

Estimação do Volume de um Espaço Interior com LiDAR

Jaqueline Bierende

Dissertação apresentada à Escola Superior de Tecnologia e de Gestão de Bragança para obtenção do Grau de Mestre em Engenharia Industrial - Eletrotécnica. Trabalho desenvolvido durante Programa de Dupla Diplomação entre Instituto Politécnico de Bragança (IPB) e Universidade Tecnológica Federal do Paraná (UTFPR).

Trabalho orientado por:

Prof. Dr. José Luís Sousa de Magalhães Lima

Prof^ª. Dr. Ana Isabel Pereira

Prof. Msc. João Braun Neto

Prof. Dr. Fábio Luiz Bertotti

Bragança

2022

Estimação do Volume de um Espaço Interior com LiDAR

Jaqueline Bierende

Dissertação apresentada à Escola Superior de Tecnologia e de Gestão de Bragança para obtenção do Grau de Mestre em Engenharia Industrial - Eletrotécnica. Trabalho desenvolvido durante Programa de Dupla Diplomação entre Instituto Politécnico de Bragança (IPB) e Universidade Tecnológica Federal do Paraná (UTFPR).

Trabalho orientado por:

Prof. Dr. José Luís Sousa de Magalhães Lima

Prof^ª. Dr. Ana Isabel Pereira

Prof. Msc. João Braun Neto

Prof. Dr. Fábio Luiz Bertotti

Bragança

2022

Dedicatória

À minha família

Anestor, Anilva e Felipe ...

Agradecimentos

A realização desse trabalho não seria possível sem o amor e o incentivo incondicional da minha família. Pai e mãe, obrigada pelo suporte e por acreditarem em mim. Ao meu irmão, Felipe, obrigada por me preparar para esse momento. A minha cunhada, Karine, por ter sido um ombro amigo quando mais precisei. Vocês são a minha base.

Gostaria de agradecer meus orientadores Dr. José Lima e Dra. Ana Pereira, e o professor João Braun pelo apoio e por oferecerem as condições necessárias para que esse trabalho fosse desenvolvido. Agradeço também meu co-orientador Dr. Fábio Bertotti pela disponibilidade.

Um obrigada especial as minhas amigas de infância, Ana, Gabriely e Eduarda, que estiveram ao meu lado desde o início e mesmo distantes se fizeram presente para dividirmos mais uma conquista.

Ana Carolina, Cristiano, Isadora e Mariah, o que construímos permanecerá para sempre no meu coração. Obrigada por tornarem a nossa casa um verdadeiro lar, pela troca de aprendizado diário e pelo apoio nos momentos difíceis.

Às minhas irmãs de vida Ana Flávia e Madu, por terem feito meus dias melhores e mais leves em Pato Branco e por compartilharmos essa experiência juntas. Vocês são meu porto seguro.

Às amigadas que fiz aqui, Elias, Lucas, Gabriel, Marcelo, Dalila, Kelly, Maicon, Filipe, vocês tornaram essa experiência ainda mais feliz, carregue vocês no meu coração e sou grata por todos os momentos vividos juntos.

A todos que se fizeram presentes de forma sincera.

A Deus.

Abstract

3D scanning is a task of great importance for our modern society and has brought great advance when it comes to precision of inventory. These sensors measure and map the material surface continuously, allowing real-time inventory monitoring.

This work presents the simulation of a low-cost 3D scanning system based on time-of-flight, that after a voxelization process, performs the volume estimation of an object filled indoor space.

The Evolid system consists of a 2D LiDAR scanner performing 180° variation. Through coupling in a stepper motor, the sensor also moves on the tilt axis, allowing three-dimensional mapping. The system and methodology developed were implemented and validated through SimTwo, a realistic simulator. Subsequently, the approach for data collection in the real environment was tested, demonstrating that a developed solution met the proposed objectives. Finally, possible works to be foreseen in future works are highlighted.

Keywords: LiDAR; scanner 2D; Simulator; Volume Estimation; Voxelization; Point Cloud;

Resumo

A digitalização tridimensional é uma tarefa de grande importância para nossa sociedade moderna e trouxe grande avanço quando se trata de precisão de inventário. Esses sensores medem e mapeiam a superfície do material continuamente, permitindo o monitoramento do produto em tempo real.

Este trabalho apresenta a simulação de um sistema de escaneamento 3D de baixo custo baseado em tempo de voo, que após um processo de voxelização, realiza a estimativa do volume de um espaço interno preenchido por um objeto.

O sistema Evolid consiste num scanner 2D LiDAR que realiza um varrimento de 180° e através de um suporte acoplado num motor de passo, o sensor também se movimenta no eixo de inclinação, permitindo o mapeamento tridimensional. O sistema e a metodologia desenvolvidos foram implementados e validados através do SimTwo, um simulador realista. Posteriormente, foi testado a abordagem para aquisição de dados no ambiente real, demonstrando que a solução desenvolvida atende aos objetivos propostos. Por fim, são destacadas possíveis melhorias a serem implementadas em trabalhos futuros.

Palavras-chave: LiDAR; scanner 2D; Simulador; Estimação de Volume; Voxelização; Nuvem de Pontos

Conteúdo

Agradecimentos	vii
Abstract	ix
Resumo	xi
1 Introdução	2
1.1 Objetivos	3
1.2 Estrutura do Documento	4
2 Estado da Arte	7
2.1 Sensores Volumétricos	7
2.1.1 LiDAR	8
2.1.2 Radar	9
2.1.3 Sonar	10
2.2 Mapeamento em Três Dimensões	11
2.3 Point Cloud	13
2.4 Laser Scanner para Estimativa de Volume	15
3 Tecnologias e Métodos Explorados	19
3.1 Equipamentos Usados	19
3.1.1 XV-11 Laser Scanner	20
3.1.2 ESP32	21

3.1.3	Servo Motor	22
3.1.4	Motor de Passo	22
3.2	Métodos e Técnicas Usadas	24
3.2.1	Mudança de Coordenadas	24
3.2.2	Máquina de Estados	25
3.3	Ambiente de Simulação	26
3.3.1	SimTwo	26
3.3.2	Estrutura da Simulação	27
4	Sistema Evolid	29
4.1	Arquitetura Evolid	29
4.2	Aquisição de Dados	30
4.2.1	Construção do Ambiente de Simulação	31
4.2.2	Máquina de Estados	36
4.2.3	Hardware	37
4.3	Processamento dos Dados	38
4.3.1	Mudança de Coordenada	39
4.3.2	Filtragem de Dados	40
4.4	Estimação do Volume	41
4.4.1	Voxelização	41
4.4.2	Cálculo do Volume	44
4.5	Modelagem dos Suportes	44
4.5.1	Suporte para o LiDAR	45
4.5.2	Suporte para o Motor de Passo	47
5	Resultados	49
5.1	Etapa Simulada	49
5.1.1	Posição do Sensor	49
5.1.2	Resolução da Grelha	52
5.1.3	Posição do Cubo	56

5.2	Etapa Real	58
5.2.1	Teste com Servo Motor	58
5.2.2	Teste Com Motor de Passo	60
5.3	Teste com LiDAR Neato XV 11	63
5.4	Validação da Máquina de Estados	65
6	Conclusão e Trabalho Futuros	69
6.1	Trabalhos Futuros	70

Lista de Tabelas

3.1	Informações Técnicas do sensor LIDAR modelo XV-11.	20
3.2	Especificações do Microcontrolador ESP32.	21
5.1	Especificações para cada simulação.	52
5.2	Resultados para escolha da resolução da Grelha	52
5.3	Posições do cubo para cada simulação	56
5.4	Posições do cubo para cada simulação	56
5.5	Resultados Motor de Passo	63

Lista de Figuras

2.1	Modelos de LiDAR [11].	8
2.2	Medição por tempo de vôo [10].	10
2.3	Sensor Ultrassônico HC-SR04 [14].	10
2.4	Diferentes acionamentos e faixas de varrimento do LRF [18].	12
2.5	Exemplo de uma <i>point cloud</i> [27].	13
2.6	Representação nuvem de pontos.	14
2.7	Resolução de uma grelha de voxel. [18]	15
2.8	Laser 3D para estimativa de armazenamento em silos [33].	17
3.1	Scanner LIDAR modelo XV-11 [39].	20
3.2	Descrição dos pinos do ESP32 [43].	22
3.3	Servo Motor modelo S06NF STD.	23
3.4	Motor de passo modelo Nema 17 42BYGHW811 [45].	23
3.5	Driver DRI0023 para o motor de passo [46].	24
3.6	Coordenada Esférica para Coordenada Cartesiana. Adaptado de [48].	25
3.7	Exemplo de uma máquina de estados genérica [50].	26
3.8	Software SimTwo: Janelas do aplicativo no sentido horário a partir do canto superior esquerdo: <i>world view</i> , <i>code editor</i> , <i>configuration</i> , <i>chart</i> , and <i>spreadsheet</i>	27
4.1	Plataforma Evolid.	30
4.2	Arquitetura do Software.	31
4.3	Arquitetura do Hardware.	31

4.4	Posição do sensor laser na simulação.	32
4.5	Visualização superior das paredes no software <i>SimTwo</i>	33
4.6	Scanner a laser.	34
4.7	Máquina de Estados do Sistema Evolid.	36
4.8	Ligações do LiDAR com a placa ESP32.	37
4.9	Ligações do Servo Motor com a placa ESP32.	38
4.10	Conexões entre o Motor de Passo, ESP32 e Driver.	39
4.11	Sistema de coordenadas do sensor liDAR.	40
4.12	Grades de voxel no plano xy	42
4.13	Grelhas de voxel em três dimensões.	43
4.14	Diagrama para resolução da grelha	43
4.15	Suporte desenvolvido no software <i>SolidWorks</i>	45
4.16	Diferentes perspectivas do módulo com o sensor LiDAR.	46
4.17	Partes da montagem do módulo.	46
4.18	Suporte para o Motor de Passo desenvolvido no software <i>SolidWorks</i>	47
5.1	Pontos de testes.	50
5.2	Ambiente construído para simulação.	50
5.3	Point Cloud obtida no software Matlab	51
5.4	Nuvem de pontos.	53
5.5	Grelhas de Voxel com resolução adaptativa.	54
5.6	Grelhas de Voxel com resolução fixa de 0,02.	55
5.7	Nuvem de pontos para diferentes posições do cubo.	57
5.8	Realização do teste com o servo motor e o laser infra-vermelho.	59
5.9	Resultado dos testes para o servo motor.	59
5.10	Diagrama representativo do teste com servo motor.	60
5.11	Realização do teste com o motor de passo e o laser infra-vermelho.	62
5.12	Diagrama representativo do teste com motor de passo.	62
5.13	Realização do teste com LiDAR.	64

5.14	Distância Medida pelo LiDAR Neato XV 11.	64
5.15	Sistema utilizado para teste.	65
5.16	Resultado da aquisição e processamento da máquina de estados.	66
5.17	Resultado da aquisição e processamento da máquina de estados com um objeto cilíndrico.	66
5.18	Resultado da aquisição e processamento da máquina de estados com uma régua.	67

Lista de Listagens

4.1	Parâmetros Paredes.	33
4.2	Parâmetros LiDAR.	33
4.3	Parâmetros Objetos.	35
5.1	Código Acionamento Servo Motor	58
5.2	Código Acionamento Motor Passo	61

Capítulo 1

Introdução

Com o constante avanço tecnológico presente nas indústrias, onde sempre se procuram buscam formas de otimizar a produção, a tecnologia de scanner destaca-se quando se trata de precisão de inventário. A digitalização 3D foi desenvolvida na tentativa de recriar com precisão as superfícies de objetos e lugares, podendo assim medir e mapear a superfície do material continuamente, permitindo o monitoramento do estoque em tempo real [1], sem a necessidade do deslocamento até o local.

A reconstrução precisa da forma 3D e a estimativa de volume são importantes em muitas aplicações, por exemplo, com estimativa de produtos e matérias-primas. O LiDAR - *Light Detection And Ranging* (detecção de luz e alcance) é um scanner a laser, também conhecido como telêmetro a laser (*Laser Rangefinder* - LRF), e é uma ferramenta atrativa e eficiente, pois o sensor delimita o volume de interesse fornecendo medidas de distância relativa do seu ponto de origem.

Para scanners LiDAR, o método de triangulação e tempo de voo (*Time of Flight* - ToF) são duas abordagens comuns. Os sistemas de triangulação têm alcance e variação de profundidade restritos, mas oferecem maior precisão de medida. Os sistemas ToF têm um amplo alcance e variação de profundidade, mas menor precisão [2], [3]. O tempo de voo é o tipo de LRF mais abundante no mercado devido à relação custo/benefício. A precisão dos sistemas baseados em ToF é determinada principalmente pela capacidade do sensor de medir com precisão a diferença entre o tempo de emissão do laser e a recepção

do sinal refletido, o que possibilita estimar a posição do objeto no espaço, o qual é o caso desenvolvido neste trabalho.

Essa precisão durante a aquisição pode ser afetada pela interferência da luz, ruído e o ângulo de incidência do feixe projetado no objeto ser muito oblíquo. Portanto, um ambiente controlado e um circuito de alta qualidade devem ser usados para realizar varrimentos em ambientes mais complexos que, em geral, esses tipos de sistemas de varredura 3D possuem um custo elevado [3]. Por esta razão, um sistema de scanner de três dimensões (3D) de baixo custo, que utiliza um LiDAR de duas dimensões (2D) com abordagem ToF, foi utilizado neste trabalho.

Os scanners a laser – como a maioria dos outros dispositivos de medição – vêm numa variedade de modelos. Dessa forma, para selecionar o modelo certo, é extremamente importante planejar a forma em que será usado os dados obtidos do dispositivo. O local de instalação também é importante, pois a tecnologia mede vários pontos, portanto, o dispositivo precisa ter uma visão clara da superfície do material [1]. O objetivo e escopo deste trabalho de pesquisa é em ambientes pequenos, como um depósito, onde não é necessário alto desempenho. Nesse contexto, por mais que existam diversos sistemas que realizem a captura de uma nuvem de pontos, os métodos ópticos se destacam para o estudo do objeto pois oferecem uma forma eficiente, sem contato e de longa distância tanto em ambientes externos quanto internos [4], [5]. As vantagens de utilizar nuvens de pontos ao invés de outras abordagens existentes na procura por representação tridimensional são que elas permitem recriar um ambiente e observá-la de qualquer ângulo [4], [6].

O sistema e a metodologia desenvolvidos são validados através de simulação proveniente do simulador SimTwo [7] e implementados em ambiente real.

1.1 Objetivos

Este trabalho visa desenvolver um sistema de mapeamento 3D capaz de efetuar a medição de volume de um determinado local a partir de uma nuvem de pontos gerada por um sensor LiDAR 2D.

Sendo este o objetivo geral, o trabalho será dividido nas seguintes tarefas específicas:

- Desenvolver um sistema de digitalização de um ambiente a 3D.
- Desenvolver uma metodologia de cálculo do volume ocupado num ambiente tridimensional.
- Testar a abordagem desenvolvida para validação da estimativa do volume em ambiente de simulação.
- Explorar ambientes de simulação.
- Validar o código para leitura dos dados laser.
- Validar a metodologia para mapeamento em três dimensões em ambiente real.

1.2 Estrutura do Documento

O presente trabalho é dividido em seis capítulos.

No **Capítulo 1** é apresentada uma introdução ao tema que será abordado, seguido dos objetivos do trabalho e da estrutura em que o documento está organizado.

No **Capítulo 2**, é realizada uma revisão bibliográfica sobre os principais sensores para medição volumétrica. Em seguida, estudos sobre mapeamento em três dimensões e nuvem de pontos e por fim, alguns trabalhos desenvolvidos para estimativa de volume através de laser scanner.

O **Capítulo 3** aborda detalhes sobre os componentes selecionados para o sistema e as técnicas desenvolvidas. O software SimTwo é apresentado, contemplando a criação da estrutura para a simulação.

No **Capítulo 4**, denominado Sistema Evolid, começa descrevendo a arquitetura do sistema. Em seguida, a aquisição de dados usadas no hardware e no software, o processamento e a estimação do volume são detalhados.

No **Capítulo 5**, apresenta a validação da abordagem no software e os testes para análise dos motores. Depois, o resultado da implementação da máquina de estados no real.

O **Capítulo 6** traz as conclusões sobre os resultados obtidos nos testes realizados e discute propostas de trabalhos futuros.

Capítulo 2

Estado da Arte

Este capítulo contextualiza através de um referencial teórico os diferentes tipos de sensores utilizados para medição de volume de um ambiente interno. Também apresenta trabalhos desenvolvidos em que o sensor LiDAR e o mapeamento em três dimensões vem sendo aplicado. Além disso, aborda a metodologia de análise de dados através da nuvem de pontos, a qual será utilizada no desenvolvimento desse projeto. Ao final do capítulo, são apresentados estudos sobre a aplicação da tecnologia de scanner a laser para determinar parâmetros volumétricos.

2.1 Sensores Volumétricos

Há uma grande diversidade de dispositivos que podem ser utilizados para a finalidade de obter uma medição precisa de forma a facilitar a representação virtual de um ambiente ou objeto. Esta seção apresenta alguns sensores usados para o cálculo volumétrico a partir do mapeamento em três dimensões e do princípio do tempo de vôo de um objeto: Radar, LiDAR e Sonar. Esses sensores são conhecidos como uma tecnologia ativa em que fornecem sua própria fonte de energia para iluminação [8].

2.1.1 LiDAR

O funcionamento de um sensor LiDAR, também conhecido como telêmetro a laser, é através da emissão de pulsos laser (*Light Amplification by Stimulated Emission of Radiation*) que associados a informação de posicionamento, produz uma nuvem de pontos bidimensionais ou tridimensionais de milhões de pontos espaciais com precisão de milímetros [9]. Uma vez que as leituras individuais são processadas, os dados ficam armazenados num arquivo de nuvem de pontos.

A medição do sensor pode ser efetuada em até três dimensões. Para a medição direta, o laser é direcionado num alvo ou refletor, obtendo então somente o valor da distância, tornando o sensor unidimensional. Ao girar ou mover o feixe de luz num plano, então além da distância será obtido um ângulo, portanto, haverá duas dimensões. Para três dimensões é necessário a inclinação do sensor, obtendo então informações sobre a distância e posição nas direções x , y e z [10]. A Figura 2.1 apresenta LiDAR comerciais para cada configuração.



