mais de três vezes a amplitude máxima do sinal que deveria ser lido. Em conclusão, este teste teve um resultado razoavelmente superior ao esperado. O descrito pode ser visto na Figura 100 abaixo:

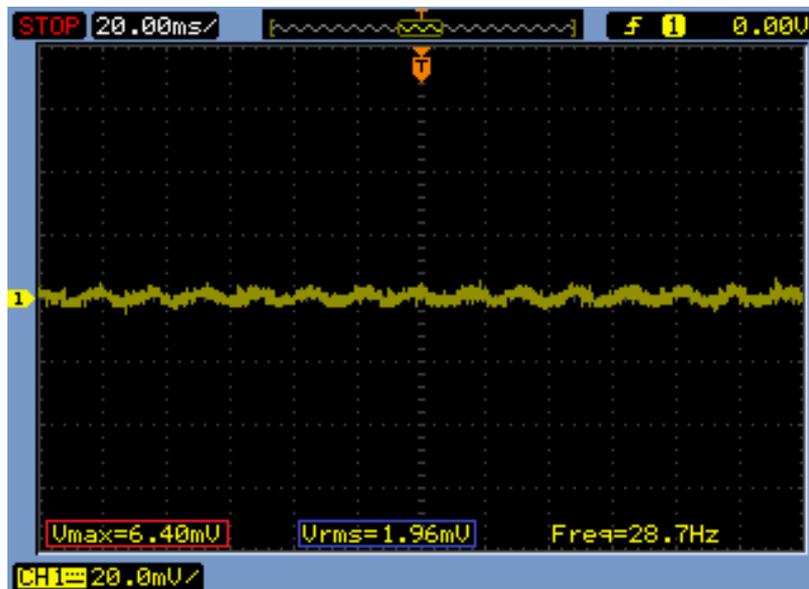


Figura 100: Teste da coleta do sinal (Imagem elaborada pelo autor)

## 4.2 Testes do tratamento primário do sinal

Após verificar que a intensidade de ruído presente no sinal mostra-se demasiadamente alta, espera-se que após a aplicação do circuito condensador o sinal de saída tenha amplitude maior (entre -4.5V e +4.5V) e o ruído esteja menos presente.

Diferentemente do teste anterior esse requer duas medições, as quais usarão o osciloscópio, o qual já foi especificado, conectado na saída do circuito e será realizada uma medição quando não há contração muscular e, em um segundo momento, com a mão fechada de forma a estimular a região monitorada. Desta forma, devem ser gerados duas formas de onda, onde na primeira aferição espera-se um sinal linear e ao contrair o músculo uma componente de amplitude e frequência fixas, porém com certa diminuição ao longo do tempo, visto que a musculatura tende a fadigar com o esforço contínuo.

Tendo os gráficos, observou-se que na ausência de contração muscular o sinal obteve certa oscilação em volta de um eixo bem definido, porém as inúmeras irregularidades demonstram a grande presença de ruídos, levando em consideração a amplitude do sinal é possível trabalhá-lo da forma como está sem maiores tratamentos. Na segunda medição, a qual mede o sinal do músculo contraído, apresente maiores problemas, pois há visível inconstância do sinal, sendo assim o nível de ruído presente neste cenário mostra-se um caminho crítico, o qual espera-se sanar com o uso da FFT. As medições aqui descritas podem ser vistas na Figura 101 e Figura 102 abaixo:

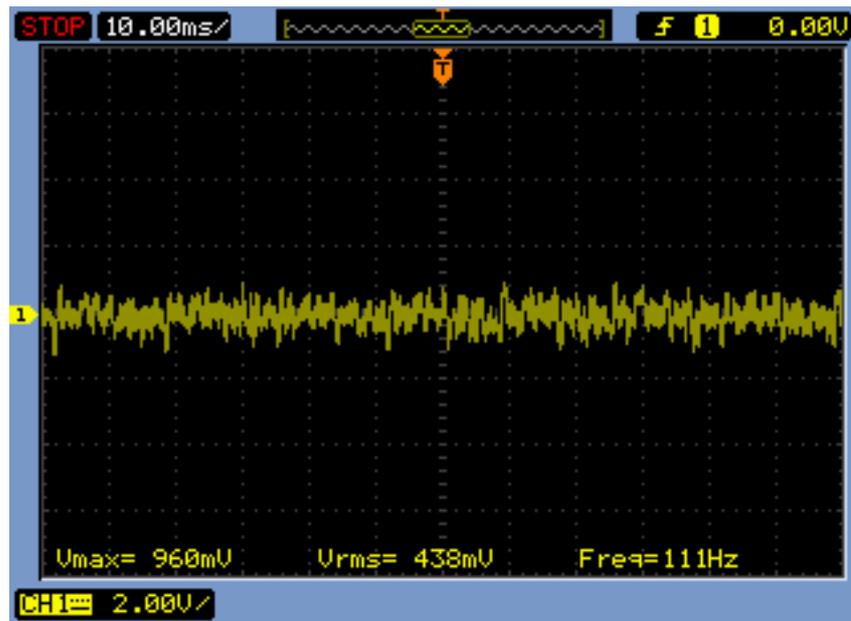


Figura 101: Teste do circuito condensador com o músculo relaxado (Imagem elaborada pelo autor)

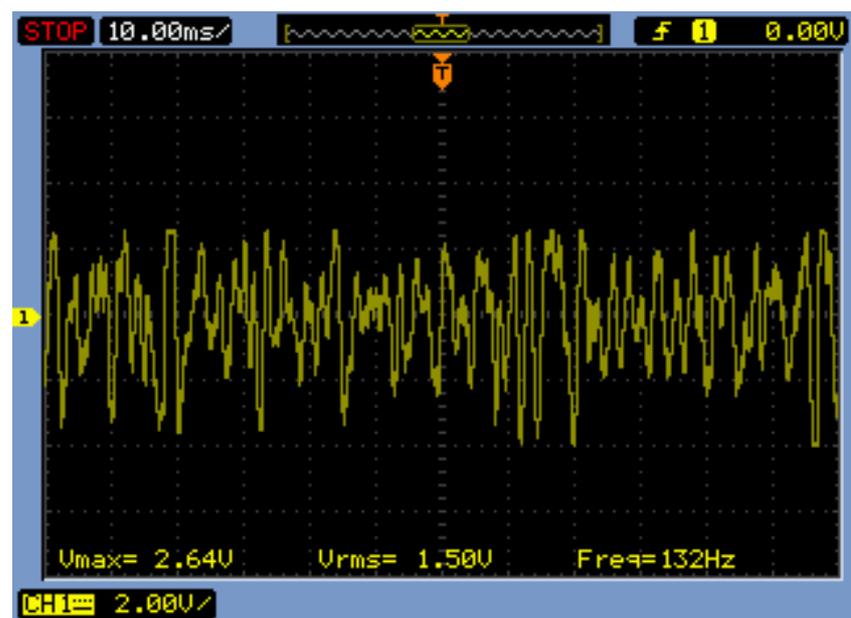


Figura 102: Teste do circuito condensador com o músculo contraído (Imagem elaborada pelo autor)

