



Universidade Federal de Ouro Preto  
Instituto de Ciências Exatas e Aplicadas  
Departamento de Engenharia Elétrica



## **Trabalho de Conclusão de Curso**

**Desenvolvimento de um sistema de leitura e análise estrutural utilizando técnicas espectrais**

**Guilherme Henrique de Souza Lima**

João Monlevade, MG  
2020

**Guilherme Henrique de Souza Lima**

**Desenvolvimento de um sistema de leitura e análise estrutural utilizando técnicas espectrais**

Trabalho de Conclusão de curso apresentado à Universidade Federal de Ouro Preto como parte dos requisitos para obtenção do Título de Bacharel em Engenharia Elétrica pelo Instituto de Ciências Exatas e Aplicadas da Universidade Federal de Ouro Preto.  
Orientador: Prof. Dr. Glauco Ferreira Gazel Yared  
Coorientador: Prof. Dr. Marcelo Moreira Tiago

**Universidade Federal de Ouro Preto**  
**João Monlevade**  
**2020**

## SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

L732d Lima, Guilherme Henrique de Souza .  
Desenvolvimento de um sistema de leitura e análise estrutural  
utilizando técnicas espectrais. [manuscrito] / Guilherme Henrique de  
Souza Lima. - 2020.  
97 f.: il.: color., tab..

Orientador: Prof. Dr. Glauco Ferreira Gazel Yared.  
Coorientador: Prof. Dr. Marcelo Moreira Tiago.  
Monografia (Bacharelado). Universidade Federal de Ouro Preto.  
Instituto de Ciências Exatas e Aplicadas. Graduação em Engenharia  
Elétrica .

1. Detectores - Falhas estruturais - Metais . 2. Detectores ópticos. 3.  
Manipuladores (Mecanismo) . 4. Microcontroladores . I. Tiago, Marcelo  
Moreira. II. Yared, Glauco Ferreira Gazel. III. Universidade Federal de Ouro  
Preto. IV. Título.

CDU 681.5

Bibliotecário(a) Responsável: Flavia Reis - CRB6-2431



MINISTÉRIO DA EDUCAÇÃO  
UNIVERSIDADE FEDERAL DE OURO PRETO  
REITORIA  
INSTITUTO DE CIÊNCIAS EXATAS E APLICADAS  
DEPARTAMENTO DE ENGENHARIA ELETRICA



## FOLHA DE APROVAÇÃO

**Guilherme Henrique de Souza Lima**

**Desenvolvimento de um sistema de leitura e  
análise estrutural utilizando técnicas espectrais**

Monografia apresentada ao Curso de Engenharia Elétrica da Universidade Federal  
de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Engenharia Elétrica

Aprovada em 16 de dezembro de 2020

### Membros da banca

Dr. Glauco Ferreira Gazel Yared - Orientador (UFOP)  
Dr. Marcelo Moreira Tiago - Co-orientador (UFOP)  
Dr. Carlos Henrique Nogueira de Resende Barbosa (UFOP)  
Dr. Renan Fernandes Bastos (UFOP)  
Dr. Ricardo Tokio Higuti (UNESP)  
Eng<sup>o</sup>. Renato Lataliza Vasconcelos (VALE)

Glauco Ferreira Gazel Yared, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 26/01/2021



Documento assinado eletronicamente por **Glauco Ferreira Gazel Yared, PROFESSOR DE MAGISTERIO SUPERIOR**, em 26/01/2021, às 19:21, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site [http://sei.ufop.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **0127843** e o código CRC **C41769E5**.

Referência: Caso responda este documento, indicar expressamente o Processo nº 23109.000747/2021-98

SEI nº 0127843

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35400-000  
Telefone: - www.ufop.br

# Agradecimentos

Primeiramente gostaria de agradecer a Deus por me dar forças e permitir continuar este trabalho, pois sem sua permissão nada disso se tornaria concreto.

De forma especial gostaria de agradecer ao meu pai Geraldo, minha mãe Hélia, minha irmã Maria Gabriela e meu irmão Samuel, por serem minha base de força, sempre acreditando em mim e me motivando a trabalhar na realização dos meus sonhos.

Agradeço aos meus orientadores Glauco Yared e Marcelo Moreira, pelo ensinamentos repassados mesmo em horários não convencionais e por se mostrarem sempre dispostos e motivados a colaborar com o desenvolvimento deste trabalho. Também gostaria de agradecer aos demais professores do ICEA, em especial aos professores Carlos Nogueira e Renan Fernandes por todo apoio e estímulo. Após todos esses anos de curso gostaria de dizer que tenho um profundo respeito e admiração pelos profissionais citados.

Agradeço também aos meus amigos de infância Carlos Gabriel e Marco Aurélio, por todo apoio e motivação dados.

Gostaria de agradecer aos meus amigos do laboratório ProcSiMoS Vitor Martins, Vitor Hugo, Natan e Leonardo pelos apoios dados, momentos de descontração e por todo crescimento que tivemos em conjunto.

Agradeço aos meus amigos de caminhada Adeilson, Flávio, Gabriel, Gustavo e Lorrán, por todos os momentos de descontração, amizade e companheirismo.

*"O sucesso é ir de fracasso em fracasso sem perder entusiasmo."*

*– Winston Churchill*

# Resumo

Os problemas envolvendo a realização de diagnósticos se perpetuam por diversas áreas, desde medicina até a construção civil. Particularmente, a necessidade e interesse no desenvolvimento de equipamentos que consigam realizar a detecção de defeitos estruturais de forma automática tem aumentado consideravelmente. Este trabalho apresenta o desenvolvimento de um equipamento de detecção de danos estruturais em objetos metálicos, com dimensões 2000x150x200 mm, a partir de uma análise geométrica espacial. Para a análise estrutural foi desenvolvido um manipulador mecânico microcontrolado capaz de realizar movimento em duas dimensões com resolução de 4mm e possuindo um sensor de distância óptico acoplado com resolução menor do que 1mm. Por fim os dados foram extraídos e a partir de suas características espectrais, os objetos são classificados como danificados ou saudáveis utilizando três diferentes classificadores, *Support Vector Machine*, *Artificial Neural Network* e *Gaussian Mixture Model*. O sistema alcançou uma taxa de acerto média de 99,2%.

**Palavras-chave:** Máquina de Vetor Suporte, Manipulador Cartesiano, Detecção de Defeitos Estruturais, Sensor de Distância Óptico, Microcontroladores.

# Abstract

The problems related to diagnosis are perpetuated in various areas, from medicine to civil construction. Particularly, the need and interest in developing an equipment that can automatically and mechanically detect structural problems has increased considerably. This work presents the stages of development of a structural damage detection equipment for metallic objects, with dimensions 2000x150x200 mm, from a spatial geometric analysis. For the structural analysis, a microcontrolled mechanical manipulator was developed, capable of moving in two dimensions with a resolution of up to 4 mm and having an optical distance sensor coupled with a resolution of less than 1 mm. Finally, the data were extracted and from their spectral characteristics, the objects are classified as damaged or healthy using three different classifiers, Support Vector Machine, Artificial Neural Network and Gaussian Mixture Model. The system reached an average accuracy rate of 99,2 %.

**Keywords:** Support Vector Machine, Cartesian Manipulator, Structural Defect Detection, Optical Distance Sensor, Microcontrollers.

# Lista de ilustrações

Figura 1 – Esquemático do método não destrutivo de detecção de trincas . . . . .	4
Figura 2 – Ponderação de cores baseada em medida de profundidade . . . . .	5
Figura 3 – Comparação com técnicas alternativas, à direita técnica proposta pelo autor . . . . .	5
Figura 4 – Reconstrução de uma pessoa falando . . . . .	6
Figura 5 – Escaneamento de um punho humano. A linha tracejada ressalta a parte do processo em que o equipamento proposto se mostra útil. . . . .	6
Figura 6 – Ilustração do escâner 3D proposto . . . . .	7
Figura 7 – Contorno do punho humano . . . . .	7
Figura 8 – Representação gráfica de um sinal decomposto pela série de Fourier . . . . .	9
Figura 9 – Razão entre a velocidade dos algoritmos de Transformada Discreta de Fourier (do inglês, <i>Discrete Fourier Transform</i> ) (DFT) e Transformada Rápida de Fourier (do inglês, <i>Fast Fourier Transform</i> ) (FFT) . . . . .	11
Figura 10 – Representação gráfica da Transformada de Fourier de Tempo Curto (do inglês, <i>Short Time Fourier Transform</i> ) (STFT) . . . . .	12
Figura 11 – Exemplos de tipos de Wavelets . . . . .	14
Figura 12 – Amostragem baseada em oitavas . . . . .	16
Figura 13 – Exemplo de separação linear . . . . .	17
Figura 14 – Possíveis hiperplanos de separação linear . . . . .	18
Figura 15 – a) Máquina de Vetor Suporte (do inglês, <i>Support Vector Machine</i> ) (SVM) Margem pequena. b) SVM Margem grande . . . . .	18
Figura 16 – Neurônio Biológico . . . . .	19
Figura 17 – Modelo do Neurônio Artificial . . . . .	20
Figura 18 – Exemplos de funções de ativação . . . . .	21
Figura 19 – Diagrama de treinamento supervisionado . . . . .	22
Figura 20 – Desenho Fresadora Controle Numérico Computadorizado (do inglês, <i>Computer Numeric Control</i> ) (CNC) de fuso roscado . . . . .	25
Figura 21 – Manipulador . . . . .	26
Figura 22 – Conjunto <i>Pillow Block</i> , Suporte 8mm e Motor – Vista A . . . . .	27
Figura 23 – Conjunto <i>Pillow Block</i> , Suporte 8mm e Motor – Vista B . . . . .	28
Figura 24 – Kit Sensor . . . . .	29
Figura 25 – Microcontrolador ESP32 DEVKITV1 . . . . .	31
Figura 26 – Diagrama de Comunicação . . . . .	32
Figura 27 – Diagrama de conexão do <i>driver</i> . . . . .	33
Figura 28 – Circuito de Acionamento dos Motores . . . . .	35
Figura 29 – Posicionamento dos fins de curso no manipulador . . . . .	36

Figura 30 – Circuito de Interrupções Externas . . . . .	36
Figura 31 – Sensor ODSL8 . . . . .	37
Figura 32 – Circuito de conversão . . . . .	38
Figura 33 – Circuito de Acionamento dos Motores . . . . .	38
Figura 34 – Características de Entrada e Saída do Sensor . . . . .	39
Figura 35 – Página Inicial do <i>Web Server</i> . . . . .	40
Figura 36 – Página de Nova Leitura do <i>Web Server</i> . . . . .	41
Figura 37 – Página de Diretório de Arquivos do <i>Web Server</i> . . . . .	41
Figura 38 – Página de <i>Download</i> de Arquivos do <i>Web Server</i> . . . . .	42
Figura 39 – Escâner Completo . . . . .	43
Figura 40 – Barra Metálica . . . . .	45
Figura 41 – Curvas Utilizadas da Barra Metálica . . . . .	46
Figura 42 – Barra Metálica Saudável . . . . .	47
Figura 43 – Barra Metálica Defeituosa . . . . .	47
Figura 44 – Comparação Entre os Espectros de Magnitude . . . . .	48
Figura 45 – Comparação entre os Coeficientes de Aproximação . . . . .	48
Figura 46 – Fluxograma algoritmo de Classificação . . . . .	50
Figura 47 – Conexões básicas de motores de passo . . . . .	61
Figura 48 – Método de controle do motor de passo . . . . .	61
Figura 49 – Arquiteturas Harvard e Von-Neumann . . . . .	63
Figura 50 – Método de Triangulação óptica do sensor . . . . .	64
Figura 51 – Montagem Barra 2500mm, suporte de 16mm e suspensão . . . . .	96
Figura 52 – Montagem Suporte 8mm e Pillow Block 16mm . . . . .	97
Figura 53 – Montagem Barra 8mm e Pillow Block 8mm . . . . .	97

# Lista de Siglas

<b>A/D</b>	Analógico-Digital
<b>ANN</b>	Rede Neural Artificial (do inglês, <i>Artificial Neural Network</i> )
<b>CTFT</b>	Transformada de Fourier de Tempo Contínuo (do inglês, <i>Countinuous Time Fourier Transform</i> )
<b>CNC</b>	Controle Numérico Computadorizado (do inglês, <i>Computer Numeric Control</i> )
<b>CWT</b>	Transformada Wavelet de Tempo Contínuo (do inglês, <i>Countinuous Wavelet Transform</i> )
<b>DFT</b>	Transformada Discreta de Fourier (do inglês, <i>Discrete Fourier Transform</i> )
<b>DTFT</b>	Transformada de Fourier de Tempo Discreto (do inglês, <i>Discrete Time Fourier Transform</i> )
<b>DWT</b>	Transformada Wavelet Discreta (do inglês, <i>Discrete Wavelet Transform</i> )
<b>EM</b>	Maximização de Esperança (do inglês, <i>Expectation Maximization</i> )
<b>EUA</b>	Estados Unidos da América
<b>FFT</b>	Transformada Rápida de Fourier (do inglês, <i>Fast Fourier Transform</i> )
$f_s$	Frequência de Amostragem
<b>GMM</b>	Modelo Mistura de Gaussianas (do inglês, <i>Gaussian Mixture Model</i> )
<b>GPU</b>	Unidade de Processamento Gráfico (do inglês, <i>Graphics Processing Unit</i> )
<b>MLE</b>	Estimativa de Máxima Verossimilhança (do inglês, <i>Maximum Likelihood Estimate</i> )
$n_{bits}$	Número de Bits
<b>OAB</b>	Ordem dos Advogados do Brasil
<b>RMS</b>	Raiz Média Quadrática (do inglês, <i>Root mean square</i> )
<b>sLIT</b>	Sistemas Lineares e Invariantes no Tempo
<b>STFT</b>	Transformada de Fourier de Tempo Curto (do inglês, <i>Short Time Fourier Transform</i> )

$V_{ref}$  Valor de Referência

**SVM** Máquina de Vetor Suporte (do inglês, *Support Vector Machine*)

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>1</b>
<b>1.1</b>	<b>Objetivos</b>	<b>2</b>
<b>1.2</b>	<b>Revisão Bibliográfica</b>	<b>2</b>
1.2.1	Problemas causados por trincas	2
1.2.2	Identificação de trincas	3
1.2.3	Escâners 3D	4
<b>1.3</b>	<b>Estrutura do Trabalho</b>	<b>8</b>
<b>2</b>	<b>REVISÃO TEÓRICA</b>	<b>9</b>
<b>2.1</b>	<b>A Classe de Ferramentas de Fourier</b>	<b>9</b>
2.1.1	Introdução	9
2.1.2	Transformada Discreta de Fourier	10
2.1.3	Comentários	11
<b>2.2</b>	<b>A Transformada Wavelet</b>	<b>11</b>
2.2.1	Introdução	12
2.2.2	A Transformada Wavelet	13
2.2.3	A Transformada Wavelet Discreta	15
2.2.4	Comentários	16
<b>2.3</b>	<b>Classificação</b>	<b>17</b>
2.3.1	<i>Support Vector Machine</i>	17
2.3.2	<i>Artificial Neural Network</i>	19
2.3.3	<i>Gaussian Mixture Model</i>	22
2.3.4	Comentários	23
<b>3</b>	<b>METODOLOGIA</b>	<b>24</b>
<b>3.1</b>	<b>Sistema Mecânico</b>	<b>24</b>
<b>3.2</b>	<b>Sistema Embarcado</b>	<b>29</b>
3.2.1	ESP32	30
3.2.2	Acionamento dos Motores	32
3.2.3	Sensores	35
3.2.4	Leitura e Salvamento dos Dados	38
<b>3.3</b>	<b>Interface Homem/Máquina</b>	<b>39</b>
<b>3.4</b>	<b>Classificação</b>	<b>42</b>
<b>4</b>	<b>RESULTADOS</b>	<b>43</b>
<b>4.1</b>	<b>Sistema Mecânico</b>	<b>43</b>

4.2	Classificação - Detecção de Trincas em Barras Metálicas . . . . .	45
5	CONSIDERAÇÕES FINAIS . . . . .	53
6	SUGESTÕES PARA TRABALHOS FUTUROS . . . . .	55
	REFERÊNCIAS . . . . .	56

# 1 Introdução

Com o aumento da demanda por produtos manufaturados, o mercado sofreu um grande aumento de produção de bens e simultaneamente se deparou com a diminuição do número de operários nas linhas de produção e montagem (STRAWN, 2016). Dentre vários aspectos proporcionados pela revolução indústria 4.0, é notável que diversos produtos sejam manufaturados sem passar pelo devido controle de qualidade e muitas vezes são vendidos e entregues aos clientes em condições não ideais. Isso tem ocorrido com mais frequência, dado que a diminuição do número de pessoas na linha de produção está diretamente relacionado com a probabilidade de um produto defeituoso ser notado. Visto que diversas vezes resulta em devolução integral do pagamento do cliente e/ou troca do produto (BRASIL, 2007; SUL, 2012), há um grande interesse industrial em equipamentos que possam detectar defeitos em produtos de forma automática. Um dos principais tipos de defeitos que se deseja detectar são as trincas, que podem ser provocadas durante o processo de fabricação ou durante o processo de manipulação dos produtos.

A capacidade de identificação de trincas (tanto internas quanto externas) tem se mostrado relevante em diversas áreas, como na análise da qualidade de rodovias, ferrovias, edificações, indústria, biomédicas (YING et al., 2010; YELLA; RAHMAN; DOUGHERTY, 2010; SHEN, 2016; CHO; YOON; JUNG, 2018) e etc, de forma que esse tipo de estudo tem aumentado consideravelmente nas últimas quatro décadas (DAS; PARHI, 2009).

Muitos trabalhos desenvolvem métodos de identificação de trincas a partir de processamento digital de imagens abordando desde detecção de trincas em rodovias a partir de imagens de câmeras (RABABAAH; VRAJITORU; WOLFER, 2005) até detecção de fissuras pulmonares a partir de imagens geradas por tomografias (GERARD et al., 2019). As imagens utilizadas nesses trabalhos podem ser representadas em duas ou três dimensões, e podem ser adquiridas através de diferentes equipamentos, tais como sensores ópticos laser, câmeras de alta resolução ou sistemas ultrassônicos, por exemplo.

O desenvolvimento de manipuladores microcontrolados possibilita diversas aplicações indústria da manufatura. Proporcionando precisão e velocidade na movimentação de materiais, equipamentos e sensores, os manipuladores são equipamentos chaves na indústria. O uso de um manipulador cartesiano microcontrolado com um sensor de distância óptico acoplado permite a criação de imagens 3D de alta resolução com dimensões limitadas apenas pelo tamanho do manipulador e alcance do sensor.

Sendo assim serão apresentados com detalhes as etapas de desenvolvimento de um sistema de escaneamento e análise de defeitos estruturais de objetos. O sistema de escaneamento foi projetado com um manipulador cartesiano de 2x1m com um sensor de distância óptico acoplado com 0.5m de alcance. Para o desenvolvimento do sistema de análise foram realizados experimentos com barras metálicas com defeitos estruturais.

O sistema foi desenvolvido com testes em combinações de extrações de características variadas e classificadores variados.

## 1.1 Objetivos

O objetivo principal deste trabalho é identificar danos estruturais em objetos através de uma imagem tridimensional formada pela leitura de sinais geométricos detectados por um sensor de distância óptico acoplados a um manipulador cartesiano.

Os objetivos específicos são:

- Projeto de um manipulador cartesiano eletromecânico controlado remotamente com resolução de 3mm para movimentação do sensor de distância laser com resolução de 1mm.
- Validar o sistema de movimentação e aquisição de sinais.
- Avaliar técnicas de técnicas de processamento de sinais para classificação de trincas em objetos.
- Validar os classificadores e métodos de parametrização analisados utilizando uma base de dados de 50 observações.

## 1.2 Revisão Bibliográfica

Como o trabalho consiste na criação de um sistema de leitura de sinais e detecção de trincas, a revisão será efetuada em três partes. Na primeira parte serão abordados problemas causados por trincas em diversas áreas, na segunda parte serão abordadas soluções disponíveis no mercado para detecção automática de trincas, e por fim na terceira parte serão abordados temas relacionados aos escâners 3D.

### 1.2.1 Problemas causados por trincas

Shen (2016) relata como as trincas em rodovias diminuem consideravelmente sua vida útil, e como grandes companhias e laboratórios nos Estados Unidos, Japão, Reino Unido bem como diversos outros pesquisadores ao redor do mundo têm proposto soluções de detecção dessas trincas com base em imagens de vídeo da estrada.

O trabalho de Chen e Jahanshahi (2018) apresenta uma relação entre os efeitos de trincas e a redução da vida útil de estruturas de concreto utilizadas na construção civil. Os autores afirmam que, por se tratarem de grandes construções, é necessário utilizar equipamentos de grande porte, como andaimes e veículos por exemplo, para que métodos de inspeção convencionais sejam implementados. E é notável que trincas em estruturas

como essas afetam consideravelmente o cotidiano das pessoas, seja no quesito mobilidade urbana ou segurança.

Cai et al. (2018) expõe como trincas em concretos são frequentes e com isso diminuem a vida útil de construções como pontes, edifícios, túneis entre outras estruturas e que por se tratarem de grandes construções os métodos convencionais por vezes necessitam do auxílio de grandes equipamentos como andaimes, veículos ou até mesmo telescópios.

### 1.2.2 Identificação de trincas

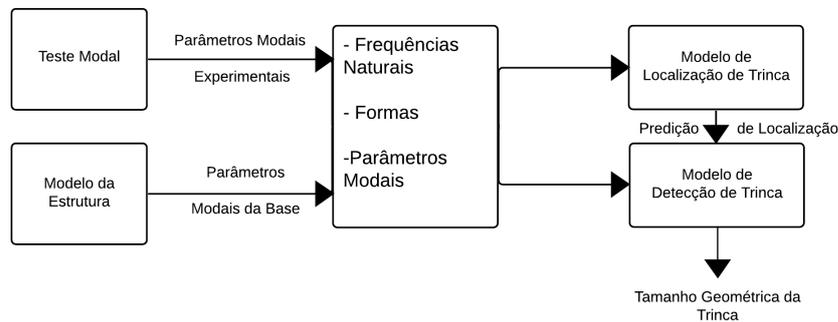
Diversas abordagens foram e são propostas pela comunidade científica com o intuito de solucionar e/ou melhorar a abordagem de identificação de trincas.

Rababaah, Vrajitoru e Wolfer (2005) abordaram o problema de trinca em grandes estruturas e em pavimentos de asfalto a partir de imagens. Sua principal motivação foi propor soluções de detecção de trincas que tivessem um custo computacional menor em relação as utilizadas atualmente pelo setor de avaliação, mantendo um bom desempenho e com uma boa velocidade de processamento que pudesse ser aplicado em sistemas de tempo real. Os autores compararam os resultados de três classificadores diferentes, dois supervisionados (*Genetic Algorithm* e o *Multilayer Perceptron*) e um não supervisionado (*self-organizing map*). Cada um dos classificadores foi testado em duas situações distintas com entradas diferentes, a primeira sendo baseada na Transformada de Hough e a segunda baseada na matriz de mosaico. Os autores afirmam que todos os classificadores obtiveram resultados acima de 98,2%, mas o método *Multilayer Perceptron* obteve a maior precisão, 98,6%, além de apresentar uma melhor eficiência computacional.

Gerard et al. (2019) analisaram fissuras pulmonares a partir de imagens de tomografia computadorizada. Os autores propõem uma ferramenta denominada FissureNet, em que a detecção de fissuras foi modelada como uma classificação probabilística de dados  $X$  e classes  $Y$ . O conjunto de classes  $Y$  foi formada por três classes com fissuras e uma sem fissura. O FissureNet utiliza duas redes neurais convolucionais como uma abordagem *coarse-to-fine* para diminuir a complexidade de treinamento do classificador. Segundo os autores, o FissureNet mostrou resultados consistentes mesmo para diferentes escâners de tomografia computadorizada, alcançando uma alta sensibilidade e tendo baixos falso positivos.

Kim e Stubbs (2003) propõem uma estratégia não destrutiva de detecção de trincas em vigas a partir de variações de frequência natural na estrutura. Para detectar as trincas os autores propõem primeiramente localizar as trincas e em seguida mensurar o seu tamanho. Essa estratégia é utilizada pois, analisando a sensibilidade modal da estrutura, é possível estimar tanto a localização quanto o tamanho da trinca. A Figura 1 demonstra o esquema proposto pelos autores para detecção de trincas.

Figura 1 – Esquemático do método não destrutivo de detecção de trincas



Fonte: Adaptado de Kim e Stubbs (2003)

Os resultados para detecção e localização de trincas em vigas se mostraram satisfatórios, sendo possível determinar o tamanho das trincas com um baixo erro, tendo apenas o conhecimento de algumas frequências naturais das vigas antes e após os danos.

### 1.2.3 Escâners 3D

O uso de escâners 3D tem se mostrado relevante para diversas aplicações, e dentre elas a detecção de trincas. Isso tem despertado o interesse da comunidade científica para produzir modelos de escâners 3D com baixo-custo, eficiência computacional, velocidade de escaneamento, mobilidade e etc.

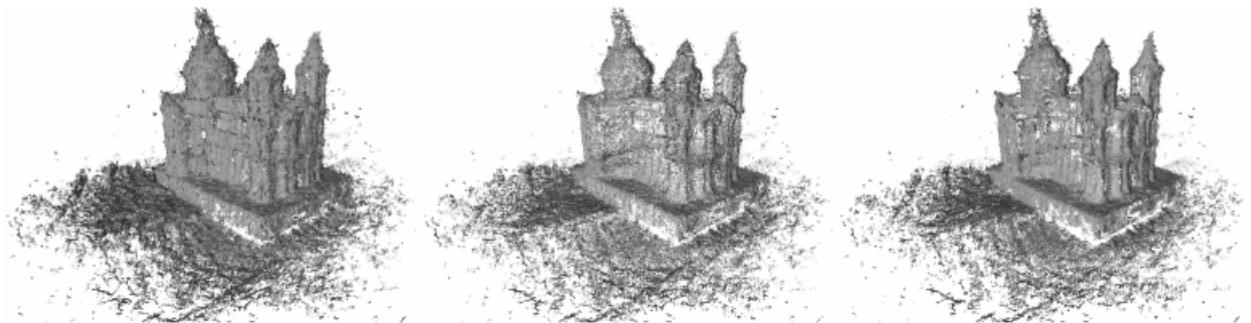
Kolev et al. (2014) propõe um projeto de escâners 3D utilizando *smartphones*. Os autores informam que muitos sistemas baseados em câmeras de vídeo e sensores de profundidade tem sido propostos, contudo esses sistemas requerem uma capacidade de processamento muito alta e conseqüentemente, ficam atrelados a um computador de mesa. A proposta dos autores é desenvolver um sistema interativo de reconstrução 3D a partir de um *smartphone* que possa ser utilizado em tempo real e com alta precisão. O sistema proposto é capaz de criar uma imagem 3D a partir da análise de conflito entre diversas fotografias do objeto e, de acordo com os autores, o método proposto se mostrou eficiente e preciso. A Figura 2 demonstra à esquerda a imagem real do objeto e à direita a imagem gerada por ponderação de cores. A Figura 3 mostra um comparação da técnica testada pelo autor com duas outras técnicas alternativas.

Figura 2 – Ponderação de cores baseada em medida de profundidade



Fonte: Retirado de Kolev et al. (2014)

Figura 3 – Comparação com técnicas alternativas, à direita técnica proposta pelo autor



Fonte: Retirado de Kolev et al. (2014)

Weise, Leibe e Gool (2007) apresentam um projeto de escâners 3D de baixo custo capaz de escanear cenas e objetos em tempo real a partir de 3 câmeras analiticamente posicionadas. Duas câmeras monocromáticas são estrategicamente alocadas, e trabalham de forma sincronizada para capturar as imagens de fase enquanto a terceira câmera simultaneamente realiza uma leitura colorida do ambiente. O sistema proposto pode realizar leitura dinâmica de ambiente baseando-se em iluminação ativa e deslocamento de fase. O autor afirma que o sistema é capaz de produzir cenas dinâmicas a 17 quadros por segundo com uma informação de profundidade precisa, tendo alta velocidade de processamento devido ao uso de métodos de processamento com Unidade de Processamento Gráfico (do inglês, *Graphics Processing Unit*) (GPU) e capacidade de superar problemas de descontinuidade de fase causados por movimentos relativos entre as câmeras e o objeto. A Figura 4 demonstra o resultado do sistema em reconstruir a imagem de uma pessoa enquanto fala.

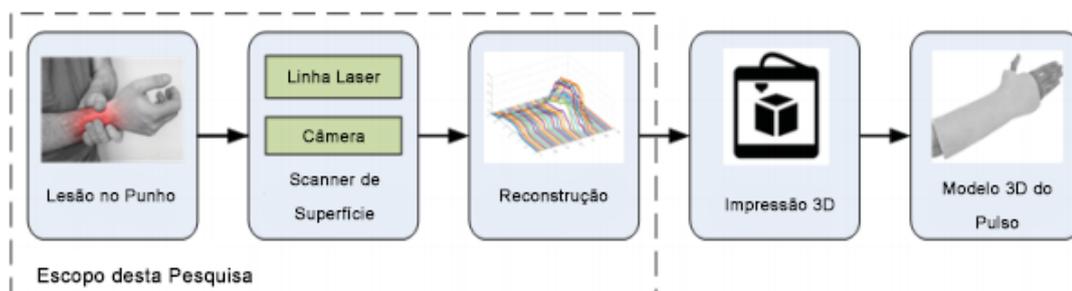
Figura 4 – Reconstrução de uma pessoa falando



Fonte: Retirado de Weise, Leibe e Gool (2007)

Yunardi e Imandiri (2018) desenvolveram um equipamento de escaneamento 3D utilizado para obter a geometria espacial do punho humano e com isso possibilitar a impressão de equipamentos ortopédicos de correta medição. O projeto é realizado utilizando um emissor de linha laser e uma câmera que utiliza a percepção do laser em varredura de linha para determinar a forma geométrica da superfície. A Figura 5 representa o diagrama de processo de produção de um suporte para punho humano.