Figura 2.1: Modelos de LiDAR [11].

As fontes de erro de um LRF incluem ruído elétrico, variações na forma do pulso e amplitude dos pulsos ópticos refletidos e mudanças no atraso dos sinais ópticos e elétricos [12]. A transmissão através de meios com índices de refração altos (por exemplo, vidro) levará a medições de profundidade incorretas, dessa forma, a precisão de um LRF depende

de cada ponto amostrado no ambiente ter refletividade suficiente para gerar um pulso de retorno detectável, isso acontece porque, pulsos muito fracos resultam em espaços vazios na nuvem de pontos resultante que precisam ser preenchidos de alguma forma. Num ambiente construído, edifícios com janelas apresentam um desafio para um LRF à medida que o feixe entra na janela [13].

Tempo de Vôo

Dentro da ampla gama de tipos desse sensor o que diferencia cada modelo é a estimação da distância e as dimensões em que o ambiente é descrito. ToF é o tipo de LRF mais abundante do mercado por sua relação custo-benefício. A precisão dos sistemas baseados em ToF é determinada principalmente pela capacidade do sensor de medir com precisão a diferença entre o tempo de emissão do laser e a recepção do sinal refletido, o que possibilita a estimativa da posição do objeto no espaço, como é o caso desenvolvido nesse trabalho.

De acordo com o conhecimento da velocidade da luz, a distância entre o objeto e o sensor pode ser encontrada pela fórmula:

$$D = \frac{c \times ToF}{2} \quad (2.1)$$

onde D é a distância do objeto, c é a velocidade da luz e ToF representa o *Time of Flight*.

A reflexão do objeto depende diretamente da constituição do objeto a ser medido. Por este motivo, é necessário a escolha correta do método de medição para o sensor LiDAR. O método de tempo de vôo faz uso do tempo da menor distância entre um pulso de luz entre emissor, objeto e receptor, conforme Figura 2.2 [10].

2.1.2 Radar

O sistema de RADAR - *Radio Detection and Ranging*, também utiliza da tecnologia de tempo de vôo, apresentada na Figura 2.2, porém os pulsos emitidos são na forma de ondas eletromagnéticas que também se propagam na velocidade da luz. Esses sensores possuem uma baixa resolução de tal forma a não se tornar possível a classificação adequada do

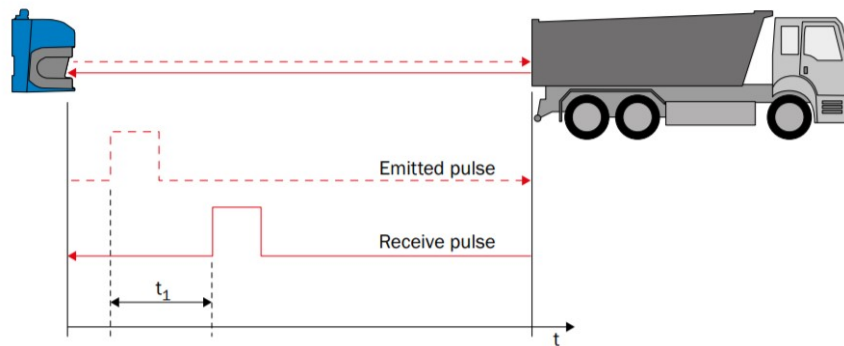


Figura 2.2: Medição por tempo de vôo [10].

objeto quando o nível de complexidade é maior.

2.1.3 Sonar

O princípio do Sonar - *Sound Navigation and Ranging* é o mesmo do LRF descrito na Equação 2.1, com a diferença da troca velocidade da luz pela velocidade do som no meio em que o sensor se encontra. Os sensores ultrassônicos emitem ondas sonoras a uma frequência de 20kHz até 40kHz para medir a distância.

O sonar é capaz de perceber através de objetos ao contrário do LiDAR e são relativamente baratos, entretanto não tem a resolução para detectar pequenos objetos. Na Figura 2.3 é visualizado um sensor ultrassônico.



Figura 2.3: Sensor Ultrassônico HC-SR04 [14].

2.2 Mapeamento em Três Dimensões

Atualmente, o sistema de varrimento laser tem sido amplamente utilizado na reconstrução em três dimensões de ambientes internos/externos e também no mapeamento de objetos, permitindo a interação com a réplica através de um software [15].

Uma otimização óbvia do sistema de medição é utilizar o método que melhor se adapta para cada aplicação. Há diversas formas de garantir um perfil de campo completo através de um sensor, as mais conhecidas são: instalar o mesmo em diferentes pontos de vistas, obter um que já possui os três eixos de rotação ou adicionar uma rotação ao laser que possui varrimento em duas dimensões.

Ao mapear o mesmo equipamento em diferentes pontos de vistas, é gerado mais de uma nuvem de pontos posicionados no mesmo local, o que pode acarretar num conjunto de desalinhamento angular e linear entre cada par de nuvem. Pavan e Santos [16] desenvolveram um modelo matemático para que através de um método automático de registro de pares de nuvens de pontos tridimensionais se torne possível a minimização dos erros do processo de varrimento.

O mapeamento em três dimensões através da utilização de um sensor horizontal é possível ao combinar a rotação no seu eixo com a rotação da posição inclinada, resultando em mais pontos de medição. Dessa forma, é possível obter a distância, ângulo no plano horizontal e ângulo nos planos do espaço tridimensional [10], [17]. Além disso, a rotação leva a uma alta densidade de medição próximas a esse eixo. Dependendo do método de varrimento, essa distribuição de densidade pode focar numa determinada área de interesse ou levar a uma exploração não desejada dos recursos do varrimento. Os sistemas de mapeamento tridimensional com dispositivos de tempo de voo a laser foram comparados em diferentes situações em [18]. A Figura 2.4 demonstra quatro possíveis formas dessas combinações.

Existem atualmente diversos sistemas de ground-truth que permitem determinar com alta precisão a posição dos robôs. Atualmente, LRFs e câmeras são amplamente aplicados na robótica móvel com o objetivo de detectar obstáculos e evitar colisões.

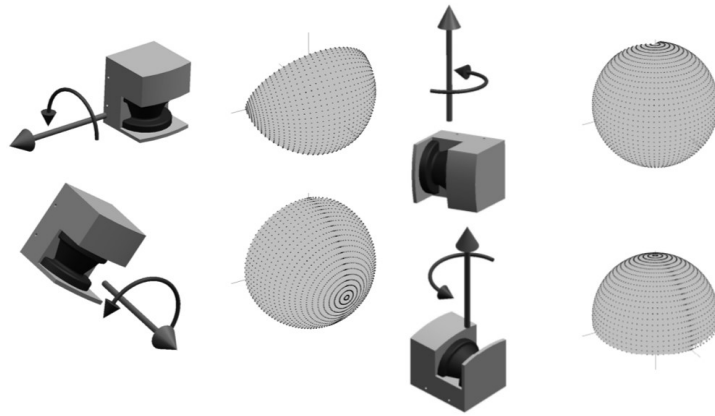


Figura 2.4: Diferentes acionamentos e faixas de varrimento do LRF [18].

O trabalho em [19] consiste no desenvolvimento de um sistema externo ao robô e portátil, facilitando sua instalação em diferentes ambientes. Moura [20] utiliza um LSF e uma câmera num ponto fixo da simulação para determinar a posição e orientação de um alvo cilíndrico com marcadores. O autor em [21] investigou uma plataforma de varrimento a laser em três dimensões denominada de LMS-3D (do inglês *Laser Measurement Systems in 3D*) para obter uma navegação autônoma de robôs. Scanners a laser também podem ser utilizados para formar medições em 3D das condições de construção. É considerada uma forma de automatizar o processo de construção civil para levantar toda a informação e característica sobre o projeto. Além do mais, se caracteriza como uma das melhores formas para fazer a monitorização do seu edifício sem a utilização de câmeras de monitoramento, preservando a privacidade das pessoas [22].

Os dispositivos de digitalização em 3D integrados por sensor 2D tem sido amplamente utilizado visando o baixo custo do sistema. Na modelagem ambiental, Li et. al [23] utiliza uma plataforma de elevação de alta precisão e um pequeno sensor de alcance a laser. Chou [17] e Surmann [24] desenvolveram um sistema de mapeamento em 3 dimensões composto por uma plataforma móvel e uma giratória composta por um telêmetro a laser. Em [25] foi avaliada a viabilidade do sensor LIDAR de baixo custo na extração de traços de folhas.

Nesse trabalho é desenvolvido um scanner a laser 2D com alcance de varrimento de 180° e um motor de acionamento que resulta em várias combinações possíveis entre o

plano de mapeamento e o eixo de rotação para obter um mapa 3D.

2.3 Point Cloud

Inúmeros instrumentos fazem mapeamento de características topográficas a partir da medição de distâncias, alturas e ângulos das posições dos objetos. Essas medições são representadas por pontos que formam um conjunto de coordenadas x , y e z . Então a partir de um conjunto desses pontos, pode ser formada uma nuvem de pontos (*point cloud*) [26]. *Point cloud*, é uma representação tridimensional de um espaço que é possível ser visualizada por qualquer ângulo desejado [6]. A Figura 2.5 exemplifica uma *point cloud* obtida por um sensor LiDAR.

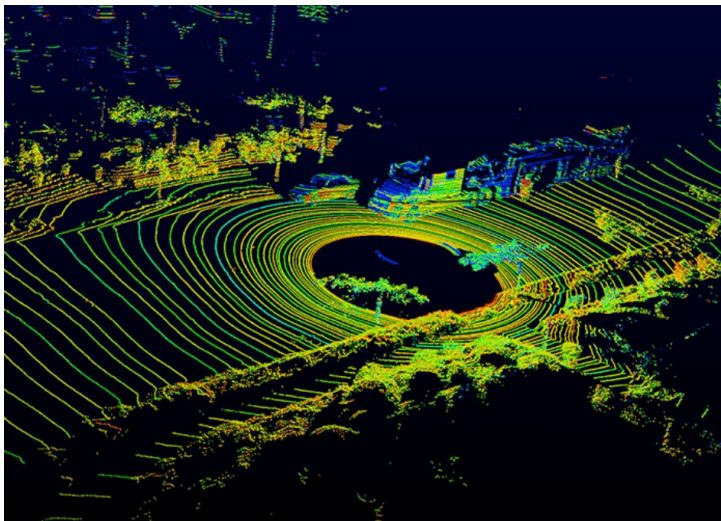


Figura 2.5: Exemplo de uma *point cloud* [27].

Existem três formatos disponíveis de dados de nuvem de pontos para quais os dados obtidos do sensor são exportados. Sendo: dados de pontos (XYZ), dados de ponto mais intensidade (XYZI) e dados de pontos mais valor de cor (XYZRGB). Moura [20] utiliza a combinação de dados de pontos mais intensidade e valor de cor (XYZIRGB) por ser o mais informativo, porém o uso computacional é maior.

Para entender a distribuição de pontos no espaço tridimensional e a análise que será realizada sobre essa geometria na metodologia deste trabalho, é necessário entender como

uma nuvem de pontos é adquirida.

Atualmente existem três maneiras principais de pré-processar a nuvem de pontos e são divididas em três subcategorias [28], [29], Figura 2.6:

- métodos baseados em projeção;
- métodos baseados em nuvens de pontos brutos;
- métodos convolucionais volumétricos (voxels).

Esses métodos são utilizados principalmente na detecção de objetos [30], porém podem ser utilizados para extrair recursos da *point cloud* como intensidade do ponto, altura e ocupação dentro da célula da grelha correspondente.

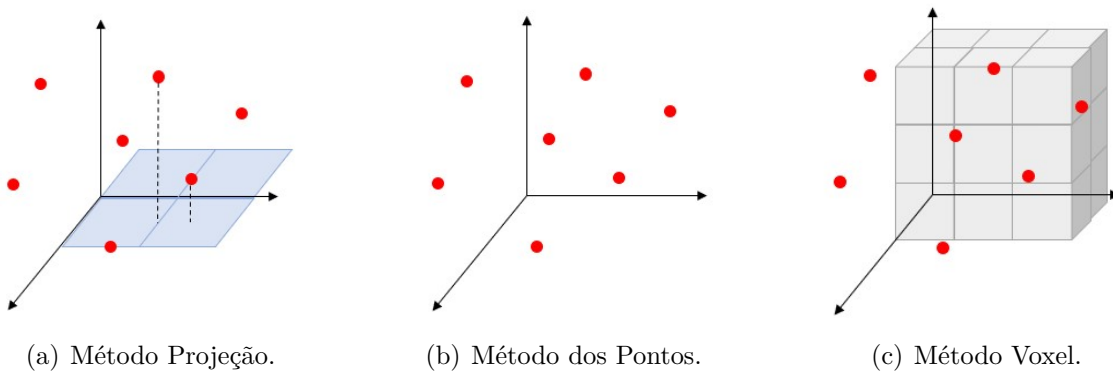


Figura 2.6: Representação nuvem de pontos.

Métodos Baseados em Projeção

Implementam uma projeção de visão única ou múltipla de uma nuvem de pontos 3D, resultando numa grade 2D. A extração de características é feita durante a projeção de pontos num plano horizontal, discretizados com uma grelha atribuída de dimensões pré-determinadas [29].

Métodos Baseados em Nuvem de Pontos Brutos

Esse método lida diretamente com nuvens de pontos para minimizar a perda de informações espaciais. A maioria dos métodos brutos baseados em nuvem de pontos são variantes do PointNet, amplamente utilizados para classificação de objetos e segmentação semântica [29], [31].

Métodos Convolucionais Volumétricos

Dada uma nuvem de pontos LiDAR e um espaço 3D particionado em voxel de tamanho fixo, voxelização é o processo de atribuição de pontos a cada grelha de voxel. Particionar o espaço 3D de acordo com um quadro de coordenadas cartesianas ou cilíndrica, resulta em voxel em formato de cubo [32].

Esse método conduz a voxelização da nuvem de pontos representando a nuvem de pontos 3D como grelhas de voxel regularmente espaçadas, conforme Figura 2.7.

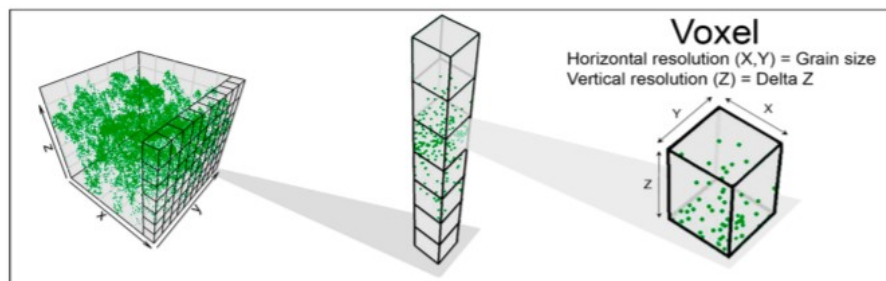


Figura 2.7: Resolução de uma grelha de voxel. [18]

Esse trabalho utilizará dados de pontos XYZ por ser utilizado somente um sensor LiDAR e processará a nuvem de pontos a partir de grades de voxel desconsiderando o particionamento ao longo do eixo Z que a metodologia mais adequada para esse sistema.

2.4 Laser Scanner para Estimativa de Volume

O trabalho de pesquisa de [1] utilizou um mobile scanner 3D GeoSLAM ZEB-Horizon dentro de um projeto para conseguir acelerar significativamente o trabalho de inventário,

esse scanner oferece uma alta cobertura da superfície devido a densa nuvem de pontos. O material analisado em questão eram minerais que se encontravam depositados em caixas de armazenamento. Esse equipamento oferece um software que pode ser acessado de um único local ou em toda a empresa, isso facilita para a informação em tempo real do inventário gerando uma eficiência operacional e financeira, apesar de que, o valor do scanner no mercado é elevado, tornando-se inviável para mapeamento de produtos em depósitos em que o utilizador não possui alto investimento.

Os sensores de nível baseados em mapeamento em três dimensões estão revolucionando o gerenciamento de produtos, adicionando a capacidade de estimar com precisão o volume sem sair da segurança de um escritório. Existem muitos dispositivos no mercado hoje para detectar o nível de sólidos em silos, porém sensores ToF é uma tecnologia avançada sem contato que supera um dos maiores desafios presente no silos, que é a presença de muita poeira. Esta tecnologia foi comprovada em diversos materiais desafiadores, incluindo alimentos e nas indústrias de energia e mineração [1]. Outro benefício da utilização de um laser é que quando o monitoramento de produto é totalmente automatizado, não há necessidade de escalar silos para fazer medições, evitando possíveis acidentes de trabalho [33]. Na Figura 2.8 é observado o posicionamento do sensor dentro do silo.

Medir a quantidade e manter o volume de produtos, cortes e aterros em empresas de terraplenagem é tarefa árdua [34]. Em [35], o trabalho consiste na utilização de um scanner 3D para avaliar as propriedades morfológicas de agregados graúdos e comparou os resultados com as técnicas convencionais.

No contexto de mineração, a estimativa de volume atualiza uma parte essencial das operações diárias da indústria, pois a maioria das empresas tenta tirar proveito da tecnologia mais recente para melhorar sua produção e maximizar o lucro. Com a ajuda da moderna tecnologia de nuvem de pontos, é possível estimar volume de forma eficaz de forma mais científica e precisa. Em sua pesquisa, Amaglo [26] procurou utilizar a nuvem de pontos e o modelo de superfície 3D de um transporte em massa para estimar o volume de produto no campo de mineração. Isso tornaria os levantamentos das minas seguros para os trabalhadores da construção, bem como a monitorização do progresso da



Figura 2.8: Laser 3D para estimativa de armazenamento em silos [33].

mineração para o gerente da construção. Um método computacional volumétrico baseado em coordenadas de nuvem de pontos foi formulado no software MATLAB para estimar o volume usando dados LiDAR capturados com um scanner a laser. A densidade de pontos e o algoritmo de filtragem desempenham um papel crítico no cálculo do volume, o que ajuda a fornecer uma boa estimativa do armazenamento [26].