Ainda no tratamento do condensador, é coerente levantar a interferência de sinais vindos de ambientes eletricamente carregados, sendo assim este próximo teste vem sendo sugerido desde o início das pesquisas em volta deste tema (2013). Sugere-se que, com os eletrodos devidamente conectados ao braço e ao circuito condensador, o usuário toque em

alguma estrutura metalizada ou ponha os pés descasos no chão. Desta forma estaria sendo inserido no circuito ruído vindos da rede elétrica (60Hz e suas devidas harmônicas) e de dispositivos eletrônicos dispostos no ambiente (lâmpadas fluorescentes, motores, celulares, fontes chaveadas, entre outros). Para obter os resultados, usou-se a mesma metodologia do teste anterior, porém adicionou-se à tela do osciloscópio o resultado da transformada rápida de Fourier. Sendo assim mostrou-se que o sinal está completamente saturado e contrair ou não a musculatura não apresenta grandes alterações na forma de onda, já ao que tange a FFT a componente de 60Hz mostrou-se dominante, mesmo tendo os filtros analógicos na placa. O descrito nesta fase de testes pode ser visto na Figura 103 abaixo:

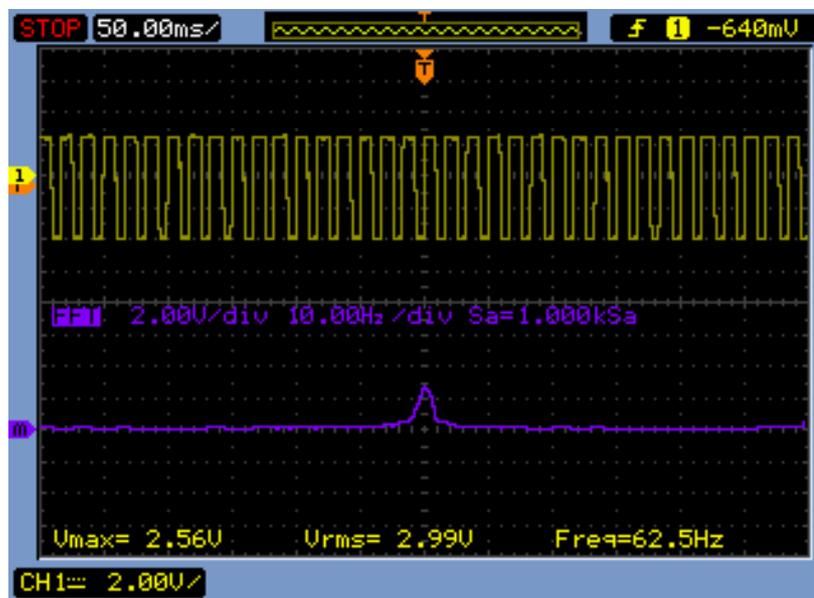


Figura 103: Influência de sinais externos no condensador (Imagem elaborada pelo autor)

### 4.3 Testes da placa principal (Processamento)

Esta fase de testes mostra-se mais complexa, visto que se trata da placa principal. Logo serão realizados três tipos de aferição, onde o primeiro se trata do controle do servo motor, a segunda referente à transmissão dos dados lidos e a terceira, e última, o tratamento dos dados.

Para o primeiro teste, referente ao controle do servomotor, utilizou-se apenas a parte do código referente ao tratamento do pulso de controle do motor, sendo assim o sistema foi posto para oscilar entre o sinal que deveria manobrar a prótese de forma a abri-la e, posteriormente, alterar o pulso para que o motor girasse no sentido contrário, a fim de fechar a mão mecânica. No primeiro momento deste teste analisou-se o sinal juntamente ao

osciloscópio e depois diretamente ligado ao motor. Quando o sistema informa ao motor que deve abrir a prótese (assumir a posição de 90º) o período de trabalho positivo deve ser por volta de 5% e o período de 20ms, no momento em que for necessário fechar a prótese (manobrando o eixo do motor à 160º) o sinal deveria alterar a duração do trabalho positivo para 8.9% e o período continuaria inalterado, sendo assim destacou-se o período em cor verde, os trabalhos positivo e negativo em vermelho e azul, respectivamente.

Ao transferir o sinal diretamente ao servo motor o mesmo apresentou a resposta como o esperado e as posições assumidas podem ser vistas na Figura 79 e Figura 80.

Os testes do sinal de controle podem ser vistos na Figura 104 e Figura 105 abaixo:

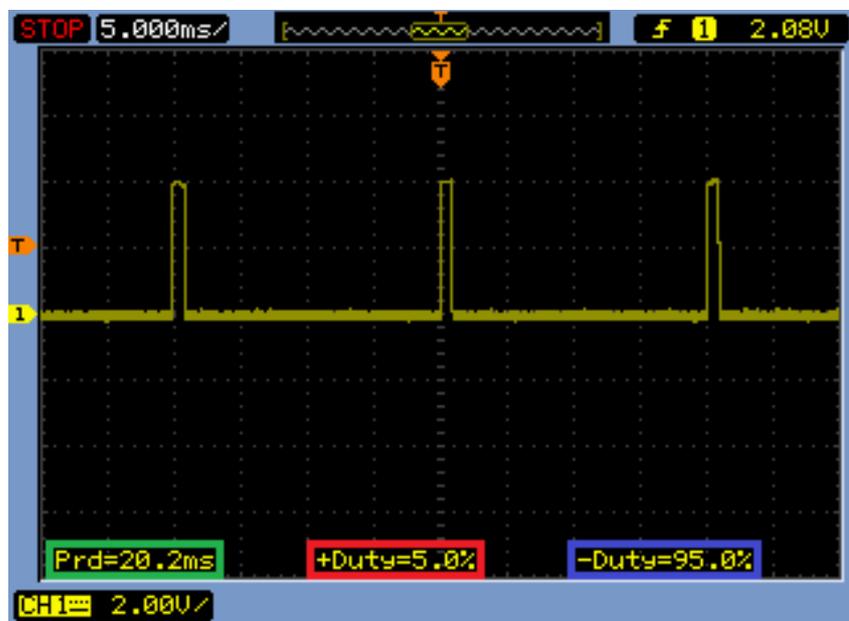


Figura 104: Sinal para abrir a prótese (Imagem elaborada pelo autor)

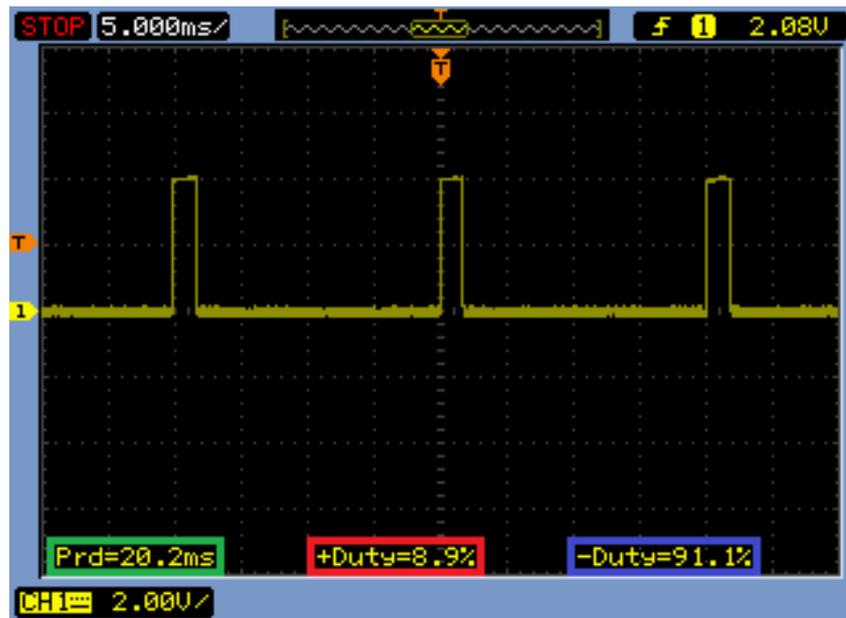


Figura 105: Sinal para fechar a prótese (Imagem elaborada pelo autor)

O próximo teste, que tange a transmissão dos dados, será feito em conjunto com o *software* de terminal virtual Hiperterminal desenvolvido pelo próprio autor deste trabalho, sendo assim foi possível verificar os dois últimos testes desta etapa, visto que foi possível visualizar a efetividade da transmissão dos sinais e os valores assumidos pelos valores lidos.