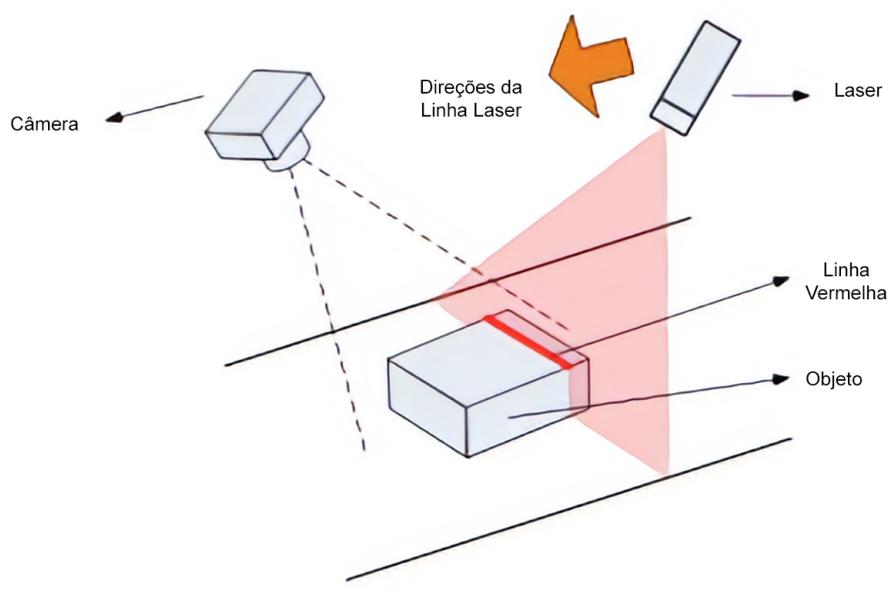
Figura 5 – Escaneamento de um punho humano. A linha tracejada ressalta a parte do processo em que o equipamento proposto se mostra útil.



Fonte: Adaptado de: Yunardi e Imandiri (2018)

A Figura 6 ilustra o equipamento produzido pelos autores e exemplifica o formato de leitura.

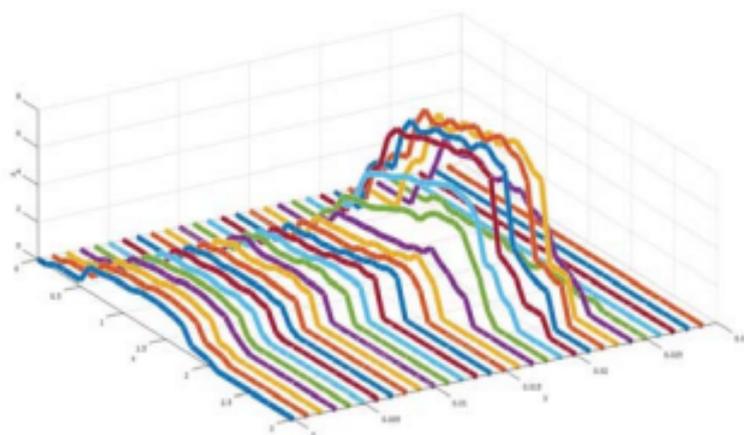
Figura 6 – Ilustração do escâner 3D proposto



Fonte: Adaptado de Yunardi e Imandiri (2018)

A Figura 7 demonstra o desenho das curvas obtidas pela varredura efetuada pelo equipamento desenvolvido.

Figura 7 – Contorno do punho humano



Fonte: Retirado de Yunardi e Imandiri (2018)

O equipamento foi testado para leitura de uma caixa de papelão e um pulso humano saudável, o sistema apresentou um erro Raiz Média Quadrática (do inglês, *Root mean square*) (RMS) de 0,47 por 0,40 centímetros.

## 1.3 Estrutura do Trabalho

Este trabalho foi estruturado em 5 capítulos da seguinte maneira:

No capítulo 1 foi realizado uma introdução sobre o tema, bem como foram apresentados os objetivos propostos. Por fim foram abordados trabalhos correlatos ao tema estudado.

No capítulo 2 é realizado uma breve revisão teórica, com conceitos teóricos fundamentais para o desenvolvimento deste trabalho. Neste capítulo ao início de cada seção será apresentado uma introdução sobre o que será tratado e ao fim é realizado uma subseção de comentários, onde são abordados os principais tópicos trabalhados.

No capítulo 3 são apresentadas as metodologias e as especificações de projeto do sistema proposto neste trabalho. Em seguida o projeto completo é dividido e apresentado em três seções, sendo elas o sistema mecânico, o sistema eletrônico e a classificação.

Por fim, nos capítulos 4 e 5 são apresentados os resultados e as considerações finais deste trabalho, respectivamente.

## 2 Revisão Teórica

Neste capítulo será apresentada uma revisão sobre os conceitos, ferramentas e técnicas necessárias para o desenvolvimento do equipamento de leitura e análise de sinais de geometria espacial proposto. Ao final de cada seção haverá uma subseção de comentários, abordando os principais pontos anteriormente descritos.

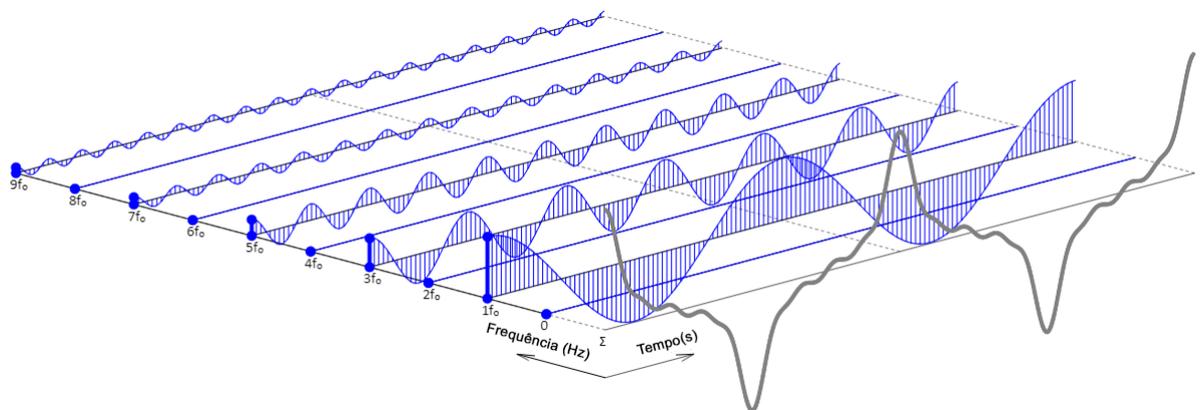
### 2.1 A Classe de Ferramentas de Fourier

Nessa seção serão apresentados as técnicas de análise espectral propostas por Fourier e alguns derivativos.

#### 2.1.1 Introdução

Jean Baptiste Joseph Fourier em 1807 esboçou um conceito que fora publicado em 1822 em seu livro, *La Théorie analytique de la chaleur* (FOURIER, 1822), traduzido para o português como A teoria analítica do calor, onde mostrou que qualquer sinal periódico de tempo contínuo, desde que pudesse satisfazer alguns básicos requisitos matemáticos, poderia ser escrito como um somatório de senos e/ou cossenos de frequências diferentes, cada uma multiplicado por um coeficiente de amplitude próprio, e com isso ele apresentou o que hoje é conhecido como a série de Fourier. A Figura 8 apresenta um sinal periódico (em cinza) e sua componentes de frequências (em azul) decompostas pela série de Fourier.

Figura 8 – Representação gráfica de um sinal decomposto pela série de Fourier



Fonte: Do Autor

Até mesmo funções aperiódicas contínuas, porém com área sob a curva finita, podem ser representadas como uma integral de senos e/ou cossenos multiplicados por uma função de ponderação. Esse conceito é conhecido como Transformada de Fourier de Tempo Contínuo (do inglês, *Continuous Time Fourier Transform*) (CTFT). (GONZALEZ; WOODS, 2008). A equação da CTFT, também chamada de equação de análise, é descrita por:

$$X(j\omega) = \int_{-\infty}^{+\infty} x(t)e^{-j\omega t} dt. \quad (2.1)$$

Em que  $X(j\omega)$  é a transformada do sinal  $x(t)$  e  $\omega$  é a frequência angular.

Como a função  $X(j\omega)$  deve conter a mesma informação do sinal  $x(t)$ , porém agora em outro domínio, é possível calcular a Transformada Inversa de Fourier. A Transformada Inversa de Fourier, também chamada de equação de síntese e pode ser calculada através da expressão:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(j\omega)e^{j\omega t} d\omega. \quad (2.2)$$

Outras transformadas também foram propostas para resolver os problemas de sinais de tempo discreto, dentre elas a Transformada de Fourier de Tempo Discreto (do inglês, *Discrete Time Fourier Transform*) (DTFT), que corresponde a uma função de uma variável contínua na frequência, e a Transformada Discreta de Fourier, do inglês *Discrete Fourier Transform* (DFT) que consiste em uma função de números complexos discreta na frequência.

### 2.1.2 Transformada Discreta de Fourier

A DFT pode ser calculada a partir da seguinte expressão:

$$X[k] = \sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi}{N}kn} \quad \text{onde } n = 0, 1, 2, \dots, N-1 \quad (2.3)$$

Em que  $N$  é o número de pontos da DFT, o qual deve ser igual ou maior do que o número de pontos do sinal analisado.

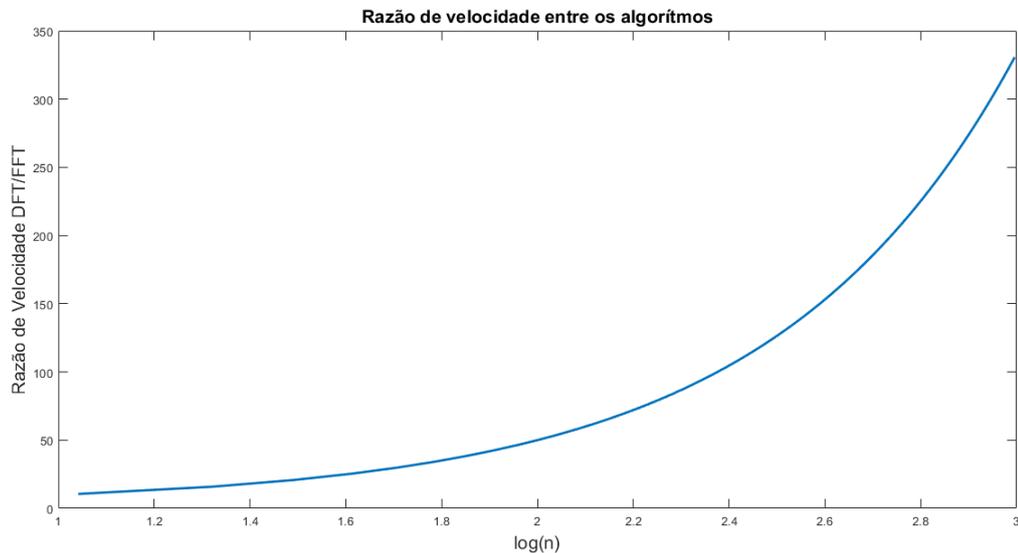
A transformada inversa de Fourier pode ser calculada por

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k]e^{j\frac{2\pi}{N}kn} \quad \text{onde } n = 0, 1, 2, \dots, N-1 \quad (2.4)$$

Se for levado em conta o número de somas e multiplicações complexas, computacionalmente essa não é uma expressão sofisticada, entretanto, ela tem um grau de complexidade computacional  $O(n^2)$ , ou seja, leva-se cerca de  $n^2$  operações para se calcular a DFT pela expressão anterior. Visando otimizar esse processo, foi desenvolvido um algoritmo denominado Transformada Rápida de Fourier, do inglês *Fast Fourier Transform* (FFT), que aproveita os cálculos já realizados explorando a periodicidade para reduzir o número

de operações necessárias para calcular a DFT, reduzindo a complexidade de  $O(n^2)$  para  $O(N\log N)$ . A Figura 9 mostra uma curva que representa a razão de velocidade entre o algoritmo de cálculo direto da DFT e algoritmo FFT.

Figura 9 – Razão entre a velocidade dos algoritmos de DFT e FFT



Fonte: Do Autor

É possível observar que quando o número de pontos é aproximadamente 1000 o algoritmo FFT é quase 350 vezes mais rápido que o algoritmo DFT.

Existem diversos algoritmos do tipo FFT. O primeiro foi proposto por Gauss no século XIX (HEIDEMAN; JOHNSON; BURRUS, 1984). Em 1965, Cooley e Tukey (1965) propuseram um algoritmo de dizimação no tempo para o cálculo da FFT dividindo sempre a sequência  $x[n]$  em duas sequências menores e, assim, sucessivamente, alcançando uma complexidade computacional  $O(\frac{N}{2}\log N)$ .

### 2.1.3 Comentários

Nesta seção foram abordados temas a respeito da análise de sinais a partir das ferramentas de Fourier. Após uma súmula geral sobre o tema, foi apresentada algumas das ferramentas desenvolvidas por Fourier e suas derivadas.

Também foi abordado a respeito dos algoritmos rápidos para o cálculo da DFT, além disso também foi pontuado e comparado a ordem de complexidade dos algoritmos.

## 2.2 A Transformada Wavelet

Nessa seção, serão abordados os principais conceitos referentes à Transformada Wavelet.

### 2.2.1 Introdução

A Transformada de Fourier é uma ferramenta para análise de sinais estacionários, ou seja, sinais cujas componentes de frequência não variam durante o tempo.

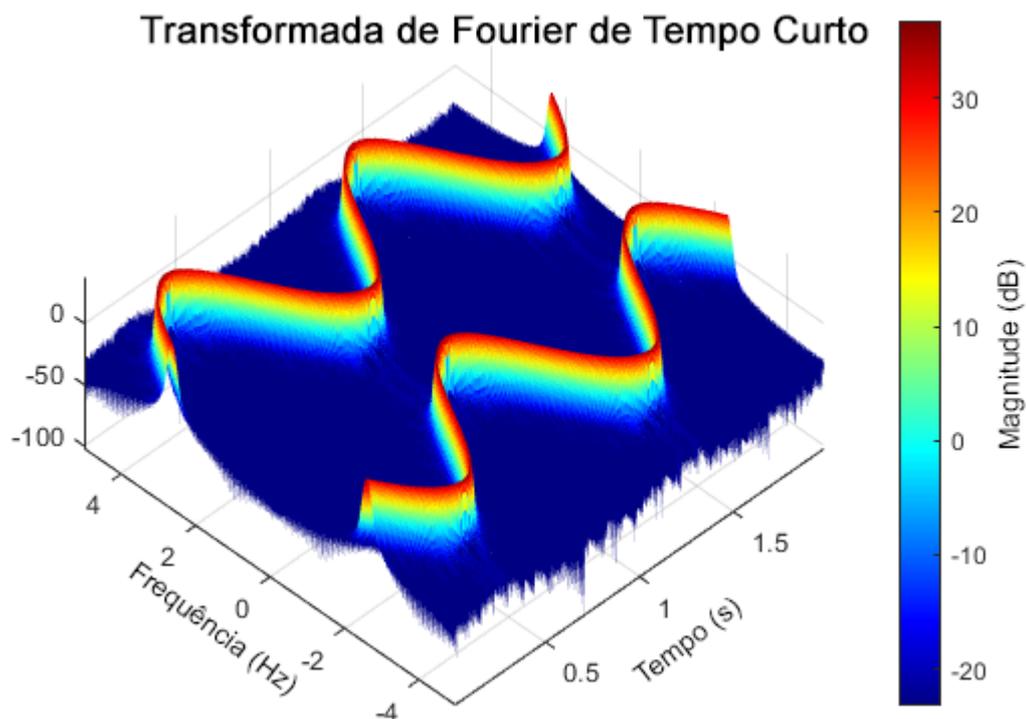
Caso seja preciso analisar um sinal variante no tempo as ferramentas de Fourier descritas até aqui devem ser aplicadas em janelas temporais com tamanho finito, de modo a garantir a estacionariedade do sinal no interior da janela.

Para então, ter uma representação espectral de sinais não estacionários Gabor (1946) propôs a Transformada de Fourier de Tempo Curto, do inglês *Short Time Fourier Transform* (STFT), definida por:

$$X(\tau, w) = \int_{-\infty}^{+\infty} x(t)g(t - \tau)e^{-jw t} dt. \quad (2.5)$$

Como visto, a equação da STFT é muito semelhante a equação da CTFT, porém multiplicada por uma função de janelamento e deslocamento. Gabor propôs calcular a transformada de Fourier para sinais não-estacionários em pequenos intervalos ou janelas, de modo que esses sinais podem ser considerados estacionários dentro de uma janela (BENTLEY; MCDONNELL, 1994). A Figura 10 demonstra uma visualização da STFT, onde as variáveis independentes são tanto tempo quanto frequência.

Figura 10 – Representação gráfica da STFT



Dessa forma, é possível analisar o espectro de frequência de sinais inicialmente não estacionários.

Da mesma forma que na Transformada de Fourier, na prática a STFT também é utilizada para sinais de tempo discreto, conforme a equação abaixo:

$$X(n, w) = \sum_{m=-\infty}^{+\infty} x[m]g[n-m]e^{-jwm}. \quad (2.6)$$

O grande problema da STFT de tempo discreto é um baixo compromisso entre a resolução temporal e a resolução espectral, pois a aproximação de sinal não estacionário para sinal estacionário fica melhor conforme o tamanho da janela diminui e por sua vez a resolução espectral fica melhor conforme o tamanho  $N$  da janela aumenta, ou seja:

$$Res = \frac{f_s}{N}. \quad (2.7)$$

Em que  $f_s$  representa a frequência de amostragem do sinal.

Para melhorar a relação entre a resolução temporal e a resolução espectral uma nova ferramenta hoje conhecida como transformada wavelet foi proposta por Morlet e Grossmann (1984), onde foi sugerido que diferentes funções janelas fossem usadas para diferentes bandas de frequências. Morlet nomeou essas funções base de Wavelets e apesar desse conceito ser introduzido no início dos anos 70, a primeira função base wavelet foi proposta em 1910, pelo matemático Hungaro Alfréd Haar (HAAR, 1910).

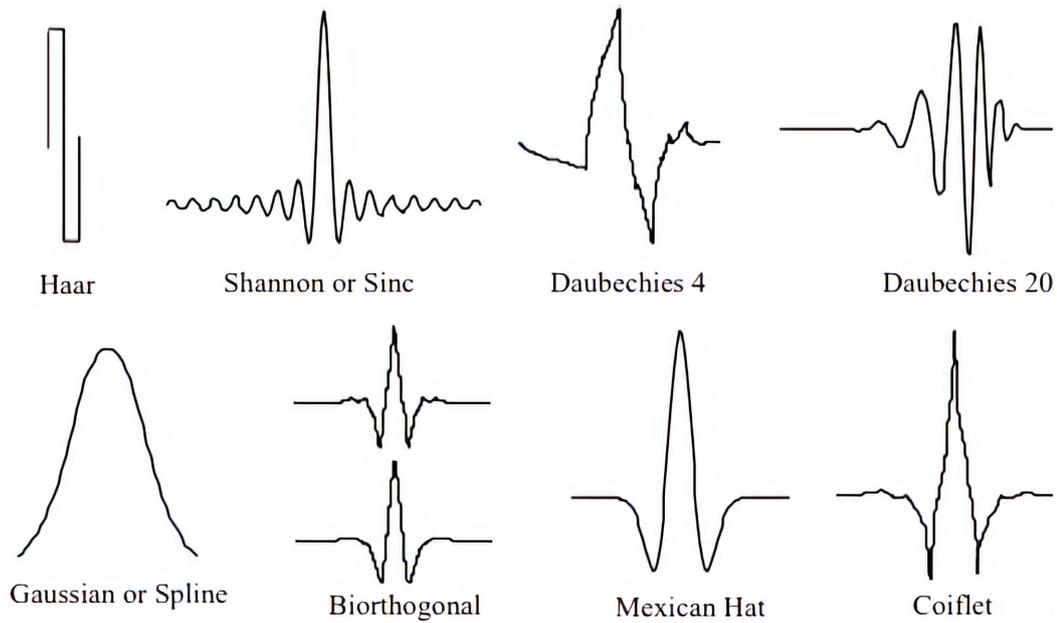
## 2.2.2 A Transformada Wavelet

Diferentemente da STFT que tem como função base assim como todas as ferramentas de Fourier a exponencial complexa, a Transformada Wavelet tem como funções básicas as ondaletas ou wavelets.

As wavelets são funções de duração finita e média zero. Ao oposto das senoides que são ondas com formato regulares e de duração infinita, as wavelets não necessariamente apresentam características simétricas.

As funções base da Transformada Wavelet podem ser separadas em famílias e membros. Dentre as principais famílias, pode-se destacar as famílias Haar, Daubechies, Biorthogonal, Coiflets, Mexican Hat, Meyer etc. A Figura 11 demonstra alguns tipos de Wavelets.

Figura 11 – Exemplos de tipos de Wavelets



Fonte: Adaptado de Fugal (2009)

A Transformada Wavelet de Tempo Contínuo (do inglês, *Continuous Wavelet Transform*) (CWT) é definida por:

$$X(\tau, a) = \int_{-\infty}^{+\infty} x(t)h_{\tau,a}^*(t)dt. \quad (2.8)$$

Onde a função base  $h_{\tau,a}$  é escalonada em frequência pelo fator  $\frac{1}{a}$  e deslocada em tempo pelo fator  $\frac{\tau}{a}$ , definida por:

$$h_{\tau,a}(t) = \frac{1}{\sqrt{a}}h\left(\frac{t-\tau}{a}\right). \quad (2.9)$$

De forma completa podemos descrever a Transformada Wavelet de tempo contínuo como:

$$X(\tau, a) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} x(t)h^*\left(\frac{t-\tau}{a}\right) dt. \quad (2.10)$$

O fator de escala  $\frac{1}{a}$  é conhecido como pseudofrequência. Quando o fator ‘a’ é grande temos uma representação em baixas frequências e grande janela temporal, gerando uma boa resolução espectral para componentes de baixas frequências. Quando o fator ‘a’ é pequeno, temos uma pequena janela temporal e uma resolução espectral um pouco mais baixa para componentes de altas frequências. Contudo, uma baixa resolução espectral em altas frequências não é um grande problema, pois em termos percentuais a resolução ainda será boa.

A Transformada Wavelet de Tempo Contínuo, assim como a Transformada de Fourier é inversível. Para a recuperação do sinal  $x(t)$  basta utilizar a equação da transformada

inversa, definida por:

$$x(t) = \frac{1}{C_h} \int_{-\infty}^{+\infty} \int_0^{+\infty} X(\tau, a) h_{\tau,a}^*(t) \frac{dad\tau}{a^2}. \quad (2.11)$$

Em que,

$$C_h = \int_{-\infty}^{+\infty} \frac{|H(jw)|^2}{|w|} dw < \infty. \quad (2.12)$$

Assim como a Transformada de Fourier necessita de algumas condições para existir a Transformada Wavelet também passa por esse processo, como por exemplo a demonstrada na equação 2.12, outras condições podem ser consultadas em Morettin (1999).

É possível também calcular a Transformada Wavelet Discreta discretizando os valores de escala ‘a’, de deslocamento ‘ $\tau$ ’ e tempo ‘t’.

As pesquisas sobre Wavelets vêm crescendo exponencialmente, e hoje é difícil acompanhar o número de artigos que vêm sendo publicados sobre o assunto em diversas áreas como acústica, ressonância magnética, reconhecimento de voz e fala, turbulência, previsão de terremotos, processamento de imagens etc. Uma revisão mais detalhada pode ser encontrada em (MEYER; RYAN, 1993) (GRAPS, 1995) (HERNÁNDEZ; WEISS, 1996) (MALLAT, 1998).

### 2.2.3 A Transformada Wavelet Discreta

Conforme abordado na subseção anterior a CWT é calculada escalonando e deslocando a sua função base, contudo nesse processo uma grande quantidade de informação redundante é calculada. Com isso, uma solução interessante é a escalonar e deslocar a função base em posições baseadas em potência de 2, essa técnica é conhecida como amostragem baseada em oitavas (NGUYEN, 1996).

Para então discretizar a CWT é necessário discretizar os coeficientes de escala a, de deslocamento  $\tau$  e de tempo t. Amostrando então a CWT em deslocamento “n” e escala “m”. Para amostrar os coeficientes uma escolha usual é a amostragem baseada em oitavas, para esse caso a Transformada Wavelet Discreta (do inglês, *Discrete Wavelet Transform*) (DWT) é definida como (QUATIERI, 2006):

$$X_w(n, m) = \sum_{p=-\infty}^{\infty} x[p] h_{n,m}^*[p]. \quad (2.13)$$

Em que,

$$h_{n,m}[p] = \frac{1}{\sqrt{a_m}} h^*\left(\frac{p - \tau_n}{a_m}\right). \quad (2.14)$$

E na amostragem baseada em oitavas, temos:

$$a_m = 2^m, \quad \text{para } m = 1, 2, 3... \quad (2.15)$$

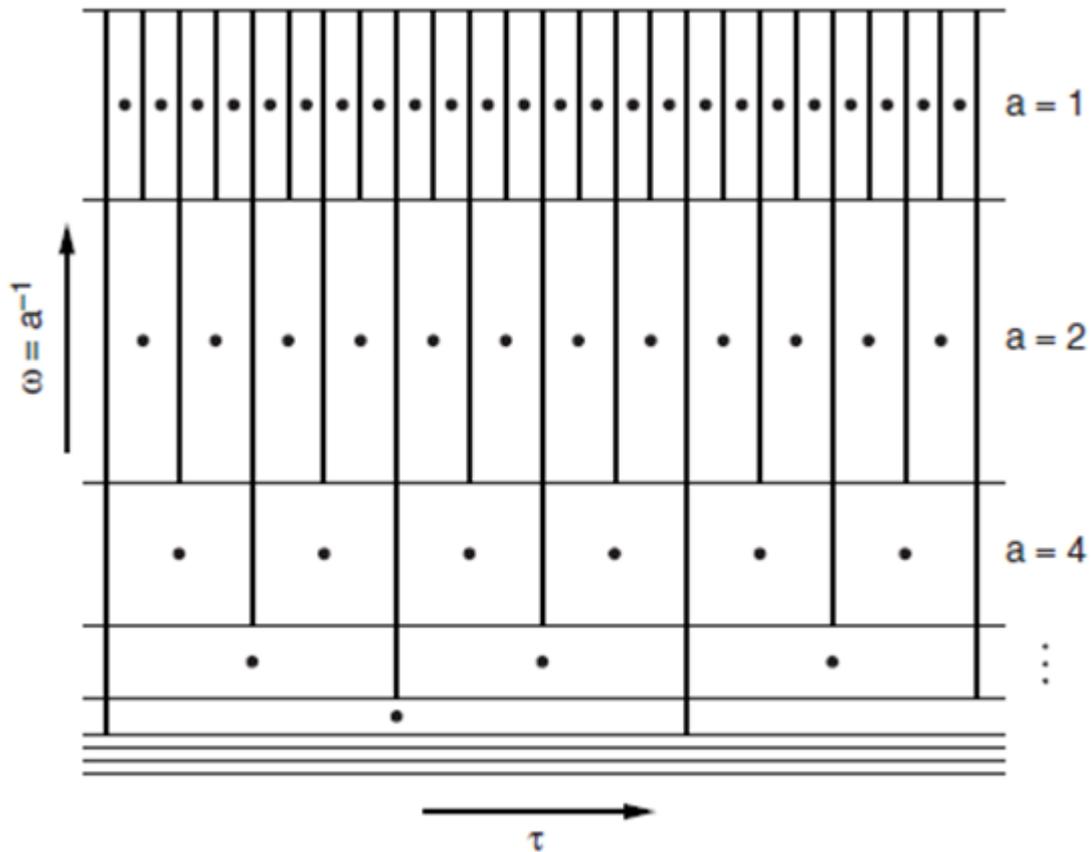
$$\tau_n = na_m, \quad \text{para } n = 1, 2, 3... \quad (2.16)$$

Portanto,

$$h_{n,m}[p] = 2^{-\frac{m}{2}} h\left(\frac{p - n2^m}{2^m}\right). \quad (2.17)$$

A Figura 12 demonstra a técnica de amostragem baseada em oitavas.

Figura 12 – Amostragem baseada em oitavas



Fonte: Adaptado de Quatieri (2006)

### 2.2.4 Comentários

Essa seção teve caráter introdutório, nela foram abordados conceitos básicos sobre o uso da Transformada Wavelet, tais como as características das funções bases, sendo elas: duração finita, média zero e algumas assimetrias. Também foi tratado sobre a melhoria do compromisso entre a resolução temporal e resolução espectral variando o tamanho da janela temporal de acordo com a pseudofrequência, garantindo sempre uma boa resolução percentual. Foi também exemplificado algumas famigeradas famílias e condições de existência da ferramenta.

Também foi abordado a respeito do crescimento dos estudos em Wavelets e as diversas áreas que essa ferramenta tem se mostrando promissora.

E por fim foi tratado o conceito de amostragem baseada em oitavas e a definição da DWT.

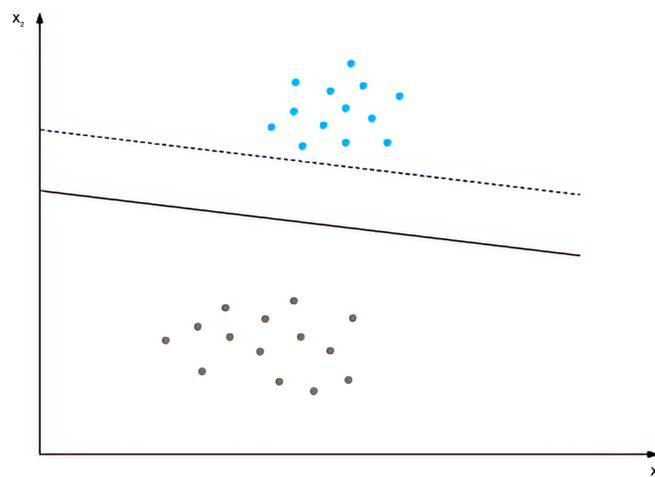
## 2.3 Classificação

A classificação deve ocorrer após a extração de características realizada pela FFT e DWT. Neste sentido, os dados devem ser classificados em dois grupos sendo eles o grupo dos elementos danificados e o grupo dos elementos saudáveis. Para efetuar a classificação serão utilizados três classificadores diferentes, um linear e dois não lineares. Sendo eles, SVM, Rede Neural Artificial (do inglês, *Artificial Neural Network*) (ANN) e Modelo Mistura de Gaussianas (do inglês, *Gaussian Mixture Model*) (GMM), respectivamente.

### 2.3.1 Support Vector Machine

O SVM é um classificador utilizado em problemas binários, quando se deseja classificar um padrão apenas em duas classes, como no nosso caso em trincado ou não trincado. O modo mais fácil de compreender o SVM é através de um sistema bidimensional. Dado um conjunto bidimensional de características como apresentado na Figura 13 é possível traçar algumas retas que dividem essas características (THEODORIDIS; KOUTROUMBAS, 2009).