O volume de uma superfície pode ser calculado rapidamente usando representações como grades de altura, voxels ou polígonos. No trabalho [36] uma nuvem de pontos criada por um telêmetro a laser é usada para criar a superfície 3D e, a partir dela, é possível formar a grade de altura que é usada para medir o volume da carga útil. Porém, atualmente os trabalhos relatados sobre método de voxel são principalmente destinados para estimar o volume de árvores. Em [37] e [38] estimaram o volume por modelagem 3D baseada em voxel usando dados LiDAR de varrimento portátil, com o objetivo de a partir de tamanho de voxels fixo, criar um modelo sólido para caracterizar o contorno de grandes seções de galhos e troncos que descreva o espaço ocupado pela árvore e assim, com base nesse voxels ocupados, voxels vazios não ocupados dentro de segmentos são identificados de forma para cálculo do volume.

Scanners a laser também têm sido usados com sucesso para medir o volume de sólidos. No artigo [13] nuvens de pontos são produzidas a partir de medições a laser e então é avaliado duas abordagens para construção de superfície 3D e estimativa de volume. O primeiro método, conhecido como *Meshed Surface*, divide os pontos recolhidos em células uniformemente espaçadas para gerar uma grelha de altura, embora conjuntos de dados esparsos necessitem de interpolação. A segunda abordagem, conhecida como Pontos Isolados, calcula o volume dividindo a área total digitalizada pelo número de pontos coletados. Devido à subamostragem em locais obscuros ou quando a refletividade é baixa, a segunda técnica produz estimativas errôneas.

Capítulo 3

Tecnologias e Métodos Explorados

Este capítulo apresenta as ferramentas e dispositivos utilizados neste trabalho, apontando algumas características e funcionalidades de todos os elementos do sistema. Começando pelo hardware, apresentar o sensor LiDAR, o microcontrolador e o motor de acionamento que formam o conjunto responsável pela aquisição de dados. Em seguida, o método e técnica utilizado para processamento dos dados. Por fim, é apresentado o software e a configuração implementada para construção da simulação.

3.1 Equipamentos Usados

Nessa seção é descrito o hardware do sistema, composto pelo sensor, motor de acionamento e microcontrolador, responsáveis por medir e adquirir os dados. Esses equipamentos foram escolhidos de acordo com os disponíveis em laboratório. O hardware do sistema é projetado de forma a fornecer controle sobre o campo de visão do varrimento. Todos os componentes do sistema estão alojados num suporte, tornando-o um instrumento pronto para uso da medição.

3.1.1 XV-11 Laser Scanner

O scanner a laser Neato XV-11, Figura 3.1, é um LRF de baixo custo, pequeno e tem baixo consumo de energia. O sensor estabelece uma comunicação série de 115200 bits por segundo (bps), enviando dados até cerca de 5 Hz e seu consumo de energia sem motor é relativamente baixo, aproximadamente 135 mA em 5 V, conforme apresentado na Tabela 3.1.

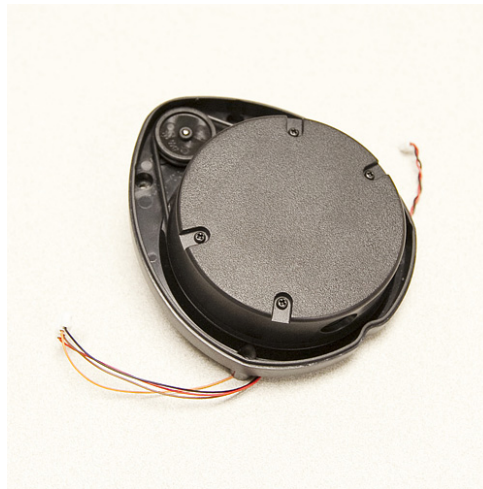


Figura 3.1: Scanner LIDAR modelo XV-11 [39].

Tabela 3.1: Informações Técnicas do sensor LIDAR modelo XV-11.

Parâmetros	Valor
Faixa de Medição	360°
Resolução Angular	1°
Alcance	0,2 - 6m
Corrente Máxima	135mA
Frequência de Escaneamento	5Hz

Uma rotação completa, proporciona uma faixa de medição de 360°, com resolução angular de 1° e alcance de 0,2 m até 6 m com erro inferior a 0,03 m. Quando o laser é retirado do robô Neato XV-11, seu motor deve ser controlado pelo utilizador, sendo necessário ser alimentado com tensão contínua de 3,0 V (60 mA) para produzir uma velocidade de rotação de 240 rpm. Para controlar e obter as medidas do sensor foi utilizado

uma fonte de alimentação, que fornece os 5 V solicitados pelo scanner a laser e pode estabelecer a comunicação serial necessária [40], [41].

Comunicação do Laser

Lima et. al. [42] realizaram o estudo sobre o protocolo de comunicação para o laser apresentado. O formato dos dados é composto por pacotes (o comprimento de um pacote é de 22 bytes) e uma volta completa enviará 90 pacotes, contendo 4 leituras consecutivas cada. Isso equivale a um total de 360 leituras em 1980 bytes [40].

3.1.2 ESP32

O microcontrolador ESP32 está disponível em várias versões entre as quais uma placa de sistema embebido, criado e produzido pela *Espressif System* e algumas de suas especificações técnicas mais relevantes são apresentados na Tabela 3.2. Possui um único combochip WiFi e Bluetooth de 2,4 GHz projetado para dispositivos móveis, eletrônicos vestíveis e aplicativos de IoT.

Tabela 3.2: Especificações do Microcontrolador ESP32.

Parâmetros	Valor
Processador	Xtensa® dual-core 32-bit LX6
Memória Flash	4 MB
Memória RAM	520 kB
Wi-Fi	2,4 GHz
Bluetooth	BLE 4.2

O módulo ESP32-WROOM-32D, utilizado neste projeto, integra o microprocessador ESP32, tornando mais fácil sua utilização em placas de circuitos impressas. A imagem do sensor e a descrição dos pinos é apresentada na Figura 3.2.

O módulo suporta uma taxa de dados de até 150 Mbps e 20 dBm de potência de saída na antena para garantir o maior alcance físico. Como tal, o módulo oferece especificações líderes do setor e o melhor desempenho para integração eletrônica, alcance, consumo de energia e conectividade [44].

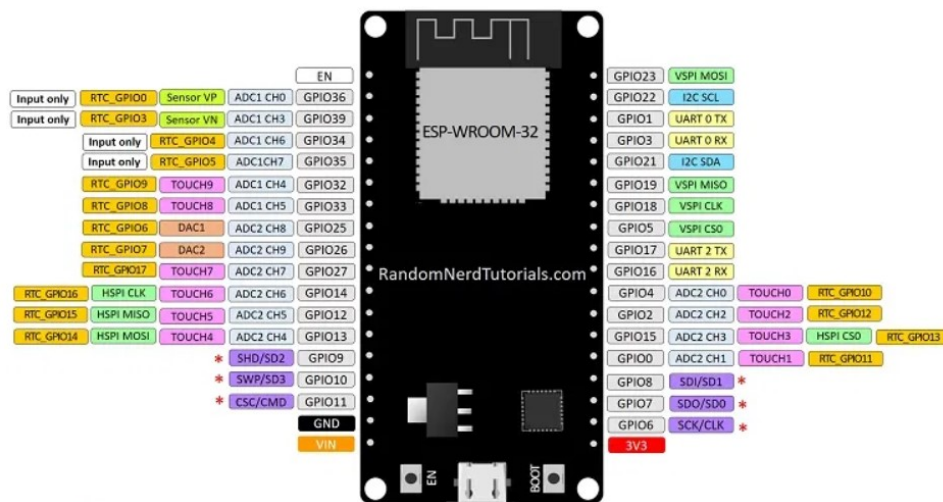


Figura 3.2: Descrição dos pinos do ESP32 [43].

O código fonte dos sistemas utilizados estão integrados no microcontrolador, e as conexões realizadas entre o ESP32 e sensores será descrita ao longo do capítulo 4.

3.1.3 Servo Motor

Dois motores de acionamento, um servo motor e um motor de passo, foram utilizados nesse trabalho para avaliar a melhor precisão para o ângulo de inclinação.

O modelo DGServo S06NF STD, Figura 3.3, é o servo motor utilizado para testar a rotação do suporte com o sensor LiDAR. Possui 3 fios para ligação e o Vcc é alimentado em 5 V. Sua rotação é limitada de no máximo 180°.

3.1.4 Motor de Passo

O motor de passo apresentado na Figura 3.4 de modelo Nema 17 tem um ângulo de passo de 1,8° (200 passos/revolução). Tensão de alimentação de 12-36 Vcc e corrente de 2 A por fase, permitindo um torque de retenção de 280 g-cm e um peso de 0,34 kg.



Figura 3.3: Servo Motor modelo S06NF STD.



Figura 3.4: Motor de passo modelo Nema 17 42BYGHW811 [45].

Driver

Motores de passo requerem alta corrente que os microcontroladores não podem fornecer. Assim, os acionadores do motor de passo são projetados para alternar alta corrente ou alterar sua direção acionada pelos sinais elétricos de entrada de baixa corrente. O driver utilizado para acionamento é de modelo DRI0023, Figura 3.5, onde através da configuração de micropassos consegue fornecer a frequência necessária para o controle desejado do motor nesse trabalho.

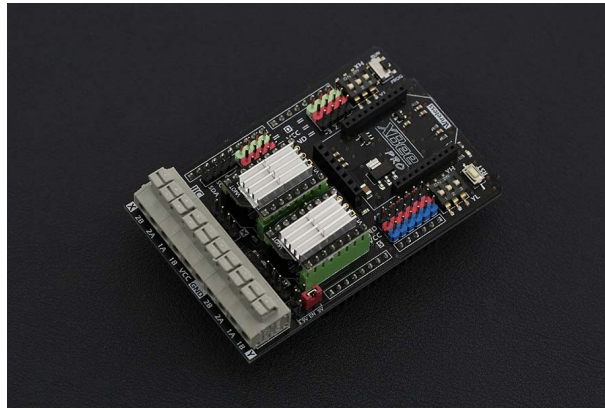


Figura 3.5: Driver DRI0023 para o motor de passo [46].

3.2 Métodos e Técnicas Usadas

Esta secção apresenta dois métodos explorados e as técnicas desenvolvidas para o processamento de dados no sistema criado nesse trabalho.

3.2.1 Mudança de Coordenadas

A conversão de coordenadas esféricas (r, θ, ϕ) para cartesianas (x, y, z) é realizado com as fórmulas do sistema de Equações 3.1 [47].

$$\begin{cases} x = r \sin(\phi) \cos(\theta) \\ y = r \sin(\phi) \sin(\theta) \\ z = r \cos(\phi) \end{cases} \quad (3.1)$$

De acordo com a Figura 3.6, r designa a distância de cada ponto de coordenadas à origem, θ é o ângulo formado entre o semi-eixo positivo de x e ϕ caracteriza o ângulo entre o semi-eixo positivo z e o vetor formado pela distância de r .

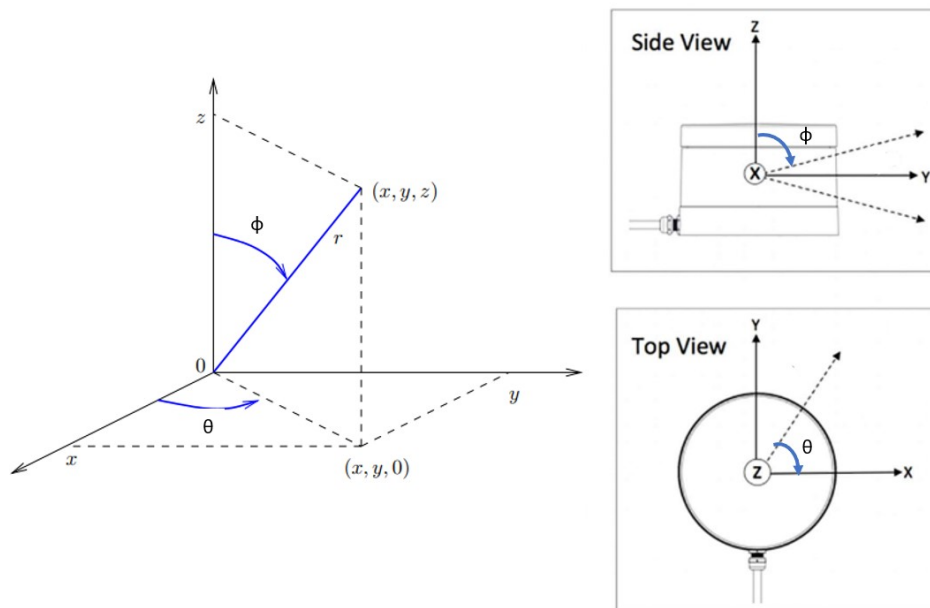


Figura 3.6: Coordenada Esférica para Coordenada Cartesiana. Adaptado de [48].

Neste trabalho, é tratada a nuvem de pontos após a projeção em coordenadas xyz .

3.2.2 Máquina de Estados

Máquinas de estados finitos, conhecidos também como máquinas sequenciais são modelos abstratos de um circuito sequencial real. Seu comportamento pode ser representados como uma sequência de eventos que ocorrem em instantes de tempo discreto [49].

Um número finito de possíveis comportamentos compõem um sistema. Conseqüentemente, a máquina de estados usada para modelar o sistema tem um número finito de

estados ligados entre si por transições que representam esses comportamentos. Assim, um estado é definido como uma situação de um sistema dependente de eventos anteriores que produz determinadas ações, enquanto uma transição é a resposta de um estado a um evento [50], como exemplo pode ser observado a Figura 3.7.

A utilização de máquinas de estados é um dos padrões mais populares para aplicações em sistemas embebidos, por ser um dos métodos mais eficazes para desenvolvimento de código orientado a eventos [50]. Esta técnica será utilizada nesse trabalho para implementação em C++ do hardware do sistema.

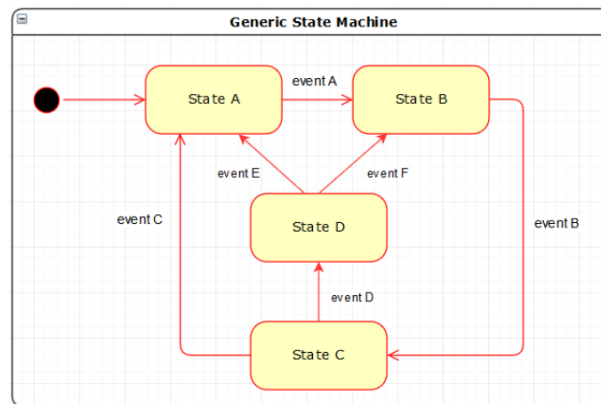


Figura 3.7: Exemplo de uma máquina de estados genérica [50].

3.3 Ambiente de Simulação

Nesta seção é apresentado o ambiente de simulação "SimTwo", descrevendo uma introdução sobre suas características, as funções de cada uma das diferentes janelas encontradas no simulador e os elementos utilizados para criar o cenário.

3.3.1 SimTwo

Desenvolvido em *Object Pascal*, SimTwo é um software gratuito de código aberto onde vários tipos de robôs podem ser modelados: diferencial, omnidirecional, industrial, humanoíde, entre outros. Foi desenvolvido em 2008 e constantemente recebe atualizações disponíveis em [7].

Este simulador possui a funcionalidade de utilizar modelos especificados de componentes caracterizados e da decomposição de um robô em corpos rígidos e em motores elétricos, isso permite um realismo dinâmico ao projeto [51]. O SimTwo permite criar e simular vários robôs podendo ser controlador quer por uma script iterna quer por diversas linguagens de programação, tais como como Matlab e LabVIEW.

Todas as ferramentas disponíveis permitem o desenvolvimento de uma ampla diversificação de trabalhos. Alguns dos trabalhos desenvolvidos incluem a simulação de um modelo de quadrotor e a avaliação da dinâmica do robô [52], ferramenta de simulação de jogos [53], a validação de um sistema de controle de um veículo autônomo [54] e robôs humanóides [55].

3.3.2 Estrutura da Simulação

O SimTwo é uma aplicação com interface de múltiplos documentos (MDI) onde as janelas “code editor”, “spreadsheet”, “configuration”, “scene editor” e “chart” estão sob o controle da janela “worldview”, como mostrado na Figura 3.8 a seguir.

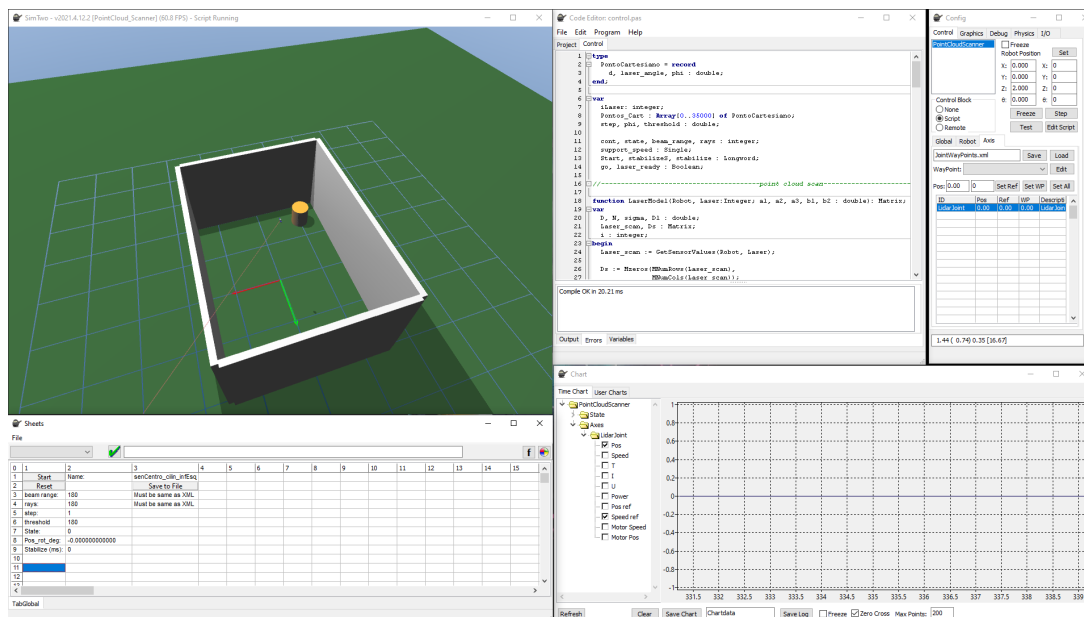


Figura 3.8: Software SimTwo: Janelas do aplicativo no sentido horário a partir do canto superior esquerdo: *world view*, *code editor*, *configuration*, *chart*, and *spreadsheet*.