O teste ocorreu de forma a monitorar os dados exibidos pelo software auxiliar, sendo assim o mesmo exibirá um caractere “A” para cada nova amostra recebida e em seguida três caracteres que vão de “000” a “200” que irão representar a amplitude da componente de 160Hz já tratada pela FFT e transmitida pelo sistema. Deve-se assumir que o valor transmitido é dado em milivolts multiplicado por 100, sendo assim quando apresentar o valor 200 estará referindo-se à 2mV.

Em um primeiro momento (representado pelo destaque na cor vermelha) aferiu-se os valores referentes à não contração muscular, onde o valor médio foi 014, ou seja, 140uV ou 0,14mV. Neste momento a amplitude compreende o valor esperado, visto que deveria estar próximo à 100uV, apresentando um erro aceitável. Na segunda parte, correspondente à mão fechada, indicada pela cor azul exibe os valores de amplitude lido com a contração muscular. Estes valores apresentam valor médio de 080, ou seja, 800uV ou 0,8mV. O valor apresentado está abaixo do esperado e a justificativa encontrada foi os diversos estágios de atenuação que

foram incluídos pelos vários estágios de filtragem. O conteúdo descrito pode ser visualizado na Figura 106 abaixo:

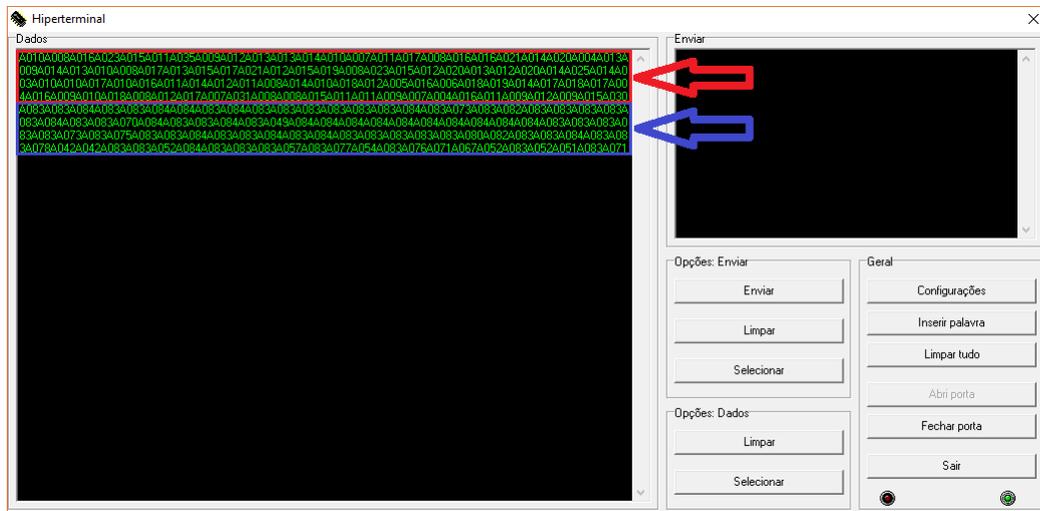
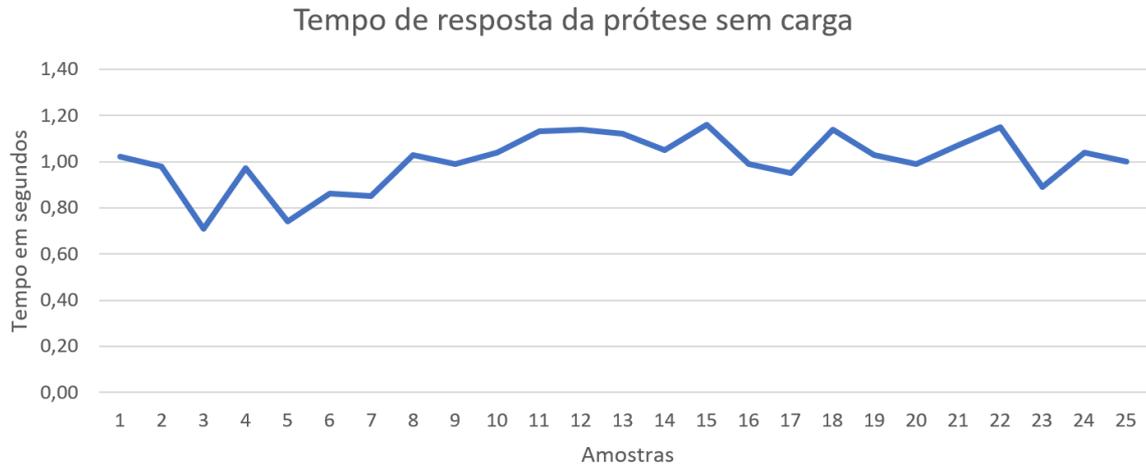


Figura 106: Dados transmitidos pelo DSP (Imagem elaborada pelo autor)

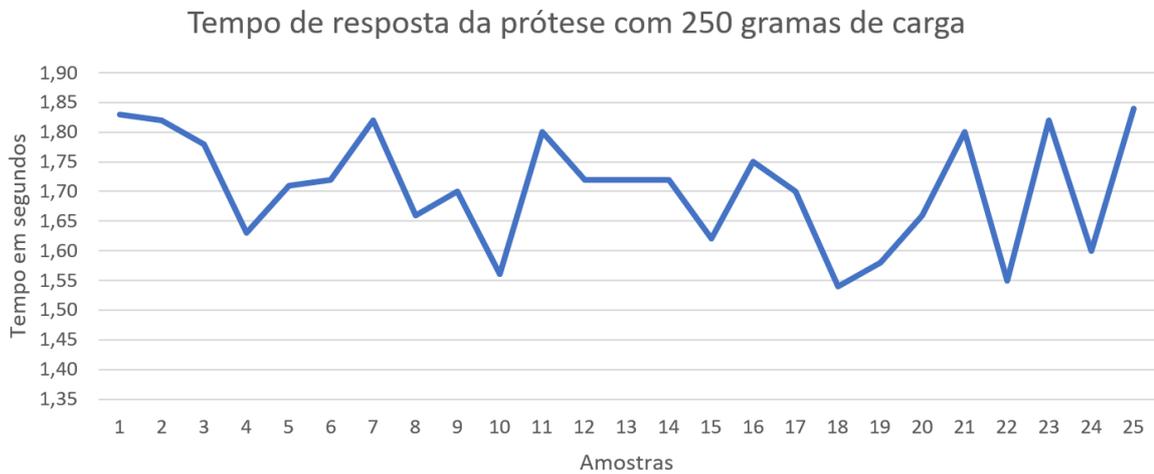
#### 4.4 Testes da estrutura mecânica

Neste ponto será levado em consideração o tempo de reação da prótese, sendo assim foram traçados gráficos que relacionam a diferença de tempo entre dar o comando para fechar a mão e a prótese concluir a ação. Para isso, o comando para abrir e fechar foi dado de forma simulada ao DSP, desta forma reduziu-se parte dos erros inerentes à aferição.