Figura 13 – Exemplo de separação linear

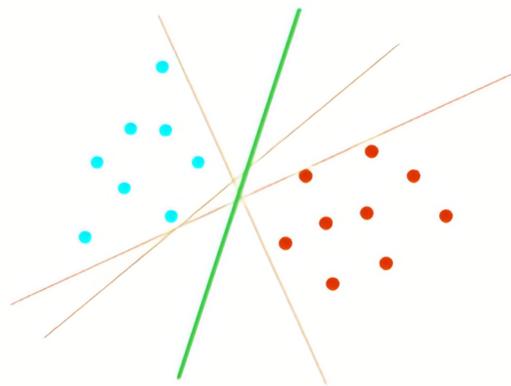


Fonte: Adaptado de Theodoridis e Koutroumbas (2009).

No caso de classificadores lineares de forma em um espaço de características unidimensional defini-se um ponto capaz de separar as duas classes, enquanto para um espaço de características bidimensional determina-se uma reta e, de forma mais geral, para um espaço n-dimensional é necessário se obter um hiperplano. Como mostrado na Figura 14 não necessariamente existirá apenas um único ponto, reta ou hiperplano que solucione o problema, contudo é possível notar que a linha tracejada possui uma confiabilidade menor do que a linha contínua. Isso ocorre, pois, a linha tracejada permite menos espaço

a classe de pontos azuis de se moverem. Uma característica importante dos classificadores é a sua capacidade de classificar elementos que não pertencem ao conjunto de dados de treinamento, ou seja, a generalização do conhecimento adquirido durante a etapa de treinamento. Alguns fatores podem comprometer a qualidade do treinamento e devem ser evitados, tais como a presença de ruídos nos dados, uma quantidade exagerada de parâmetros do modelo que pode provocar o *overfitting*, a insuficiência de dados para caracterizar as classes existentes no problema, dentre outros. O objetivo do SVM é justamente encontrar o hiperplano ótimo para a solução do problema, como vemos a Figura 14 demonstra possíveis hiperplanos e em verde o hiperplano ótimo (JUNIOR, 2010).

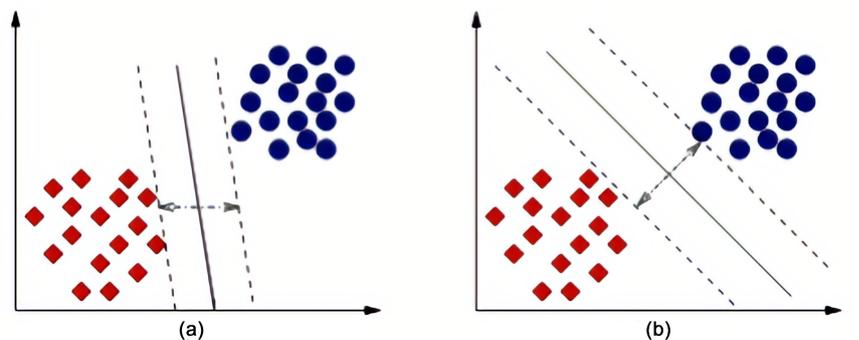
Figura 14 – Possíveis hiperplanos de separação linear



Fonte: Adaptado de Junior (2010).

O SVM traça o hiperplano médio entre as classes de treinamento e cria dois vetores de suporte de modo a encontrar o hiperplano que fornece a maior margem para a classificação, como mostrado na Figura 15.

Figura 15 – a) SVM Margem pequena. b) SVM Margem grande

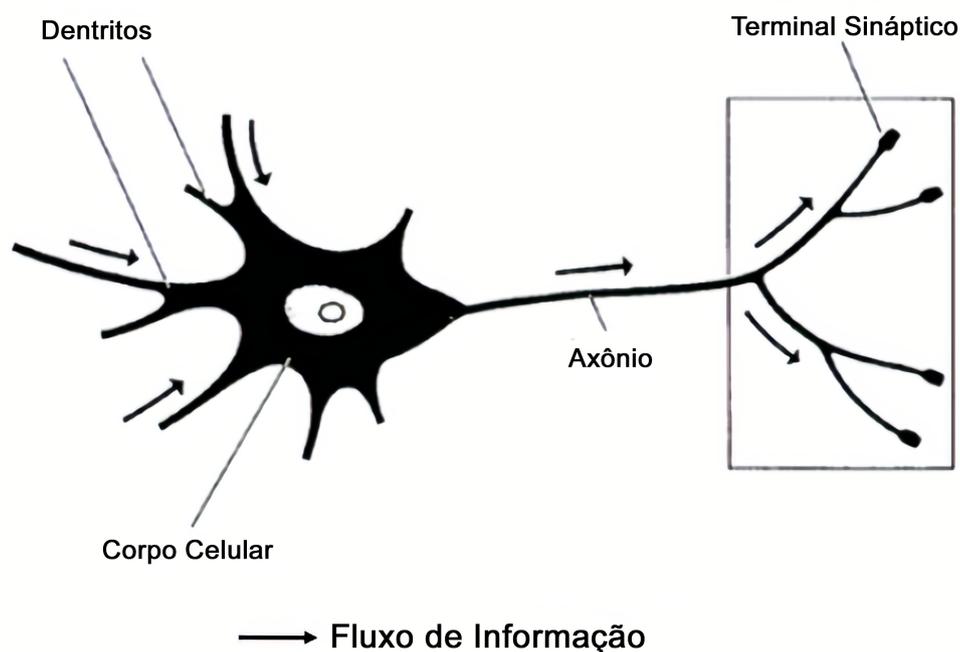


Fonte: Adaptado de Junior (2010).

### 2.3.2 Artificial Neural Network

O cérebro humano possui uma imensa capacidade de aprendizagem, absorção de dados e generalização de padrões, o objetivo da ANN é conseguir reproduzir essas características inerentes ao cérebro humano. É importante visualizar de forma básica o funcionamento do neurônio humano, dado que a ANN visa replicar o mesmo princípio. A Figura 16 apresenta uma ilustração do neurônio humano.

Figura 16 – Neurônio Biológico



Fonte: Adaptado de SIMAS (2019).

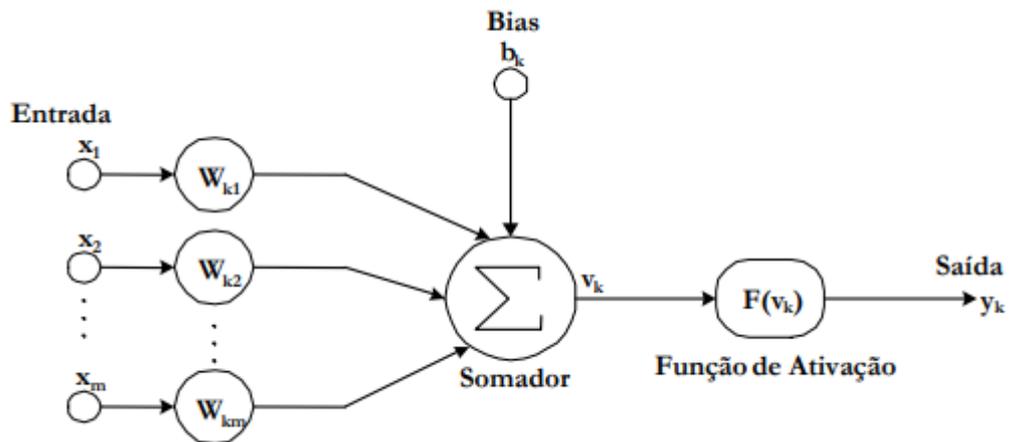
Na Figura 16 pode-se notar que os dendritos conduzem estímulos e/ou informação para dentro do corpo da célula, conhecido como *somma*. O axônio é uma extensa fibra que se ramifica formando a arborização axonal, cuja última terminação tem a finalidade de se conectar com outros neurônios e é denominado terminal sináptico.

Como dito, a ANN visa reproduzir as características do cérebro humano, dentre as quais é importante citar:

- Capacidade de adquirir conhecimento do ambiente a qual a rede está inserida;
- Armazenamento do conhecimento obtido nos pesos sinápticos.

A Figura 17 apresenta o modelo do neurônio artificial.

Figura 17 – Modelo do Neurônio Artificial



Fonte: Retirado de Oliveira (2005).

Na Figura 17 as sinapses são as ligações dos sinais de entrada  $x_m$  ao somador, cada sinapse tem associado um peso  $w_m$  e ao somador é incluído o Bias, a saída é produzida pelo resultado do somador como argumento para a função de ativação (OLIVEIRA, 2005).

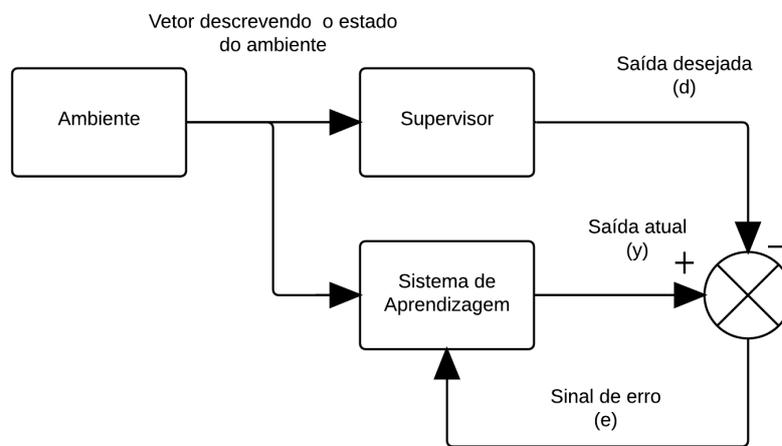
A função de ativação tem como função restringir o valor do saída, existem várias funções de ativação e algumas são apresentadas na Figura 18.



A ANN pode ser de camada única ou múltiplas camadas, os neurônios internos de uma rede de múltiplas camadas são denominados neurônios ocultos.

O método de aprendizagem de máquina utilizado nesse trabalho é o método supervisionado, onde o algoritmo receberá alguns pares de entrada e saída, ou seja, na fase de treinamento do classificador, o mesmo receberá a informação tanto do ambiente ao qual está inserido quanto da resposta correta para aquele caso. A Figura 19 apresenta o diagrama de treinamento supervisionado.

Figura 19 – Diagrama de treinamento supervisionado



Fonte: Do autor.

Como mostrado na Figura 19 o neurônio de saída  $k$ , recebe a informação  $x[n]$  do conjunto de neurônios ocultos que são acionados pelo estímulo denominado vetor de entrada. O neurônio  $k$  produz uma saída  $y_k[n]$  que é comparada com a resposta esperada  $d_k[n]$ , um sinal de erro denominado  $e_k[n]$  é gerado e utilizado para acionar o mecanismo de controle de modo a reajustar os ganhos sinápticos do neurônio de saída  $k$ .

### 2.3.3 Gaussian Mixture Model

O *GMM* é um classificador que se baseia em definir uma função densidade de probabilidade formada por  $K$  distribuições gaussianas ponderadas capazes de tratar o agrupamento dos dados. Pode-se definir a *GMM* como a seguinte função densidade de probabilidade (PORTELA, 2015):

$$f(x_n) = \sum_{k=1}^K \pi_k N(x_n | \mu_k, \Sigma_k). \quad (2.20)$$

Onde  $\pi_k$  são conhecidos como coeficientes de mistura, e serão os responsáveis por fazer as ponderações entre as gaussianas do modelo,  $\pi_k$  é definido como:

$$0 \leq \pi_k \leq 1. \quad (2.21)$$

e,

$$\sum_{k=1}^K \pi_k = 1. \quad (2.22)$$

Onde  $K$  é número de componentes de misturas.

Os parâmetros de cada gaussiana, portanto, serão as médias e as matrizes de covariância. Definindo então:

$$N(x_n | \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{\frac{M}{2}} |\Sigma_k|^{\frac{1}{2}}} e^{-\frac{1}{2}(x_n - \mu_k)^T \Sigma_k^{-1} (x_n - \mu_k)}. \quad (2.23)$$

Portanto, o GMM é definido através de alguns parâmetros que necessitam ser estimados, sendo eles  $\pi_k$ ,  $\mu_k$  e  $\Sigma_k$ , o primeiro caracteriza a ponderação das gaussianas e os dois últimos caracterizam as próprias distribuições. Definindo então  $\Psi = (\pi_k, \mu_k, \Sigma_k)$  agora de forma explícita pode-se escrever:

$$f(x_n | \Psi) = \sum_{k=1}^K \pi_k N(x_n | \mu_k, \Sigma_k). \quad (2.24)$$

Como dito, os parâmetros  $\Psi$  necessitam ser estimados e para este trabalho será usado o método conhecido como Maximização de Esperança (do inglês, *Expectation Maximization*) (EM), este algoritmo trata de forma iterativa a técnica de Máxima Verossimilhança – Estimativa de Máxima Verossimilhança (do inglês, *Maximum Likelihood Estimate*) (MLE), algoritmo com extensa usabilidade em problemas com dados incompletos (COVÕES, 2014).

Após a determinação do modelo, por se tratar de um modelo probabilístico o algoritmo classificará os dados de teste com base na verossimilhança daquele dado sob cada gaussiana.

### 2.3.4 Comentários

Esta seção revisou conceitos a cerca dos classificadores que foram utilizados nesse trabalho. Foram abordados temas como linearidade e método de treinamento dos classificadores, bem com os ideias que são base para o seu funcionamento.

## 3 Metodologia

Nesta seção serão apresentados os métodos aplicados para a realização deste trabalho. A construção do sistema foi realizado em três etapas, sendo elas o projeto do sistema mecânico de movimentação (manipulador), o projeto do sistema eletrônico de controle e leitura de sinais e a classificação realizada de forma *offline* em ambiente MATLAB. Caso o leitor não tenha familiaridade com os equipamentos utilizados neste trabalho, o apêndice A apresenta um breve conceito sobre o assunto, é possível também visualizar os algoritmos implementados no apêndice B e os desenhos computacionais no apêndice C. O sistema foi desenvolvido de acordo com as especificações apresentadas na Tabela 1.

Tabela 1 – Especificações de Projeto.

Parâmetro	Valor	Unidade
Tamanho Total	2 x 1	m
Tamanho Útil/Tamanho Total	> 90	%
Precisão Longitudinal	< 3	mm
Precisão Transversal	< 3	mm
Precisão Vertical	< 1	mm
Frequência de Amostragem do AD	> 1000	Hz
Resolução de Quantização	< 0.1	%
Tempo para Leitura Completa	< 10	minutos
Armazenamento em SD	SIM	–
Taxa de Acerto na Classificação	> 80	%
Custo Total do Projeto	< 4.500,00	R\$

Fonte: Do autor.

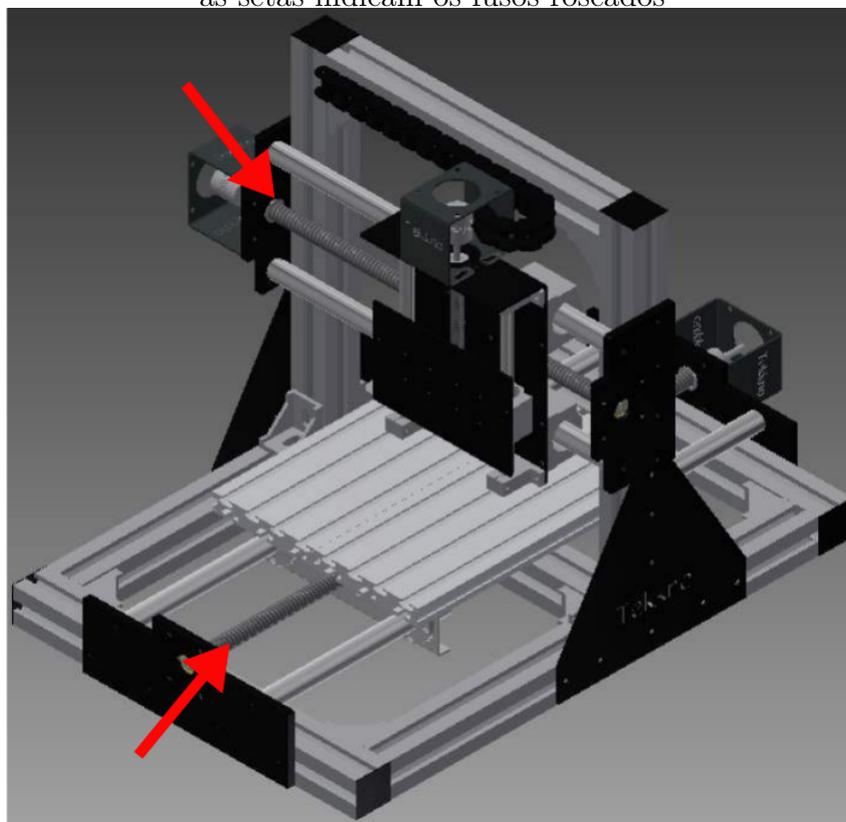
A especificação de Tamanho Útil/Tamanho Total é necessária pois o manipulador não é capaz de transportar o sensor por toda a sua extensão.

### 3.1 Sistema Mecânico

O sistema mecânico é de fundamental importância para o desenvolvimento do trabalho, pois por meio dele ocorrerá a movimentação do sensor durante todo o período de medição. O projeto do manipulador se aproxima muito de um projeto de uma fresadora CNC, contudo para atender os requisitos de precisão e custo foi necessário elaborar métodos alternativos de desenvolvimento. Um projeto comum de CNC tipicamente apresenta um fuso roscado ou parafuso de avanço para movimentação do sistema. Porém, considerando-se o preço e as dimensões do sistema optou-se por elaborar um manipulador com controle de posição utilizando correias dentadas, motores de passo e eixos lisos. A

Figura 20 apresenta uma fresadora CNC com seus fusos roscados sendo indicados pelas setas.

Figura 20 – Desenho Fresadora CNC de fuso roscado  
as setas indicam os fusos roscados



Fonte: Adaptado de (TEKKNO, 2019)

Para a execução do projeto foram adquiridos alguns equipamentos dimensionados de forma a possibilitar o acoplamento entre os objetos. Os equipamentos adquiridos estão descritos na Tabela 2.

O conjunto mancal e rolamento linear é conhecido como *Pillow Block*, e essa nomenclatura será utilizada para descrever o conjunto ao longo do texto. É importante destacar que as barras de 2500mm tem o dobro do raio da barras de 1000mm. Esse dimensionamento ocorreu pois o torque exercido sobre as barras de 2500mm é superior ao exercido sobre a barra de 1000mm e como ambas são feitas do mesmo aço, apresentam as mesmas características de flambagem e ruptura, é necessário uma bitola maior para a barra de 2500mm para garantir a durabilidade do sistema. As figuras a seguir apresentam em conjuntos as peças utilizadas nesse trabalho para a montagem do manipulador. Todos os desenhos realizados foram feitos em escala real através do software Autodesk Inventor 2020 utilizando os desenhos técnicos fornecidos pelos próprios fabricantes.

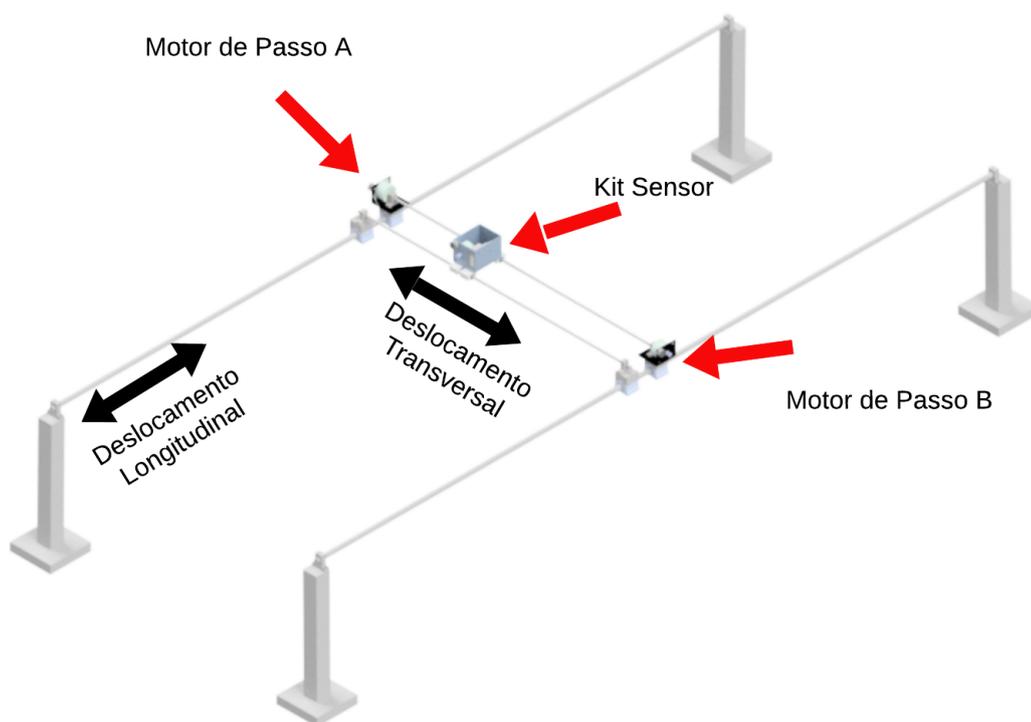
De forma completa, o desenho do manipulador cartesiano apresentado na Figura 21.

Tabela 2 – Componentes Mecânicos Adquiridos.

Descrição	Quantidade
MANCAL PARA ROLAMENTO LINEAR LM 16UU	4
ROLAMENTO LINEAR 16UU	4
MANCAL PARA ROLAMENTO LINEAR LM 8UU	4
ROLAMENTO LINEAR 8UU	4
SUPORTE DE EIXO LINEAR SK 16mm	4
SUPORTE DE EIXO LINEAR SK 8mm	4
EIXO RETIFICADO E CROMADO 16mmx2500mm AÇO 1045	2
EIXO RETIFICADO E CROMADO 8mmx1000mm AÇO 1045	2
POLIA GT2 16 DENTES FURO 5mm	3
8M CORREIA ABERTA GT2 6mm NUCLEO EM AÇO	1

Fonte: Do autor.

Figura 21 – Manipulador



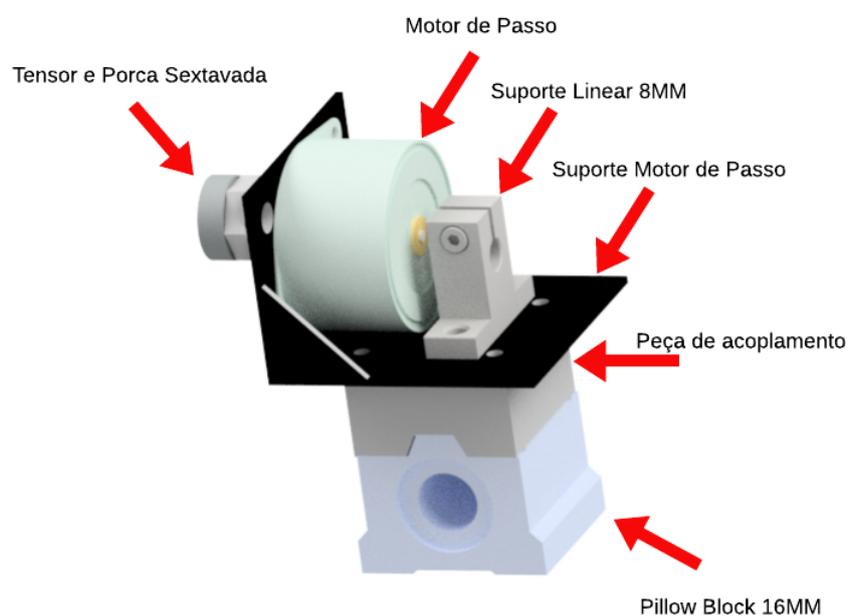
Fonte: Do Autor

A correia dentada adquirida possui 6mm de largura e um passo entre dentes de 2mm, foi escolhida de modo a garantir as especificações de precisão longitudinal e transversal.

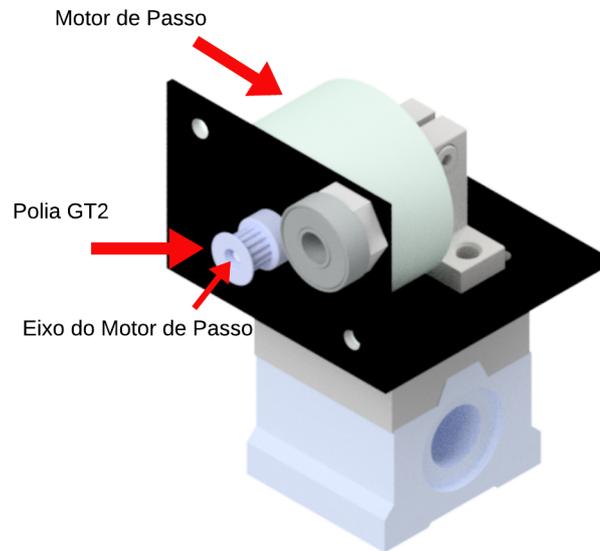
Para fornecer o torque necessário para movimento transversal serão utilizados mo-

tores de passo, o motor escolhido foi (Mitsumi, modelo M49SP-1), e deve estar acoplado ao sistema mecânico (MITSUMI, 2008). Para isso, foi necessário elaborar um suporte para o motor de passo, evitando que ele transmita vibrações ao sistema e garantindo que opere de modo correto. O suporte e o motor foram acoplados ao *Pillow Block* 16mm através de uma peça de acoplamento. Com o intuito de garantir o alinhamento entre o tensor e o eixo de deslizamento da polia, foi utilizado uma porca sextavada de 6mm de largura.

Figura 22 – Conjunto *Pillow Block*, Suporte 8mm e Motor – Vista A



Fonte: Do Autor

Figura 23 – Conjunto *Pillow Block*, Suporte 8mm e Motor – Vista B

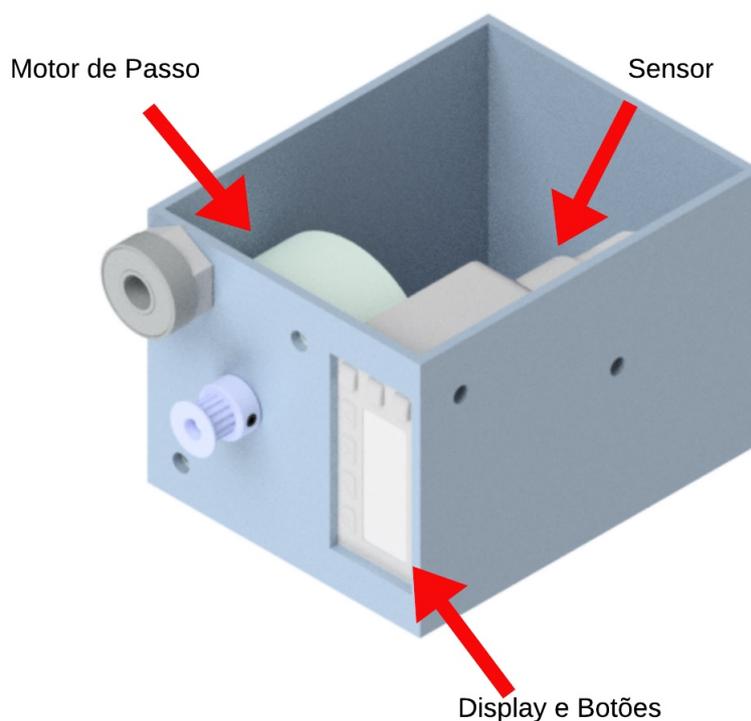
Fonte: Do Autor

Foram utilizados dois conjuntos Pillow Block - Motor de forma paralela no manipulador, pois como o projeto foi feito para um sistema de comprimento elevado, é importante que ambos os lados tenham um motor fornecendo torque, garantindo assim um alinhamento correto.

Para fornecer suporte a outra barra de 8mm é feito um outro conjunto de Pillow block e suporte, porém sem o motor de passo.

O deslocamento longitudinal é feito tal como é feito o transversal, contudo agora basta ser utilizado um único motor, pois tanto o motor quanto o sensor e o circuito eletrônico serão movimentados sob o mesmo suporte que é um único corpo. Denominaremos o conjunto motor, sensor, circuito e suporte como kit sensor, demonstrado na Figura 24.

Figura 24 – Kit Sensor



Fonte: Do Autor

Todos os desenhos técnicos das peças estão disponibilizados no apêndice C.

## 3.2 Sistema Embarcado

Para gerenciar os processos de aquisição de dados do sensor de distância óptico, efetuar o controle de acionamento dos três motores de passo, efetuar a leitura dos sensores e realizar a gravação dos dados no cartão SD, foi desenvolvido um sistema eletrônico embarcado microcontrolado. Para isso foram utilizados os componentes apresentados na Tabela 3.

Tabela 3 – Componentes Eletrônicos Adquiridos.

Descrição	Quantidade
MOTOR DE PASSO BIPOLAR 48 PPR	3
DRV8825	3
CAPACITORES ELETROLÍTICOS 100UF 50V	2
CONVERSORES STEP-DOWN LM2596	2
FONTE CHAVEADA ESTABILIZADA CA/CC 24V/10A	1
ESP32 DEVKIT-V1	1
CHAVES FIM DE CURSO 1A	4
DISCOS DE ENCODER 100 FUIROS	3
LM393	3
RESISTORES DE PRECISÃO 250 $\Omega$	3
ADS1115	1
MÓDULO <i>SD CARD</i>	1
<i>SD CARD</i> 2GB	1
SENSOR DE DISTÂNCIA LASER – ODSL 8/C66-500-S12	1
CABO DE CONEXÃO M12 - 5 VIAS	1
PCI DOIS LAYERS	1
CABOS PARA CHICOTE	-

Fonte: Do autor.

### 3.2.1 ESP32

O principal componente do sistema eletrônico é o microcontrolador ESP32 da Espressif, utilizado como plataforma de processamento para o sistema embarcado proposto.

O modelo de microcontrolador utilizado (ESP32 DEVKITV1) faz parte de um kit de desenvolvimento chamado NodeMCU, opera com tensões de alimentação entre 4,5V e 9V, e possui um sistema de comunicação sem fio Wi-Fi padrão 802.11 b/g/n, que suporta até 5 conexões TCP/IP simultâneas. O módulo também possui interfaces de comunicação SPI e I2C, e pode ser programado através de um ambiente de desenvolvimento (IDE, do inglês Integrated Development Environment) fornecida pelo fabricante Espressif ou através da IDE desenvolvida para a plataforma Arduino. O ESP32 juntamente com o NodeMCU é apresentado na Figura 25.