A janela “*configuration*” oferece controle sobre vários parâmetros, entre elas a posição da câmera e da iluminação. Através da aba “*I/O*”, é possível a comunicação externa pela Porta Serial.

O “*code editor*” oferece um ambiente de desenvolvimento integrado (IDE) para programação em linguagem Pascal.

Na janela “*chart*” é possível plotar todas as variáveis disponíveis para cada robô, como sua posição e velocidade. “*Spreadsheet*” é uma tabela que permite mostrar em tempo real o valor de qualquer variável desejada.

Em “*Scene Editor*” é possível criar robôs, trilhas, paredes e quaisquer outras entidades no ambiente. Consiste no arquivo principal “*scene.xml*” e outros arquivos XML.

Capítulo 4

Sistema Evolid

Este capítulo inicia apresentando o conceito do sistema Evolid desenvolvido ao longo deste trabalho. Na sequência, será apresentado um diagrama de blocos com todas as etapas apresentadas no processo de implementação da abordagem desenvolvida neste trabalho. Começando pela arquitetura do sistema mostrando um passo a passo da aquisição, processamento dos dados e do cálculo do volume. Por fim, a modelação dos suportes necessárias para implementação do sistema em ambiente no real.

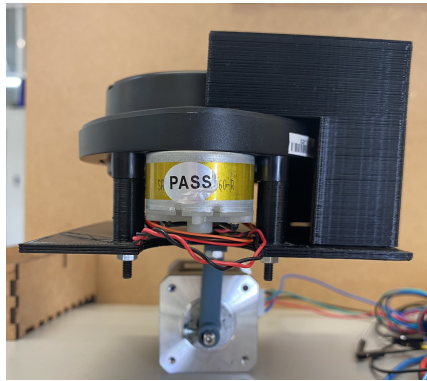
4.1 Arquitetura Evolid

Evolid é um sistema formado pela modelação de uma plataforma giratória de LiDAR de baixo custo real responsável pelo mapeamento e aquisição de dados e por um sistema de tratamento de dados que seja capaz de calcular o volume ocupado por objetos num depósito.

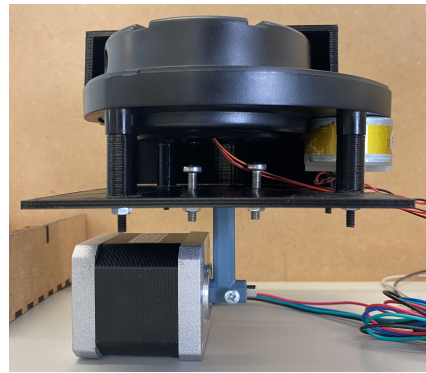
A plataforma projetada ilustrada na Figura 4.1 é composta por 3 partes principais e descrita durante esse capítulo:

- Sensor rotativo: parte do LiDAR é responsável pela orientação no eixo correspondente ao ângulo azimutal. Neste projeto, o sensor possui liberdade para girar 360° , contudo a aquisição será limitada até 180° .

- Motor: Responsável pela orientação do LiDAR no eixo de inclinação. Tem liberdade de rotação de 180°.
- Suporte: parte na qual o sensor e o motor são fixados, é onde se projetou também para localizar a maior parte da eletrônica do projeto incluindo o microcontrolador.



(a)



(b)

Figura 4.1: Plataforma Evolid.

O trabalho pode ser definido em três etapas principais:

1) Aquisição de dados: a simulação é realizada no software SimTwo e são adquiridos os dados por meio do mapeamento a laser de todo o local.

2) Processamento dos dados: os dados são convertidos para coordenadas cartesianas gerando então uma única *point cloud*, e posteriormente são filtrados para remover as paredes.

3) Estimação do Volume: a nuvem passa pelo processo de voxelização para que fosse possível o cálculo do volume ocupado pelo objeto.

A arquitetura do hardware é constituída pela etapa 1 e a do software pelas etapas 2 e 3 visualizada no diagrama da Figura 4.2.

4.2 Aquisição de Dados

O conjunto proposto consistirá de um subsistema de varrimento a laser bidimensional acionado por um motor, que irá adquirir a distância dos objetos do cenário juntamente

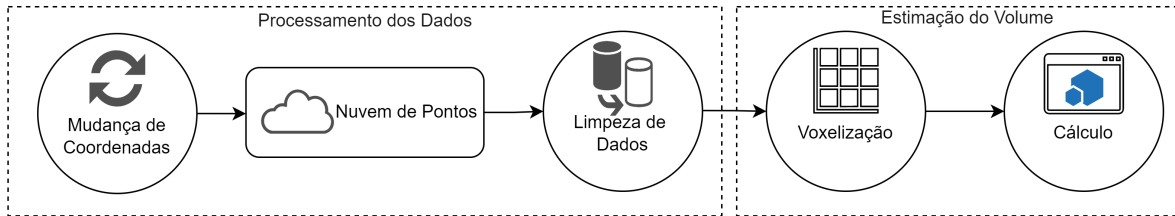


Figura 4.2: Arquitetura do Software.

com a informação da orientação dos ângulos do sistema. A Figura 4.3 mostra as fases da implementação do processo.

Quando o processo de varrimento começa, a plataforma começa a girar numa velocidade constante conhecida. Ao mesmo tempo, o dispositivo LiDAR varre o ambiente e a cada revolução da plataforma de 360° o motor incrementa o grau desejado. Desta forma, a cada rotação o aparelho irá identificar novas informações até atingir 180° . Nesse sentido, como a velocidade angular e a faixa de medição do LiDAR não podem ser alteradas, o tamanho do incremento na estrutura é uma forma de regular a resolução do escaneamento. Ao final é obtido um arquivo texto (.txt) contendo informações de distância, ângulo azimutal e ângulo de inclinação.

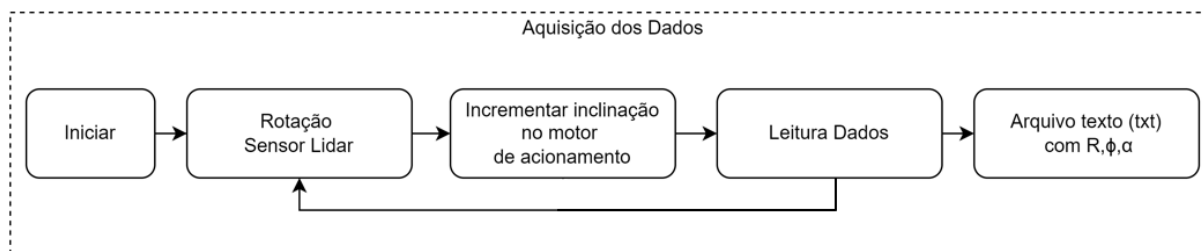


Figura 4.3: Arquitetura do Hardware.

A aquisição de dados pode ser dividida entre a etapa simulada e a etapa real constituída pela configuração do sensor LiDAR e pelo motor de acionamento com o microcontrolador.

4.2.1 Construção do Ambiente de Simulação

Esta subseção apresenta o processo de criação do ambiente. A descrição física de cada entidade é feita por meio de arquivos XML. Antes de iniciar a construção, é importante

identificar quais entidades estarão presentes na simulação e como eles irão interagir, para que seja possível definir como será o ambiente virtual.

O conjunto proposto consistirá de um subsistema de mapeamento a laser bidimensional acionado por um servomotor, que adquire em tempo real a distância dos objetos do ambiente juntamente com a informação da orientação dos ângulos do sistema. Pretende-se que o subsistema a laser seja capaz de detectar as paredes e os objetos localizados no centro, criando então um arquivo texto para ser possível estimar a nuvem de pontos do ambiente. Para que o sensor laser (círculo vermelho) tenha uma boa cobertura de toda a cena, foi proposto como exemplo o layout mostrado na Figura 4.4 onde sua localização é no centro e no teto do ambiente, o retângulo laranja representa as paredes.

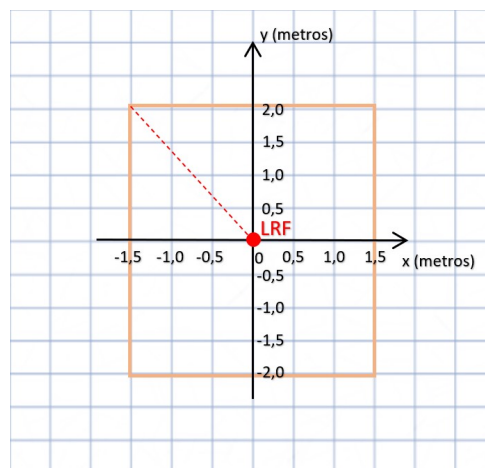


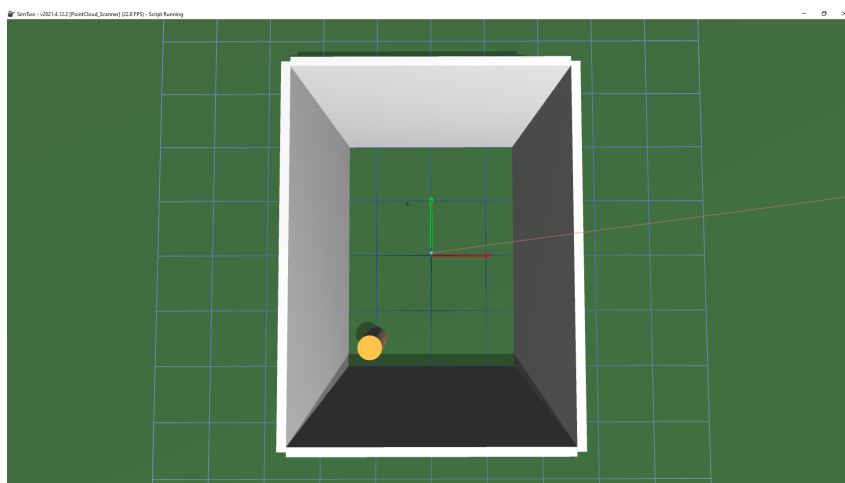
Figura 4.4: Posição do sensor laser na simulação.

Configuração Paredes

Iniciando a construção, foram utilizadas paredes configuradas em XML de espessura 0,1m para criar um espaço retangular e a título de exemplo, na Listagem 4.1 foi configurado um ambiente interno de $3 \times 4 \text{m}^2$ e com altura de 2m, onde o sistema e o objeto estarão contidos.

Listagem 4.1: Parâmetros Paredes.

```
1 <defines>
2   <const value="3" name="wall_length_x"/>
3   <const value="4" name="wall_length_y"/>
4   <const value="0.1" name="wall_width"/>
5   <const value="2" name="wall_height"/>
6   <const value="1" name="wall_Z_pos"/>
7 </defines>
```

Figura 4.5: Visualização superior das paredes no software *SimTwo*.

Configuração Scanner a Laser

Para o sensor LiDAR é necessária uma configuração na seção de feixe em sensores do código XML, onde é descrito um ranger 2D. Nesta seção é possível descrever configurações como o comprimento do feixe (6 metros neste caso particular) com uma largura inicial e final de 5 milímetros, valores de ruído, deslocamento, ganhos, período e a cor do feixe. Todos esses parâmetros são definidos no setor de sensores do arquivo XML, conforme descrito na Listagem 4.2. O modelo do scanner a laser Neato XV-11 e suas equações de desvio padrão são descritos no artigo [41] e usados como base nesse trabalho, tornando possível a simulação em SimTwo.

Listagem 4.2: Parâmetros LiDAR.

```
1 <ranger2d>
2   <ID value="ranger2d"/>
3   <beam final_width="0.005" initial_width="0.005" length="6"/>
4   <period value="0.04"/>
5   <pos z="0" y="0" x="0"/>
6   <rot_deg z="0" y="0" x="0"/>
7   <tag value="00"/>
8   <beam rays="180" angle="180"/>
9   <noise gain="1" offset="0" stdev_p="0" stdev="0.00"/>
10  <color_rgb b="0" g="0" r="255"/>
11 </ranger2d>
```

O sensor foi configurado para ter um tempo de varrimento de 25ms e um campo de visão de 180°. Em conformidade com o Neato XV11, na simulação é adquirida uma 1 amostra por grau, totalizando 180 amostras durante um varrimento completo. A Figura 4.6 mostra o subsistema do scanner a laser, caracterizado por uma base fixa e o LRF.

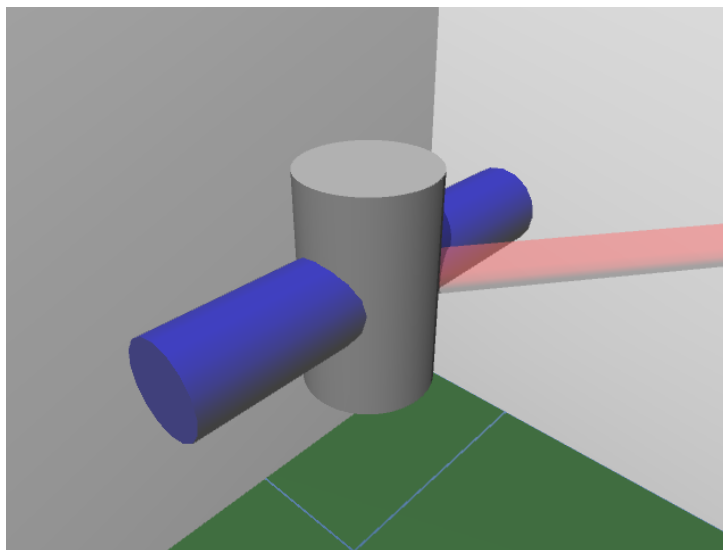


Figura 4.6: Scanner a laser.

Configuração Objetos

Na Listagem 4.3, tendo como exemplo, estão descritos 3 objetos em código XML para a simulação: cubo, cilindro, esfera. O cubo nesse caso possui dimensão $0,25 \times 0,5 \times 0,5 \text{m}^3$, a esfera possui um raio de $0,2 \text{m}$ e o cilindro possui um raio de $0,1 \text{m}$ e altura de $0,5 \text{m}$.

Listagem 4.3: Parâmetros Objetos.

```
1 <cuboid>
2   <ID value='box1' />
3   <size x='0.25' y='.5' z='0.5' />
4   <pos x='-1' y='-1.5' z='0.25' />
5   <rot_deg x='0' y='0' z='90' />
6   <color_rgb r='250' g='140' b='10' />
7 </cuboid>
8
9 <sphere>
10  <ID value='Ball1' />
11  <radius value='0.2' />
12  <mass value='0.3' />
13  <pos x='-1' y='-1.5' z='0.2' />
14  <surface mu='0.1' softness='0.1' />
15  <drag coefficient='0' stokes='1e-4' roll='5e-4' />
16 </sphere>
17
18 <cylinder>
19  <ID value='cylinder1' />
20  <size x='.2' y='.2' z='.5' />
21  <pos x='-1' y='-1.5' z='0.25' />
22  <rot_deg x='0' y='0' z='0' />
23  <color_rgb r='250' g='140' b='10' />
24 </cylinder>
```

A posição dos objetos é configurada a partir do seu centro de massa, então para o objeto não ficar submerso no eixo z , o mesmo precisa ser configurado para metade da sua altura estar com um offset do chão da simulação. Para objetos rígidos simples com

densidade uniforme, o centro de massa está localizado no centroide.

4.2.2 Máquina de Estados

A máquina de estados executada no microcontrolador possui 6 estados e foi modelada com o intuito de facilitar a implementação do código na plataforma Arduino IDE. O diagrama, Figura 4.7, ilustra o comportamento do sistema com os estados e as transições associadas a cada um.

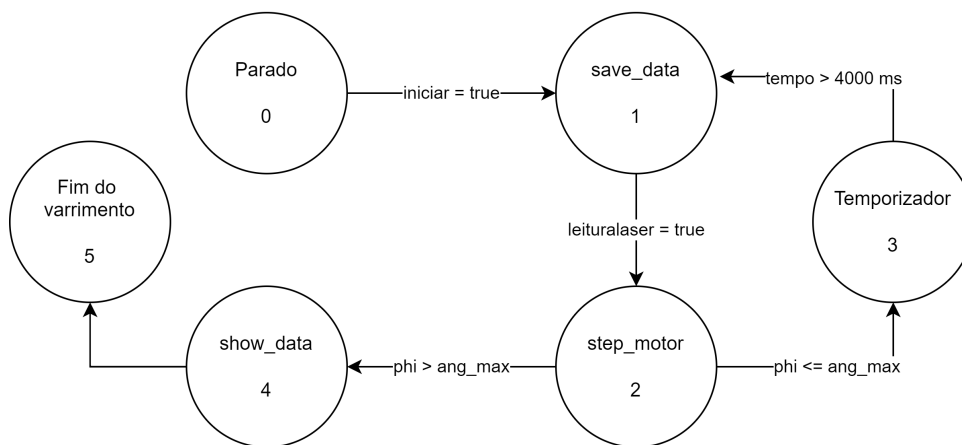


Figura 4.7: Máquina de Estados do Sistema Evolid.

No estado 0 a plataforma está parada e processando os dados adquiridos do sensor. O comando *iniciar* é dado ao pressionar uma tecla e salva em três variáveis a distância, o ângulo azimutal e o ângulo de inclinação daquela medição e após esses dados serem salvos, o comando *leituralaser* passa a ser verdadeiro dando seguimento para o estado *step_motor*, o qual é realizado o incremento de 1° no ângulo de inclinação (φ). Se φ for menor que o ângulo máximo de 180° , um temporizador é iniciado para esperar a estabilidade do suporte, voltando ao estado *save_data*. Caso contrário, com φ maior que ângulo máximo, os dados são expostos no monitor serial do software, passando para o último estado, finalizando o varrimento.

4.2.3 Hardware

Cada sensor tem suas particularidades quanto ao PINOUT, interface do protocolo de comunicação e tensão de alimentação. A linguagem de programação C++ foi usada para implementação no software Arduino IDE e a ligação dos ESP32 com os sensores está presente nessa sessão.