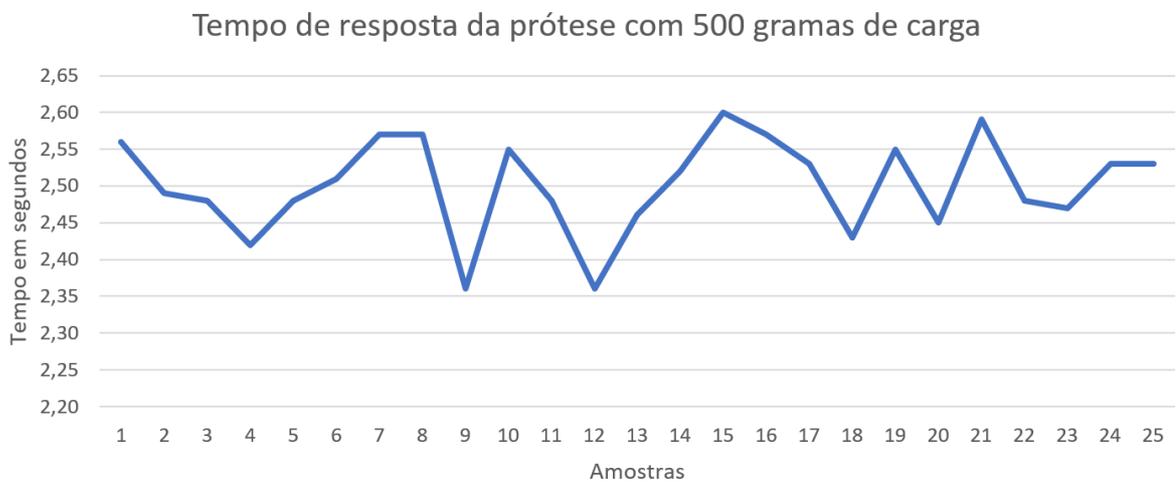
Em trabalhos anteriores já foi definido que a prótese em conjunto com o servo dimensionado é capaz de erguer massa de até 500 gramas, desta forma os testes foram definidos três cenários, sendo eles, sem carga, com 250 gramas e, por último, com 500 gramas. Para isso foram colhidas 25 amostras de cada um e criados os respectivos gráficos. Pode-se observar que o tempo de fechamento da prótese aumenta de acordo com que se incrementa a massa do objeto manipulado, notou-se também que o nível de oscilação por volta o valor médio avança de forma inversamente proporcional a massa que deve ser levantada. Como valores anotados pode-se obter o tempo médio de 1 segundo quando não há carga, 1,71 segundo com 250 gramas e, por fim, 2,5 segundos ao operar no seu limite de 500 gramas. Os resultados podem ser vistos na Figura 107, Figura 108 e Figura 109 abaixo:



*Figura 107: Tempo de resposta da prótese sem carga (Imagem elaborada pelo autor)*



*Figura 108: Tempo de resposta da prótese para 250 gramas (Imagem elaborada pelo autor)*



*Figura 109: Tempo de resposta da prótese para 500 gramas (Imagem elaborada pelo autor)*

#### 4.5 Testes da recepção dos sinais sem fio

Para o teste da placa responsável por receber os dados sem fio enviados pela placa principal, serão definidos dois cenários, confirmação de recebimento dos dados que foram enviados e defasagem de tempo entre o envio e o recebimento da mensagem.

Para o primeiro tipo de testes, será utilizado novamente o *software* Hiperterminal onde no mesmo serão escritas seis mensagens e enviadas, em seguida será aferido se todas as mensagens foram devidamente recebidas. Logo após enviar as mensagens observou-se que há, de fato, um atraso entre o envio e o recebimento, parâmetro este que será testado mais à frente. Marcados em cor azul, tem-se as mensagens enviadas e em vermelho as recebidas, o teste teve atendeu ao esperado e recebeu todos os dados e na ordem em que foram enviados. O descrito pode ser visto na Figura 110 abaixo:

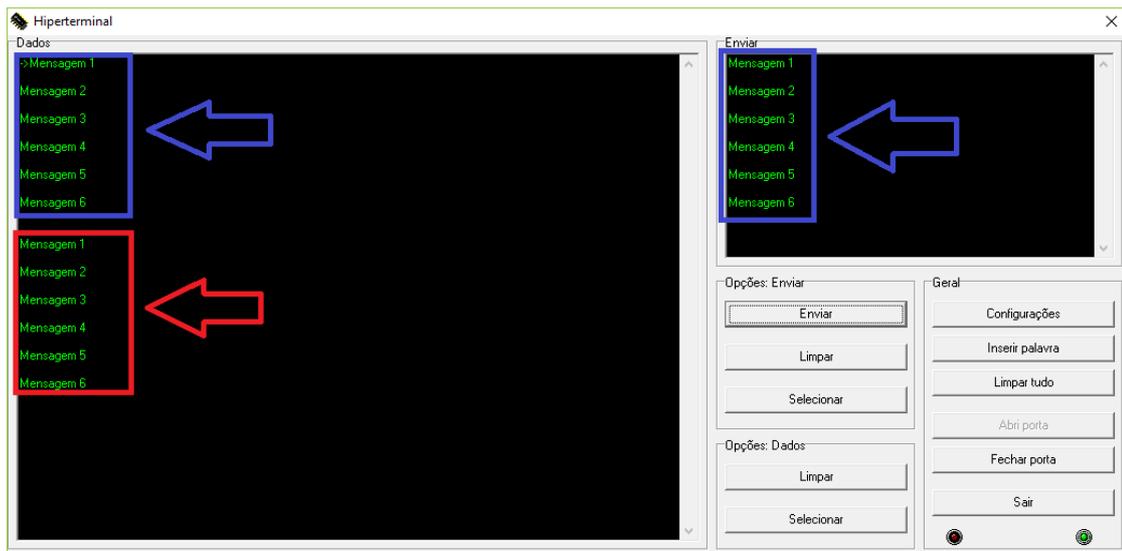


Figura 110: Teste do envio e recebimento de dados pela placa sem fio (Imagem elaborada pelo autor)

Em seguida foi necessário aferir o atraso que há entre enviar o dado e recebê-lo. Para isso programou-se o DSP para repetir a mensagem recebida, logo a placa receptora, por meio do *software* Hiperterminal, envia uma informação e mede-se o tempo que essa mensagem demora para ir e voltar. Desta forma é possível estimar a defasagem entre o sinal enviado e o que seria exibido no *software* responsável por desenhar o gráfico. Novamente, foram colhidas 25 amostras, e o teste apresentou variação máxima de 100ms entre a informação que mais demorou e a que gastou menos tempo, tendo, ainda, um valor médio de 310ms. Como o valor

apresentado demonstra-se baixo para ser facilmente percebido a olho nu, a resposta obtida neste cenário pode ser considerada boa, de forma a não prejudicar a exibição dos dados em tempo real. O teste descrito pode ser visto na Figura 111 abaixo:



Figura 111: Gráfico do tempo de envio das informações (Imagem elaborada pelo autor)

#### 4.6 Testes do *software*

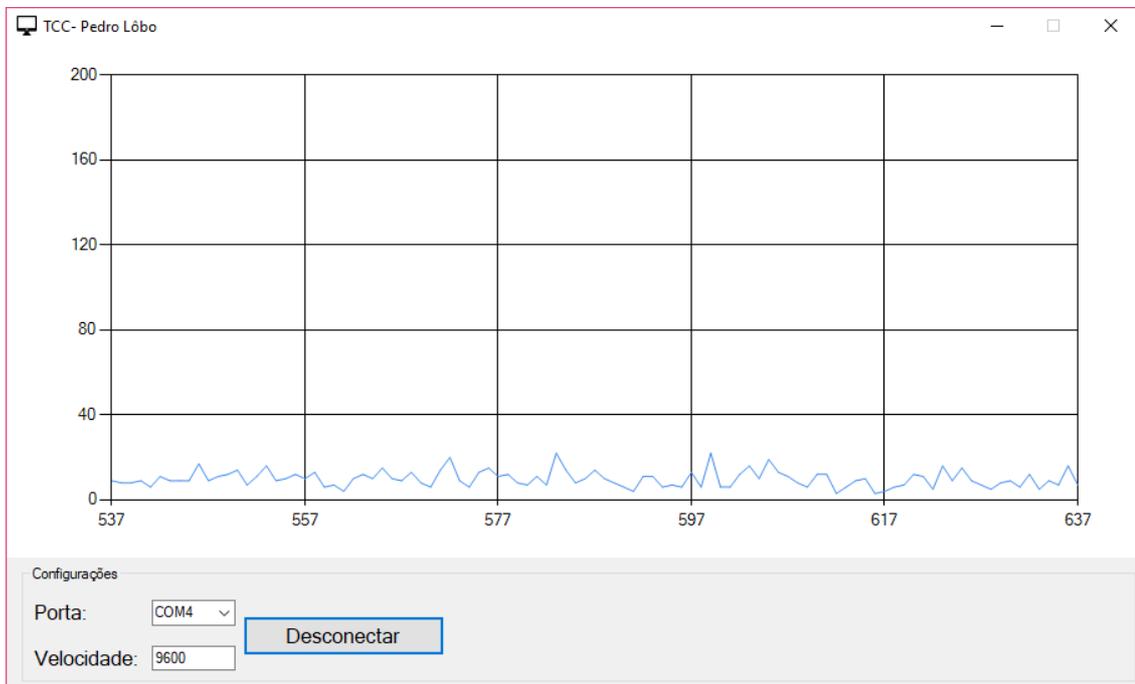
O teste final se dá pelo *software* pois para o funcionamento deste, todas as demais estruturas do projeto devem estar operando de forma correta, logo admite-se que todas as etapas estão desempenhando suas devidas funções. Para isso, os testes serão feitos em quatro cenários, sendo eles, sem contração, sem contração com os pés descalços e no chão, com contração e, por último, com movimentos periódicos entre contração e relaxamento da musculatura.

No primeiro cenário, o sinal mostrou baixa intensidade e pouca presença de sinais indesejados, sendo assim o sinal apresenta boas características para que possa ser usado a fim de manter a prótese aberta. Em um segundo momento, foi feito o mesmo teste anterior, porém considerando interferências externas de maior intensidade, logo, optou-se por um cenário mais realista onde o usuário estaria caminhando descalço ou com sapatos de baixa isolamento. Ao fim do segundo teste, constatou-se que a amplitude do sinal sofreu leve aumento em relação ao primeiro cenário, porém manteve boa estabilidade e continua suficientemente bom para a aplicação desejada.

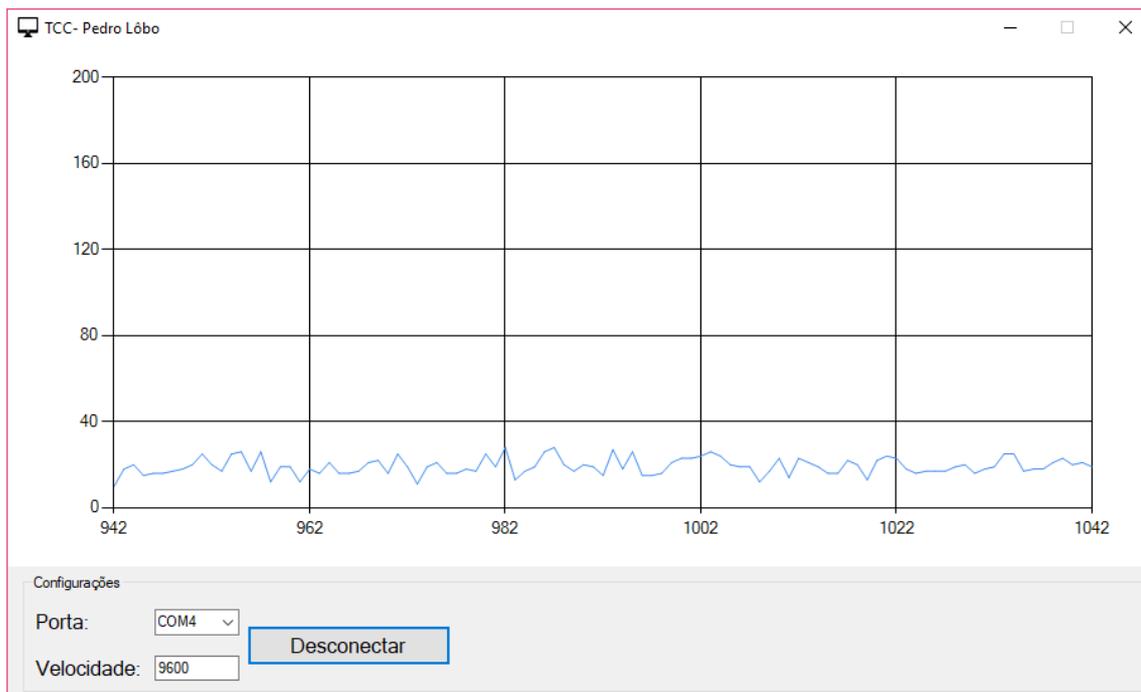
No terceiro tipo de teste, manteve-se a musculatura pressionada por aproximadamente 10 segundos e após isso começou-se a gerar o gráfico. Pode-se notar que o sinal ficou estável por boa parte do tempo, sofrendo em poucos momentos com uma pequena queda na intensidade, fato esse que é esperado pela fadiga que a musculatura trabalhada

sofre com o tempo de contração. Logo, o sinal obtido, apresenta uma amplitude inferior à aquela primeiramente esperada, porém a estabilidade é tamanha que torna o controle por essa modalidade viável. Por fim, o último teste procura trazer os resultados de um cenário onde o usuário estaria constantemente variando o estado da prótese, logo alternou-se de forma periódica entre mão aberta e mão fechada. A resposta obtida foi rápida e estável, logo os testes mostraram-se promissores e atendem ao objetivo previamente estabelecido.