Figura 25 – Microcontrolador ESP32 DEVKITV1

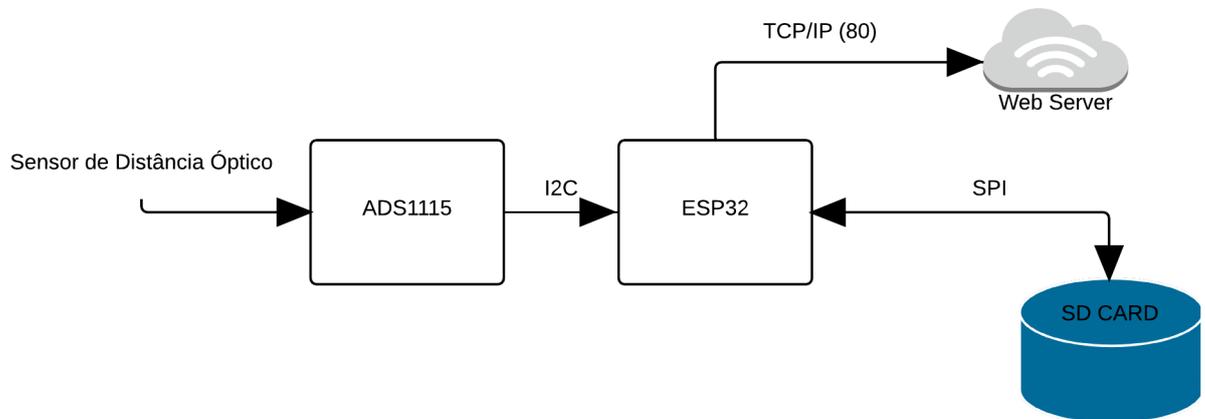


Fonte: Retirado de FILIPEFLOP (2020).

Um módulo de cartões SD, conectado ao barramento SPI, é utilizado para armazenar as informações adquiridas pelo microcontrolador.

Um conversor analógico digital (AD)(Texas Instruments, modelo ADS1115) foi utilizado para adquirir os sinais de saída do sensor óptico. A opção pela utilização de um conversor AD externo deve-se ao fato de sua resolução vertical ser mais alta do que a resolução vertical do conversor AD do micro utilizado (16 bits para o ADS1115, 12bits para o ESP32), embora o conversor AD do ESP32 funcione como redundância no sistema. O barramento I2C foi utilizado para conectar o micro ao conversor AD. A Figura 26 representa o diagrama de comunicação do circuito.

Figura 26 – Diagrama de Comunicação



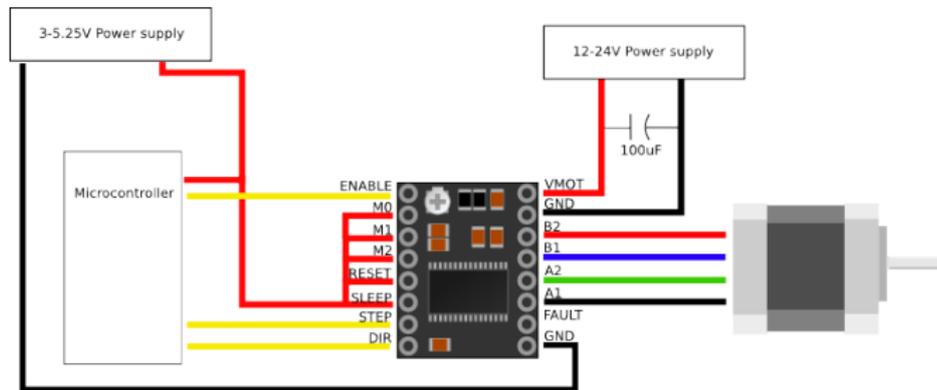
Fonte: Do autor.

Para simplificar o entendimento do circuito elaborado dividiremos ele em três partes, sendo elas acionamentos dos motores, leitura e salvamento de dados e circuito de interrupções externas.

### 3.2.2 Acionamento dos Motores

O acionamento dos motores deve ser feito de forma coordenada, para garantir que o sistema operará de forma consistente. O deslocamento do kit sensor deve ser feito de forma longitudinal ou transversal, não sendo permitida a movimentação simultânea nos dois sentidos. O controle de alimentação das fases dos motores será feito pelo ESP32, que enviará as instruções de comando para o *driver* de acionamento DRV8825.

O DRV8825 possui alimentação lógica entre 3V e 5,25V, e pode alimentar motores com tensões de alimentação entre 12V e 24V, fornecendo correntes de até 2,5 A. A Figura 27 apresenta o diagrama de conexões do conjunto microcontrolador, *driver* e motor.

Figura 27 – Diagrama de conexão do *driver*

Fonte: Retirado de Bart Meijer (2013)

O driver possui 6 modos de operação, que podem ser determinados através dos pinos de controle M0, M1 e M2. A configuração dos pinos de controle determinará a resolução do passo do motor, como apresentado na tabela 4.

Tabela 4 – Modos de operação DRV8825.

M0	M1	M2	Resolução
0	0	0	<i>FULL STEP</i>
1	0	0	<i>HALF STEP</i>
0	1	0	$\frac{1}{4}$ <i>STEP</i>
1	1	0	$\frac{1}{8}$ <i>STEP</i>
0	0	1	$\frac{1}{16}$ <i>STEP</i>
1	0	1	$\frac{1}{32}$ <i>STEP</i>
0	1	1	$\frac{1}{32}$ <i>STEP</i>
1	1	1	$\frac{1}{32}$ <i>STEP</i>

Fonte: Do autor.

A Figura 21 mostra como os motores de passo estarão alocados no manipulador, bem como demonstra quais serão os seus respectivos sentidos de deslocamento.

Para reduzir os custos do projeto, os motores de passo utilizados no desenvolvimento desse sistema foram reutilizados de impressoras em desuso. A desvantagem de se empregar motores usados é que nem sempre as características físicas são iguais, de modo que é necessário adaptar o circuito de acionamento de forma individual. Para controlar então os motores com eficiência, algumas premissas foram consideradas:

- Os motores longitudinais devem trabalhar em sincronismo, movimentando em mesma direção e velocidade iguais;

- Para alcançar velocidades altas de rotação dos motores é importante desenvolver um controle de velocidade de partida, pois alta variação na inércia de uma estrutura tão grande pode provocar problemas mecânicos e redução de vida útil do equipamento;
- Os motores devem conseguir fornecer torque suficiente para gerar movimento;
- É necessário calibrar o sistema, para que seja possível relacionar o número de pulsos por volta do motor com a distância linear percorrida pelo kit sensor;
- Velocidades de rotação baixas demais podem saturar a corrente no enrolamento e diminuir a vida útil dos motores.

Os motores longitudinais utilizados possuem o mesmo passo ( $7,5^\circ$ ), de modo que para garantir o sincronismo entre eles, foi necessário que os dois *drivers* dos motores longitudinais recebessem os mesmos pulsos de controle. Como apresentado na Figura 21, os motores de passo que controlam o movimento longitudinal foram posicionados em lados opostos do kit sensor. Dessa forma, para garantir que o kit se deslocasse de maneira correta, foi necessário ajustar o sentido de rotação dos motores de maneira complementar, ou seja, enquanto um deles gira no sentido horário, o outro gira no sentido anti-horário. Para resolver esse problema, optou-se por inverter uma fase de um dos motores de passo.

Aumentar a velocidade de rotação dos motores torna todo o processo de leitura ligeiramente mais rápido, mas em contra partida, ocasiona perda em torque e requer um controle de partida, então buscou-se encontrar velocidades mais baixas de rotação.

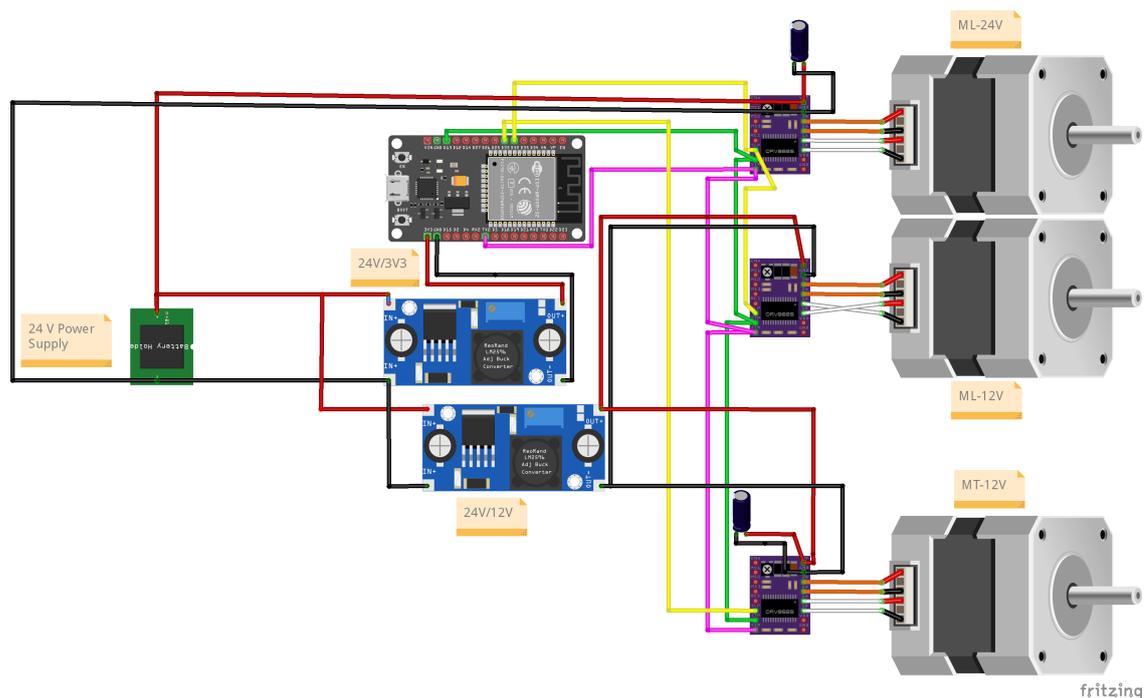
Empiricamente, as velocidades mais adequadas de rotação para os motores encontradas foram de 250 RPM longitudinalmente e 180 RPM transversalmente. Infelizmente um dos motores longitudinais em condição nominal de tensão (24V) não conseguia produzir torque em 250 RPM, e para solucionar esse problema foi necessário reduzir a tensão de operação dele para 12V.

As polias acopladas ao eixo dos motores possuem 16 dentes espaçados de 2mm cada, portanto uma revolução do eixo equivale a um deslocamento de 32mm. Todos os motores possuem resolução de passo de  $7,5^\circ$ , equivalente a 48 passos por revolução. O nosso passo mínimo é limitado pela resolução dos dentes da polia (2mm), e a relação de passos do motor equivalente a essa resolução pode ser definido como:

$$x = \frac{48(\text{passos})}{16(\text{dentes})} = 3(\text{passos/dente}). \quad (3.1)$$

A fonte de alimentação do sistema possui 24V, porém 2 motores (um longitudinal e um transversal) operam em 12V. Para efetuar a conversão desse nível de tensão (24V/12V) foi utilizado um conversor *step-down* CC/CC LM2596 com regulador de tensão. A Figura 28 representa o esquemático do circuito utilizado no acionamento dos motores.

Figura 28 – Circuito de Acionamento dos Motores

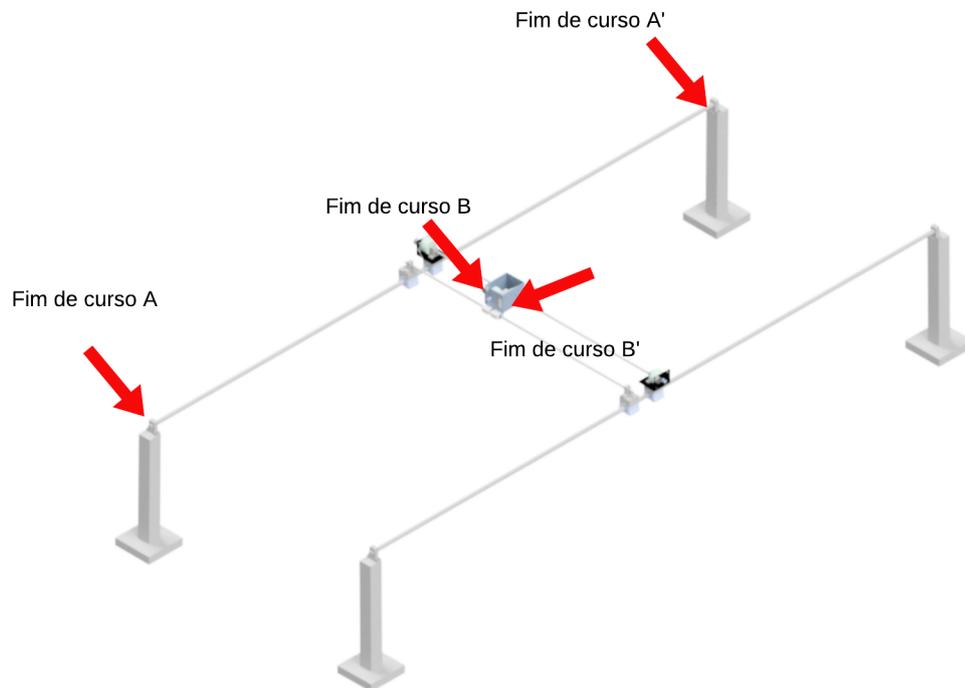


Fonte: Do Autor.

### 3.2.3 Sensores

O sistema completo é composto por dois tipos de sensores: fim de curso e o mediador de distância óptico. O fim de curso é posicionado de modo a garantir que os motores de passo parem de produzir torque quando os eixos atingirem o limite máximo de deslocamento, como mostrado na Figura 29.

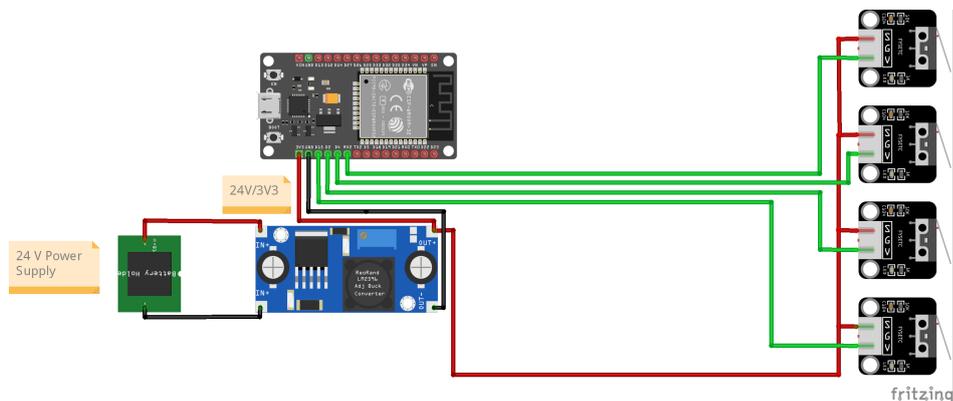
Figura 29 – Posicionamento dos fins de curso no manipulador



Fonte: Do autor.

Os fins de curso são utilizados tanto para zerar a posição do sistema ao início da leitura quanto para impedir que qualquer erro possa acarretar em danos na estrutura. Sempre que o fim de curso é acionado, uma interrupção externa é habilitada desligando os motores e os fazendo rotacionar 30 mm em sentido oposto para garantir que nenhuma pressão continue a ocorrer entre os componentes mecânicos. A Figura 30 representa a porção do circuito referente aos sensores que acionam interrupções externas.

Figura 30 – Circuito de Interrupções Externas



Fonte: Do Autor.

O segundo sensor que será utilizado neste trabalho é o sensor de distância óptico ODSL8, ele ficará responsável por efetuar a medição da terceira dimensão do sistema, recomenda-se que o leitor leia o tópico do sobre o ODSL8 disponibilizado no apêndice A. A Figura 31 apresenta uma foto do sensor escolhido.

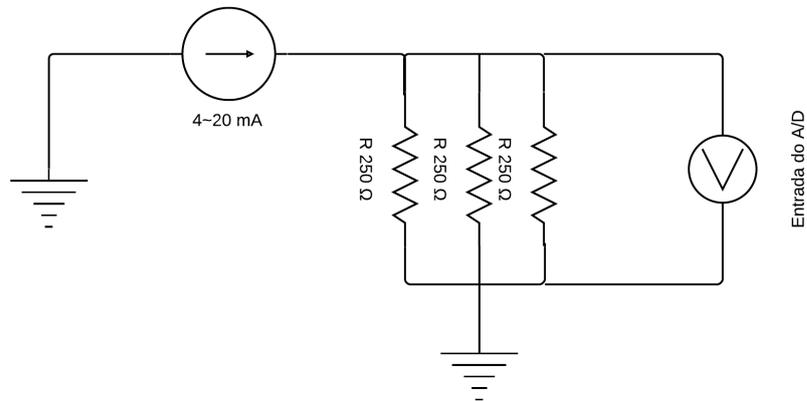
Figura 31 – Sensor ODSL8



Fonte: Retirado de Leuze electronic (2018).

O sensor utilizado apresenta em sua saída uma corrente entre 4mA e 20mA, proporcional a uma leitura de distância entre 20mm e 500mm. Para adquirir os sinais, utilizou-se o conversor analógico-digital (AD) integrado ao microcontrolador ESP32. Esse conversor possui uma tensão de fundo de escala de 1,66V e uma resolução vertical de 12 bits. Para converter os sinais de saída de corrente do sensor em tensão, utilizou-se um resistor de precisão de 83,3  $\Omega$ , montado a partir de uma associação em paralelo de 3 resistores de 250  $\Omega$ . O circuito utilizado é apresentado na Figura 32.

Figura 32 – Circuito de conversão

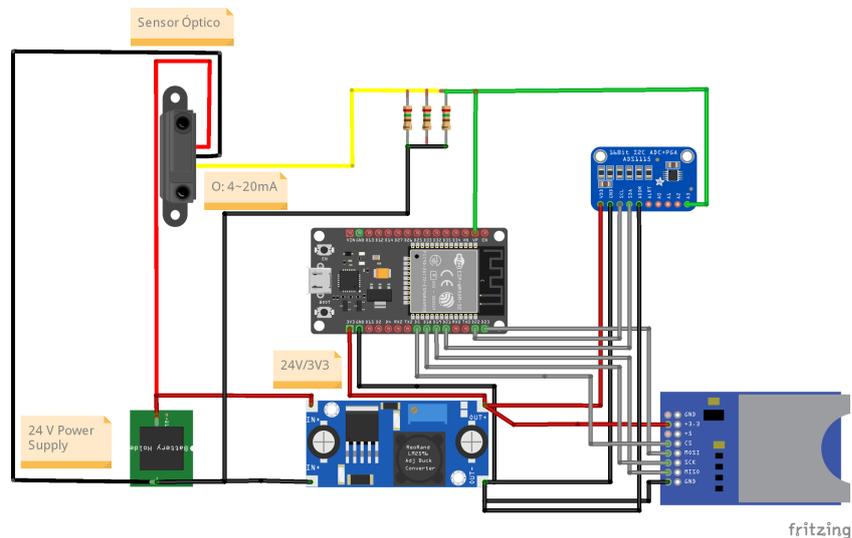


Fonte: Do autor.

### 3.2.4 Leitura e Salvamento dos Dados

A porção do circuito relativa a leitura e salvamento dos dados é apresentada na Figura 33.

Figura 33 – Circuito de Acionamento dos Motores



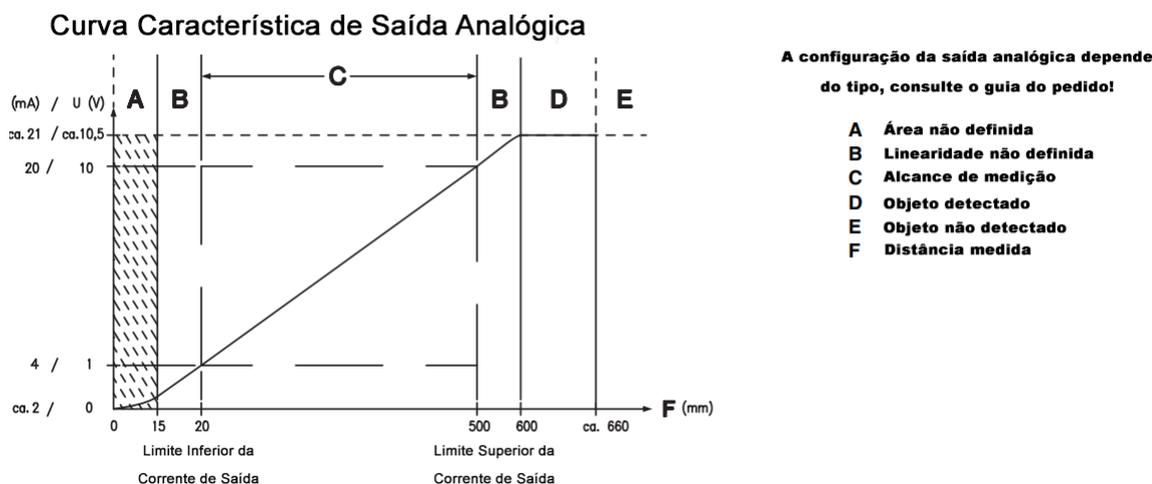
Fonte: Do Autor.

Com exceção do sensor de distância óptico que tem sua alimentação em uma faixa de 18 a 30V, todos os outros elementos desse circuito possuem alimentação em 3,3V ou 5V. Dessa forma, o sensor de distância óptico é alimentado diretamente pela fonte de 24V enquanto o restante do circuito recebe a alimentação da saída de um outro LM2596 que

realiza a conversão de 24V para 3,3v. Optou-se por 3,3V ao invés de 5V pois embora o ESP32 possa ser alimentado com 5V o restante dos seus pinos operam em 3,3V.

O sensor de distância óptica fornecer um valor de corrente relativo à distância medida, conforme Figura 34.

Figura 34 – Características de Entrada e Saída do Sensor



Fonte: Adaptado de Leuze electronic (2018).

Os três resistores de precisão em paralelo fornecem uma resistência equivalente de  $83,33 \Omega$  e conseqüentemente um fundo de escala de tensão de 1,66V. O uso do ADS1115 permite uma boa resolução de leitura dado que ele efetua a amostragem com 16 bits (sendo 1 de sinal e 15 de dados) e possui ganho de entrada ajustável. Como o fundo de escala do sistema é 1,66V, o ganho de entrada foi ajustado para 2x podendo efetuar leitura de até 2,048V. A comunicação dos dados entre o módulo A/D e o microcontrolador é feito via I2C, um pino do microcontrolador também é conectado a saída do sensor para garantir redundância em caso de problemas futuros com o ADS1115.

Após a aquisição, os dados são salvos no cartão de memória acoplado no módulo SD CARD, que se comunica com o microcontrolador via SPI. Os dados serão salvos em um arquivo de tabela nomeado como `_datalog_xxxxx.csv`, onde “xxxxx” é um numeral que se inicia em 1 e acresce de um em um conforme o sistema efetua novas leituras.

### 3.3 Interface Homem/Máquina

Para facilitar a usabilidade do sistema, foi implementado um sistema de comunicação usuário/máquina. O sistema funciona com base em um *Web Server* configurado no próprio microcontrolador, e para acessar o *Web Server* é necessário conectar-se à rede wi-fi do ESP32.

Embora o ESP32 conte com o protocolo wi-fi 802.11 b/g/n, ele não possui a potência de um roteador de uso comercial. Dessa forma, para minimizar as perdas de pacotes TCP é recomendado que se esteja no mesmo ambiente do manipulador ao tentar controlá-lo. Além disso, para diminuir as interferências com redes locais próximas, a rede wi-fi do ESP32 teve seu canal escolhido levando em conta as disponibilidades de canais em desuso nas proximidades do laboratório.

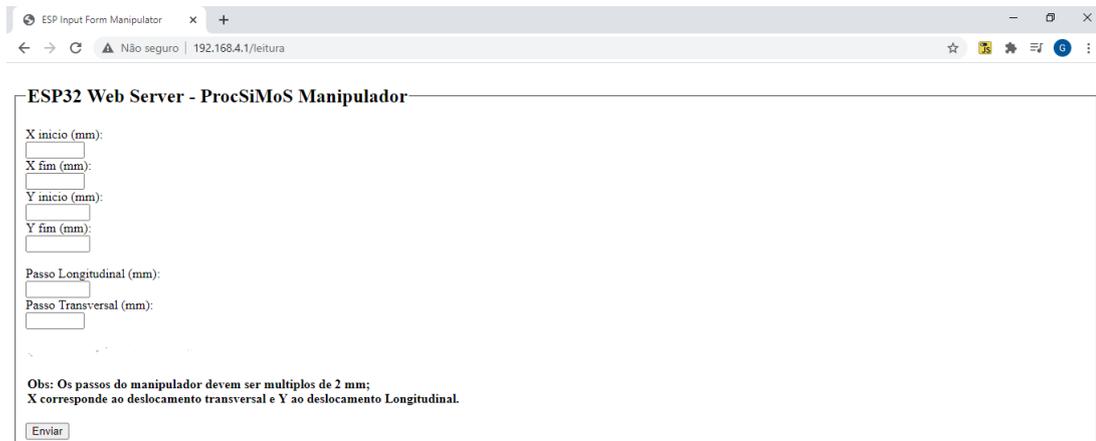
A interface de utilização do *Scanner 3D* possui duas funcionalidades principais, sendo elas efetuar uma nova leitura e fazer o *download* de um arquivo anteriormente lido. A página inicial do *Web Server* é demonstrada na Figura 35.

Figura 35 – Página Inicial do *Web Server*



Fonte: Do Autor.

Na página de nova leitura o usuário pode especificar a área que o usuário deseja escanear, bem como com quais passos (longitudinal e transversal) serão utilizados entre uma leitura e outra. A página foi toda elaborada em HTML5 e os dados são enviados para o *Web Server* via método *GET*. A Figura 36 demonstra a página de nova leitura.

Figura 36 – Página de Nova Leitura do *Web Server*

ESP32 Web Server - ProcSiMoS Manipulador

X inicio (mm):

X fim (mm):

Y inicio (mm):

Y fim (mm):

Passo Longitudinal (mm):

Passo Transversal (mm):

Obs: Os passos do manipulador devem ser mltiplos de 2 mm;  
X corresponde ao deslocamento transversal e Y ao deslocamento Longitudinal.

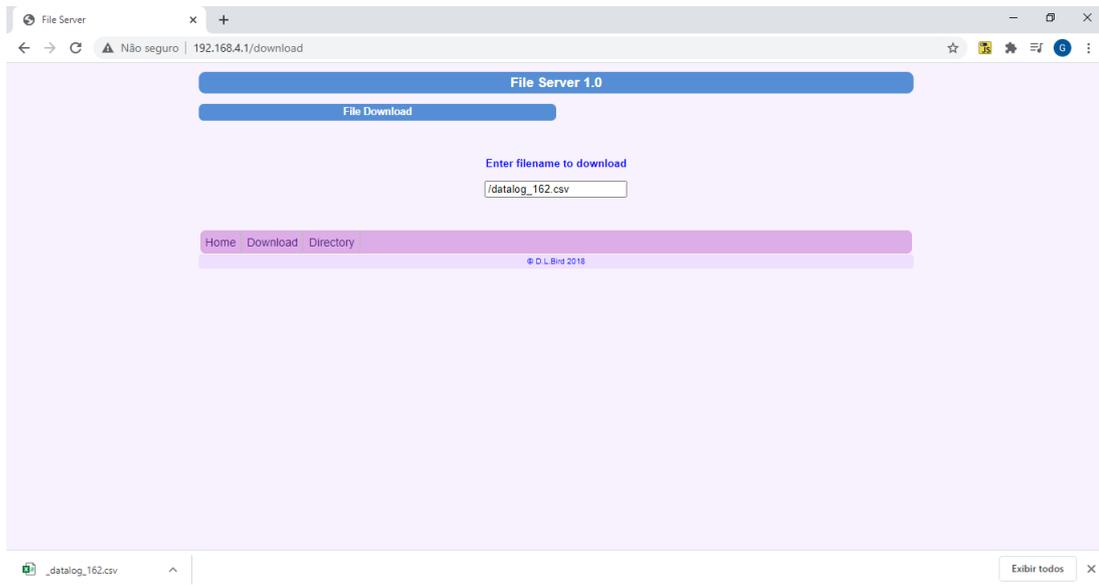
Fonte: Do Autor.

Na página de *Download* o usuário tem acesso ao diretório de dados do *SD CARD* e pode realizar o *download* de um arquivo já existente. A Figura 37 demonstra a página de diretório de dados e a Figura 38 demonstra a página de *download* de um arquivo.

Figura 37 – Página de Diretório de Arquivos do *Web Server*

Name/Type	Type File/Dir	File Size
Dir	/System Volume Information	
/System Volume Information/WPSettings.dat	File	12 B
/System Volume Information/IndexerVolumeGuid	File	76 B
/test.txt	File	18 B
/test.csv	File	23 B
/datalog_1.csv	File	0 B
/datalog_2.csv	File	0 B
/datalog_3.csv	File	0 B
/datalog_4.csv	File	0 B
/datalog_5.csv	File	0 B
/datalog_6.csv	File	0 B
/datalog_7.csv	File	0 B
/datalog_8.csv	File	0 B
/datalog_9.csv	File	0 B
/datalog_10.csv	File	0 B
/datalog_11.csv	File	0 B
/datalog_12.csv	File	0 B
/datalog_13.csv	File	0 B
/configuracao.csv	File	0 B
/datalog_14.csv	File	0 B

Fonte: Do Autor.

Figura 38 – Página de *Download* de Arquivos do *Web Server*

Fonte: Do Autor.

### 3.4 Classificação

O sistema de classificação foi implementado utilizando o Matlab. Os algoritmos foram desenvolvidos utilizando a *Wavelet Toolbox* para extração de características e *Statistics and Machine Learning Toolbox* para treinamento e validação dos classificadores.

A *Wavelet Toolbox* proporciona ferramentas de análise e síntese de sinais e imagens, incluindo algoritmos para os cálculos de coeficientes de aproximações e detalhes que são o foco desse trabalho.

A *Statistics and Machine Learning Toolbox* provê funções e aplicações para descrever, analisar e modelar dados, incluindo ferramentas de redução de dimensionalidade, análise de componentes principais, métodos de seleção de características e algoritmos de treinamento de classificadores.