Neato XV 11

De acordo com [40], os fios do LiDAR possuem a seguinte pinagem: Vermelho: 5 V, Laranja: Tx, Marrom: Rx e Preto: Terra. O fio marrom não é utilizado nesse caso por não haver necessidade de recepção de dados e a porta IO17 do microcontrolador é conectada ao Tx do sensor. O fio vermelho do motor é ligado ao Vcc de 3V e o preto em comum com o terra do sensor.

A Figura 4.8 apresenta como Neato XV 11 é conectado ao ESP32.

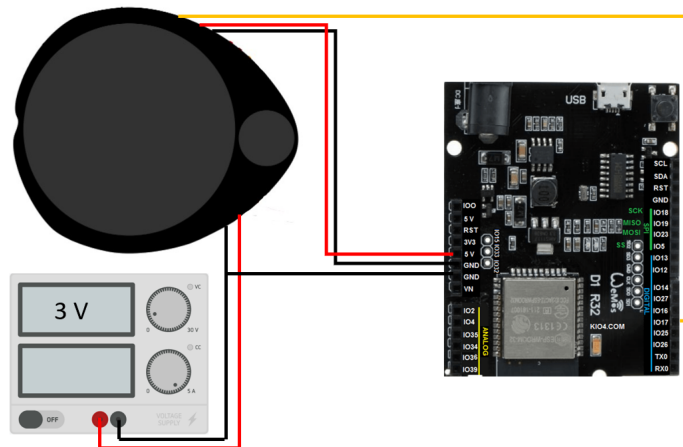


Figura 4.8: Ligações do LiDAR com a placa ESP32.

ServoMotor

O motor é acionado pelo pino de alimentação Vcc de 3,3 V presente no ESP32 e possui a ligação descrita na Figura 4.9.

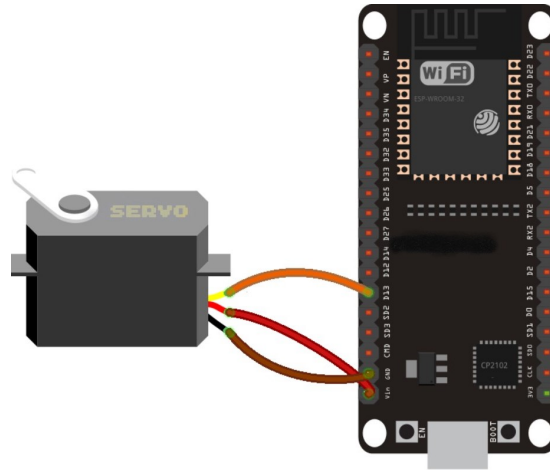


Figura 4.9: Ligações do Servo Motor com a placa ESP32.

Motor de Passo

Figura 4.10 mostra a conexão do motor de passo com o ESP32 e o Driver, onde os 4 fios representando as duas bobinas que saem do motor são ligados ao driver que possui Vcc de 3,3 V e alimenta o motor com 12 V através da fonte de alimentação. O pino Terra é comum em todo o circuito.

Para esse circuito, três portas foram configuradas, sendo *IO27* utilizada para realizar o valor de passo do motor, a porta *IO14* controla a direção e *IO12* possui a função de *enable* (habilitar e desabilitar),

4.3 Processamento dos Dados

O arquivo txt é importado para o ambiente MATLAB. O arquivo contém os parâmetros em coordenadas esféricas e, portanto, precisa ser convertido para o sistema de coordenadas cartesianas.

Depois da conversão é gerado a *point cloud* do ambiente. Passando então pelo processo de filtragem das paredes obtendo a nuvem de pontos somente dos objetos.

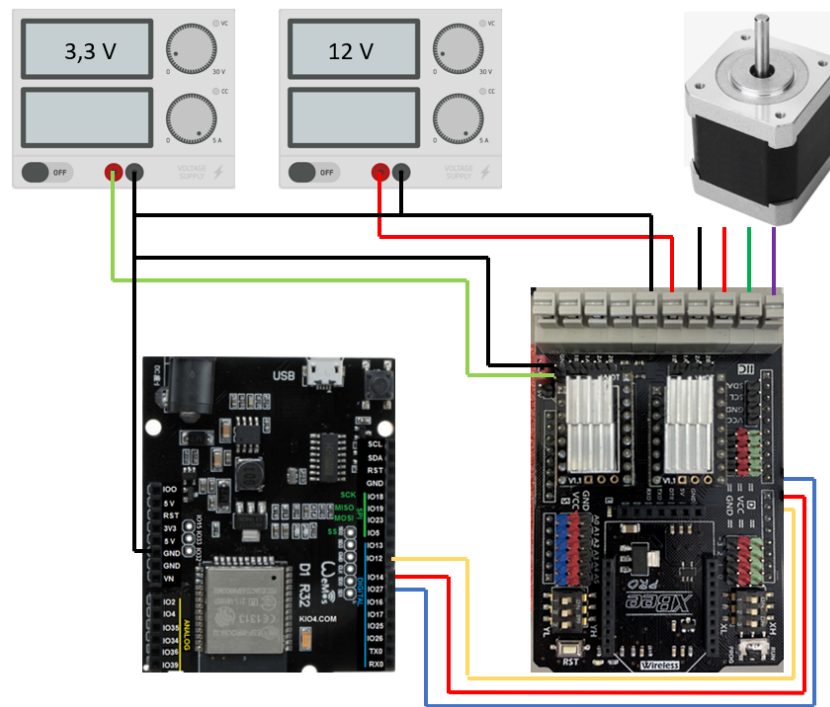


Figura 4.10: Conexões entre o Motor de Passo, ESP32 e Driver.

4.3.1 Mudança de Coordenada

A Figura 4.11 apresenta a geometria associada a um ponto P . É observado dois pontos para a característica da estrutura: A posição do sensor (x_L, y_L, z_L) e o ponto adquirido pelo mapeamento (x_1, y_1, z_1) . Para efeitos de cálculo, a origem do sistema é considerada na posição do sensor S_L (círculo vermelho).

As informações recebidas em cada etapa são ressaltadas na imagem com a cor azul, dessa forma:

- φ é a inclinação do plano do sensor paralelo ao plano xy com um intervalo de $[0, \pi]$.
- α é o ângulo azimutal medido a partir do eixo y para R e possui um intervalo $[0, \pi]$.
- R é o parâmetro que representa a distância do sensor até o objeto encontrado.

Em seguida, essas três variáveis são utilizadas para o cálculo da transformação de coordenadas tridimensionais em coordenadas cartesianas (xyz) e assim tornando possível a construção de um mapa de nuvem de pontos.

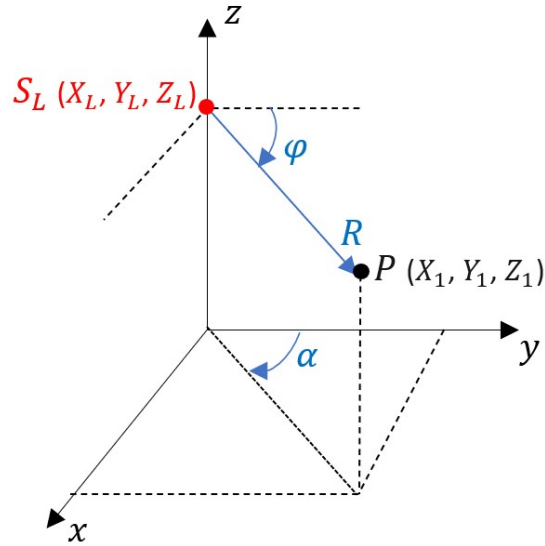


Figura 4.11: Sistema de coordenadas do sensor liDAR.

A posição P (círculo preto) num dado tempo em relação ao sensor em coordenadas cartesianas (X_1, Y_1, Z_1) pode ser definida em termos dos parâmetros (R, φ, α) :

$$\begin{cases} x = R \cos(\varphi) \sin(\alpha) \\ y = R \cos(\alpha) \\ z = -R \sin(\varphi) \sin(\alpha) \end{cases} \quad (4.1)$$

Essa relação pode ser comprovada através do teorema de mudança de variáveis em integrais triplas.

Quando o cálculo do valor da coordenada é concluído, no caso da simulação é feita a construção e exibição da imagem da nuvem de pontos. Uma vez convertido para o sistema de coordenadas cartesianas, X , Y e Z representam as coordenadas correspondentes em metros do sistema do mundo real.

4.3.2 Filtragem de Dados

Nessa etapa, o plano de fundo necessita ser removido da nuvem de pontos para não ser contabilizado na estimativa do volume. O método utilizado consiste na definição de um

limite aceitável da variação dos pontos que representam as paredes.

Como a origem do sistema é o sensor e ele está localizado no centro do ambiente, Figura 4.4, os extremos da parede possuem um valor máximo de x, y , e z positivo enquanto na outra ponta o valor mínimo dessas variáveis serão negativas. Dessa forma, serão considerados pontos menores que x_{max} , y_{max} e z_{max} e pontos maiores que x_{min} , y_{min} e z_{min} . A Equação 4.2 mostra a relação utilizada em software, onde ρ representa o limite escolhido e sign o sinal que acompanha o valor da variável.

$$\left\{ \begin{array}{l} x_{max} = \max(x) - \text{sign}(\max(x)) \cdot \rho \\ x_{min} = \min(x) - \text{sign}(\min(x)) \cdot \rho \\ y_{max} = \max(y) - \text{sign}(\max(y)) \cdot \rho \\ y_{min} = \min(y) - \text{sign}(\min(y)) \cdot \rho \\ z_{max} = \max(z) - \text{sign}(\max(z)) \cdot \rho \\ z_{min} = \min(z) - \text{sign}(\min(z)) \cdot \rho \end{array} \right. \quad (4.2)$$

4.4 Estimação do Volume

Para estimativa do volume a partir da nuvem de pontos foi utilizada a abordagem da voxelização.

4.4.1 Voxelização

Essa abordagem caracteriza-se, em primeiro lugar por particionar quadrados de tamanho iguais no plano xy com o objetivo de unificar todos os pontos naquela região. Em seguida, realizar a média da altura desses pontos que representa o valor da altura para cada grade, dessa forma obtendo como resultado o formato de um cubo. Cada cubo terá uma altura média pelo qual será possível o cálculo de volume daquela região.

As Figuras 4.12 e 4.13 apresentam um exemplo de uma nuvem de pontos desenvolvida em duas e em três dimensões, respectivamente. Na Figura 4.13 é possível analisar que cada cubo possui seu particionamento no eixo Z dependendo da média da altura e do

número de pontos ocupados naquele espaço.

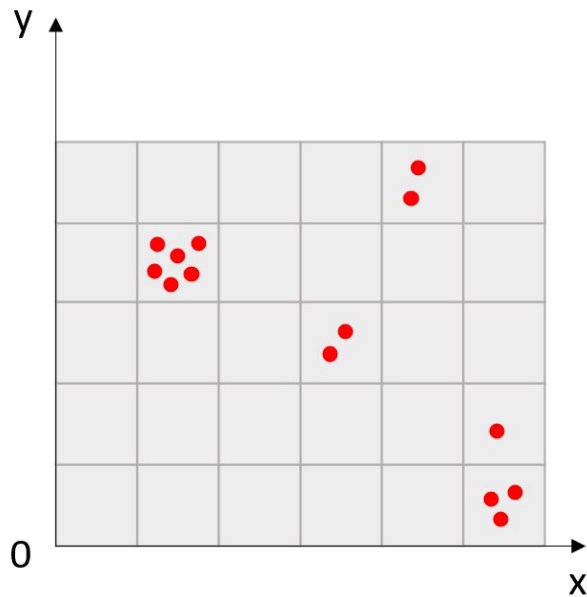


Figura 4.12: Grades de voxel no plano xy .

Resolução da Grelha

A resolução da grelha se faz necessária por conta da variação da densidade de pontos do sistema. É importante que tamanho do voxel seja determinado adequadamente, ele deve ser grande o suficiente para não haver grelhas não preenchidas mas também pequena o suficiente para representar adequadamente os dados. Assim, foi analisado uma relação entre densidade de pontos que depende diretamente da altura do sensor com a variação do degrau do motor. Essa relação é encontrada pela distância entre dois feixes distanciados pelo grau de inclinação escolhido.

Para melhor entendimento, a Figura 4.14 apresenta a proposta escolhida com um cubo no meio do cenário.

A letra a caracteriza a altura do sensor até o chão, b a diferença da altura do sensor e do cubo e c corresponde o valor mínimo que a grelha precisa ter. O valor de c pode ser calculado pela relação encontra na Equação 4.3

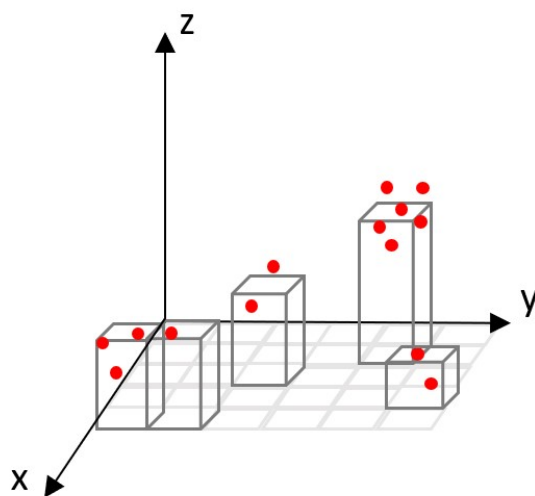


Figura 4.13: Grelhas de voxel em três dimensões.

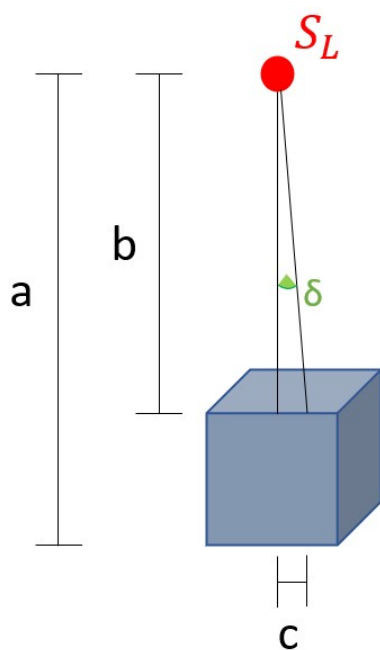


Figura 4.14: Diagrama para resolução da grade

$$c = \tan(\delta)b \quad (4.3)$$

onde o ângulo de inclinação δ representa o passo escolhido para o motor de acionamento.

4.4.2 Cálculo do Volume

Ao final, o volume total deduzido na Equação 4.4 será a soma de todas as grelhas de voxel no plano xyz .

$$V_{TOTAL} = \sum_{j=1}^M \sum_{i=1}^N x'_{i,j} \cdot y'_{i,j} \cdot z_{i,j} \quad (4.4)$$

onde M e N são a posição de cada grade de voxel:

- x' representa a largura da grelha.
- y' representa o comprimento da grelha.
- z representa a altura e é calculado pela soma da altura de todos os pontos dividido pela quantidade.

O valores de x e y são fixos para a mesma nuvem de pontos, porém o valor de z é alterado e depende dos pontos. Assim, a equação pode ser simplificada como:

$$V_{TOTAL} = x' \cdot y' \cdot \sum_{j=1}^M \sum_{i=1}^N z_{i,j} \quad (4.5)$$

4.5 Modelagem dos Suportes

As estruturas foram desenvolvidos no software *SolidWorks* e foram fabricadas utilizando uma impressora 3D.

São montadas em plástico para engenharia conhecido por PLA (polylactide), próprio para impressão em 3D. Este material foi escolhido para conferir a resistência mecânica necessária e levando em consideração o baixo custo e a acessibilidade.

4.5.1 Suporte para o LiDAR

Toda a modelagem da estrutura foi pensada de forma a se otimizar em termos de espaço e peso. O intuito é fixar num suporte: o sensor, o microcontrolador, o servo-motor e os demais circuitos integrados. Para isso, foi feita a análise da melhor forma para se conectar o LiDAR ao motor a fim de possibilitar a rotação de todo o módulo em 180° . Também foi limitada a visão do sensor em 180° impedindo o sensor de adquirir os feixes até 360° , conforme analisado na Figura 4.15.

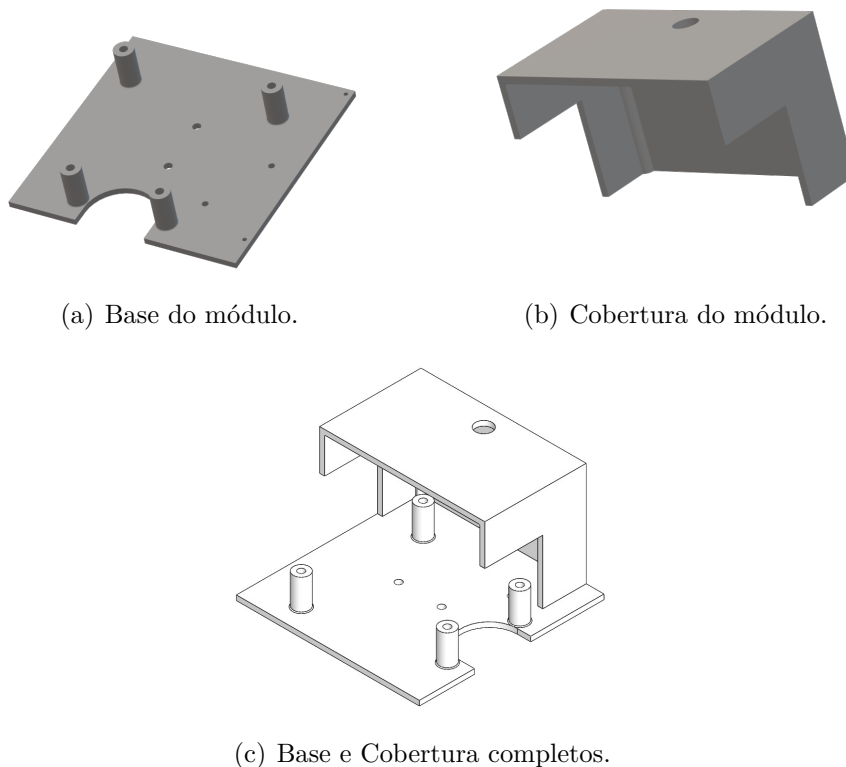


Figura 4.15: Suporte desenvolvido no software *SolidWorks*

A Figura 4.16 apresenta o resultado final e a Figura 4.17 ilustra as partes da montagem do módulo, dessa forma possibilitando a etapa de aquisição de dados do sistema.