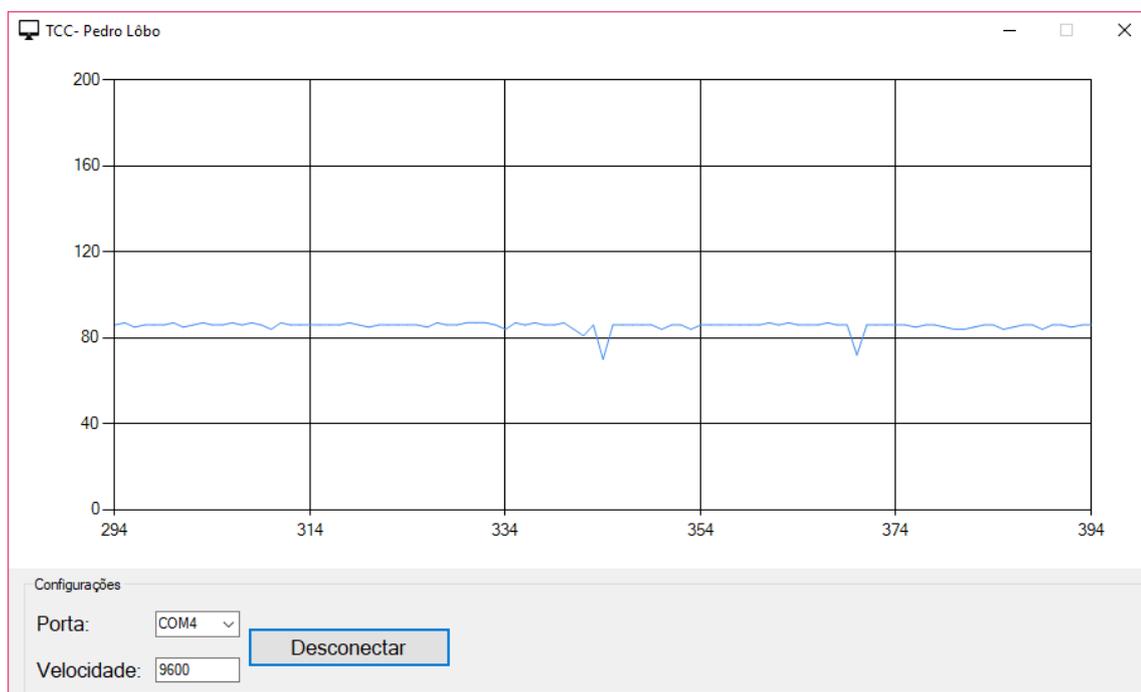
Os resultados descritos podem ser respectivamente vistos na Figura 112 Figura 113, Figura 114 e Figura 115 abaixo:



*Figura 112: Sinal sem contração (Imagem elaborada pelo autor)*



*Figura 113: Sinal sem contração com pés descalços (Imagem elaborada pelo autor)*



*Figura 114: Sinal com contração (Imagem elaborada pelo autor)*

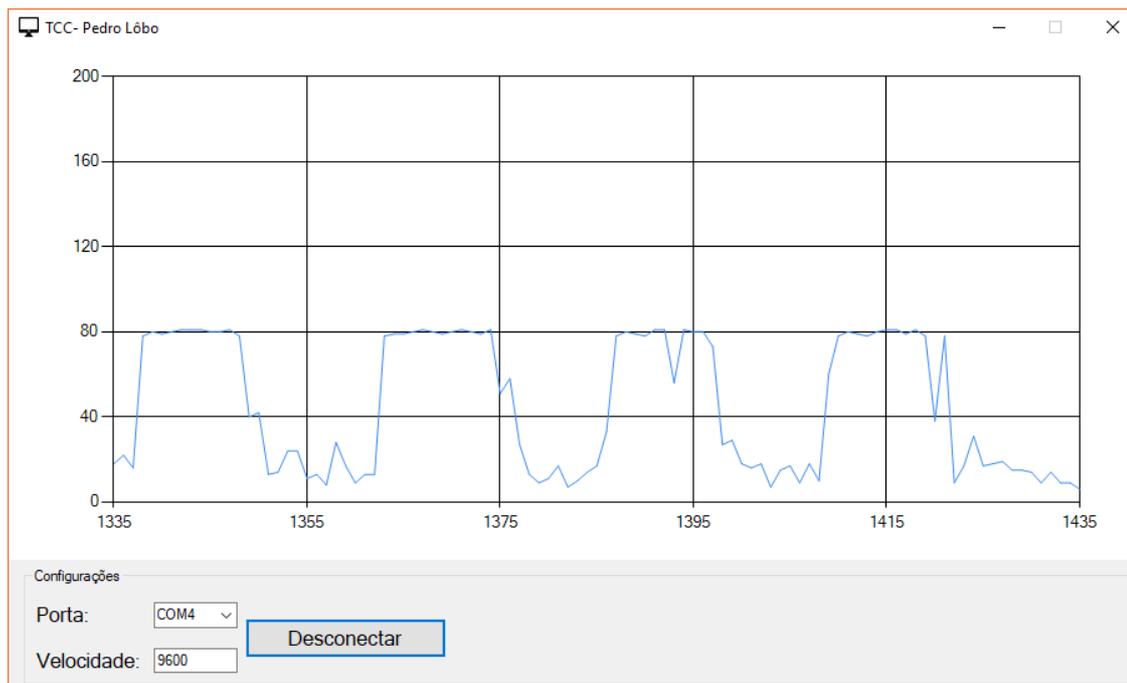


Figura 115: Sinal alternado entre contração e relaxamento (Imagem elaborada pelo autor)

## 5 Considerações finais

Ao longo dos milênios observou-se que o ser humano sempre empenhou esforços em tentar repor o membro perdido por acidentes, doenças ou guerras, fosse para recuperar a mobilidade ou apenas por fins estéticos.

Da forma com que a tecnologia evoluiu, buscou-se por órteses e próteses de materiais mais resistentes, duráveis, confortáveis e funcionais. Sendo assim, buscou-se atender ao máximo a necessidade do paciente, porém as limitações mecânicas trouxeram a necessidade de implementação computacional para a caminhada tecnológica desta área. Resolvendo este problema do avanço, surge a questão econômica que, ao implementar um computador no dispositivo, o valor de mercado acaba sendo razoavelmente impactado.

Com o novo problema aparente, a pesquisa proposta procurou trazer um equilíbrio em viabilidade econômica e boa eficiência no tratamento dos sinais eletromiográficos a fim de obter resultados praticáveis em ambiente de uso diário. Sendo assim, esperava-se que ao encaixar a necessidade em um *hardware* de baixo custo, e conseqüentemente baixo processamento, pudesse obter um sistema capaz de realizar o movimento básico de pinça, bem como auxiliar o estágio de reabilitação e treinamento do paciente a fim de aumentar a naturalidade de uso do equipamento.

No primeiro resultado proposto, movimento de pinça da prótese, o sistema executou bem a ação com boa relação entre velocidade e força. Observou-se que o uso da transformada rápida de Fourier aplicou uma filtragem significativa no sinal, porém pouco impactou na técnica de controle da prótese, resultando em uma diminuição de apenas 15% do valor de *offset* do controlador ON-OFF. A FFT mostrou-se impactante ao reduzir a interferência de ambientes eletricamente carregados e ruidosos, o que facilita a aplicação em ambientes reais de uso diário.