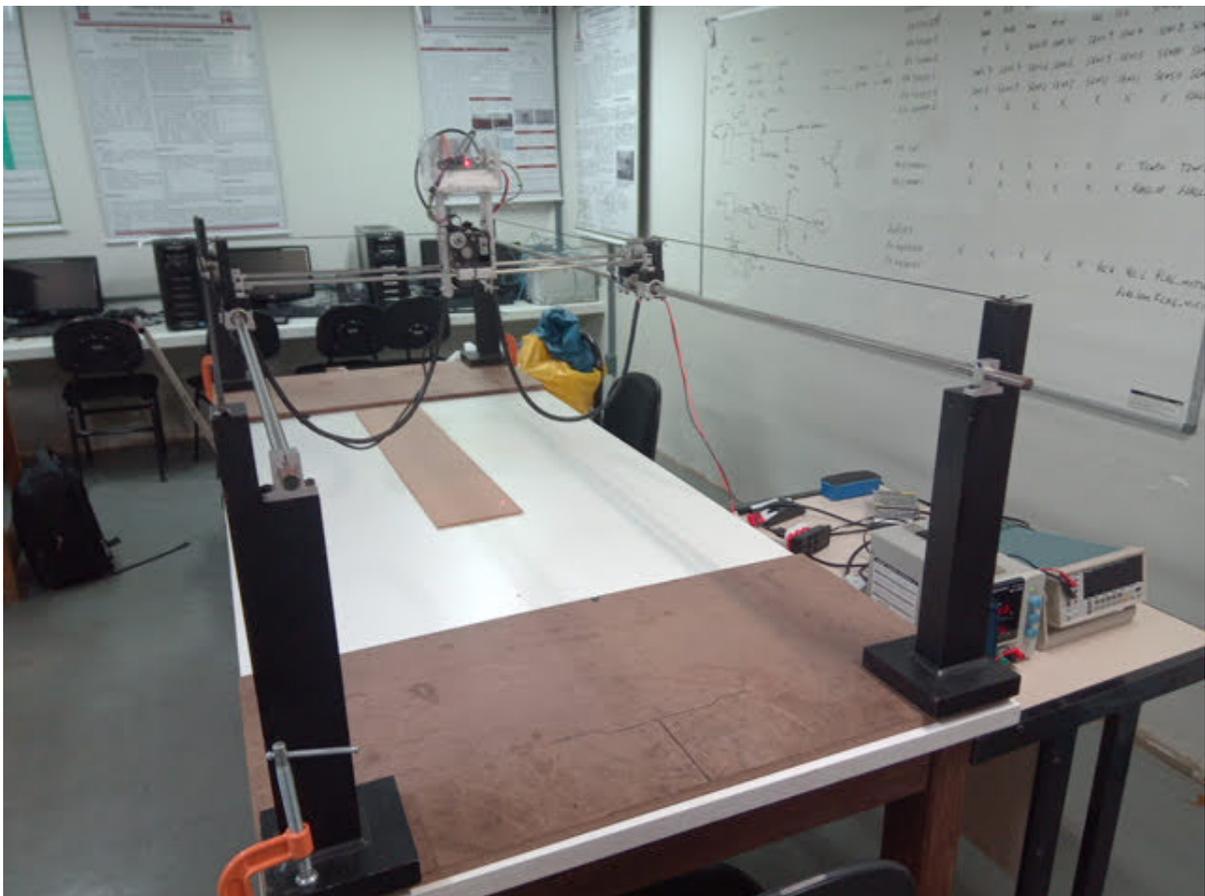
A metodologia empregada para o desenvolvimento do sistema de classificação foi testar combinações diferentes entre extrações de características, classificadores e parâmetros do classificador, com o intuito de encontrar a sequência de atributos que alcance o melhor desempenho para a situação.

## 4 Resultados

### 4.1 Sistema Mecânico

Como anteriormente descrito a primeira etapa deste trabalho consiste no desenvolvimento de um manipulador mecânico que possa ser utilizado para transportar o Kit Sensor. Para a construção do manipulador foi necessário a usinagem de algumas peças descritas na seção 3.1 e também de algumas peças anteriormente não previstas. A Figura 39 apresenta o equipamento desenvolvido.

Figura 39 – Escâner Completo



Fonte: Do Autor.

Para que o deslocamento transversal e longitudinal possa ocorrer é necessário que haja paralelismo entre os eixos que passam pelos rolamentos. O deslocamento longitudinal tem seu paralelismo garantido pois os pés foram alocados na mesa de forma paralela e são fixados por morsas de bancada, além de contarem com seu próprio elevado peso.

Os testes práticos demonstraram grande perda de paralelismo entre os eixos transversais sempre que qualquer tipo de deslocamento ocorria, tanto transversal quanto longitudinal. Esse foi um problema não notado no projeto mecânico e solucionado com a usinagem de uma peça que pudesse fornecer acoplamento entre os dois Pillow Block's 16 mm.

Para que a correia possa ser tensionada e tenha bastante contato com os dentes da polia, foi necessário o uso de um rolamento tensor que no projeto era apoiado por uma porca sextavada de 6mm, conforme Figura 22. Contudo, na prática a porca sextavada era muito extensa e impedia o tensor de girar. Para solucionar isso foram usinados pequenos eixos vazados de bronze que servem como apoio para o rolamento tensor.

Os Pillow's Block 16mm que contém o suporte dos motores longitudinais, recebe graças a espessura dos suportes dos motores uma elevação de 2 mm no suporte linear 8mm sobre eles. Essa elevação causa uma diferença de paralelismo entre a parte frontal e traseira do kit sensor e foi solucionada utilizando duas arruelas de 1mm como apoio nos Pillow's Blocks que não suspendem motores.

Por fim, o último problema de paralelismo que ocorreu entre o projeto e a fabricação ocorreu no Kit Sensor. Os furos para encaixe dos Pillow Block's 8mm deveriam ser feitos de forma perfeitamente linear e paralela, durante a fabricação das peças os furos tiveram leves variações que impediam o Kit Sensor de se movimentar sempre que os parafusos eram fortemente apertados. A solução encontrada para esse problema foi manter alguns Pillow Block's com parafusos levemente frouxos.

Os últimos ajustes mecânicos realizados não projetados foram referentes a estabilidade da correia. Mesmo que o projeto e usinagem das peças tenham sido feitas de forma a garantir bom alinhamento entre os furos das correias, devido as grandes dimensões do sistema os testes práticos demonstraram que as correias por diversas vezes escorregavam para fora do rolamento tensor, impedindo a movimentação da estrutura. Para solucionar isso foi necessária usinar uma pequena peça de nylon que serviu como um revestimento para o tensor e guia para a correia.

As polias utilizadas nesse projeto são fabricadas no padrão dos motores NEMA 17 e possuem parafusos Allen M1,5 utilizados para fornecer pressão entre a polia e o eixo do motor. Entretanto, os motores utilizados nesse projeto foram reutilizados de diversas impressoras e não possuem o eixo no mesmo padrão das polias. Com isso a pressão exercida pelos parafusos não era suficiente para garantir acoplamento entre o eixo do motor e a polia, fazendo com que o eixo do motor girasse de forma livre dentro do furo da polia. A solução para esse problema foi a utilização de uma cola trava rosca alto toque no eixo do motor.

## 4.2 Classificação - Detecção de Trincas em Barras Metálicas

O objetivo deste trabalho é detectar trincas em barras metálicas a partir da leitura superficial das mesmas. As barras metálicas utilizadas nesse procedimento possuem cerca de 350 mm de largura e 2300 mm de comprimento. A partir do sistema supervisorio descrito anteriormente, o escâner foi ajustado para realizar leituras das peças metálicas, seguindo as especificações apresentadas na Tabela 5.

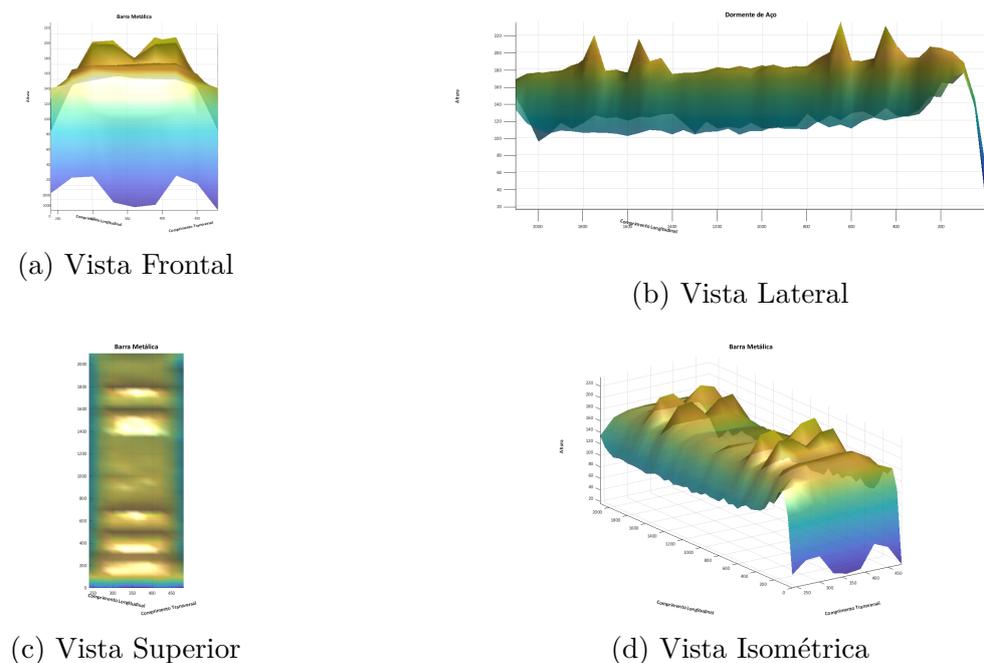
Tabela 5 – Configuração de Leitura Solicitada.

Configuração de leitura	
X início	150 mm
X fim	550 mm
Y início	0 mm
Y fim	2100 mm
Passo Longitudinal	50 mm
Passo Transversal	30mm

Fonte: Do autor.

Essa configuração define um *grid* de 14 x 43 aquisições, totalizando 602 medições de distância. As Figura 40 apresenta o *surface plot* de uma das barras metálicas lidas pelo Escâner 3D.

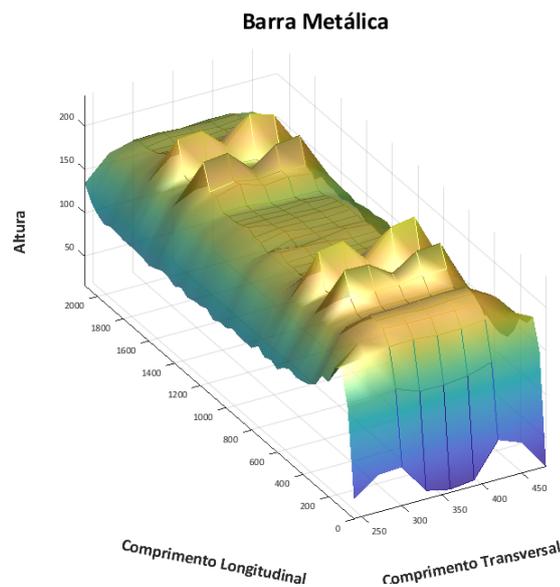
Figura 40 – Barra Metálica



Fonte: Do Autor.

A base de dados gerada contou com 10 leituras de barras metálicas sendo 5 delas em bom estado de uso e 5 contendo trincas em uma ou duas extremidades. A abordagem proposta utiliza as curvas superiores das barras metálicas para detectar as trincas, como demonstrado na Figura 41. Com o passo transversal utilizado de 30 mm foi possível detectar 5 dessas curvas por barra metálica, portanto a base de dados final possuiu 50 observações.

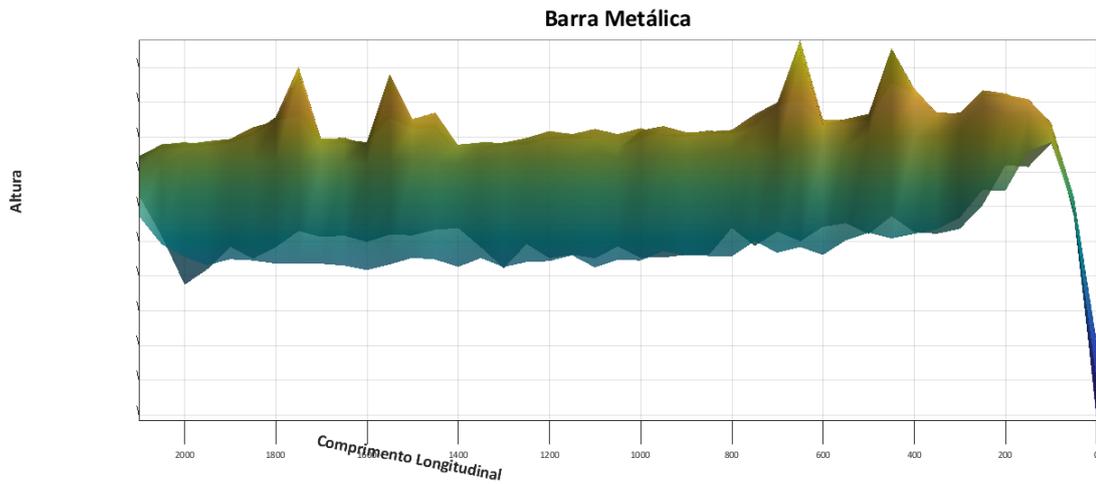
Figura 41 – Curvas Utilizadas da Barra Metálica



Fonte: Do Autor.

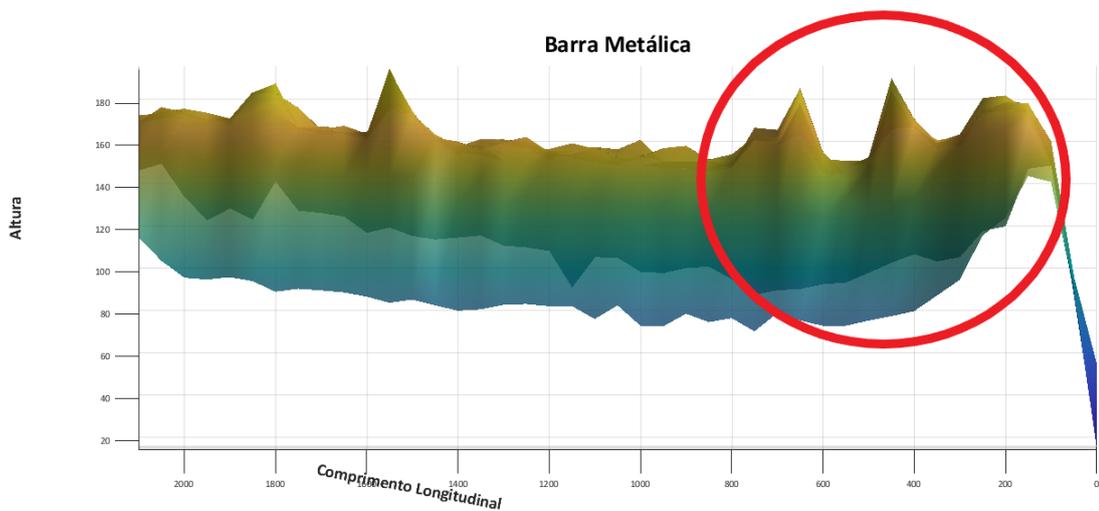
As Figuras 42 e 43 apresentam, respectivamente, duas barras metálicas escanêdas sendo a primeira uma barra metálica saudável e a segunda uma barra metálica defeituosa. É possível notar na Figura 43 uma curva mais acentuada ocasionada pela presença da trinca e destacada por um círculo vermelho.

Figura 42 – Barra Metálica Saudável



Fonte: Do Autor.

Figura 43 – Barra Metálica Defeituosa



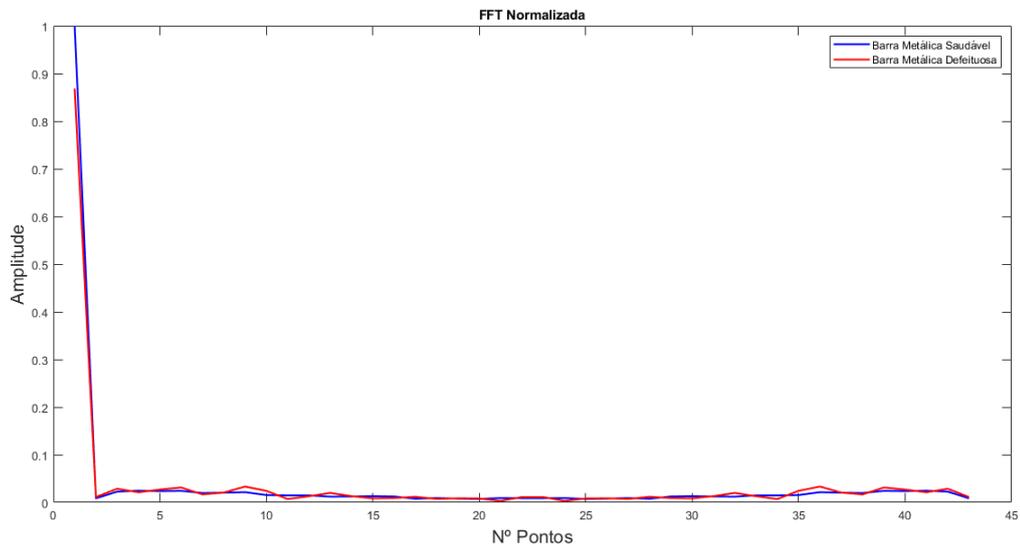
Fonte: Do Autor.

Neste trabalho as curvas das barras metálicas foram parametrizadas utilizando a magnitude da DFT e DWT. No caso da parametrização Wavelet foram utilizados todos os possíveis membros das famílias Coiflets, Biorthogonal, Symlet e Daubechies variando dentre todos os níveis de decomposição possíveis e alternando entre coeficientes de aproximação e detalhe. Essa abordagem foi escolhida pois a priori não se conhecia qual seria a configuração de família, membro, nível de decomposição e tipo de coeficiente que melhor representaria o problema.

As Figuras 44 e 45 apresentam, respectivamente, o espectro de magnitude das DFTs e os coeficientes de aproximação no terceiro nível da wavelet Bior2.2 de duas barras

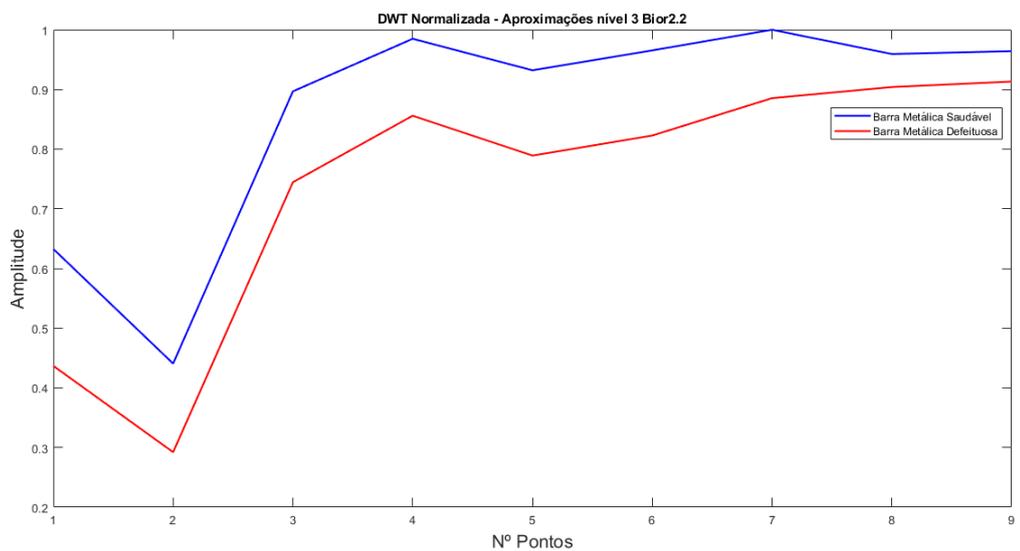
metálicas. As curvas em vermelho apresentam uma barra metálica defeituosa e as em azul apresentam uma barra metálica saudável. É possível notar diferenças nos padrões de comportamento das barras metálicas tanto para DFT quanto para DWT.

Figura 44 – Comparação Entre os Espectros de Magnitude



Fonte: Do Autor.

Figura 45 – Comparação entre os Coeficientes de Aproximação



Fonte: Do Autor.

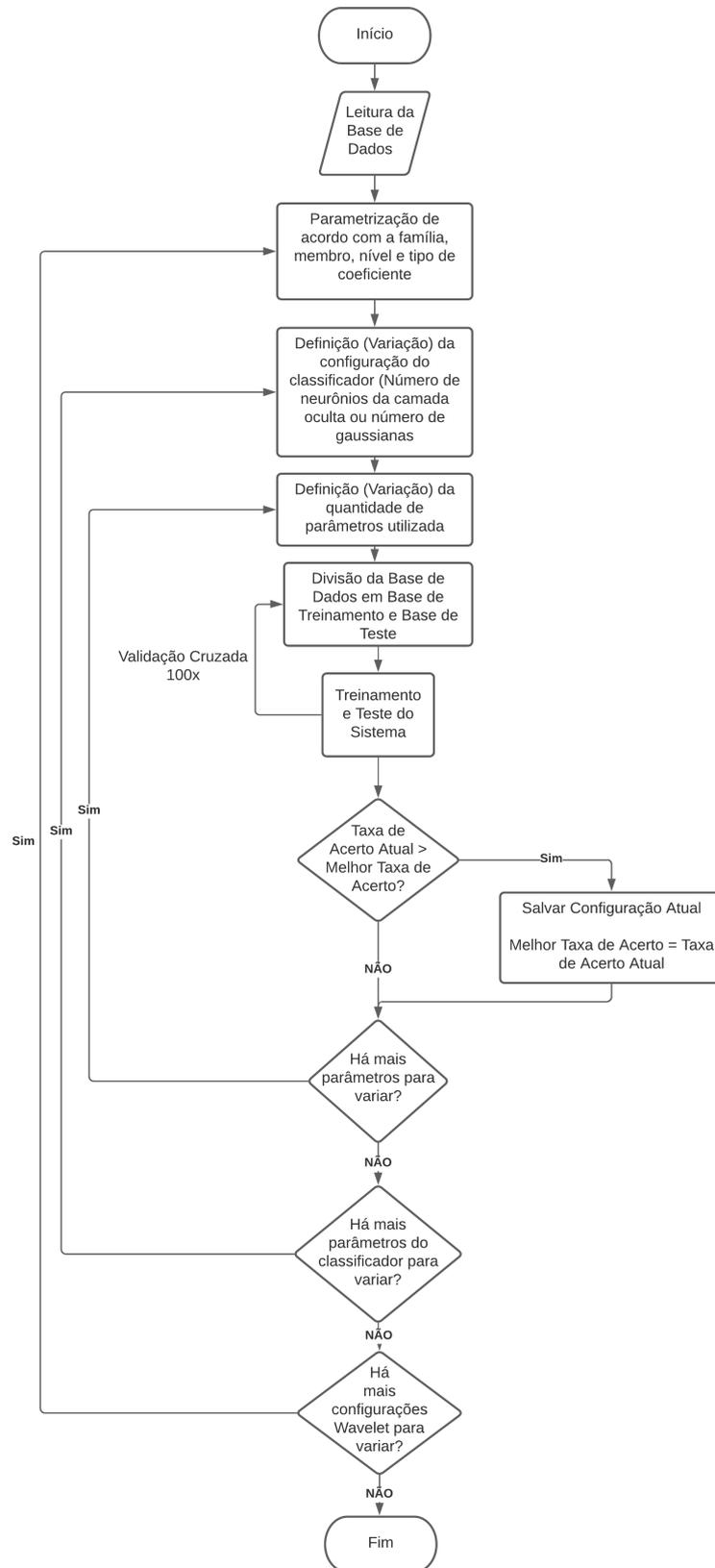
Além da variação de configuração na parametrização dos dados também ocorreram variações em configurações intrínsecas a cada classificador como por exemplo números de neurônios para ANN e número de gaussianas para GMM.

Uma vez que toda configuração do sistema de reconhecimento é definida, o classificador foi treinado utilizando 60% da base de dados e testado com os 40% restante. Para cada configuração o processo de treinamento e teste foi repetido 100 vezes, variando aleatoriamente as barras metálicas que compõe a base de treinamento e a de teste a fim de minimizar os efeitos do sorteio sobre o desempenho do sistema (validação cruzada). O desempenho final é a média dos desempenhos de cada uma das 100 validações.

É importante destacar que a função wavelet utilizada bem como seu nível e tipo de coeficiente (aproximação ou detalhe) foi determinada a partir de testes empíricos. Os testes foram realizados em todos os possíveis membros das famílias Daubechies, Biorthogonal, Symlet e Coiflets. Outra importante observação é que para o cálculo da DFT seria possível utilizar um número maior de pontos do que obtido no sinal espacial para maior resolução espectral, contudo um maior número de pontos também significaria um maior número de parâmetros para o classificador o que conseqüentemente tornaria o procedimento mais lento.

A Figura 46 apresenta o processo genérico de reconhecimento com parametrização Wavelet, o processo para reconhecimento com parametrização FFT é similar, porém não contém as variações de membro, nível de decomposição e tipo de coeficiente.

Figura 46 – Fluxograma algoritmo de Classificação



Fonte: Do Autor.

Empiricamente observou-se que o desempenho do sistema melhorava conforme se diminuía o número de parâmetros utilizados, portanto um algoritmo de força bruta foi desenvolvido para variar os parâmetros utilizados para a geração do modelo

O desempenho do sistema foi medido pela sua taxa de acerto médio e intervalo de confiança, demonstrado pela equação abaixo:

$$Desempenho = \bar{x} \pm 1,96 \frac{\sigma}{\sqrt{100}}. \quad (4.1)$$

Em que  $\bar{x}$  representa a média das taxas de acerto de cada uma das 100 validações cruzadas.

A tabela 6 demonstra o melhor desempenho para cada método de parametrização e classificador com um nível de confiança de 95%.

Tabela 6 – Melhores Desempenhos do Sistema.

Parametrização	Classificador	Desempenho (%)
FFT	SVM	91,30 ± 1,98
	GMM	85,80 ± 2,42
	ANN	87,50 ± 2,43
DWT	SVM	99,20 ± 0,50
	GMM	98,55 ± 0,83
	ANN	96,90 ± 1,70

Fonte: Do autor.

A tabela 7 apresenta a configuração do melhor resultado obtido para o sistema de detecção de trincas em barras metálicas.

Tabela 7 – Configuração do melhor resultado.

Configuração do Sistema	Valor
Coefficientes de Aproximações ou Detalhes	Aproximações
Família/Membro	Biorthogonal/Bior2.2
Nível de Decomposição	3
Número de parâmetros utilizados	2
Taxa de Desempenho	99,20 ± 0.50
Nível de confiança	95%

Fonte: Do autor.

A tabela 8 apresenta os valores reais obtidos no processo construtivo do escâner 3D alcançados em comparação com aqueles utilizados nas especificações do projeto e que se encontram descritos na tabela 1.

Tabela 8 – Especificações de Projeto Final.

Parâmetro	Valor Proposto	Valor Real	Unidade
Tamanho Total	2 x 1	2,5 x 1	m
Tamanho Útil/Tamanho Total	>90	70	%
Precisão Longitudinal	<3	4	mm
Precisão Transversal	<3	4	mm
Precisão Vertical	<1	<1	mm
Frequência de Amostragem do AD	>1000	<1000 Hz para ADS1115 >1000 Hz para ESP32	Hz
Resolução de Quantização	<0.1	<0.1	%
Tempo para Leitura Completa	<10	Variável	minutos
Armazenamento em SD	SIM	SIM	–
Taxa de Acerto na Classificação	>80	>99	%
Custo Total do Projeto	<4.500,00	Cerca de 6.000,00	R\$

Fonte: Do autor.

## 5 Considerações Finais

Os resultados mostraram que o sistema de detecção de trincas em barras metálicas proposto obteve êxito bem como mostraram sua utilidade em aplicações práticas, visto que atinge cerca de 99% da taxa de classificação correta.

Embora as peças mecânicas tenham sido devidamente projetadas antes de usadas, a montagem final de um sistema dessa dimensão foi trabalhosa. Visto que os rolamentos dos mancais necessitam de um excelente paralelismo entre os eixos. Além do uso de eixos de bitola elevadas, pés da estrutura com bastante peso também foram adicionados morsas de bancada em cada um dos pés do manipulador, garantindo redução de vibração da estrutura e flambagem dos eixos. O processo de usinagem das peças não é perfeito, portanto alguns deslocamentos em furos ou desvios de medida tiveram que ser compensados durante a montagem do equipamento.

Ao longo do estudo de equipamentos de escaneamento tridimensional foi possível observar a importância desse equipamento em diversas áreas da indústria, abrindo um leque para desenvolvimento de pesquisas em linhas médicas, tecnológicas e industriais.

O sistema apresentou interface amigável e de fácil utilização, possibilitando que um novo usuário possa elaborar leituras ou baixar leituras anteriores sem maiores dificuldades. O projeto da interface teve suas limitações, pois sua criação almejava uma interface intuitiva, agradável e rápida, para isso foi necessário aprofundar os estudos a cerca de HTML5 e CSS dado que não são conteúdos abordados ao longo do curso. Apesar do ESP32 possuir um módulo wi-fi, o mesmo apresenta baixa potência, que torna a distância um fator preocupante ao se conectar na interface, a recomendação mais coerente é conectar o microcontrolador a um roteador próximo e o usuário se conectar diretamente ao roteador, contudo o laboratório não possui uma rede própria, o que inviabilizou essa abordagem.

Apesar do microcontrolador utilizado possuir dois núcleos, apenas um deles foi utilizado uma vez que todo algoritmo funcionava de forma serial. Isso acarreta em um drástico aumento de tempo no processo de leitura, a implementação do algoritmo utilizando os dois núcleos possibilitaria que a leitura fosse efetuada de forma mais veloz.

A metodologia empregada no sistema de classificação buscou testar diferentes ferramentas de reconhecimento de padrões com o intuito de encontrar uma melhor configuração para o sistema. Os três classificadores foram escolhidos por possuírem diferentes características quanto a linearidade e quanto ao treinamento. E embora o SVM tenha apresentado o melhor desempenho, isso apenas significa que o classificador performou melhor para este problema e não que é uma ferramenta melhor ou pior que as demais.

Por ser um ambiente dotado de ferramentas e *toolboxes*, neste trabalho o sistema de classificação *offline* foi implementado no MATLAB. Para essa aplicação específica e já

conhecendo a configuração descrita na Tabela 7 é possível que o sistema de classificação seja implementado em um processador embarcado como o Raspberry Pi.