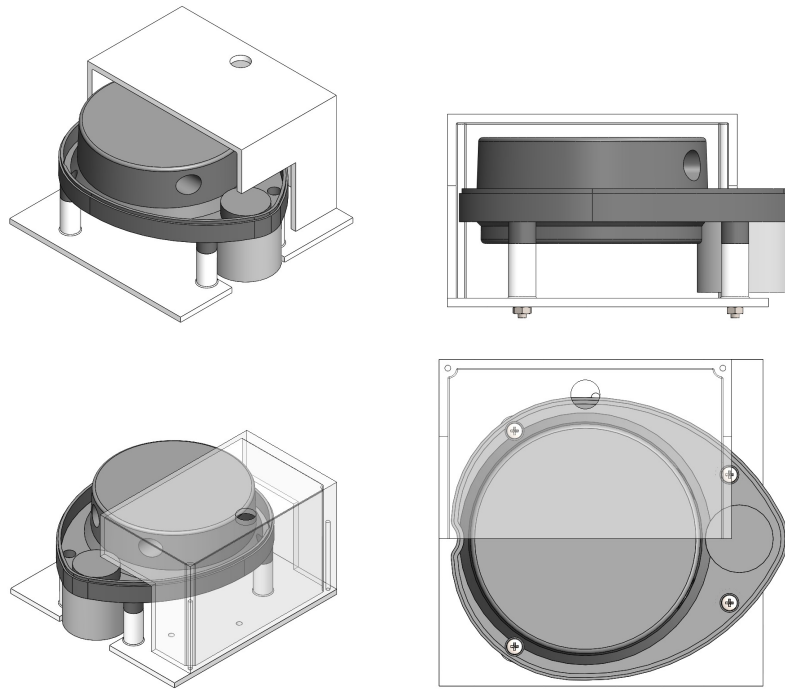


Figura 4.16: Diferentes perspectivas do módulo com o sensor LiDAR.

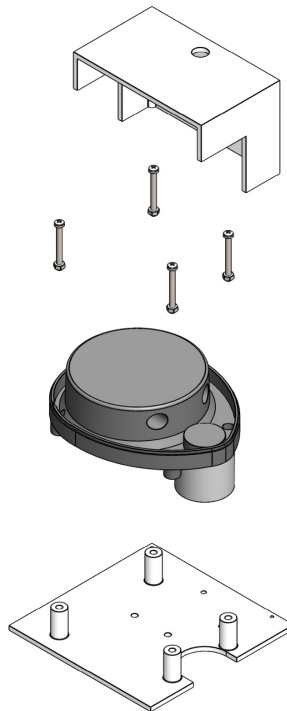


Figura 4.17: Partes da montagem do módulo.

4.5.2 Suporte para o Motor de Passo

O suporte para o motor de passo, Figura 4.18, foi projetado com a finalidade de fixar o suporte para o LiDAR no eixo do motor, de forma a possibilitar a rotação do sistema em 180°.

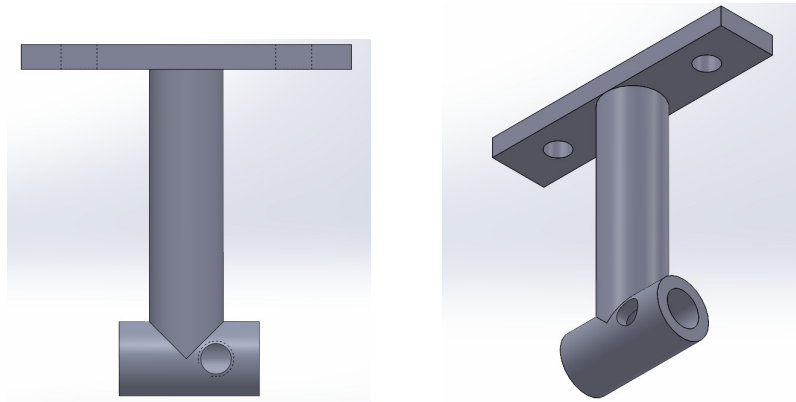


Figura 4.18: Suporte para o Motor de Passo desenvolvido no software *SolidWorks*.

Capítulo 5

Resultados

Este capítulo é dividido entre os resultados obtidos na estimação do volume em ambiente simulado e os resultados alcançados na etapa real.

5.1 Etapa Simulada

Este capítulo apresenta os resultados de simulação obtidos nesta tese. Para realizar a validação do sistema Evolid, o SimTwo foi configurado para realizar diversas simulações e posteriormente utilizando o software Matlab foi possível analisar o comportamento dos dados obtidos.

5.1.1 Posição do Sensor

Esta seção contém os resultados obtidos nos testes realizados com o objetivo de analisar a melhor posição para o sensor dentro do espaço. Para isso, foram realizados dois testes em diferentes posições (A, B), conforme mostra a Figura 5.1. Também é visualizado na Figura 5.2 que o cenário é formada por um ambiente de $4 \times 6 \times 2 \text{ m}^3$ constituído por 4 objetos, sendo um cilindro, uma esfera e dois paralelepípedos, nos quais as dimensões não importa, tendo em vista que, a finalidade é observar a melhor distribuição dos pontos da nuvem.

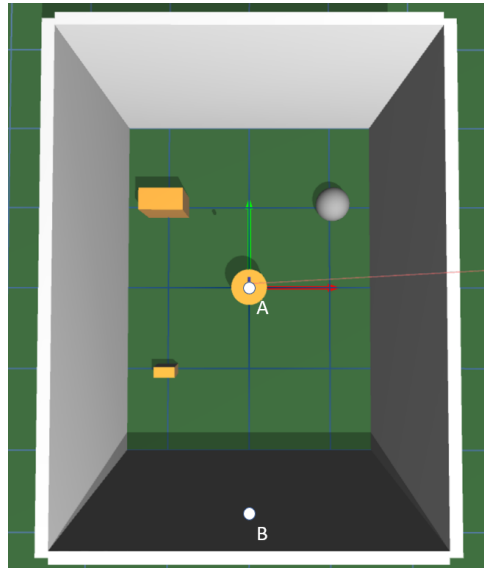


Figura 5.1: Pontos de testes.

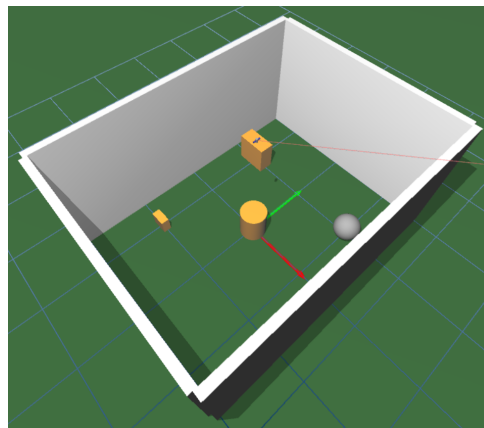


Figura 5.2: Ambiente construído para simulação.

Figura 5.3 ilustra os resultados obtidos para cada posição. A origem do sistema é considerada sempre a posição do sensor, logo observando a Figura 5.3a e Figura 5.3b a origem é no ponto com coordenada 0 em x , y e z .

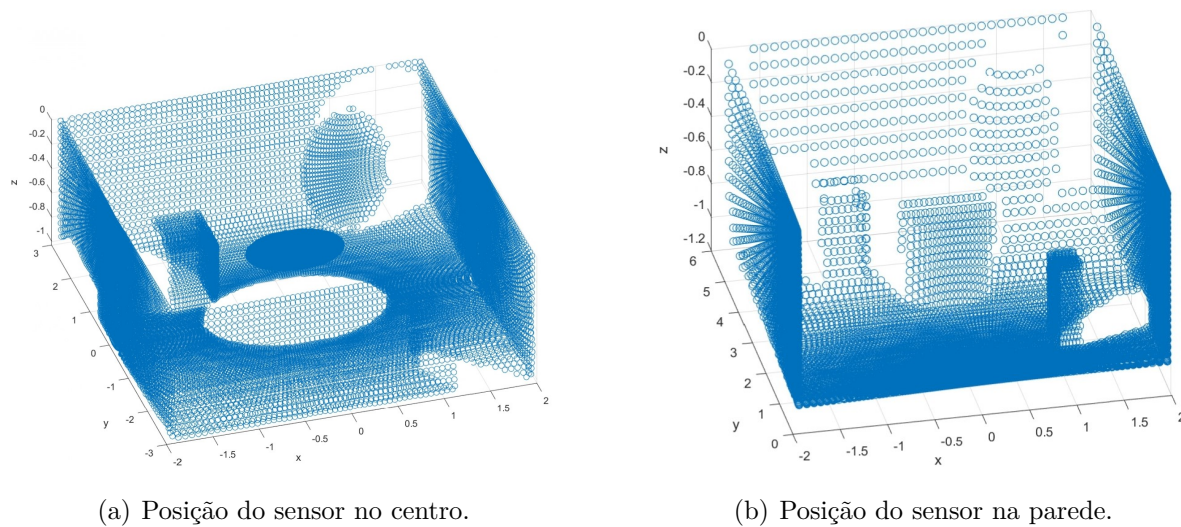


Figura 5.3: Point Cloud obtida no software Matlab

Ao analisar as figuras, é possível notar que o sistema no centro obtém uma melhor uniformidade na distribuição da *point cloud*, enquanto que para a posição na parede a densidade de pontos próximos ao sensor é muito maior e quanto mais distante do sensor, a densidade diminui, isso se deve ao formato do varrimento, que gera um padrão onde a medida que a distância do sensor aumenta os pontos são distorcidos e perdem a resolução. Também é perceptível a diferença na quantidade de pontos, visto que a nuvem formada na posição A possui 32400 pontos e na posição B 16200 pontos, o que pode ser justificado pois quando o sensor está na parede, sua faixa de inclinação é limitada em 90° enquanto para o sensor no centro ela ocorre até 180° .

Como a posição do sensor A apresentou o melhor resultado, nas simulações que se seguiram considerou-se que o sensor estaria posicionado em A.

5.1.2 Resolução da Grelha

Esta seção contém os resultados obtidos das três simulações realizadas para aplicar a abordagem tratada na seção 4.4.1.

A Tabela 5.1 apresenta os resultados adquiridos para as três simulações. Conforme ilustrado na Figura 4.14 a coluna a , b e c representam, respectivamente, a altura do sensor, a diferença entre a altura do sensor e do cubo e a distância entre os pontos.

Tabela 5.1: Especificações para cada simulação.

Simulação	Dimensões Sala (m)	Dimensões Cubo (m)	a (m)	b (m)	c (m)	Volume Real (m ³)
1	4 x 3 x 2	0,25 x 0,5 x 0,5	2,0	1,5	0,0261	0,0625
2	4 x 3 x 2	0,25 x 0,125 x 0,25	2,0	1,75	0,0305	0,0078
3	4 x 3 x 2,5	0,25 x 0,5 x 0,5	2,5	2	0,0349	0,0625

A simulação consiste no mapeamento de um cubo localizado no centro do ambiente em que variou-se na simulação 2 o tamanho do objeto e na simulação 3 as dimensões da sala. O motor foi configurado com um passo de 1°. A nuvem de pontos resultante de cada simulação é visualizada na Figura 5.4.

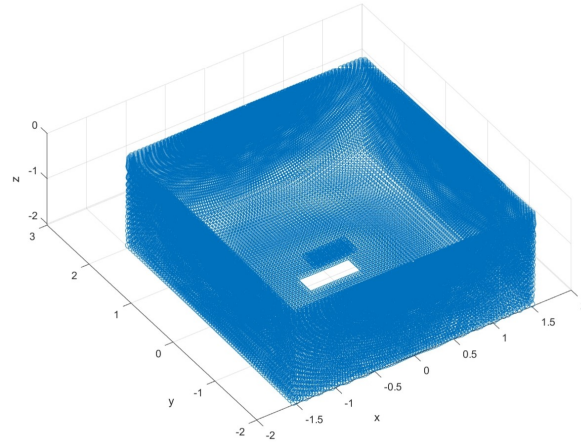
A Tabela 5.2 apresenta os resultados obtidos e também apresenta o resultado do cálculo do erro absoluto entre o volume real e o volume estimado. Para cada simulação foi realizado um teste com uma grelha fixa de 0,02 m e com a grelha adaptativa o mais próximo do valor de c .

Tabela 5.2: Resultados para escolha da resolução da Grelha

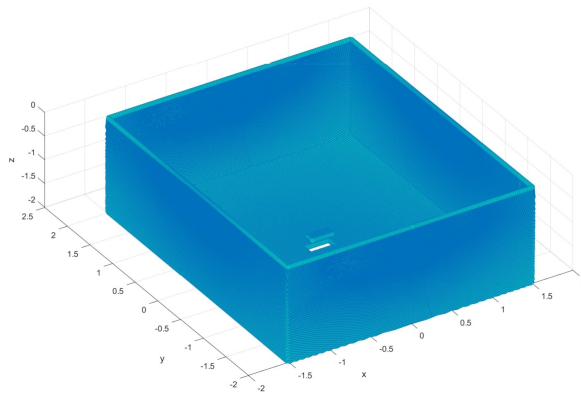
Simulação	Grelha Fixa			Grelha Adaptativa		
	Resolução (m)	Volume Estimado (m ³)	Erro (%)	Resolução (m)	Volume Estimado (m ³)	Erro (%)
1	0,02	0,0380	39,20	0,027	0,0623	0,32
2	0,02	0,0065	16,67	0,0306	0,0084	7,69
3	0,02	0,0390	37,60	0,035	0,0643	2,88

Na Figura 5.5 é visualizado a grelha de voxel adaptativa para cada simulação e na Figura 5.6 a grelha de voxel com resolução fixa em 0,02 m.

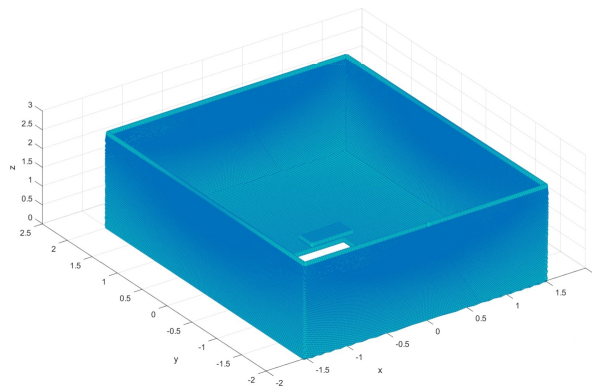
Analisando a tabela e os erros encontrados, a grelha adaptativa possui resultados melhores quando comparados a fixa. Percebe-se que a grande diferença é justificada ao



(a) Simulação 1

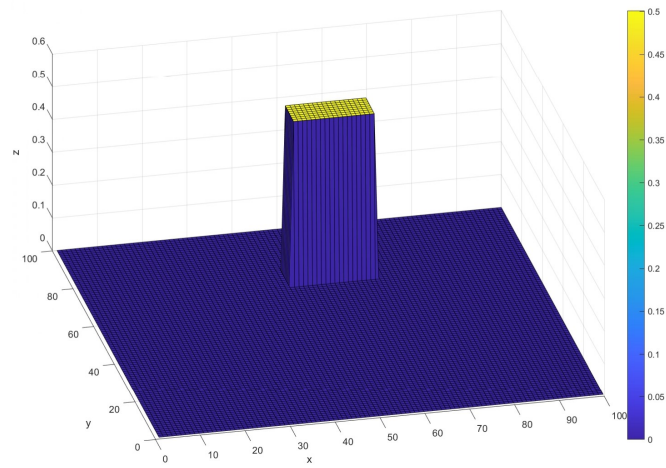


(b) Simulação 2

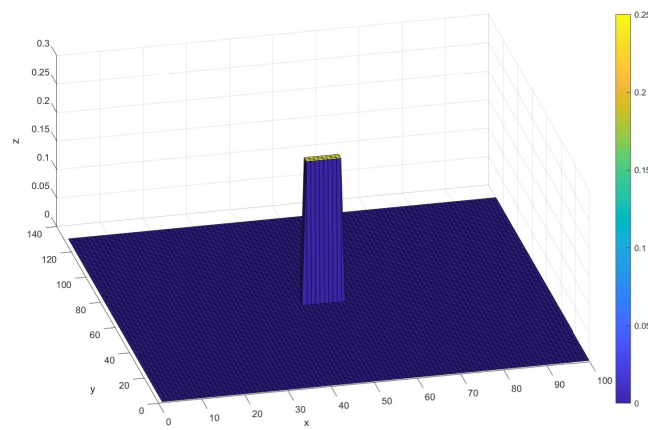


(c) Simulação 3

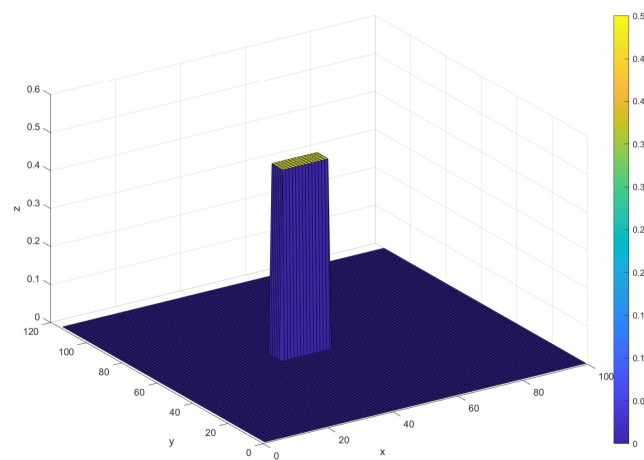
Figura 5.4: Nuvem de pontos.