Ainda ao que tange o primeiro objetivo, a estrutura mecânica, trabalhando dentro da sua margem de especificações técnicas, cumpriu o objeto que se refere a segurar pequenos objetos como uma caneta, copo, garrafas, entre outros.

Pacientes que sofrem a amputações, tendem a ter a reação muscular da área danificada, prejudicada ao longo do tratamento por pouco uso, seja por alto grau de deformação ou por dores ao exercitar a região. Desde o início a reabilitação é aconselhada para a melhor manutenção da saúde muscular da região tratada, para isso usa-se órteses e

próteses no auxílio do tratamento. O movimento necessário para operar o dispositivo aqui desenvolvido assemelha-se ao esforço natural para abrir e/ou fechar a mão, porém quando a musculatura sobre alguma lesão ou limitação, o usuário precisa passar por um estágio de aprendizagem para que consiga operar o mecanismo de forma natural e intuitiva. Não se espera que toda a mobilidade seja retomada, porém procura devolver para da autoestima do paciente e dá-lhe maior independência para manipular pequenos objetos.

Por último, observou-se que ao longo dos anos de pesquisa deste trabalho, houve a queixa, por parte dos profissionais de fisioterapia que aplicaram o dispositivo em paciente reais, que não podiam observar a intensidade e melhora do sinal muscular ao passo em que a prótese era usada. Dito isso, buscou-se o desenvolvimento do *software* para que atendesse a essa demanda e pudesse gerar uma resposta à qual o profissional que acompanhe a reabilitação possa extrair maiores informações em tempo real e traçar estratégias que atendam melhor cada caso a ser atendido.

Após a modelagem/impressão 3D da prótese apresentado no item 3.5.2 conforme sugestões anteriores deste projeto, foram destacados diversos problemas em se tratando de materiais para adaptação dos sinais mioelétricos no protótipo, deixando este passo como sugestões para próximas pesquisas, e meio aos testes, algumas observações foram feitas, as quais geraram propostas que são:

- Miniaturização dos circuitos a fim de caber em uma prótese real;
- Aprimorar os estudos em torno das modelagens e impressões 3D, viabilizando economicamente os resultados;
- Desenvolvimento profissional das PCBs a fim de trazer maior resistência à ruídos;
- Montagem em caixa metálica para que, por efeito de gaiola de Faraday, o sistema tenha maior blindagem e sofra menos interferências externas;
- Adicionar ao *software* medidor de esforço a fim de parametrizar a evolução da reabilitação do paciente;
- Fazer o estudo de consumo de energia para o correto dimensionamento das baterias necessárias para manter o sistema por um dia completo;
- Aplicar uma luva de silicone para humanizar a estrutura mecânica.

## Referências

- Alecrim, E. (16 de Julho de 2005). *Memórias ROM e RAM*. Fonte: Engenharia Elétrica UFRJ: <http://www.dee.ufrj.br/cirlog/TextosCompl/memorias02.PDF>
- Araújo, F. M. (Fevereiro de 2007). *SISTEMAS DE CONTROLE*. Fonte: DEPARTAMENTO DE ENGENHARIA DE COMPUTAÇÃO E AUTOMAÇÃO: <https://www.dca.ufrn.br/~meneghet/FTP/Controle/scv20071.pdf>
- Arndt, D. M. (Abril de 2016). *Filtragem Analógica Filtragem Analógica*. Fonte: Instituto Federal de Santa Catarina: <https://wiki.sj.ifsc.edu.br/wiki/images/5/5d/FiltroAnalogico1.pdf>
- Bonfim, M. (Junho de 2010). *FILTROS ATIVOS*. Fonte: Universidade Federal do Paraná: <http://www.eletrica.ufpr.br/marlio/te054/capitulo5.pdf>
- Braskem. (21 de Setembro de 2017). *Braskem*. Fonte: Braskem: <https://www.braskem.com.br/paratletismo-infografico>
- Camara, R. C. (Novembro de 2013). *ULA- Unidade Lógica Aritmética*. Fonte: UniVASF: <http://www.univasf.edu.br/~romulo.camara/novo/wp-content/uploads/2013/11/Unidade-L%C3%B3gica-Aritm%C3%A9tica.pdf>
- Carvalho, G. L. (2004). *PROPOSTA DE UM MÉTODO DE PROJETO DE PRÓTESE DE MEMBROS SUPERIORES COM A UTILIZAÇÃO DA ENGENHARIA E ANÁLISE DO VALOR*. São Paulo: Escola Politécnica da Universidade de São Paulo.
- Cerqueira, A. S. (2007). *Filtros Digitais Aplicados em Sinais de Áudio*. Juiz de Fora: Universidade Federal de Juiz de Fora.
- CHANG, Y. C. (2011). *A Kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities. Research in Developmental Disabilities. Volume 32(6)*.
- Codá, L. M. (2013). *MEMÓRIA FLASH*. Fonte: Universidade de São Paulo: [https://edisciplinas.usp.br/pluginfile.php/343991/mod\\_resource/content/1/MEM%C3%93RIA%20FLASH\\_2013.pdf](https://edisciplinas.usp.br/pluginfile.php/343991/mod_resource/content/1/MEM%C3%93RIA%20FLASH_2013.pdf)
- Cordeiro, E. (3 de Março de 2010). *Fisioterapia e Saúde*. Fonte: Fisioterapia e Saúde: <http://blog.opovo.com.br/fisoterapiaesaude/amputados-de-membros-inferioress/>
- Duarte, P. H. (2006). Instrumentação em Eletromiografia. *Laboratório de Biofísica, Escolar de Educação Física e Esporte, Universidade de São Paulo*.
- Duque, L. H. (2013). *Gamificação dos sinais Mioelétricos e Proposta de um Controle inteligente de Prótese Mioelétrica para Membros Superiores*. Brasília: UniCEUB.
- Felipe Gabardo Gonçalves, M. A. (2015). *Frequência de operação de microcontroladores comerciais*. Fonte: Media Metrologia: [http://media.metrologia2015.org.br/media/uploads/trabalhos/freq\\_clock\\_290715\\_v3.pdf](http://media.metrologia2015.org.br/media/uploads/trabalhos/freq_clock_290715_v3.pdf)
- FERREIRA, C. S. (2016). *SISTEMAS DE CONTROLE SERVOMECANISMOS*. Brasília: UniCEUB.
- Galdino, J. (Janeiro de 2009). *ÓRTESE E PRÓTESE*.
- Galhardo, C. (2012). *Torque em um exoesqueleto estimado por sinais eletroneuromiográficos*. Fonte: Maxwell: <https://www.maxwell.vrac.puc-rio.br/20892/20892.PDF>