## 6 Sugestões para trabalhos futuros

Como sugestão ao prosseguimento dos estudos que foram abordados neste trabalho, ficam as seguintes sugestões:

- Implementação do controle e leitura de forma simultânea (núcleos diferentes);
- Aplicação de técnicas de seleção de características;
- Aplicação de pré-processamento da Base de Dados;
- Utilização do sistema de leitura tridimensional desenvolvido para reconhecimento de indivíduos;
- Utilização do sistema de leitura tridimensional desenvolvido para elaboração de próteses;
- Implementação do sistema de classificação de forma embarcada;
- Alocação dos encoders e implementação do controle de posição em malha fechada.

# Referências

- BANNER. *L-GAGE LE250/550 Analog/Discrete Laser Gauging Sensors*. [S.l.], 2015. Rev. E. 64
- BART MEIJER. *STEPSTICK DRV8825 v1.0 DATASHEET*. Klopperman 7 2292JE Wateringen The Netherlands, 2013. 33
- BENTLEY, P. M.; MCDONNELL, J. T. E. Wavelet transforms: an introduction. *Electronics Communication Engineering Journal*, v. 6, n. 4, p. 175–186, Aug 1994. 12
- BRASIL. *Lei nº. 8.078, de 11 de setembro de 1990. Código de Defesa do Consumidor. Dispõe sobre a proteção do consumidor e dá outras providências*. 2007. Último acesso em 29 de Setembro de 2019. Disponível em: <[http://www.planalto.gov.br/ccivil\\_03/leis/l8078.htm](http://www.planalto.gov.br/ccivil_03/leis/l8078.htm)>. 1
- CAI, Y. et al. Methods for long-distance crack location and detection of concrete bridge structures. In: *2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC)*. Chongqing, China: IEEE, 2018. v. 1, n. 18167158, p. 576–580. 3
- CHEN, F.; JAHANSHAHI, M. R. Nb-cnn: Deep learning-based crack detection using convolutional neural network and naïve bayes data fusion. *IEEE Transactions on Industrial Electronics*, v. 65, n. 5, p. 4392–4400, May 2018. 2
- CHO, H.; YOON, H.; JUNG, J. Image-based crack detection using crack width transform (cwt) algorithm. *IEEE Access*, v. 6, p. 60100–60114, 2018. 1
- CONSTANDINOU, T. G. Stepper motors uncovered (1). 2003. 60, 61
- COOLEY, J.; TUKEY, J. An algorithm for the machine calculation of complex fourier series. *Mathematics of Computation*, v. 19, n. 90, p. 297–301, 1965. 11
- COVÕES, T. F. *Algoritmos evolutivos para modelos de mistura de gaussianas em problemas com e sem restrições*. Tese (Doutorado) — Universidade de São Paulo, 2014. 23
- DAS, H.; PARHI, D. Application of neural network for fault diagnosis of cracked cantilever beam. In: *2009 World Congress on Nature Biologically Inspired Computing (NaBIC)*. Coimbatore, India: IEEE, 2009. p. 1303–1308. 1
- FILIPEFLOP. *Módulo WiFi ESP32 Bluetooth*. 2020. Último acesso em 14 de Setembro de 2020. Disponível em: <<https://www.filipeflop.com/produto/modulo-wifi-esp32-bluetooth/>>. 31
- FOURIER, J. B. J. baron. *Théorie analytique de la chaleur*. [S.l.]: F. Didot, 1822. 9
- FUGAL, D. L. *Conceptual wavelets in digital signal processing : an in-depth, practical approach for the non-mathematician*. University of California, USA: Space Signals Technical Publishing, 2009. 14
- GABOR, D. *Theory of communication*. India: IET, 1946. v. 93. 429–457 p. 12

- GERARD, S. E. et al. Fissurenet: A deep learning approach for pulmonary fissure detection in ct images. *IEEE Transactions on Medical Imaging*, v. 38, n. 1, p. 156–166, Jan 2019. 1, 3
- GONZALEZ, R. C.; WOODS, R. E. *Digital image processing*. 3rd. ed. New Jersey, USA: Prentice Hall, 2008. ISBN 9780131687288 013168728X 9780135052679 013505267X. 10
- GRAPS, A. An introduction to wavelets. *IEEE computational science and engineering*, IEEE, v. 2, n. 2, p. 50–61, 1995. 15
- HAAR, A. Zur theorie der orthogonalen funktionensysteme. *Mathematische Annalen*, v. 69, n. 3, p. 331–371, Sep 1910. ISSN 1432-1807. Disponível em: <<https://doi.org/10.1007/BF01456326>>. 13
- HEIDEMAN, M.; JOHNSON, D.; BURRUS, C. Gauss and the history of the fast fourier transform. *IEEE ASSP Magazine*, v. 1, n. 4, p. 14–21, October 1984. 11
- HERNÁNDEZ, E.; WEISS, G. *A first course on wavelets*. Clevelândia, Ohio, EUA: CRC press, 1996. 15
- JUNIOR, G. Máquina de vetores suporte: estudo e análise de parâmetros para otimização de resultado. *Ciência da Computação, Universidade Federal de Pernambuco*, 2010. 18
- KEYENCE. *1D Laser Displacement Sensors*. 2019. Último acesso em 08 de Novembro de 2019. Disponível em: <[https://www.keyence.com/ss/products/measure/measurement\\_library/type/laser\\_1d/](https://www.keyence.com/ss/products/measure/measurement_library/type/laser_1d/)>. 64
- KIM, J.-T.; STUBBS, N. Crack detection in beam-type structures using frequency data. *Journal of sound and vibration*, Elsevier, v. 259, n. 1, p. 145–160, 2003. 3, 4
- KOLEV, K. et al. Turning mobile phones into 3d scanners. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. Columbus, OH, USA: IEEE, 2014. p. 3946–3953. 4, 5
- LEUZE ELETRONIC. *ODSL*. New Hudson, MI, USA, 2018. 37, 39
- MALLAT, S. A wavelet tour of signal processing (academic press, new york 1999); i. *Daubechies Ten Lectures on Wavelets (SIAM, Philadelphia, 1992)*. [10]. Jouault, F. *S é bille and V. de la Mota, Nucl. Phys. A*, v. 628, n. 1, p. 998, 1998. 15
- MATHWORKS. *STFT: Short-time Fourier transform*. 2019. <<https://www.mathworks.com/help/signal/ref/stft.html>>. 12
- MEYER, Y.; RYAN, R. Wavelets: Algorithms and applications. *SIAM, Philadelphia, PA*, 1993. 15
- MITSUMI. *Stepping Motors M49SP-1*. Perdizes, São Paulo, 2008. 27
- MIYADAIRA, A. N. *Microcontroladores PIC18: Aprenda e programe em linguagem C*. [S.l.]: Saraiva Educação SA, 2009. 63
- MORETTIN, P. A. *Ondas e ondaletas: da análise de Fourier à análise de ondaletas*. São Paulo, São Paulo: Edusp, 1999. 15

- MORLET, J.; GROSSMANN, A. Decomposition of hardy functions into square integrable wavelets of constant shape. *SIAM Journal on Mathematical Analysis*, v. 15, p. 723–736, 07 1984. 13
- NGUYEN, T. *Wavelets and filter banks*. 7 Southgate Rd, Wellesley, MA 02482, USA: Wellesley-Cambridge Press, 1996. 15
- OLIVEIRA, A. d. Redes neurais artificiais aplicadas na detecção, classificação e localização de defeitos em linhas de transmissão. *UFJF. Juiz de Fora*, 2005. 20
- PORTELA, N. M. Modelo de mistura de gaussianas fuzzy contextual. Universidade Federal de Pernambuco, 2015. 22
- QUATIERI, T. F. *Discrete-time speech signal processing: principles and practice*. India: Pearson Education India, 2006. 15, 16
- RABABAAH, H.; VRAJITORU, D.; WOLFER, J. Asphalt pavement crack classification: a comparison of ga, mlp, and som. In: *Genetic and Evolutionary Computation Conference*. Indiana University South Bend: Indiana University South Bend, 2005. 1, 3
- SHEN, G. Road crack detection based on video image processing. In: *2016 3rd International Conference on Systems and Informatics (ICSAI)*. Shanghai, China: IEEE, 2016. p. 912–917. 1, 2
- SIMAS, E. *Disciplina: Processamento Estatístico de Sinais Aula 04: Introdução às Redes Neurais Artificiais Prof. Eduardo Simas*. 2019. Último acesso em 04 de Dezembro de 2019. Disponível em: <<https://slideplayer.com.br/slide/292589/>>. 19, 21
- STRAWN, G. Automation and future unemployment. *IT Professional*, v. 18, n. 1, p. 62–64, Jan 2016. 1
- SUL, O. M. G. D. *Direito do Consumidor: lojas têm 30 dias para trocar produto com defeito*. 2012. Último acesso em 29 de Setembro de 2019. Disponível em: <<https://oab-ms.jusbrasil.com.br/noticias/100381546/direito-do-consumidor-lojas-tem-30-dias-para-trocar-produto-com-defeito>>. 1
- TEKKNO. *Fresadora Router Cnc - Usina Aço - Fuso de Esferas*. 2019. Último acesso em 03 de Dezembro de 2019. Disponível em: <<https://www.tekkno.com.br/produto/3678/fresadora-router-cnc---usina-aco---fuso-de-esferas>>. 25
- THEODORIDIS, S.; KOUTROUMBAS, K. *Pattern recognition. 2003*. Cambridge, Massachusetts, USA: Academic Press, 2009. 17
- WEISE, T.; LEIBE, B.; GOOL, L. V. Fast 3d scanning with automatic motion compensation. In: *2007 IEEE Conference on Computer Vision and Pattern Recognition*. Minneapolis, MN, USA: IEEE, 2007. p. 1–8. 5, 6
- YELLA, S.; RAHMAN, A. S.; DOUGHERTY, M. Pattern recognition for classifying the condition of wooden railway sleepers. In: *2010 International Conference on Multimedia Computing and Information Technology (MCIT)*. Sharjah, United Arab Emirates: IEEE, 2010. p. 61–64. 1

- YING, X. et al. Experiment studies on crack phenomena of hot stamping parts. In: *International Conference on Advanced Technology of Design and Manufacture (ATDM 2010)*. Beijing, China: IET, 2010. p. 388–390. 1
- YUNARDI, R. T.; IMANDIRI, A. Design of the 3d surface scanning system for human wrist contour using laser line imaging. In: *2018 5th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*. Semarang, Indonesia: IEEE, 2018. p. 1–5. 6, 7

# Apêndice A

Neste apêndice serão abordados temas sobre os equipamentos utilizados para a realização deste trabalho.

## Motor de Passo

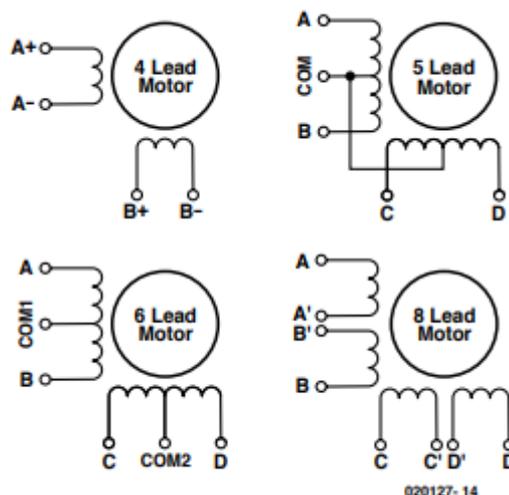
Abordaremos aqui de forma resumida a teoria de funcionamento do motor de passo, bem como a utilização do driver de funcionamento e do encoder.

### Funcionamento

O uso de motores é comum no dia-a-dia das pessoas, mesmo que indiretamente. Por diversas vezes poder ser necessário garantir um deslocamento específico em algum equipamento e embora isso possa ser possível de ser feito com um motor DC o controle de deslocamento fica muito melhor quando se usa um motor de passo.

O motor de passo costuma ser muito utilizado em equipamentos como escâners, impressoras, CNCs e em diversos processos industriais. O motor de passo pode ser de relutância variável, de imã permanente ou híbrido. Ao introduzir corrente elétrica em um enrolamento, é produzido um campo magnético no estator que atrai o campo magnético do rotor e faz com que o rotor se alinhe, com isso identificamos que para fazer o rotor girar, basta introduzir corrente elétrica alternadamente nos terminais do estator (CONSTANDINOU, 2003).

Figura 47 – Conexões básicas de motores de passo



Fonte: Retirado de Constandinou (2003)

Como citado o rotor do motor de passo se move ao introduzir corrente nos terminais do estator, porém o rotor se move até os campos magnéticos se alinharem e então fica estático. Para que o rotor fique em movimento constante é necessário aplicar uma série de pulsos de corrente nos terminais do estator, o diagrama abaixo mostra uma das possíveis maneiras de se fazer isso para motores de bipolares a 4 fios.

Figura 48 – Método de controle do motor de passo

Order	Phase A	Phase B	Phase $\bar{A}$	Phase $\bar{B}$
1				
2				
3				
4				

Fonte: Retirado de Constandinou (2003)

Algumas expressões são comuns no uso de motores de passo e de importante conhecimento:

- **Holding Torque:** Máximo torque possível de aplicar no rotor sem produzir rotação do eixo.
- **Pull-in Torque:** Torque máximo que o rotor pode arrancar, parar e inverter seu sentido de movimento sem que sejam causados erros de posicionamento.
- **Pull-out Torque:** Torque máximo que é possível aplicar para que o rotor rode em velocidade constante.

Existem também outros importantes conceitos para o leitor que deseja se aprofundar no estudo de motores de passo, como precisão, erro de histerese, ressonância, curva velocidade/torque, máxima frequência de arranque.

## Encoder

Embora o controle de posição com o motor de passo seja muito bom, pro diversas vezes acontece do motor “engasgar”, ou seja, embora ocorra o passo elétrico o motor não corresponde com o passo mecânico e isso ocasiona um erro de posição. Para solucionar esse problema, um solução comum é o uso de um equipamento denominado encoder.

O encoder é um equipamento capaz de emitir um sinal elétrico relacionado com a posição angular do sistema, o conceito é utilizar de um sistema luminoso para identificar a posição do eixo. Dentre os tipos de encoder existentes, os mais comuns são o encoder incremental e o encoder absoluto, a diferença básica entre os dois é que o encoder incremental mede a posição relativa do eixo que depende da posição inicial do sistema, já o encoder absoluto determina a posição absoluta do encoder a partir de códigos luminosos.

## Driver de Funcionamento

Em geral, é necessário o uso de um microprocessador ou similar para o controle da rotação do motor de passo, porém devido a alta corrente drenada pelo motor o microprocessador não é capaz de fornecer a potência necessária para que o motor exerça sua função, portanto é necessário o uso de circuitos de alimentação que forneçam potência para o motor.

Contudo esses circuitos podem ser um pouco complicados ou não tão interessantes de serem implementados, para isso costuma-se utilizar os drivers de motor de passo que recebem os pulsos do microcontrolador e a partir deles fornecem corrente para que o motor funcione.

## Microcontroladores

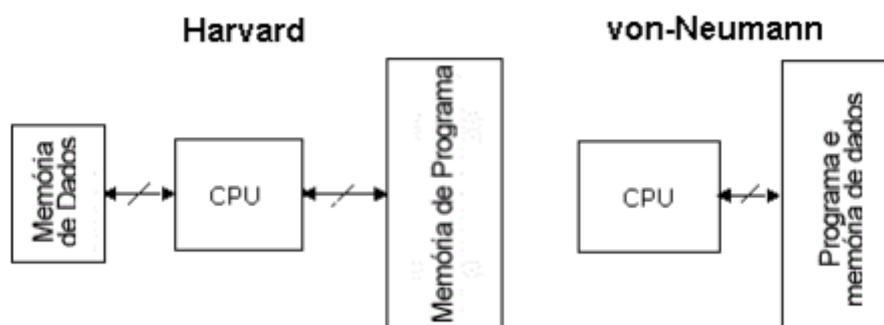
Nesta seção serão apresentados algumas características a respeito dos microcontroladores.

## Funcionamento

Microcontroladores são pequenos dispositivos capazes de realizar execuções de algoritmo em alta velocidade. Em geral um microcontrolador é composto por uma unidade de processamento, memórias de dados e de programa e periféricos, como portas *I/O*, *SPI*, *USART*, *RTC* e etc.

Os microcontroladores têm um espaço significado em muitos projetos de engenharia, dado a sua capacidade de reproduzir as características de um computador em um espaço muito pequeno. As duas principais arquiteturas de microcontroladores são Harvard e Von-Neumann como apresentado na figura abaixo:

Figura 49 – Arquiteturas Harvard e Von-Neumann



Fonte: Retirado de Miyadaira (2009)

Como citados, temos dois tipos de memórias presentes em um microcontrolador, sendo elas de programa e de dados. A primeira tem como característica ser não-volátil, ou seja, mesmo após desenergizado o microcontrolador não perde os dados registrados, essa memória é utilizada para armazenar o algoritmo programado no microcontrolador. A segunda memória especificada é a memória de dados e tem como principal característica ser volátil, essa memória é utilizada para salvar temporariamente dados de variáveis e cálculos efetuados pela unidade de processamento (MIYADAIRA, 2009).

Dentre os periféricos é importante elucidar a respeito das portas *I/O* que possibilitam o microcontrolador ler e escrever dados ao mundo exterior. Existem outros periféricos também muito utilizados como *TIMER*, *SPI*, *P.C.* Também é necessário esclarecer que a frequência do microcontrolador é um fator muito determinante na velocidade de processamento, sendo possível utilizar o oscilador interno do microcontrolador ou também utilizar um oscilador externo através de um cristal de quartzo ou um ressonador.

## Sensor de Distância Óptico

Nessa seção será abordado algumas características do sensor de distância óptico utilizado para a realização deste trabalho.

### Funcionamento

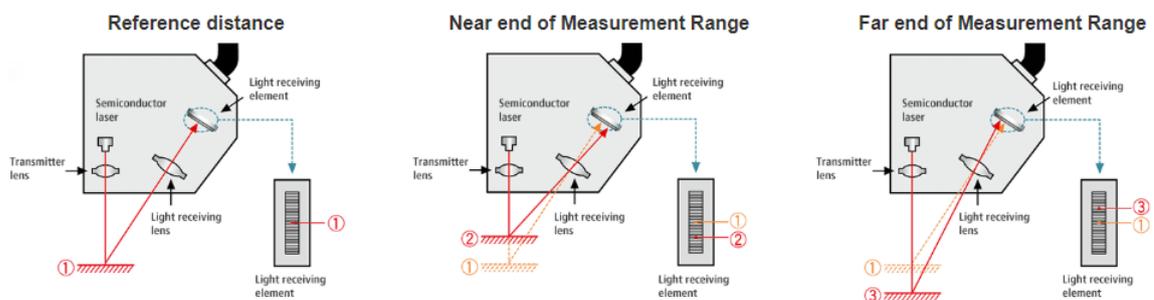
Neste trabalho será utilizado o sensor de distância óptico ODSL8. Esse sensor conta com duas saídas, sendo uma delas analógica que emite um sinal de tensão (10V até 30V)

ou corrente (4mA a 20mA) com uma relação linear à distância que o sensor tem do objeto. Optou-se por utilizar o sensor com saída de corrente, para evitar perda de informação por queda de tensão nos condutores. A outra saída é digital, e emite um sinal nível alto ou baixo (pnp ou npn) quando detectar um objeto há uma distância previamente setada.

O ODSL8 não é programável, toda configuração é feita através dos parafusos de interface, após ligado o sensor faz leitura à todo momento. O seu range de leitura é de 20 milímetros até 0,5 metro, sendo que os primeiros 20 milímetros são considerados “zona morta” (BANNER, 2015).

O sensor funciona a partir de triangulação óptica, um laser de luz é emitido e refletido pelo objeto que é concentrada na lente receptora. Conforme a distância entre o laser e o objeto varia, a posição em que o feixe é recebido na lente receptora também varia.

Figura 50 – Método de Triangulação óptica do sensor



Fonte: Retirado de KEYENCE (2019)

# Apêndice B

## Algoritmo do ESP32

```
1 // #include <ESP32WebServer.h> // https://github.com/  
    Pedroalbuquerque/ESP32WebServer download and place in your  
    Libraries folder  
2 #include <Wire.h>  
3 #include <Adafruit_ADS1015.h>  
4 #include <ESPmDNS.h>  
5 #include "FS.h"  
6 #include "Network.h"  
7 #include "Sys_Variables.h"  
8 #include "CSS.h"  
9 #include <SD.h>  
10 #include <SPI.h>  
11 // -----ate aqui ---  
12 #include <WiFi.h>  
13 #include <WebServer.h>  
14 #include <string.h>  
15 //Motores  
16 #include "DRV8834.h"  
17 #include "A4988.h"  
18 #include "DRV8825.h"  
19  
20  
21 /* Put your SSID & Password */  
22 const char* ssid = "ESP32-ProcSiMoS-Manipulador"; // Enter  
    SSID here  
23 const char* password = "12345678"; //Enter Password here  
24  
25 /* Put IP Address details */  
26 //IPAddress local_ip(192,168,1,1);  
27 //IPAddress gateway(192,168,1,1);  
28 //IPAddress subnet(255,255,255,0);  
29  
30 WebServer server(80);
```

```
31
32 //ESP32WebServer server(80);
33
34 String xin, xfi, yin, yfi, pl, pt, encoder_rec;
35 int working = 1; //1 para dispon vel, 0 para trabalhando
36 int usa_encoder = 1; //1 para usar informa es do encoder,
    0 para n o usar;
37 int x_i, x_f, y_i, y_f, p_l, p_t, recalibra_encoder = 0,
    zerandoposicao;
38 float ajuste_encodert = 0, ajuste_encoderl1 = 0,
    ajuste_encoderl2 = 0;
39 volatile int fim_curso = 0, encoder_t = 0, encoder_l1 = 0,
    encoder_l2 = 0;
40 volatile unsigned long tempo_atual = -10000; //iniciando
    vari vel para aceitar a primeira interrup o
41 int meio_da_pista = 350;
42
43
44 // declarando html
45 const char index_html[] = R"rawliteral(
46 <!DOCTYPE HTML><html><head>
47   <title>ESP Input Form Manipulator </title>
48   <meta name="viewport" content="width=device-width, initial-
    scale=1">
49   </head><body>
50 <form action="/get">
51   <fieldset>
52     <legend><h2>ESP32 Web Server - ProcSiMoS Manipulador</h2
    ></legend>
53     X inicio (mm):<br><input type="number" min="0.00" max
    ="700" step="2" name="xi" required ><br>
54     X fim (mm): <br><input type="number" min="0.00" max="700"
    step="2" name="xf" required><br>
55     Y inicio (mm): <br><input type="number" min="0.00" max
    ="2400" step="2" name="yi" required><br>
56     Y fim (mm): <br><input type="number" min="0.00" max
    ="2400" step="2" name="yf" required><br><br>
57     Passo Longitudinal (mm): <br><input type="number" min="2"
    max="2000" step="2" name="pl" required><br>
```

```

58     Passo Transversal (mm): <br><input type="number" min="2"
        max="700" step="2" name="pt" required><br><br>
59     <input type="checkbox" name="encoders" value="1">
60     <label for="encoders"> Efetuar Calibração dos Encoders
        </label><br>
61     <legend><h4>Obs: Os passos do manipulador devem ser
        multiplos de 2 mm;<br>X corresponde ao deslocamento
        transversal e Y ao deslocamento Longitudinal.</h4></
        legend>
62     <input type="submit" value="Enviar">
63     </fieldset>
64 </form>
65 </body></html>)rawliteral";
66
67
68 // fim html
69 // gerando class file
70 File myFile;
71 File myFile_e;
72 String filename = "/datalog_0000001.csv"; //"datalog_0000001.
        csv";
73
74 // declarando variáveis i2c
75 //Adafruit_ADS1115 ads(0x48);
76 Adafruit_ADS1115 ads;
77
78 //
79
80     ADS1015  ADS1115
81 //
82     -----  -----
83 // ads.setGain(GAIN_TWOTHIRDS); // 2/3x gain +/- 6.144V  1
84     bit = 3mV      0.1875mV (default)
85 // ads.setGain(GAIN_ONE); // 1x gain +/- 4.096V  1
86     bit = 2mV      0.125mV
87 // ads.setGain(GAIN_TWO); // 2x gain +/- 2.048V  1
88     bit = 1mV      0.0625mV