(a) Simulação 1 - Resolução de 0,027

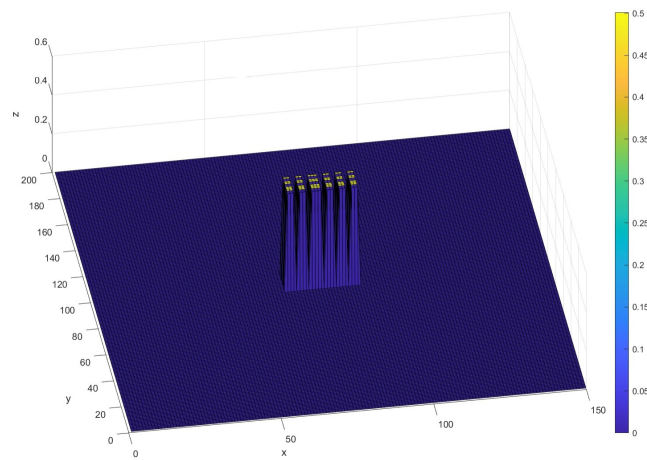


(b) Simulação 2 - Resolução de 0,031

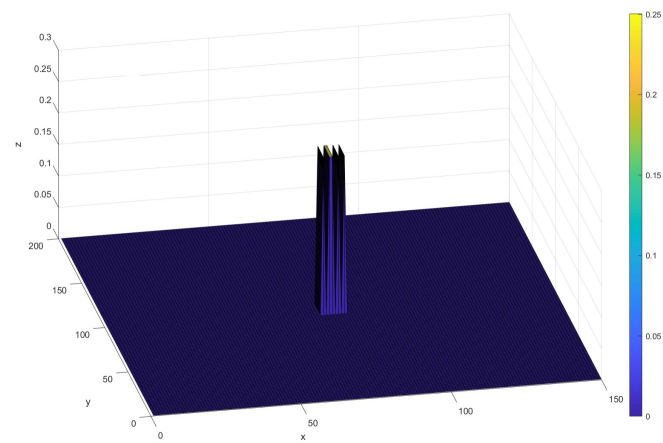


(c) Simulação 3 - Resolução de 0,035

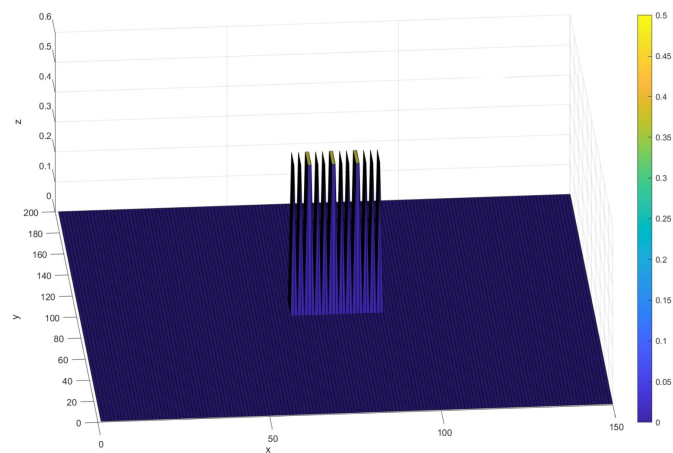
Figura 5.5: Grelhas de Voxel com resolução adaptativa.



(a) Simulação 1



(b) Simulação 2



(c) Simulação 3

Figura 5.6: Grelhas de Voxel com resolução fixa de 0,02.

analisar as figuras da grelha fixa onde a resolução da grelha é tão baixa que há espaços vazios entre o objeto.

5.1.3 Posição do Cubo

Na terceira etapa, foram realizadas 3 simulações com um cubo de $0,25 \times 0,5 \times 0,5 \text{ m}^3$ variando sua posição dentro do ambiente de simulação. A Tabela 5.3 contém as coordenadas da posição do cubo para cada simulação. Essas distâncias são medidas a partir do laser scanner, que é considerado a origem das coordenadas do sistema.

Tabela 5.3: Posições do cubo para cada simulação

	x (m)	y (m)
Simulação 1	-0,5	0
Simulação 2	0,25	0,5
Simulação 3	0	1

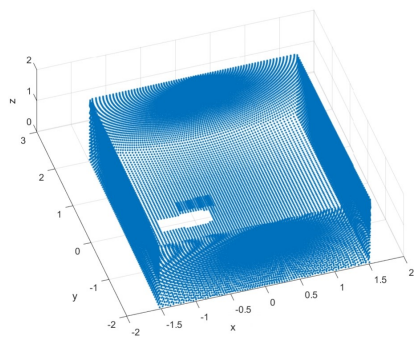
Ao adquirir a nuvem de pontos em coordenadas esféricas de cada simulação, o arquivo texto foi exportado num formato compatível com o Matlab e realizado a mudança de coordenada, obtendo então, as nuvem de pontos da Figura 5.7.

O volume do cubo corresponde a $0,0625 \text{ m}^3$ e o resultado para cada simulação está presente na Tabela 5.4.

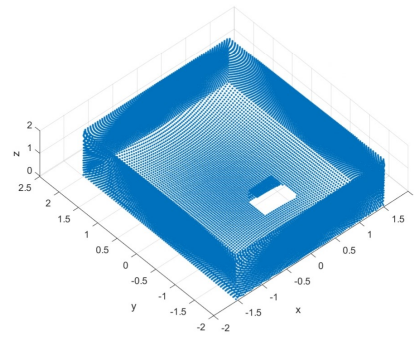
Tabela 5.4: Posições do cubo para cada simulação

	Volume Estimado (m^3)	Erro (%)
Simulação 1	0,0623	0,32
Simulação 2	0,0626	0,16
Simulação 3	0,0621	0,64

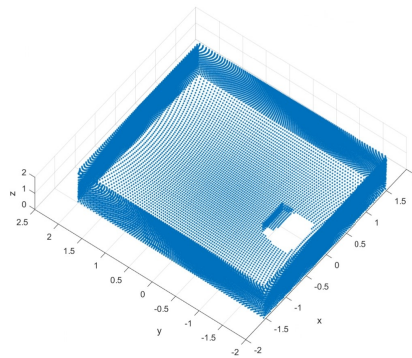
Através do baixo erro encontrado na Tabela 5.4 percebe-se a validação da abordagem. Notando-se que o erro cresce a medida em que a distância do objeto ao sensor aumenta, visto que, com o objeto mais distante do sensor, perde a precisão no formato de um cubo.



(a) Simulação 1



(b) Simulação 2



(c) Simulação 3

Figura 5.7: Nuvem de pontos para diferentes posições do cubo.

5.2 Etapa Real

O resultado no ambiente simulado comprova o funcionamento dos métodos aplicados até o momento, permitindo o início da criação do protótipo para aplicação em ambiente real.

5.2.1 Teste com Servo Motor

Para realização do teste com servo motor foi implementado o código apresentado na Listagem 5.1 no ESP32. Esse código representa que a cada tecla pressionada no teclado é enviada a informação para incremento em um grau no eixo do motor.

Listagem 5.1: Código Acionamento Servo Motor

```
1  int angle = 0;
2  int angleStep = 1;
3
4  void setup() {
5      Serial.begin(115200);
6      servo1.attach(servoPin);
7  }
8
9  void loop() {
10     if (Serial.available()) {
11         char xx = Serial.read();
12         angle += angleStep;
13         servo1.write(angle);
14     }
15 }
```

O teste realizado é observado na Figura 5.8, onde o motor é acionado com um feixe infra-vermelho. Para observar o resultado foi colado na parede uma folha juntamente com uma régua e assim após o comando de incremento de um grau foi medido o tamanho do deslocamento do feixe. A Figura 5.9a mostra o primeiro feixe e na Figura 5.9b o feixe depois do comando enviado.

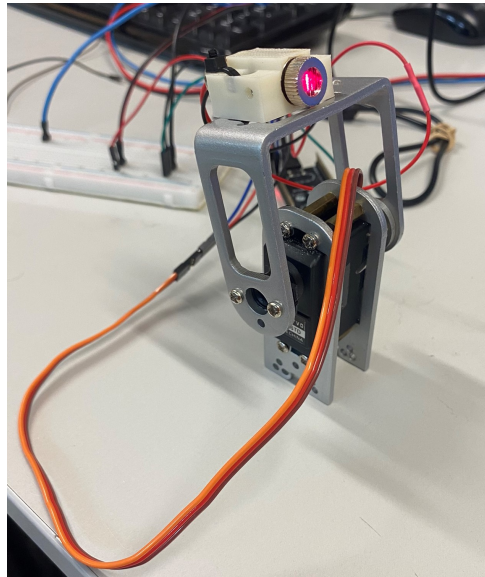
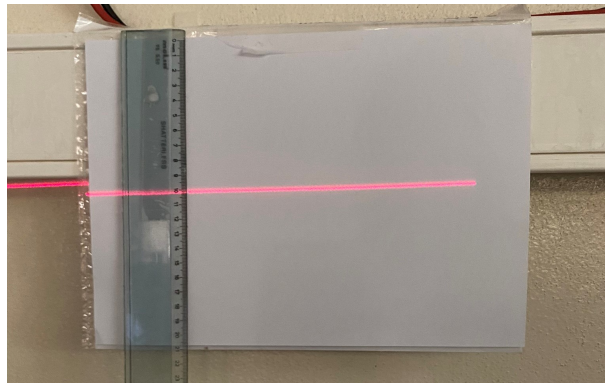
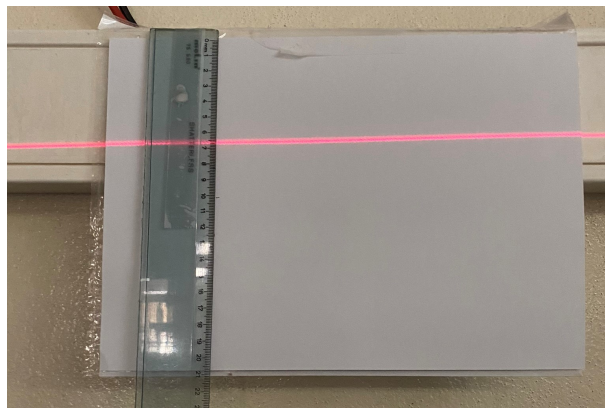


Figura 5.8: Realização do teste com o servo motor e o laser infra-vermelho.



(a) Primeiro feixe



(b) Segundo feixe

Figura 5.9: Resultado dos testes para o servo motor.

Dessa forma, pelo diagrama visto na Figura 5.10, é possível encontrar a relação do ângulo α , pela Equação 5.1:

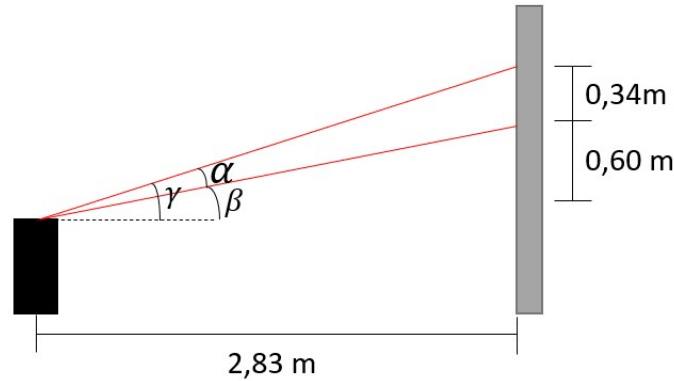


Figura 5.10: Diagrama representativo do teste com servo motor.

$$\tan(\beta) = \frac{0,60m}{2,83m} \rightarrow \beta = 11,97^\circ$$

$$\tan(\gamma) = \frac{(0,60m + 0,34m)}{2,83m} \rightarrow \gamma = 18,37^\circ$$

$$\alpha = \gamma - \beta = 6,4^\circ \quad (5.1)$$

O resultado de $6,4^\circ$ encontrado implica na imprecisão do servo motor, visto que, era proposto apenas 1° de variação. Assim tornando a sua utilização inviável para o projeto, em razão de que, não havia outro servomotor disponível no laboratório e o controle desse ângulo é determinante para a qualidade da conversão em coordenadas cartesianas. Com isso, optou-se por trabalhar com um motor de passo, realizando antes um teste para validação da precisão do motor.

5.2.2 Teste Com Motor de Passo

A partir do mesmo método utilizado na seção 5.2.1, o código da Listagem 5.2 foi aplicado no microcontrolador ESP32. O driver foi configurado com a resolução do microstep

em $1/32$. Conforme já descrito, o motor de passo realiza $1,8^\circ/step$, o que necessita da operação da equação 5.2 com o objetivo de encontrar o valor de passos (p) necessários no microcontrolador para manter a variação do passo do motor em 1° .

$$p = \frac{1^\circ \cdot 32 \text{ passos}}{1,8^\circ} = 17,78 \simeq 18 \text{ passos} \quad (5.2)$$

Listagem 5.2: Código Acionamento Motor Passo

```

1  const int M1dirpin = 14; //Motor X direction pin
2  const int M1steppin = 27; //Motor X step pin
3  const int M1en=12; //Motor X enable pin
4
5  void setup(){
6      pinMode(M1dirpin,OUTPUT);
7      pinMode(M1steppin,OUTPUT);
8      pinMode(M1en,OUTPUT);
9      Serial.begin(115200);
10     digitalWrite(M1en,LOW);// Low Level Enable
11 }
12
13 void loop(){
14     int j;
15     delayMicroseconds(2);
16     digitalWrite(M1dirpin,LOW);
17     if (Serial.available()) {
18         char xx = Serial.read();
19         for(j=0;j<18;j++){
20             digitalWrite(M1steppin,LOW);
21             delayMicroseconds(2000);
22
23             digitalWrite(M1steppin,HIGH); // step
24             delayMicroseconds(2000);
25             digitalWrite(M1steppin,LOW);
26             delayMicroseconds(2000);

```

27 }
 28 }

Um laser infra-vermelho foi fixado ao suporte acoplado no eixo do motor, Figura 5.11, e a cada tecla pressionada recebe o comando para inclinar em 1° .

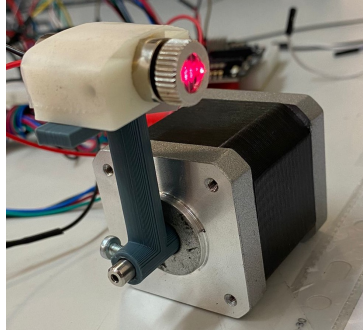


Figura 5.11: Realização do teste com o motor de passo e o laser infra-vermelho.

Dessa forma, foram retiradas cinco medidas com o auxílio de um paquímetro e de uma régua, como exemplo, as duas primeiras visualizadas no diagrama da Figura 5.12.

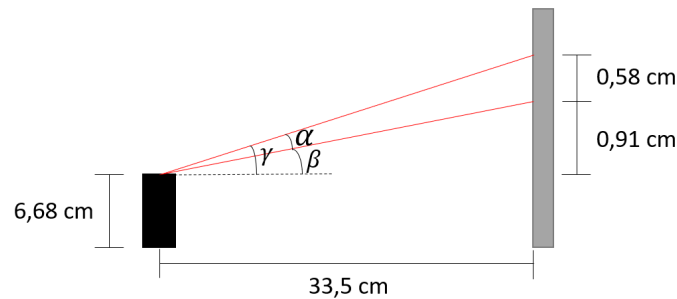


Figura 5.12: Diagrama representativo do teste com motor de passo.

O passo realizado pelo motor (α) é desenvolvido na Equação 5.3, em que, $d1$ e $d2$ representam a distância do primeiro e do segundo feixe ao chão.

$$\tan(\gamma) = \frac{(d1 - 6,68)}{33,5}$$

$$\tan(\beta) = \frac{(d2 - 6,68)}{33,5}$$

$$\alpha = \gamma - \beta \quad (5.3)$$

A distância dos lasers medidos ao chão são de: 8,17 cm, 7,59 cm, 7,01 cm, 6,43 cm, 5,84 cm. Na Tabela 5.5 é demonstrado o valor de α para cada diferença de distância entre dois feixes.

Tabela 5.5: Resultados Motor de Passo

	Distância (cm)	Ângulo α
feixe 1 - feixe 2	0,58	0,991°
feixe 2 - feixe 3	0,58	0,991°
feixe 3 - feixe 4	0,58	0,991°
feixe 4 - feixe 5	0,59	1,009°

Como é possível notar, a precisão do motor de passo é maior comparada ao servo motor devido a resolução do ângulo ter sido muito próxima a pretendida e as medidas não oscilarem entre si. Tendo em vista esse resultado, o motor de passo será utilizado para o Sistema Evolid.

5.3 Teste com LiDAR Neato XV 11

Em primeiro lugar, foi validado o código para verificação dos dados enviado pelo Neato XV 11, onde se colocou um objeto circular à volta do sensor. Os papéis utilizados variavam a distância com o laser em aproximadamente 22 cm, conforme Figura 5.13. As medidas constituíram de 0 até 179°, sendo então adquiridos ao final 180 feixes.

Os dados foram adquiridos no monitor serial do software Arduino IDE com duas colunas, uma contendo o número do feixe (que coincide com o ângulo azimutal) e a distância medida, dessa forma gerando o gráfico da Figura 5.14. Como os valores de distância não foram medidos precisamente, o objetivo era analisar se os dados recebidos permanecessem perto da distância entre o objeto e o laser, logo, analisando o gráfico, foi comprovado o funcionamento do código utilizado para leitura dos dados enviados pelo laser.

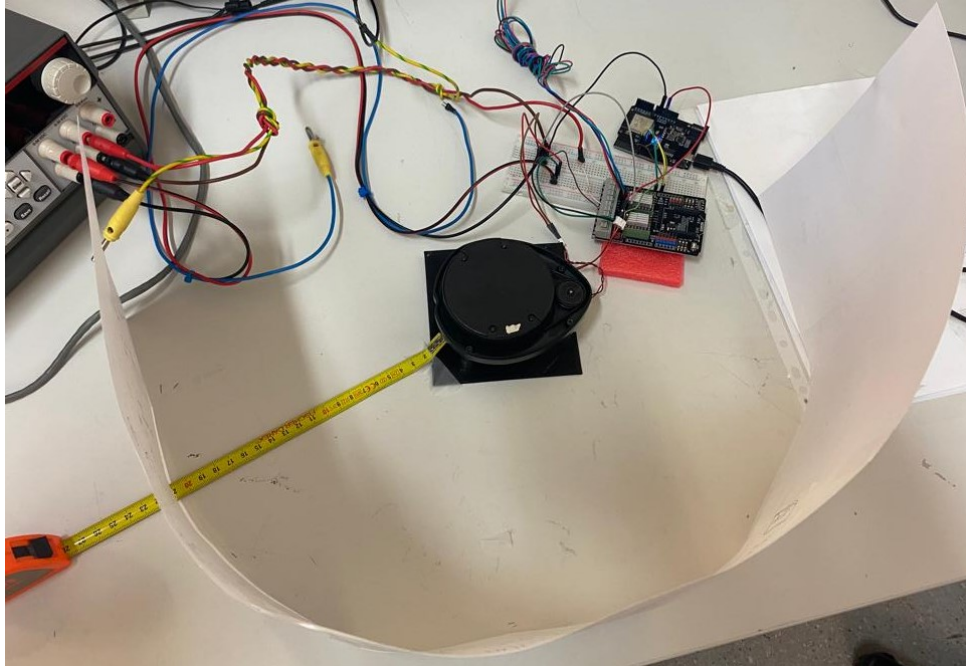


Figura 5.13: Realização do teste com LiDAR.