- Heckbert, P. (1995). *Fourier Transforms and the Fast Fourier Transform (FFT) Algorithm*. Carnegie Mellon University.
- JOHNSON, H. (1991). *Análise Básica de Circuitos Elétricos*. MEXICO: Prentice Hall.
- Lôbo, P. H. (2014). *Uso de rede neural em sinais mioelétricos para controle inteligente de prótese mioelétrica de Membros Superiores*. Brasília: CNPq.
- Malvino, A. P. (1985). *Microcomputadores e Microprocessadores*. McGraw-Hill.
- Microchip. (2004). *dsPIC30F3014, dsPIC30F4013 Data Sheet*. Fonte: <http://ww1.microchip.com/>: <http://ww1.microchip.com/downloads/en/devicedoc/70138c.pdf>
- Microchip, T. (2003). <http://www.microchip.com/>. Fonte: Microchip: <http://ww1.microchip.com/downloads/en/devicedoc/41190c.pdf>
- Mitra, S. K. (1998). *Digital Signal Processing: A Computer Based Approach*. McGraw-Hill.
- MONKMAN G. J., H. S. (2007). *Robot Grippers*. WILEY-VCH.
- Monteiro, H. M. (2016). *SISTEMA PORTÁTIL PARA DIAGNÓSTICO E PROCESSAMENTO DA VARIABILIDADE DA FREQUÊNCIA CARDÍACA BASEADA NAS TRANSFORMADAS DE WAVELET E COM ENVIO DOS RESULTADOS VIA INTERNET*. Brasília: UniCEUB.
- Morimoto, C. E. (2007). MEMÓRIA RAM. *Guia do hardware Ano 1 - Nº 4 - Abril*, 5-31.
- Ogata, K. (2011). *Engenharia de Controle Moderno*. São Paulo: Pearson.
- Oliveira, I. S. (31 de Outubro de 2010). *Módulo Comparador Analógico*. Fonte: <http://www.labs.eletrica.unir.br/>: [http://www.labs.eletrica.unir.br/uploads/41414141/arquivos/ARQUIVOS%20MICROPROCESSADOS/ROTEIRO\\_5\\_Comparador.pdf](http://www.labs.eletrica.unir.br/uploads/41414141/arquivos/ARQUIVOS%20MICROPROCESSADOS/ROTEIRO_5_Comparador.pdf)
- ONUBR. (06 de Maio de 2017). *Número de pobres no Brasil terá aumento de no mínimo 2,5 milhões em 2017, aponta Banco Mundial*. Fonte: ONUBR: <https://nacoesunidas.org/numero-de-pobres-no-brasil-tera-aumento-de-no-minimo-25-milhoes-em-2017-aponta-banco-mundial/>
- PAIXÃO, J. P. (2009). *CONTROLE DE VELOCIDADE DE MOTORES ELÉTRICOS*. Curitiba: UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ .
- Pazeto, A. C. (2014). DESENVOLVIMENTO DE UM SOFTWARE PARA ANÁLISE DE MARCHA UTILIZANDO ELETROMIOGRAFIA E ACELEROMETRIA. *Revista Eletrônica de Biologia*.
- Peña, A. (16 de Junho de 2016). *ÉPOCA Negócios*. Fonte: ÉPOCA Negócios: <http://epocanegocios.globo.com/Tecnologia/noticia/2016/06/braco-biomecanico-protese-de-baixo-custo-made-paraguai.html>
- Pereira, S. d. (2009). *Lingua C*.
- Prentice, W. E. (2002). *Modalidade Terapêutica em Medicina Esportiva*. Carolina do Norte: Manole.
- Pupin, J. R. (2011). *Introdução às Séries e Transformadas de Fourier e Aplicações no Processamento de Sinais e Imagens*. São Carlos: Universidade Federal de São Carlos.
- Resende, A. P. (2011). *Eletromiografia de superfície para avaliação dos músculos do assoalho pélvico feminino: revisão de literatura*. São Paulo: Fisioterapia e Pesquisa.

- Santos, A. (2007). *Servomotores*. Porto Alegre.
- Scandelari, L. (Novembro de 2013). *Filtros Digitais*. Universidade Tecnológica Federal do Paraná. Fonte: Universidade Tecnológica Federal do Paraná.
- Silva, B. F. (2014). *Princípios de Telecomunicações*. Fonte: Instituto Federal de Santa Catarina: [https://wiki.sj.ifsc.edu.br/wiki/images/5/5a/PRT\\_60806\\_Aula\\_11\\_-\\_Filtros\\_Anal%C3%B3gicos\\_%28Conceitos%2C\\_Par%C3%A2metros\\_e\\_Topologias%29.pdf](https://wiki.sj.ifsc.edu.br/wiki/images/5/5a/PRT_60806_Aula_11_-_Filtros_Anal%C3%B3gicos_%28Conceitos%2C_Par%C3%A2metros_e_Topologias%29.pdf)
- Silveira, D. D. (Junho de 2011). *Circuitos Lógicos - Portas Lógicas*. Fonte: Universidade Federal de Juiz de Fora: [http://www.ufjf.br/daniel\\_silveira/files/2011/06/aula\\_2.pdf](http://www.ufjf.br/daniel_silveira/files/2011/06/aula_2.pdf)
- Smith, J. O. (2002). *Mathematics of the Discrete Fourier Transform (DFT)*. Stanford: Stanford University.
- Soares, F. A. (Março de 2013). PROCESSAMENTO DE SINAIS ELETROMIOGRÁFICOS DE SUPERFÍCIE UTILIZANDO TÉCNICAS DE PROCESSAMENTO DE IMAGENS. *TESE DE DOUTORADO EM ENGENHARIA DE SISTEMAS ELETRÔNICOS E DE AUTOMAÇÃO*.
- Takahashi, L. S. (2006). *Análise da relação entre eletromiografia e força do músculo quadríceps em exercícios resistidos*. São Carlos: Universidade de São Paulo.
- Tavares, P. (2000). *INTERVENÇÃO PROTÉTICA EM AMPUTAÇÃO DE MEMBROS SUPERIORES (MMSS)*. João Pessoa: Universidade Federal da Paraíba.
- Torres, A. E. (2004). *INTRODUÇÃO AO FUNCIONAMENTO E AO ACIONAMENTO DE MOTORES DC*. Rio de Janeiro: UNIVERSIDADE FEDERAL DO RIO DE JANEIRO.
- Torres, R. C. (20 de Outubro de 2004). <http://www.cbpf.br/>. Fonte: PROCESSADORES DIGITAIS DE PROCESSADORES DIGITAIS DE SINAIS (DSPs) E SUAS SINAIS (DSPs) E SUAS APLICAÇÕES: [http://www.cbpf.br/cat/pdsi/downloads/DSPs\\_E\\_Suas\\_Aplicacoes\\_Em\\_DSP.pdf](http://www.cbpf.br/cat/pdsi/downloads/DSPs_E_Suas_Aplicacoes_Em_DSP.pdf)
- Vega, A. S. (Agosto de 2017). *Processamento Digital de Sinais*. Fonte: <http://www.telecom.uff.br/>: [http://www.telecom.uff.br/~delavega/public/DSP/apostila\\_teo\\_dsp.pdf](http://www.telecom.uff.br/~delavega/public/DSP/apostila_teo_dsp.pdf)
- WEG. (2003). *Varição de velocidade*. Jaraguá do Sul: WEB.
- Wendling, M. (2010). *Amplificadores Operacionais*. São Paulo: Unesp.
- Winkler, G. (2009). Controle de uma prótese experimental do segmento mão-braço por sinais mioelétricos e redes neurais artificiais. *Repositório Digital LUME UFRGS*.
- MOHAMMED M. , TATIENI J., CADD B. "Advanced auricular prosthesis development by 3D modelling and multi-material printing"
- NORTON K. "A brief history of prosthetics" InMotion 2007, Amputee Coalition.