```

```
83 // ads.setGain(GAIN_FOUR);          // 4x gain   +/- 1.024V  1
    bit = 0.5mV    0.03125mV
84 // ads.setGain(GAIN_EIGHT);        // 8x gain   +/- 0.512V  1
    bit = 0.25mV   0.015625mV
85 // ads.setGain(GAIN_SIXTEEN);     // 16x gain  +/- 0.256V  1
    bit = 0.125mV  0.0078125mV
86
87 //----INSTANCIAS DOS MOTORES ---
88
89 // Quantidade de passos por volta completa
90 #define MOTOR_STEPS 48
91 // Definição de pinos DIR e STEP
92 #define DIR 17
93 #define STEP 13
94
95 // Pinos de controle do DRV8825
96 #define MODE0 26
97 #define MODE1 27
98 #define MODE2 14
99
100 DRV8825 stepper(MOTOR_STEPS, DIR, STEP, MODE0, MODE1, MODE2)
    ;//motor transversal
101 DRV8825 stepper2(MOTOR_STEPS, DIR, STEP, MODE0, MODE1, MODE2)
    ;//dois motores longitudinais
102 DRV8825 stepper3(MOTOR_STEPS, DIR, STEP, MODE0, MODE1, MODE2)
    ;//motor transversal operando com poucos pulsos
103
104 void setup() {
105     Serial.begin(115200); //115200
106     while (!Serial) {
107         ; // wait for serial port to connect.
108     }
109     WiFi.softAP(ssid, password, 6); //6 anteriormente
110     // WiFi.softAPConfig(local_ip, gateway, subnet);
111     delay(100);
112
113     server.on("/", handle_OnConnect);
114     server.on("/get", handle_get);
115     server.on("/downloads", handle_downloads);
```

```
116 server.on("/leitura", handle_leitura);
117 server.on("/cancel", handle_cancel);
118 server.on("/download", File_Download);
119 server.on("/dir", SD_dir);
120 server.on("/concluido", leitura_finalizada);
121
122 server.onNotFound(handle_NotFound);
123
124 server.begin();
125 Serial.println("HTTP server started");
126
127 // configurando download
128
129 Serial.println(MISO);
130 pinMode(19, INPUT_PULLUP);
131 Serial.print(F("Initializing SD card..."));
132 if (!SD.begin(SD_CS_pin)) { // see if the card is present
    and can be initialised. Wemos SD-Card CS uses D8
133     Serial.println(F("Card failed or not present, no SD Card
        data logging possible..."));
134     SD_present = false;
135 }
136 else
137 {
138     Serial.println(F("Card initialised... file access enabled
        ..."));
139     SD_present = true;
140 }
141
142 // Note: Using the ESP32 and SD_Card readers requires a 1K
    to 4K7 pull-up to 3,3V on the MISO line, otherwise they
    do-not function.
143 ads.setGain(GAIN_TW0);
144 ads.begin(); //iniciando o ads1115
145 // motores
146
147 stepper.begin(180, 1); //180 RPM microsteps = 1
148 stepper2.begin(250, 1); //250 RPM microsteps = 1
149 stepper3.begin(350, 1); //350RPM microsteps = 1;
```

```
150 pinMode(25, OUTPUT); //2 motores logitudinais
151 pinMode(33, OUTPUT); //1 motor transversal
152 pinMode(15, INPUT); //pino interrup o fins de curso
153 pinMode(4, INPUT);
154 pinMode(2, INPUT);
155 pinMode(16, INPUT);
156 pinMode(34, INPUT); //pino interrup o encoder motor
    transversal
157 pinMode(35, INPUT); //pino interrup o encoder motor
    longitudinal 12V
158 pinMode(32, INPUT); //pino interrup o encoder motor
    longitudinal 24V
159 attachInterrupt(digitalPinToInterrupt(15), fimdecurso,
    FALLING); // configura o interrup o do fim de curso
    NF
160 attachInterrupt(digitalPinToInterrupt(4), fimdecurso,
    FALLING); // configura o interrup o do fim de curso
    NF
161 //attachInterrupt(digitalPinToInterrupt(2), fimdecurso,
    FALLING); // configura o interrup o do fim de curso
    NF
162 //attachInterrupt(digitalPinToInterrupt(16), fimdecurso,
    FALLING); // configura o interrup o do fim de curso
    NF
163 attachInterrupt(digitalPinToInterrupt(34), ler_encodert,
    RISING); //configurando interrup o encodere
    transversal
164 attachInterrupt(digitalPinToInterrupt(35), ler_encoderl1,
    RISING); //configurando interrup o encodere
    longitudinal 12v
165 attachInterrupt(digitalPinToInterrupt(32), ler_encoderl2,
    RISING); //configurando interrup o encodere
    longitudinal 24v
166 }
167
168
169 void loop() {
170     server.handleClient();
171     if (!working) {
```

```
172     scanear();
173   }
174 }
175
176 void scanear() {
177   //int x_i, x_f,y_i,y_f,p_l,p_t;
178   calibracao();
179   String Vetor = "";
180   int y, x, i, idx, idy, j, ajuste;
181   int passot, passol;
182   passot = (p_t / 3) * 2;
183   passol = (p_l / 3) * 2;
184   float altura = 0;
185   myFile = SD.open(filename.c_str(), FILE_WRITE); //abre o
        arquivo criado
186   x = 0;
187   idx = (x_f - x_i) / passot;
188   idy = (y_f - y_i) / passol;
189   for (i = 0; i <= idx; i++) {
190     Vetor = String(Vetor + String(x) + ";"); //ESCREVE
        PRIMEIRA LINHA REFERENTE AOS PASSOS
191     x = x_i + i * passot;
192   }
193   Vetor = String(Vetor + String(x) + ";");
194   myFile.println(Vetor);
195   //Chamar funcao que zera a posicao dos motores; (JOGAR
        KIT SENSOR NO MEIO, ANDAR DE VOLTA COM OS DOIS MOTORES E
        EM SEGUIDA ZERAR KIT SENSOR)
196   usa_encoder = 0; //desliga encoders
197   tempo_atual = millis() - 9000;
198   movimenta_motor(1275, -1, 1); //Zera posicao do motor
        translacional
199   ajuste = (meio_da_pista / 2) * 3;
200   movimenta_motor(ajuste, 1, 1); // Coloca motor
        translacional no meio da pista
201   tempo_atual = millis() - 9000;
202   delay(5000);
203   zerandoposicao=1;
```

```
204  movimenta_motor(4000, -1, 0); // Zera posição do motor
      longitudinal
205  zerandoposicao=0;
206  //usa_encoder = 1;//liga encoders
207  y = 0;
208  for (j = 0; j <= idy; j++) { //INICIA PROCESSO DE LEITURA
209      delay(5000);
210      if (j == 0) {
211          if (y_i != 0) {
212              movimenta_motor(((y_i / 2) * 3) , 1, 0); //Movimenta
                os motores longitudinais para a posição inicial
213              y = y_i;//ajusta o valor atual de y
214          }
215      } else {
216          movimenta_motor(p_l , 1, 0);//Anda um passo adiante
217          y = y + passol;//Ajusta o valor atual de y
218      }
219      delay(5000);
220      ajuste = (meio_da_pista / 2) * 3;
221      tempo_atual = millis() - 9500;
222      movimenta_motor((ajuste + 100 + (x_i / 2) * 3), -1, 1);
            // Coloca motor translacional no meio da pista//Chamar
            funcao que zera posição do kit sensor
223      x = 0;
224      //Inicia movimento do kit sensor
225      delay(5000);
226      Vetor = String(y) + ";";
227      for (i = 0; i <= idx; i++) {
228          if (i == 0) {
229              if (x_i != 0) {
230                  movimenta_motor(((x_i / 2) * 3) , 1, 1); //
                    Movimenta os motor translacional para a
                    posição inicial
231                  x = x_i;//ajusta o valor atual de x;
232              }
233          } else {
234              movimenta_motor(p_t , 1, 1);//Anda um passo adiante
235              x = x + passot;//Ajusta o valor atual de x
236          }
```

```
237     altura = ler_ad();
238     Vetor = String(Vetor + String(altura) + ";");
239 }
240 myFile.println(Vetor); //Salvar dados no cartão SD
241 ajuste = ((meio_da_pista - x) / 2) * 3; //Calculando
        numero de pulsos necessários para jogar o sensor pro
        meio da pista
242 movimenta_motor(ajuste, 1, 1); //Chamar função que
        aloca kit sensor no meio da pista
243 delay(500);
244 }
245 movimenta_motor(1275, -1, 1); //zera posição do motor;
246 myFile.close();
247 working = 1;
248 }
249
250 float ler_ad() {
251     int16_t adc0;
252     float voltg = 0.0;
253     int i = 0;
254     if (((p_t / 3) * 2) > 40) {
255         delay(5000);
256     }else{
257         delay(2000);
258     }
259     for (i = 0; i < 1000; i++) {
260         //adc0 = ads.readADC_SingleEnded(3); //pino 3 do ad
261         adc0 = analogRead(36);
262         voltg = voltg + adc0 * 0.001;
263         delay(1);
264     }
265     //voltg = analogRead(36);
266     //voltg = (voltg * 0.0625) / 1000; //calculando o valor de
        tens o real;
267     voltg = (voltg * 3.3) / 4095;
268     voltg = 360 * voltg - 100; //convertendo tens o (V) em
        dist ncia (mm)
269
270
```

```
271     return voltg;
272 }
273
274 void movimenta_motor(int pulsos, int direcao, int motor) {
275     //motor = 0 = longitudinal, 1 = transversal.
276     //direcao = 1 = frente, -1 = tr s
277     if (motor) {
278         //direcao = direcao*(-1);
279         if (direcao * pulsos > 10 || direcao * pulsos < -10) {
280             digitalWrite(33, HIGH);
281             delay(2);
282             stepper.move(direcao * pulsos);
283             digitalWrite(33, LOW);
284             if (fim_curso) { //Caso o fim de curso seja acionado
285                 delay(5000);
286                 digitalWrite(33, HIGH);
287                 delay(2);
288                 stepper.move(-1 * direcao * 45); //Anda 3CM pro lado
289                 digitalWrite(33, LOW);
290                 fim_curso = 0;
291             }
292         } else {
293             digitalWrite(33, HIGH);
294             delay(2);
295             stepper3.move(direcao * pulsos);
296             digitalWrite(33, LOW);
297             if (fim_curso) { //Caso o fim de curso seja acionado
298                 delay(5000);
299                 digitalWrite(33, HIGH);
300                 delay(2);
301                 stepper3.move(-1 * direcao * 45); //Anda 3CM pro lado
302                 digitalWrite(33, LOW);
303                 fim_curso = 0;
304             }
305         }
306     } else {
307         digitalWrite(25, HIGH);
308         delay(2);
309         stepper2.move(direcao * pulsos);
```

```
310     digitalWrite(25, LOW);
311     if(zerandoposicao){
312     if (fim_curso) { //Caso o fim de curso seja acionado
313         delay(5000);
314         digitalWrite(25, HIGH);
315         delay(2);
316         stepper2.move(-1 * direcao * 45); //Anda 3CM pro lado
317         digitalWrite(25, LOW);
318         fim_curso = 0;
319     }}
320 }
321
322 if (usa_encoder) {
323     int pulsos_e = 100 * pulsos / 48; //numero de pulsos que
        o encoder deveria ter lido
324     int pulsos_novos = 0;
325     if (motor) {
326         encoder_t = (int)(ajuste_encodert * encoder_t); //
            ajustando valor do encoder com cast for ando o
            retorno como inteiro
327         if ((pulsos_e - encoder_t) > 0.3 * pulsos_e) {
328             encoder_t = 0;
329             pulsos_novos = (pulsos_e - encoder_t);
330             pulsos_novos = 48 * pulsos_novos / 100;
331             movimenta_motor(pulsos_novos , direcao, motor);
332         }
333     } else {
334         encoder_l1 = (int)(ajuste_encoderl1 * encoder_l1);
335         encoder_l2 = (int)(ajuste_encoderl2 * encoder_l2);
336         if ((pulsos_e - encoder_l1) > 0.3 * pulsos_e || (
            pulsos_e - encoder_l2) > 0.3 * pulsos_e) {
337             encoder_l1 = 0;
338             encoder_l2 = 0;
339             pulsos_novos = ((pulsos_e - encoder_l1) + (pulsos_e -
                encoder_l2)) / 2;
340             pulsos_novos = 48 * pulsos_novos / 100;
341             movimenta_motor(pulsos_novos , direcao, motor);
342         }
343     }
```

```
344     }
345
346   }
347
348 }
349
350 void fimdecurso() { //interrup o fim de curso
351   if (( millis() - tempo_atual) > 10000) {
352     tempo_atual = millis();
353     fim_curso = 1;
354     digitalWrite(33, LOW);
355     if(zerandoposicao){
356       digitalWrite(25, LOW);}
357   }
358 }
359
360 // interrup es encoder
361 void ler_encodert() {
362   encoder_t++;
363 }
364
365 void ler_encoderl1() {
366   encoder_l1++;
367 }
368
369 void ler_encoderl2() {
370   encoder_l2++;
371 }
372
373 void calibracao() {
374   float valor2[3], valor1;
375   int cont = 0, valor, ajuste;
376   String leitura = "", atual = "";
377   Serial.print("Recalibra encoder: ");
378   Serial.print(recalibra_encoder);
379   Serial.println();
380   if (recalibra_encoder) {
381     //escrever fun o que recalibra encoder
382     usa_encoder = 0;//desliga encoders
```

```
383     tempo_atual = millis() - 9000;
384     movimenta_motor(1275, -1, 1); //Zera posição do motor
        translacional
385     ajuste = (meio_da_pista / 2) * 3;
386     movimenta_motor(ajuste, 1, 1); // Coloca motor
        translacional no meio da pista
387     tempo_atual = millis() - 9000;
388     movimenta_motor(4000, -1, 0); // Zera posição do motor
        longitudinal
389     //calibrando encoders longitudinais
390     encoder_l1 = 0;
391     encoder_l2 = 0;
392     movimenta_motor(2016, 1, 0); //movimenta motores 42
        voltas = 4200 pulsos do encoders
393     if (encoder_l1 != 0 && encoder_l2 != 0) {
394         ajuste_encoderl1 = 4200 / encoder_l1;
395         ajuste_encoderl2 = 4200 / encoder_l2;
396     }
397     //calibrando encoder transversal
398     movimenta_motor(ajuste, -1, 1); // Zera posição do
        motor
399     encoder_t = 0;
400     movimenta_motor(1008, 1, 1); //movimenta motor
        translacional 21 voltas = 2100 pulsos do encoder
401     if (encoder_t != 0) {
402         ajuste_encodert = 2100 / encoder_t;
403     }
404     //salvando no cartão SD os novos dados
405
406     myFile_e = SD.open("/configuracao.csv", FILE_WRITE);
407     if (myFile_e) {
408         leitura = String(String(ajuste_encodert) + ";" + String
            (ajuste_encoderl1) + ";" + String(ajuste_encoderl2)
            + ";");
409         myFile_e.println(leitura);
410         myFile_e.close();
411         Serial.println(leitura);
412     } else {
413         Serial.println("error opening /configuracao.csv");
```

```
414     }
415
416
417     usa_encoder = 1; //liga encoders
418     recalibra_encoder = 0;
419 } else {
420     //escrever fun o que l calibra o atual
421     myFile_e = SD.open("/configuracao.csv", FILE_WRITE);
422
423     if (myFile_e) {
424         while (myFile_e.available()) {
425             atual = myFile.read();
426             if (cont < 3) {
427                 if (!atual.equals(String(59))) {
428                     if (atual.equals(String(46))) {
429                         leitura = leitura + ".";
430                     } else {
431                         valor = atual.toInt() - 48;
432                         leitura = leitura + String(valor);
433                     }
434                 }
435             } else {
436                 valor1 = leitura.toFloat();
437                 valor2[cont] = valor1;
438                 leitura = "";
439                 cont++;
440             }
441         }
442         myFile_e.close();
443     }
444     ajuste_encoder_t = valor2[0];
445     ajuste_encoder_l1 = valor2[1];
446     ajuste_encoder_l2 = valor2[2];
447 }
448 }
449
450 void handle_get() {
451     xin = server.arg("xi");
452     xfi = server.arg("xf");
```

```
453     yin = server.arg("yi");
454     yfi = server.arg("yf");
455     pl = server.arg("pl");
456     pt = server.arg("pt");
457     encoder_rec = server.arg("encoders");
458     if (server.arg("encoders")) {
459         recalibra_encoder = encoder_rec.toInt();
460     }
461     x_i = xin.toInt();
462     x_f = xfi.toInt();
463     y_i = yin.toInt();
464     y_f = yfi.toInt();
465     p_l = (pl.toInt() / 2) * 3;
466     p_t = (pt.toInt() / 2) * 3;
467     working = 0;
468     // In cio da configura o do nome do arquivo.
469
470
471     int contname = 1;
472     int test_name = 1;
473     int tamanho;
474
475     while (test_name) {
476         tamanho = filename.length() - 8;
477         filename.remove(8, tamanho);
478         filename.concat('_');
479         filename.concat(String(contname));
480         filename.concat(".csv");
481         Serial.println(filename);
482
483
484         if (SD.exists(filename.c_str())) {
485             contname++;
486         } else {
487             myFile = SD.open(filename.c_str(), FILE_WRITE);
488             test_name = 0;
489             //Test
490             //myFile.println("Salve");
491             myFile.close();
```

```
492     }
493 }
494     server.send(200, "text/html", Send_HTML_working());
495
496 }
497
498 void handle_OnConnect() {
499     if (working)
500         server.send(200, "text/html", SendHTML());
501     else
502         server.send(200, "text/html", SendHTML_leiturafinalizada
503             ());
504 }
505
506 void handle_leitura() {
507     if (working)
508         server.send(200, "text/html", index_html);
509     else
510         server.send(200, "text/html", SendHTML_leiturafinalizada
511             ());
512 }
513
514 void handle_NotFound() {
515     if (working)
516         server.send(404, "text/plain", "Not found");
517     else
518         server.send(200, "text/html", SendHTML_leiturafinalizada
519             ());
520 }
521 void handle_cancel() {
522     if (!working) {
523         working = 1;
524         SD.remove(filename.c_str());
525         server.send(200, "text/html", SendHTML());
526     }
527     else {
```

```
528     server.send(200, "text/html", SendHTML());
529 }
530 }
531
532 void leitura_finalizada() {
533     server.send(200, "text/html", SendHTML_leiturafinalizada())
534     ;
535 }
536 String SendHTML() {
537     String ptr = "<!DOCTYPE html> <html>\n";
538     ptr += "<head><meta name=\"viewport\" content=\"width=
539         device-width, initial-scale=1.0, user-scalable=no\">\n";
540     ptr += "<title>ProcSiMoS - Inicio</title>\n";
541     ptr += "<style>html { font-family: Helvetica; display:
542         inline-block; margin: 0px auto; text-align: center;}\n";
543     ptr += "body{margin-top: 50px;} h1 {color: #444444;margin:
544         50px auto 30px;} h3 {color: #444444;margin-bottom: 50px
545         ;}\n";
546     ptr += ".button {display: block;width: 100px;background-
547         color: #3498db;border: none;color: white;padding: 13px
548         30px;text-decoration: none;font-size: 25px;margin: 0px
549         auto 35px;cursor: pointer;border-radius: 4px;}\n";
550     ptr += ".button-on {background-color: #3498db;}\n";
551     ptr += ".button-on:active {background-color: #2980b9;}\n";
552     ptr += ".button-off {background-color: #34495e;}\n";
553     ptr += ".button-off:active {background-color: #2c3e50;}\n";
554     ptr += "p {font-size: 14px;color: #888;margin-bottom: 10px
555         ;}\n";
556     ptr += "</style>\n";
557     ptr += "</head>\n";
558     ptr += "<body>\n";
559     ptr += "<h1>ESP32 Web Server - ProcSiMoS Manipulador</h1>\n
560         ";
561     ptr += "<h3>Escolha a opcao desejada</h3>\n";
562     ptr += "<a class=\"button button-on\" href=\"/leitura\">
563         Leitura 3D</a>\n";
564     ptr += "<a class=\"button button-on\" href=\"/downloads\">
565         Download </a>\n";
```

```
555
556     ptr += "</body>\n";
557     ptr += "</html>\n";
558     return ptr;
559 }
560
561 String Send_HTML_working() {
562     String ptr = "<!DOCTYPE html> <html>\n";
563     ptr += "<head><meta name=\"viewport\" content=\"width=
           device-width, initial-scale=1.0, user-scalable=no\">\n";
564     ptr += "<meta http-equiv=\"Content-Type\" content=\"text/
           html;charset=utf-8\">\n";
565     ptr += "<title>ProcSiMoS - Aguarde</title>\n";
566     ptr += "<style>html { font-family: Helvetica; display:
           inline-block; margin: 0px auto; text-align: center;} \n";
567     ptr += "body{margin-top: 50px;} h1 {color: #444444;margin:
           50px auto 30px;} h3 {color: #444444;margin-bottom: 50px
           ;} \n";
568     ptr += ".button {display: block;width: 120px;background-
           color: #3498db;border: none;color: white;padding: 13px
           30px;text-decoration: none;font-size: 25px;margin: 0px
           auto 35px;cursor: pointer;border-radius: 4px;} \n";
569     ptr += ".button-on {background-color: #3498db;} \n";
570     ptr += ".button-on:active {background-color: #2980b9;} \n";
571     ptr += ".button-off {background-color: #34495e;} \n";
572     ptr += ".button-off:active {background-color: #2c3e50;} \n";
573     ptr += "p {font-size: 14px;color: #888;margin-bottom: 10px
           ;} \n";
574     ptr += "</style>\n";
575     ptr += "</head>\n";
576     ptr += "<body>\n";
577     ptr += "<h1>ESP32 Web Server - ProcSiMoS Manipulador</h1>\n
           ";
578     ptr += "<h3>O Manipulador est efetuando a leitura.</h3>\n
           ";
579     ptr += "<a class=\"button button-on:active\" href=\"/
           concluido\">Continuar</a>\n";
580
581     ptr += "</body>\n";
```

```
582     ptr += "</html>\n";
583     return ptr;
584 }
585
586 // fun    es do servidor de download
587
588
589 void handle_downloadsd() {
590     if (working) {
591         SendHTML_Header();
592         webpage += F("<a href='/download'><button>Download</button></a>");
593         webpage += F("<a href='/dir'><button>Directory</button></a>");
594         append_page_footer();
595         SendHTML_Content();
596         SendHTML_Stop(); // Stop is needed because no content
                           length was sent
597     }
598
599     else
600         server.send(200, "text/html", Send_HTML_working());
601
602 }
603
604
605 //-----
606 void File_Download() { // This gets called twice, the first
                           pass selects the input, the second pass then processes the
                           command line arguments
607     if (working) {
608         if (server.args() > 0 ) { // Arguments were received
609             if (server.hasArg("download")) SD_file_download(server.
                           arg(0));
610         }
611         else SelectInput("File Download", "Enter filename to
                           download", "download", "download");
612     } else {
```

```
613     server.send(200, "text/html", Send_HTML_working());
614 }
615 }
616 //~~~~~

617 void SD_file_download(String filename) {
618     if (SD_present) {
619         File download = SD.open("/") + filename);
620         if (download) {
621             server.setHeader("Content-Type", "text/text");
622             server.setHeader("Content-Disposition", "attachment;
                filename=" + filename);
623             server.setHeader("Connection", "close");
624             server.streamFile(download, "application/octet-stream")
                ;
625             download.close();
626         } else ReportFileNotPresent("download");
627     } else ReportSDNotPresent();
628 }
629 //~~~~~

630 void SendHTML_Header() {
631     server.setHeader("Cache-Control", "no-cache, no-store,
                must-revalidate");
632     server.setHeader("Pragma", "no-cache");
633     server.setHeader("Expires", "-1");
634     server.setContentLength(CONTENT_LENGTH_UNKNOWN);
635     server.send(200, "text/html", ""); // Empty content
        inhibits Content-length header so we have to close the
        socket ourselves.
636     append_page_header();
637     server.sendContent(webpage);
638     webpage = "";
639 }
640 //~~~~~

641 void SendHTML_Content() {
642     server.sendContent(webpage);
643     webpage = "";
```

```
644 }
645 //~~~~~

646 void SendHTML_Stop() {
647     server.sendContent("");
648     server.client().stop(); // Stop is needed because no
        content length was sent
649 }
650 //~~~~~

651 void SelectInput(String heading1, String heading2, String
        command, String arg_calling_name) {
652     SendHTML_Header();
653     webpage += F("<h3 class='rcorners_m'>"); webpage +=
        heading1 + "</h3><br>";
654     webpage += F("<h3>"); webpage += heading2 + "</h3>";
655     webpage += F("<FORM action='/'>"); webpage += command + "'
        method='post'>"; // Must match the calling argument e.g.
        '/chart' calls '/chart' after selection but with
        arguments!
656     webpage += F("<input type='text' name='"); webpage +=
        arg_calling_name; webpage += F("' value=' '><br>");
657     webpage += F("<type='submit' name='"); webpage +=
        arg_calling_name; webpage += F("' value=' '><br><br>");
658     append_page_footer();
659     SendHTML_Content();
660     SendHTML_Stop();
661 }
662 //~~~~~

663 void ReportSDNotPresent() {
664     SendHTML_Header();
665     webpage += F("<h3>No SD Card present</h3>");
666     webpage += F("<a href='/'>[Back]</a><br><br>");
667     append_page_footer();
668     SendHTML_Content();
669     SendHTML_Stop();
670 }
```

```
671 // ~~~~~
672 void ReportFileNotPresent(String target) {
673     SendHTML_Header();
674     webpage += F("<h3>File does not exist</h3>");
675     webpage += F("<a href='/'>"); webpage += target + "'>[Back]</
        a><br><br>";
676     append_page_footer();
677     SendHTML_Content();
678     SendHTML_Stop();
679 }
680
681 // fun es do diretorio
682
683 void SD_dir() {
684     if (SD_present) {
685         File root = SD.open("/");
686         if (root) {
687             root.rewindDirectory();
688             SendHTML_Header();
689             webpage += F("<h3 class='rcorners_m'>SD Card Contents</
                h3><br>");
690             webpage += F("<table align='center'>");
691             webpage += F("<tr><th>Name/Type</th><th style='width
                :20%'>Type File/Dir</th><th>File Size</th></tr>");
692             printDirectory("/", 0);
693             webpage += F("</table>");
694             SendHTML_Content();
695             root.close();
696         }
697         else
698         {
699             SendHTML_Header();
700             webpage += F("<h3>No Files Found</h3>");
701         }
702         append_page_footer();
703         SendHTML_Content();
704         SendHTML_Stop(); // Stop is needed because no content
                length was sent
```

```
705     } else ReportSDNotPresent();
706 }
707 //~~~~~

708 void printDirectory(const char * dirname, uint8_t levels) {
709     File root = SD.open(dirname);
710 #ifndef ESP8266
711     root.rewindDirectory(); //Only needed for ESP8266
712 #endif
713     if (!root) {
714         return;
715     }
716     if (!root.isDirectory()) {
717         return;
718     }
719     File file = root.openNextFile();
720     while (file) {
721         if (webpage.length() > 1000) {
722             SendHTML_Content();
723         }
724         if (file.isDirectory()) {
725             webpage += "<tr><td>" + String(file.isDirectory() ? "
726                 Dir" : "File") + "</td><td>" + String(file.name()) +
727                 "</td><td></td></tr>";
728             printDirectory(file.name(), levels - 1);
729         }
730         else
731         {
732             webpage += "<tr><td>" + String(file.name()) + "</td>";
733             webpage += "<td>" + String(file.isDirectory() ? "Dir" :
734                 "File") + "</td>";
735             int bytes = file.size();
736             String fsize = "";
737             if (bytes < 1024) fsize = String(
738                 bytes) + " B";
739             else if (bytes < (1024 * 1024)) fsize = String(
740                 bytes / 1024.0, 3) + " KB";
741             else if (bytes < (1024 * 1024 * 1024)) fsize = String(
742                 bytes / 1024.0 / 1024.0, 3) + " MB";
```

```
737     else                                     fsize = String(  
738         bytes / 1024.0 / 1024.0 / 1024.0, 3) + " GB";  
739     webpage += "<td>" + fsize + "</td></tr>";  
740     }  
741     file = root.openNextFile();  
742     }  
743     file.close();  
744 }  
745 String SendHTML_leiturafinalizada() {  
746     String ptr = "<!DOCTYPE html> <html>\n";  
747     ptr += "<head><meta name=\"viewport\" content=\"width=  
748         device-width, initial-scale=1.0, user-scalable=no\">\n";  
749     ptr += "<meta http-equiv=\"Content-Type\" content=\"text/  
750         html; charset=utf-8\">\n";  
751     ptr += "<meta http-equiv=\"refresh\" content=\"30\">\n";  
752     ptr += "<title>ProcSiMoS - Inicio</title>\n";  
753     ptr += "<style>html { font-family: Helvetica; display:  
754         inline-block; margin: 0px auto; text-align: center;}\n";  
755     ptr += "body{margin-top: 50px;} h1 {color: #444444;margin:  
756         50px auto 30px;} h3 {color: #444444;margin-bottom: 50px  
757         ;}\n";  
758     ptr += ".button {display: block;width: 100px;background-  
759         color: #3498db;border: none;color: white;padding: 13px  
760         30px;text-decoration: none;font-size: 25px;margin: 0px  
761         auto 35px;cursor: pointer;border-radius: 4px;}\n";  
762     ptr += ".button-on {background-color: #5de227;}\n";  
763     ptr += ".button-on:active {background-color: #2980b9;}\n";  
764     ptr += ".button-off {background-color: #34495e;}\n";  
765     ptr += ".button-off:active {background-color: #2c3e50;}\n";  
766     ptr += "p {font-size: 14px;color: #888;margin-bottom: 10px  
767         ;}\n";  
768     ptr += "</style>\n";  
769     ptr += "</head>\n";  
770     ptr += "<body>\n";  
771     ptr += "<h1>ESP32 Web Server - ProcSiMoS Manipulador</h1>\n  
772     ";  
773     if (working) {
```

```
764 ptr += "<h3>Sua Leitura foi finalizada com Sucesso. O
      nome do seu arquivo      " + filename + "</h3>\n";
765 ptr += "<a class=\"button button-on\" href=\"/\">Home</a
      >\n";
766 ptr += "<a class=\"button button-on\" href=\"/downloads
      \">Download </a>\n";
767 } else {
768 ptr += "<h3>O manipulador est  efetuando sua leitura.
      Por favor AGUARDE.</h3>\n";
769 ptr += "<a class=\"button button-off\" href=\"/cancel\">
      Cancelar </a>\n";
770 }
771
772 ptr += "</body>\n";
773 ptr += "</html>\n";
774 return ptr;
775 }
776
777 //CSS.h
778 void append_page_header() {
779     webpage = F("<!DOCTYPE html><html>");
780     webpage += F("<head>");
781     webpage += F("<title>File Server</title>"); // NOTE: 1em =
      16px
782     webpage += F("<meta name='viewport' content='user-scalable=
      yes,initial-scale=1.0,width=device-width'>");
783     webpage += F("<style>");
784     webpage += F("body{max-width:65%;margin:0 auto;font-family:
      arial;font-size:105%;text-align:center;color:blue;
      background-color:#F7F2Fd;}");
785     webpage += F("ul{list-style-type:none;margin:0.1em;padding
      :0;border-radius:0.375em;overflow:hidden;background-
      color:#dcade6;font-size:1em;}");
786     webpage += F("li{float:left;border-radius:0.375em;border-
      right:0.06em solid #bbb;}last-child {border-right:none;
      font-size:85%}");
787     webpage += F("li a{display: block;border-radius:0.375em;
      padding:0.44em 0.44em;text-decoration:none;font-size:85%
      }");
```

```
788 webpage += F("li a:hover{background-color:#EAE3EA;border-  
radius:0.375em;font-size:85%}");  
789 webpage += F("section {font-size:0.88em;}");  
790 webpage += F("h1{color:white;border-radius:0.5em;font-size  
:1em;padding:0.2em 0.2em;background:#558ED5;}");  
791 webpage += F("h2{color:orange;font-size:1.0em;}");  
792 webpage += F("h3{font-size:0.8em;}");  
793 webpage += F("table{font-family:arial,sans-serif;font-size  
:0.9em;border-collapse:collapse;width:85%;}");  
794 webpage += F("th,td {border:0.06em solid #dddddd;text-align  
:left;padding:0.3em;border-bottom:0.06em solid #dddddd  
;}");  
795 webpage += F("tr:nth-child(odd) {background-color:#eeeeee  
;}");  
796 webpage += F(".rcorners_n {border-radius:0.5em;background  
:#558ED5;padding:0.3em 0.3em;width:20%;color:white;font-  
size:75%;}");  
797 webpage += F(".rcorners_m {border-radius:0.5em;background  
:#558ED5;padding:0.3em 0.3em;width:50%;color:white;font-  
size:75%;}");  
798 webpage += F(".rcorners_w {border-radius:0.5em;background  
:#558ED5;padding:0.3em 0.3em;width:70%;color:white;font-  
size:75%;}");  
799 webpage += F(".column{float:left;width:50%;height:45%;}");  
800 webpage += F(".row:after{content:'';display:table;clear:  
both;}");  
801 webpage += F("*{box-sizing:border-box;}");  
802 webpage += F("footer{background-color:#eedfff;text-align:  
center;padding:0.3em 0.3em;border-radius:0.375em;font-  
size:60%;}");  
803 webpage += F("button{border-radius:0.5em;background:#558ED5  
;padding:0.3em 0.3em;width:20%;color:white;font-size  
:130%;}");  
804 webpage += F(".buttonsm{border-radius:0.5em;background:#558  
ED5;padding:0.3em 0.3em;width:9%; color:white;font-size  
:70%;}");  
805 webpage += F(".buttonm {border-radius:0.5em;background:#558  
ED5;padding:0.3em 0.3em;width:15%;color:white;font-size  
:70%;}");
```

```
806     webpage += F(".buttonw {border-radius:0.5em;background:#558
        ED5;padding:0.3em 0.3em;width:40%;color:white;font-size
            :70%;}");
807     webpage += F("a{font-size:75%;}");
808     webpage += F("p{font-size:75%;}");
809     webpage += F("</style></head><body><h1>File Server ");
        webpage += String(ServerVersion) + "</h1>";
810 }
811 //~~~~~

812 void append_page_footer(){ // Saves repeating many lines of
        code for HTML page footers
813     webpage += F("<ul>");
814     webpage += F("<li><a href='/'>Home</a></li>"); // Lower
        Menu bar command entries
815     webpage += F("<li><a href='/download'>Download</a></li>");
816     webpage += F("<li><a href='/dir'>Directory</a></li>");
817     webpage += F("</ul>");
818     webpage += "<footer>&copy;"+String(char(byte(0x40>>1)))+
        String(char(byte(0x88>>1)))+String(char(byte(0x5c>>1)))+
        String(char(byte(0x98>>1)))+String(char(byte(0x5c>>1)));
819     webpage += String(char((0x84>>1)))+String(char(byte(0xd2
        >>1)))+String(char(0xe4>>1))+String(char(0xc8>>1))+
        String(char(byte(0x40>>1)));
820     webpage += String(char(byte(0x64/2)))+String(char(byte(0x60
        >>1)))+String(char(byte(0x62>>1)))+String(char(0x70>>1))
        + "</footer>";
821     webpage += F("</body></html>");
822 }
823
824 //Network.h
825 // Adjust the following values to match your needs
826 // -----
827 #define     servername "fileserv" // Set your server's
        logical name here e.g. if myserver then address is http://
        myserver.local/
828 IPAddress local_ip(192,168,1,1);
829 IPAddress gateway(192,168,1,1);
830 IPAddress subnet(255,255,255,0);
```

```
831 IPAddress dns(192,168,1,1);          // Set your network DNS
      usually your Router base address
832 const char ssid_1[]      = "your_SSID1";
833 const char password_1[] = "your_PASSWORD_for SSID1";
834
835 const char ssid_2[]      = "your_SSID2";
836 const char password_2[] = "your_PASSWORD_for SSID2";
837
838 const char ssid_3[]      = "your_SSID3";
839 const char password_3[] = "your_PASSWORD_for SSID3";
840
841 const char ssid_4[]      = "your_SSID4";
842 const char password_4[] = "your_PASSWORD_for SSID4";
843
844 //Sys_Variables.h
845 #define ServerVersion "1.0"
846 String webpage = "";
847 bool    SD_present = false;
848
849 #define SD_CS_pin      5          // Use pin 5 on MH-T
      Live ESP32 version of Wemos D1 Mini for SD Card Chip-
      Select
```

## Algoritmo de Classificação SVM com parametrização Wavelet

```
1 clear;clc;close all;
2 load('base_de_dados.mat'); %Testando para db, sym, coif, bior
3 Max_taxa_de_acerto=0;
4
5 % load('membros_wavelets.mat');
6 load('novos_membros.mat');
7 membro_wav = novos_membros;
8 xxx = waitbar(0, 'Iniciando...');
9 contbar =0;
10
11 for fam =1:length(membro_wav)
12
13     maximo_level = wmaxlev(43, membro_wav{fam});
```

```
14 clear C L
15 if maximo_level > 0
16     for sd=1:50
17         [C(:,sd) L(:,sd)] = wavedec(z_total(:,sd),
18                                     maximo_level, membro_wav{fam});
19     end
20     for nivel_dec=3%1:maximo_level %N vel de
21         decomposi o
22         for aprox_ou_det =1%1:2 %1 aproxima o , 2
23             detalhe
24             clear hs
25             if isequal(aprox_ou_det,1)
26                 for sd=1:50
27                     hs(:,sd) = appcoef(C(:,sd), L(:,sd),
28                                         membro_wav{fam}, nivel_dec);
29                 end
30             else
31                 for sd=1:50
32                     hs(:,sd) = detcoef(C(:,sd), L(:,sd),
33                                         nivel_dec);
34                 end
35             end
36             [tam1 tam2] = size(hs);
37             for num_para=1:tam1
38
39                 Taxa_de_acerto = 0;
40                 for k=1:100
41                     seq_bons_ant= randsample(5,5);
42                     seq_ruins_ant = 5+randsample(5,5);
43                     contador = 0;
44                     seq_bons_ant = seq_bons_ant*5 - 5;
45                     seq_ruins_ant = seq_ruins_ant*5 - 5;
46                     for s=1:5
47                         for f=1:5
48                             contador = contador +1;
49                             seq_bons(contador) =
50                                 seq_bons_ant(s) + f;
51                             seq_ruins(contador) =
52                                 seq_ruins_ant(s) + f;
```

```
46         end
47     end
48     hs_treinamento = [hs(:,seq_bons(1:15))
49         ) hs(:,seq_ruins(1:15))];
50     hs_test = [hs(:,seq_bons(16:end)) hs
51         (:,seq_ruins(16:end))];
52     t_treinamento = [t(seq_bons(1:15)) t(
53         seq_ruins(1:15))];
54     t_test = [t(seq_bons(16:end)) t(
55         seq_ruins(16:end))];
56
57     hs_treinamento=hs_treinamento(1:
58         num_para,:);
59     hs_test=hs_test(1:num_para,:);
60
61     SVMModel = fitcsvm(hs_treinamento',
62         t_treinamento');
63
64     y=predict(SVMModel,hs_test');
65
66     %     y_2 = round(y);
67     error= y - t_test';
68     n_error = length(find(error~=0));
69     error_rate = n_error / length(t_test)
70         ;
71     Taxa_de_acerto_atual = 1-error_rate;
72     resultados(k) = Taxa_de_acerto_atual;
73
74     Taxa_de_acerto = Taxa_de_acerto +
75         Taxa_de_acerto_atual/100;
76     %     Taxa_de_acerto =
77         Taxa_de_acerto_atual;
78     contbar = contbar+1;
79     waitbar((contbar/562400),xxx,[num2str
80         (contbar/(5624)) '%']);
81 end
82 if Taxa_de_acerto > Max_taxa_de_acerto
83     Max_taxa_de_acerto = Taxa_de_acerto
84     ap_ou_det = aprox_ou_det;
```

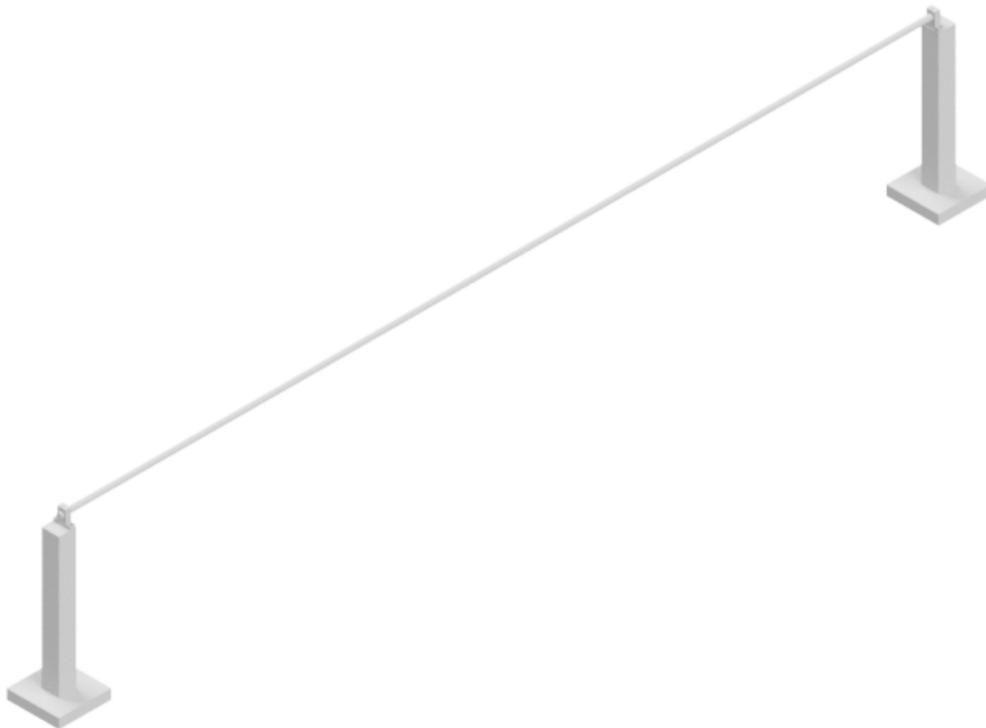
```
75         nivel_usado = nivel_dec;
76         resultados_melhores = resultados;
77         membro_correto = membro_wav{fam};
78         num_para_usado = num_para;
79         save('configuracao_svm_wavelet.mat', '
           Max_taxa_de_acerto', 'ap_ou_det', '
           nivel_usado', 'membro_correto', '
           resultados_melhores', '
           num_para_usado');
80
81         end
82     end
83 end
84
85     end
86 else
87     contbar=contbar+200;
88     waitbar((contbar/21800),xxx,[num2str(contbar/(21800))
           '%']);
89     fprintf('N vel m ximo igual a zero\n');
90 end
91 end
```

## Apêndice C

Neste apêndice serão apresentados detalhes sobre o projeto do manipulador cartesiano.

A Figura 51 a seguir mostra uma montagem formada por uma barra de 2500mm com dois conjuntos formados por um suporte de 16mm acoplado em uma estrutura de suspensão que foi usinada para o sistema.

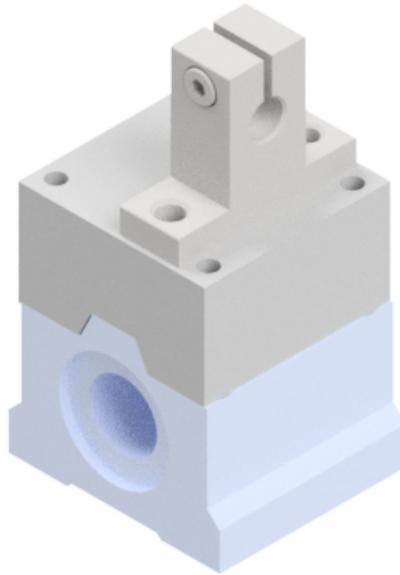
Figura 51 – Montagem Barra 2500mm, suporte de 16mm e suspensão



Fonte: Do Autor

Para fornecer suporte a um dos eixos de 8mm é feito um outro conjunto de Pillow block e suporte, porém sem o motor de passo. A Figura 52 demonstra o conjunto criado.

Figura 52 – Montagem Suporte 8mm e Pillow Block 16mm



Fonte: Do Autor

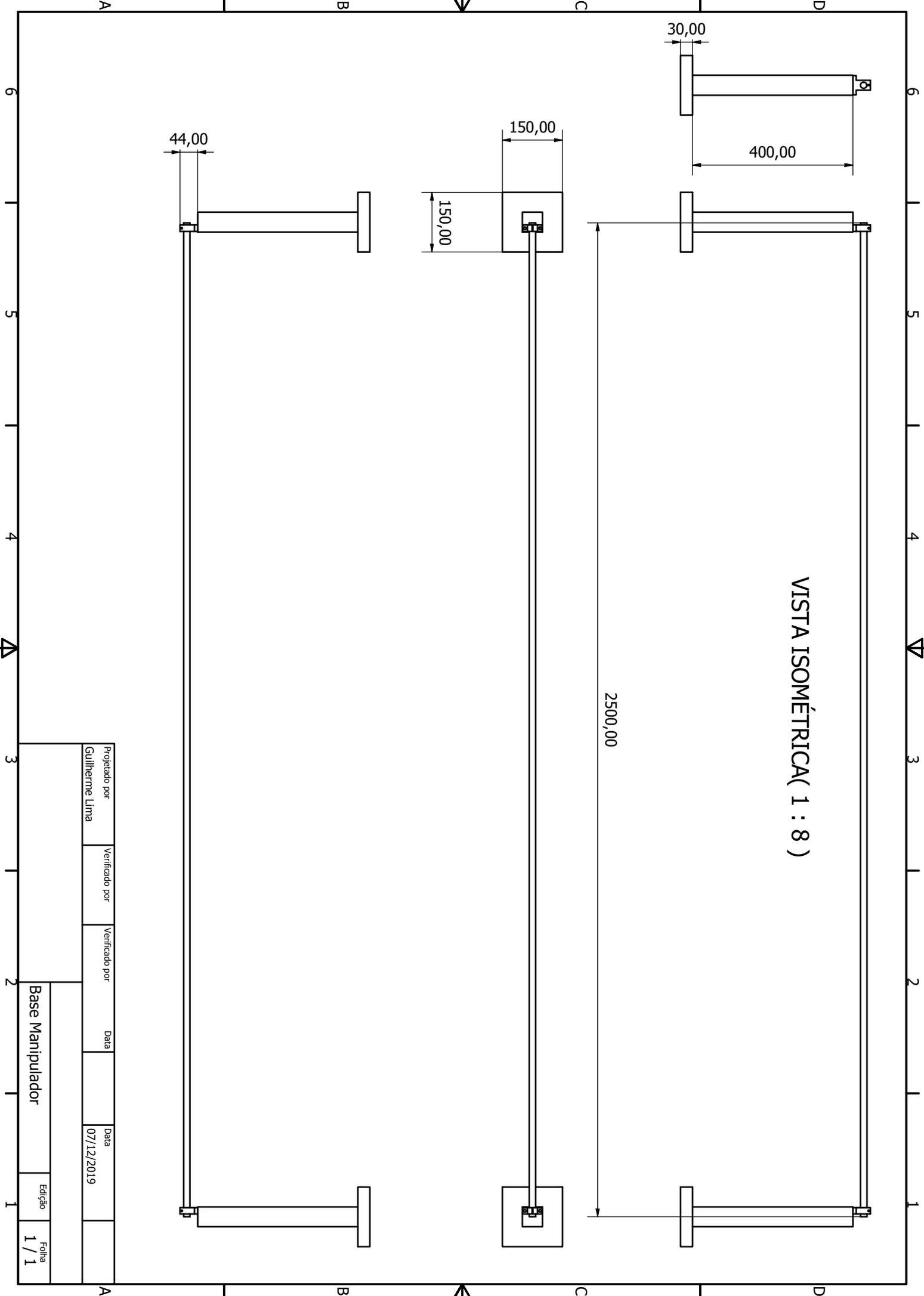
A seguir serão apresentados os desenhos técnicos das montagens realizadas no *Autodesk Inventor*, os desenhos são os mesmos apresentados visualmente nas figuras 22, 24, 51, 52, 53.

Para servir de suporte ao movimento longitudinal do sensor, serão utilizadas duas barras de 1000mm com dois Pillow Blocks 8mm, como mostrado na Figura 53.

Figura 53 – Montagem Barra 8mm e Pillow Block 8mm



Fonte: Do Autor

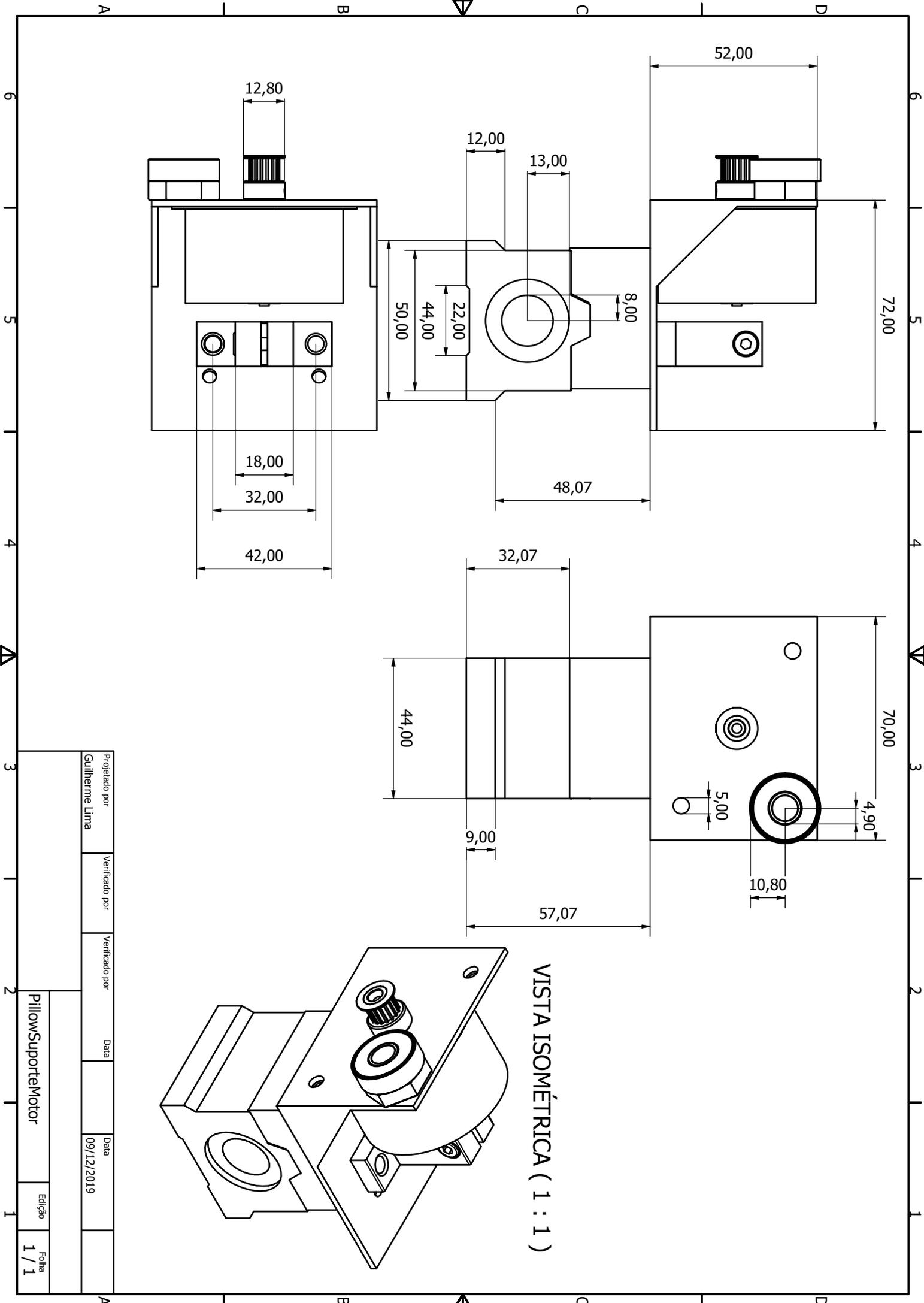


VISTA ISOMÉTRICA ( 1 : 8 )

Projetado por	Verificado por	Verificado por	Data	Data
Guilherme Lima				07/12/2019

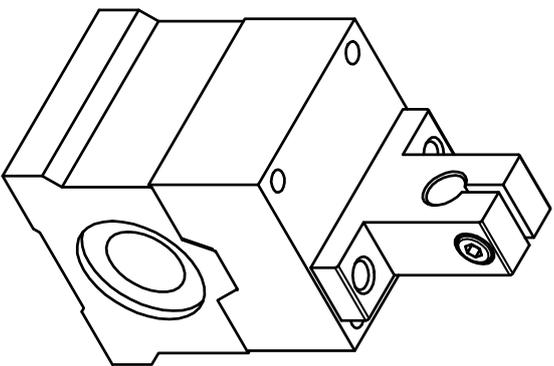
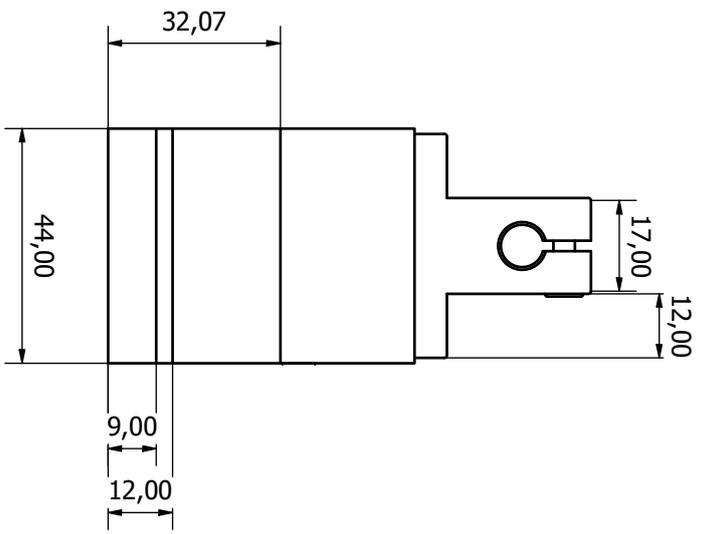
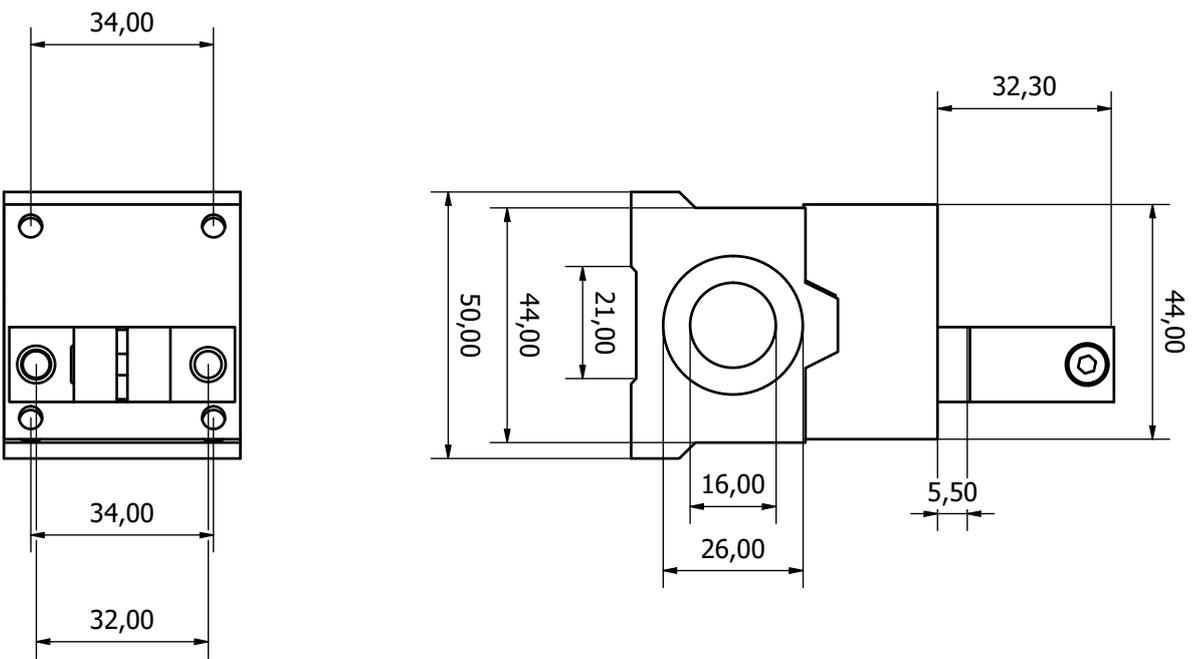
Base Manipulador

Edição  
1 / 1



Projetado por	Verificado por	Verificado por	Data	Data	Edição	Folha
Guilherme Lima				09/12/2019	1	1 / 1

PillowsuporteMotor



VISTA ISOMÉTRICA ( 1 : 1 )

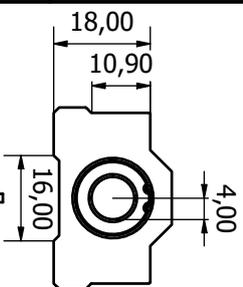
Projetado por	Verificado por	Verificado por	Data	Data	
Guilherme Lima				09/12/2019	

Pillowsuporte

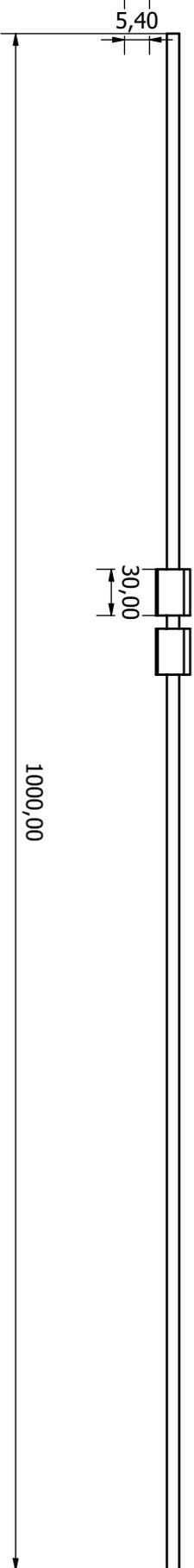
Edição

Folha  
1 / 1

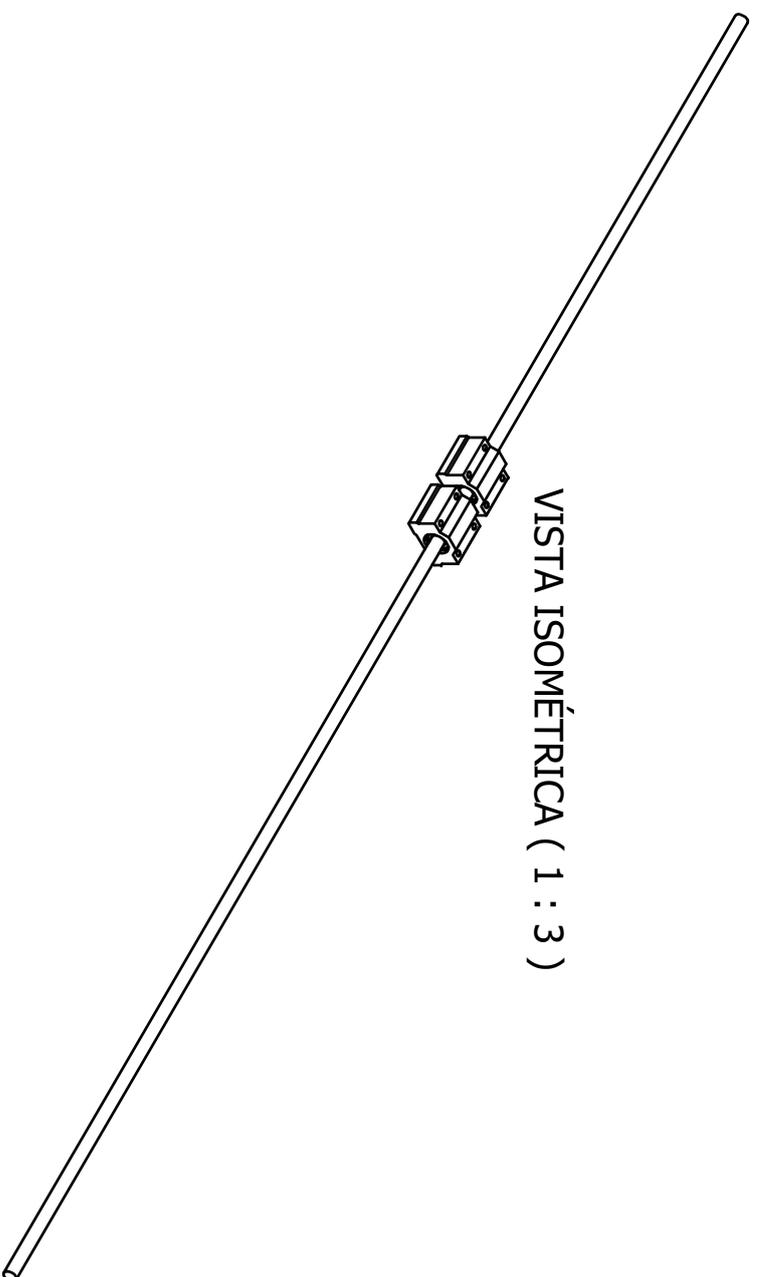
VISTA FRONTAL ( 1 : 1 )



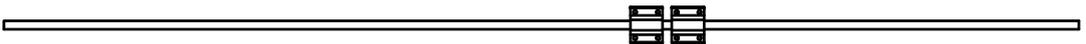
VISTA LATERAL ( 1 : 3 )



VISTA ISOMÉTRICA ( 1 : 3 )

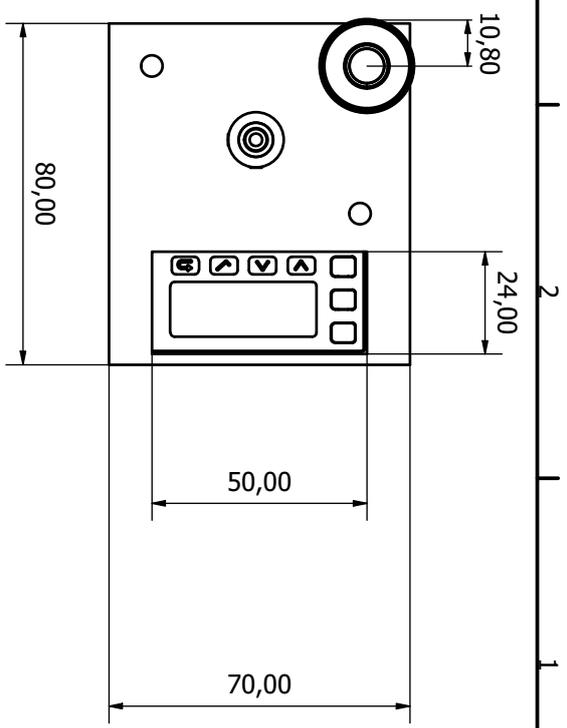
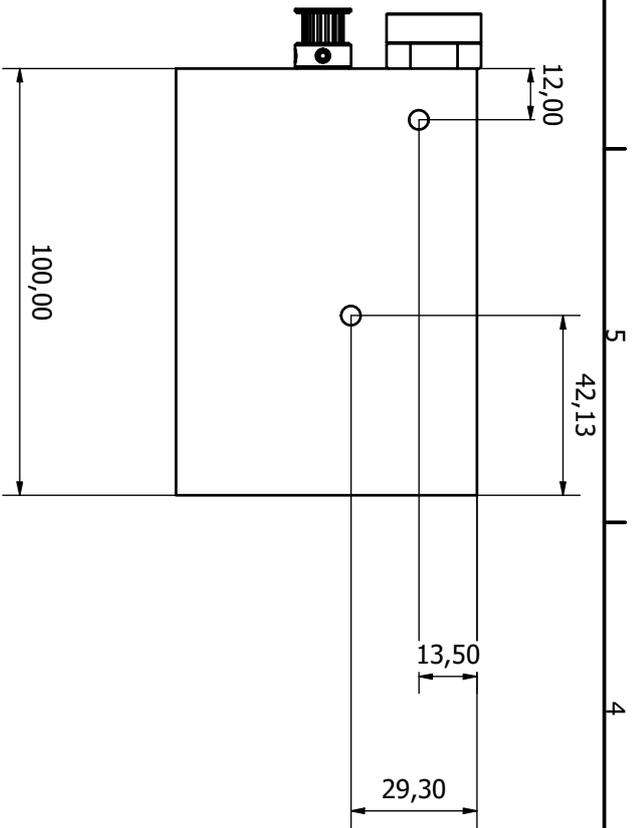


VISTA SUPERIOR ( 1 : 5 )

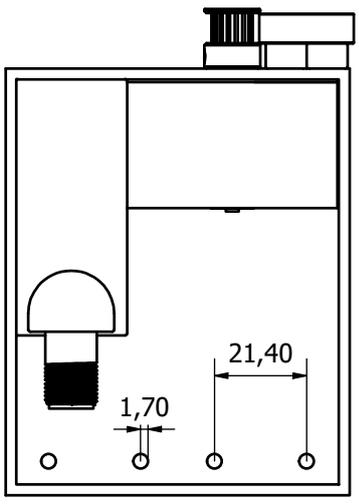
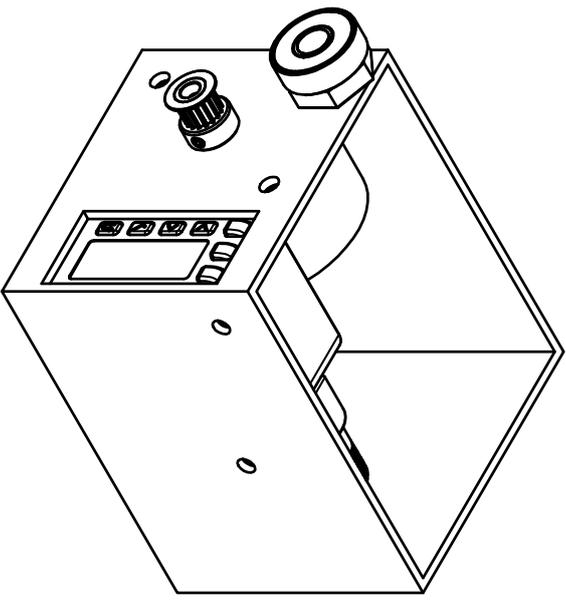


Projetado por	Verificado por	Verificado por	Data	Data	Edição	Folha
Guilherme Lima				09/12/2019		1 / 1

Barra8Pillow8



VISTA ISOMÉTRICA ( 4:5 )



Projetado por	Verificado por	Verificado por	Data	Data	Edição	Folha
Guilherme Lima				09/12/2019		1 / 1

Kit Sensor