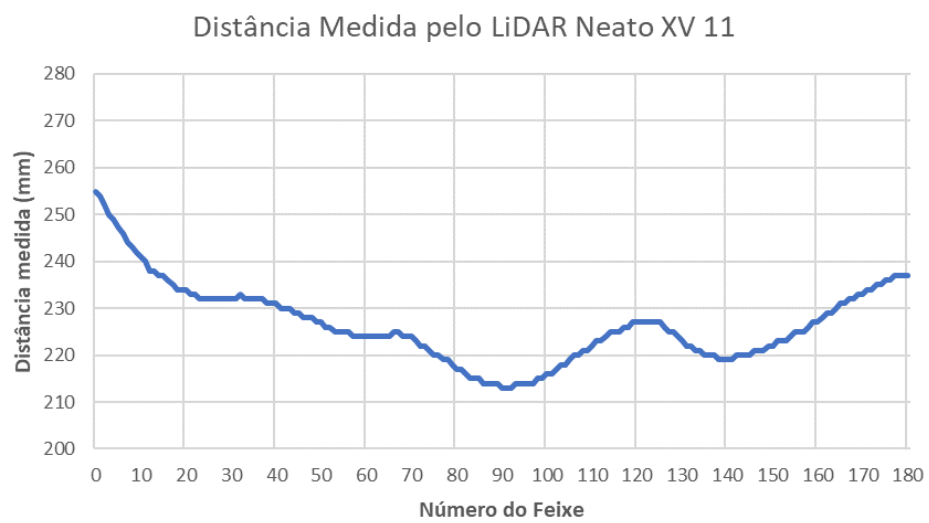


Figura 5.14: Distância Medida pelo LiDAR Neato XV 11.

5.4 Validação da Máquina de Estados

Como proposta para análise da máquina de estados, foram executados três mapeamentos com o sistema da Figura 5.15, tendo como objetivo a visualização da nuvem de pontos adquiridas no software Matlab. O sensor foi configurado para realizar 20 varrimentos com ângulo de inclinação variando de 0° até 19° , o que resulta em 19 passos gerados pelo motor.



Figura 5.15: Sistema utilizado para teste.

Em seguida, os resultados são ilustrados nas Figuras 5.16, 5.17 e 5.18, em que, é comparado na esquerda o ambiente real e na direita a *point cloud* criada depois da conversão de coordenadas.

Comparando as Figuras, é possível concluir que a máquina de estados teve seu funcionamento comprovado através da semelhança encontrada entre o real e a nuvem de pontos.

Os métodos apresentados se mostraram eficazes na fase simulada, principalmente porque os dados foram gerados por software, ou seja, sem a presença de ruído, o que garantiu um mapeamento preciso. Durante a realização em ambiente real, algumas dificuldades

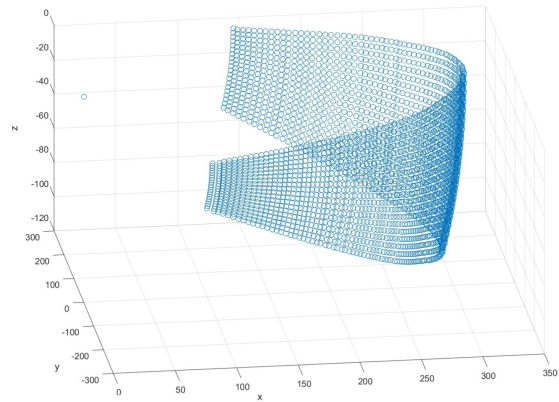
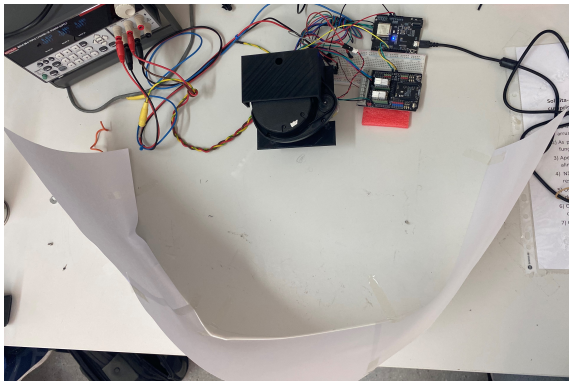


Figura 5.16: Resultado da aquisição e processamento da máquina de estados.

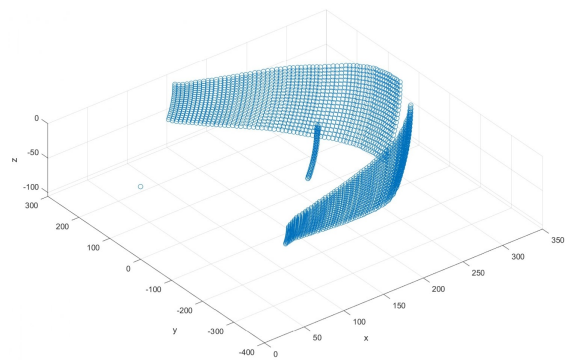


Figura 5.17: Resultado da aquisição e processamento da máquina de estados com um objeto cilíndrico.

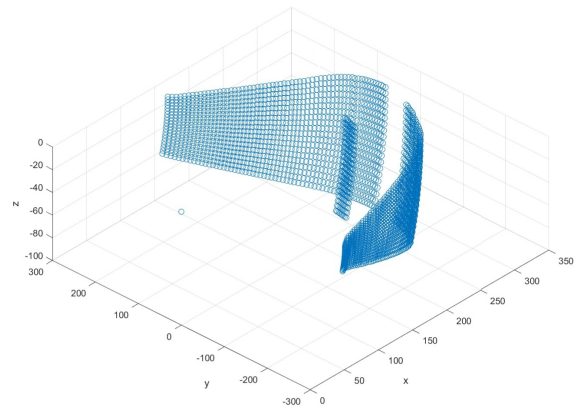
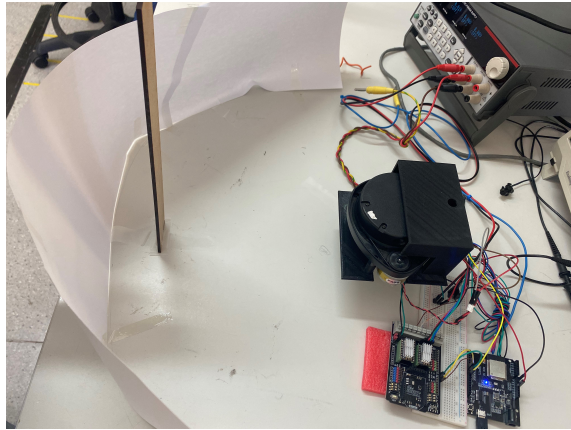


Figura 5.18: Resultado da aquisição e processamento da máquina de estados com uma régua.

foram percebidas como a falta de controle preciso do servo motor, assim como, um estudo aprofundado para realizar um algoritmo para correção dos pontos, realizando mais de uma medida por varrimento, evitando dados ruidosos.

Capítulo 6

Conclusão e Trabalho Futuros

A estimação de volume de materiais é extremamente útil para tornar as operações mais eficientes para as empresas por meio do rastreamento de estoques. Várias indústrias estão a mudar para sensores automatizados, de alta tecnologia e sem contato, que normalmente têm custos mais altos e, portanto, estão fora do alcance das pequenas empresas. O principal objetivo desse trabalho foi desenvolver um sistema de mapeamento em 3D formado por um LiDAR 2D de baixo custo capaz de estimar o volume de um espaço interior.

O processo começou com a escolha de um sensor LiDAR capaz de medir a distância desejada. Em seguida, utilizando o simulador SimTwo, foi criado um ambiente de simulação para representar um cenário real, implementando características reais do sensor Neato XV11. A partir do simulador a nuvem de pontos foi gerada e posteriormente os métodos para o processamento dos dados foram testados no software Matlab.

Na etapa simulada foram realizados três tipos de teste. Em primeiro lugar, verificou-se que a melhor posição para o sensor é no centro do cenário. Em seguida, através de três simulações, a grelha adaptativa apresenta erros menores no cálculo de volume comparado a grelha fixa. E por último, a abordagem para estimativa de volume foi validada para o cubo em diferentes posições, sendo possível a implementação no real.

Na etapa real, é notável pelos testes que o motor de passo possui o controle desejado do ângulo de inclinação ao contrário da resposta do servo motor. Além disso, é realizada a validação do código para leitura dos dados do sensor laser.

A aquisição de dados da etapa real foi modelada numa máquina de estados para auxiliar na implementação do sistema. O código foi implementado no sistema Evolid e verificou-se sua efetividade.

De acordo com os resultados apresentados, conclui-se que os métodos foram suficientes para estimação do volume em ambiente simulado, validando o objetivo inicialmente teorizada. Em ambiente real, os métodos propostos foram suficiente para testar a abordagem necessária para aquisição dos dados. Algumas limitações são encontradas devido a necessidade de precisão das variáveis adquiridas, tornado o mapeamento tridimensional incompleto. Isso poderia ser corrigido com o desenvolvimento de um filtro para ruído e de um algoritmo de correção, que preferiu-se deixar como trabalho futuro.

6.1 Trabalhos Futuros

Como próximos passos, sugere-se dar continuidade a este mesmo projeto aumentando sua complexidade. Com isso, algumas implementações para trabalhos futuros são apresentadas:

- Implementar a automatização da resolução da grelha.
- Implementar um algoritmo para correção dos pontos, utilizando mais de um varrimento por *step*.
- Validar o ambiente de simulação em ambiente real.

Bibliografia

- [1] *Volume determination for inventory purposes by means of laser scanning*. URL: <https://www.laserscanning-europe.com/en/news/volume-determination-inventory-purposes-means-laser-scanning>, Acessado em: 2022-05-20.
- [2] J. Franca, M. Gazziro, A. Ide e J. Saito, “A 3D scanning system based on laser triangulation and variable field of view,” em *IEEE International Conference on Image Processing 2005*, vol. 1, 2005, pp. I–425. DOI: 10.1109/ICIP.2005.1529778.
- [3] J. Braun, J. Lima, A. I. Pereira, C. Rocha e P. Costa, “Searching the Optimal Parameters of a 3D Scanner Through Particle Swarm Optimization,” em *International Conference on Optimization, Learning Algorithms and Applications*, Springer, 2021, pp. 138–152.
- [4] M. Weinmann, “Preliminaries of 3D point cloud processing,” em *Reconstruction and Analysis of 3D Scenes*, Springer, 2016, pp. 17–38.
- [5] N. Namitha, S. Vaitheeswaran, V. Jayasree e M. Bharat, “Point cloud mapping measurements using kinect RGB-D sensor and kinect fusion for visual odometry,” *Procedia Computer Science*, vol. 89, pp. 209–212, 2016.
- [6] B. J. B. K. Reis, “Codificação das cores de uma point cloud através da sua divisão em filamentos,” 2018.
- [7] *Simtwo*. URL: <https://github.com/P33a/SimTwo>, Accessed: 2021-08-17.

- [8] A. Discant, A. Rogozan, C. Rusu e A. Bensrhair, “Sensors for obstacle detection-a survey,” em *2007 30th International Spring Seminar on Electronics Technology (ISSE)*, IEEE, 2007, pp. 100–105.
- [9] T. X. Rossi, *Mapeamento tridimensional de ambientes internos utilizando um sensor LIDAR*, <http://www.monografias.ufop.br/handle/35400000/2439>, 2020.
- [10] H. Weber, *Modo de Funcionamento e Variantes dos Sensores LiDAR*, 2018.
- [11] *Sick sensor intelligence*. URL: www.sick.com, Acessado em: 2022-04-11.
- [12] G. Perchet e M. Lescure, “Magnification of phase shift for a laser rangefinder: intrinsic resolution and improved circuit,” *Journal of optics*, vol. 29, n.º 3, p. 229, 1998.
- [13] X. Zhang, J. Morris e R. Klette, *Volume measurement using a laser scanner*, 2005.
- [14] *SparkFun*. URL: <https://www.sparkfun.com/>, Acessado em: 2021-11-29.
- [15] J. A. B. Neto, J. L. Lima, A. I. Pereira e P. Costa, “Low-cost 3d lidar-based scanning system for small objects,” *International Conference on Industrial Technology*, 2021.
- [16] N. L. Pavan e D. R. dos Santos, “Um método automático para registro de dados laser scanning terrestre usando superfícies planas,” *Boletim de Ciências Geodésicas*, vol. 21, n.º 3, pp. 572–589, 2015, ISSN: 1982-2170. DOI: <http://dx.doi.org/10.1590/S1982-21702015000300032>.
- [17] Y.-S. Chou e J.-S. Liu, “A Robotic Indoor 3D Mapping System Using a 2D Laser,” *International Journal of Advanced Robotic Systems*, 2012. DOI: 10.5772/54655.
- [18] O. Wulf e B. Wagner, “Fast 3D scanning methods for laser measurement systems,” *International Conference on Control Systems and Computer Science*, pp. 312–317, 2011.
- [19] R. Marchant, P. Guerrero e J. Ruiz-del-Solar, “A portable ground-truth system based on a laser sensor,” *RoboCup*, pp. 234, 235, 2011.
- [20] G. S. de Moura, *Development and Simulation of a Low-Cost Ground-Truth Localization System for Mobile Robot*, <http://hdl.handle.net/10198/23511>, 2020.

- [21] J. E. A. Jacobo, *Geração de Mapas de ambientes utilizando um Sistema de Percepção LIDAR – 3D*, 2013.
- [22] P. Tang, D. Huber, B. Akinici, R. Lipman e A. Lytle, “Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques,” *Automation in Construction*, vol. 19, n.º 7, pp. 829–843, 2010, ISSN: 0926-5805. DOI: <https://doi.org/10.1016/j.autcon.2010.06.007>. URL: <https://www.sciencedirect.com/science/article/pii/S0926580510000907>.
- [23] K. Li, J. Wang e D. Qi, “The development and application of an original 3D laser scanning: a precise and nondestructive structural measurements system,” *Frattura ed Integrità Strutturale*, 2020. DOI: 10.3221/IGF-ESIS.51.28.
- [24] H. Surmann, A. Nüchter e J. Hertzberg, “An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments,” *Robotics and Autonomous Systems*, 2003. DOI: 10.1016/j.robot.2003.09.004.
- [25] P. Karim, D. A. V. e W. K. A., “LiDARPheno – A Low-Cost LiDAR-Based 3D Scanning System for Leaf Morphological Trait Extraction,” *Frontiers in Plant Science*, vol. 10, 2019.
- [26] W. Y. Amaglo, *Volume Calculation Based on LiDAR Data*, 2021.
- [27] *Google’s Street View cars are now giant, mobile 3D scanners*. URL: <https://arstechnica.com/gadgets/2017/09/googles-street-view-cars-are-now-giant-mobile-3d-scanners/>, Acessado em: 2022-04-12.
- [28] H. Li, X. Zhou, Y. Chen, Q. Zhang, D. Zhao e D. Qian, “Comparison of 3D Object Detection Based on LiDAR Point Cloud,” *IEEE 8th Data Driven Control and Learning Systems Conference*, pp. 678–685, 2019.
- [29] G. Yang, a. M. B. Simone Mentasti, Y. Wang, F. Braghin e F. Cheli, “LiDAR point-cloud processing based on projection methods: a comparison,” 2020.
- [30] R. Huang, W. Zhang, A. Kundu et al., “An LSTM Approach to Temporal 3D Object Detection in LiDAR Point Clouds,” *CoRR*, vol. abs/2007.12392, 2020.

- [31] G. Yang, S. Mentasti, M. Bersani, Y. Wang, F. Braghin e F. Cheli, “A Survey on 3D Object Detection Methods for Autonomous Driving Applications,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 3782–3795, 2019.
- [32] G. Zamanakos, L. Tsochatzidis, A. Amanatiadis e I. Pratikakis, “A comprehensive survey of LIDAR-based 3D object detection methods with deep learning for autonomous driving,” *Computers & Graphics*, vol. 99, pp. 153–181, 2021, ISSN: 0097-8493. DOI: <https://doi.org/10.1016/j.cag.2021.07.003>.
- [33] J. N. Christensen, *3D scanners take volume measurement to a new level*. 2020, pp. 98–102.
- [34] N. Nehruji, *Evaluation of 3D Laser Scanning for Volume Calculations*, 2020. URL: <https://www.conservesolution.com/evaluation-of-3d-laser-scanning-for-volume-calculations/>, Acessado em: 2022-05-24.
- [35] P. H. Loz, S. C. Angulo, M. S. Rebmann e E. Tutumluer, “Use of a 3D Structured-Light Scanner to Determine Volume, Surface Area, and Shape of Aggregates,” *Journal of Materials in Civil Engineering*, vol. 33, n.º 9, p. 04 021 240, 2021.
- [36] A. Bewley, R. Shekhar, S. Leonard, B. Upcroft e P. Lever, “Real-time volume estimation of a dragline payload,” em *2011 IEEE International Conference on Robotics and Automation*, IEEE, 2011, pp. 1571–1576.
- [37] F. Hosoi, Y. Nakai e K. Omasa, “3-D voxel-based solid modeling of a broad-leaved tree for accurate volume estimation using portable scanning lidar,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 82, pp. 41–48, 2013, ISSN: 0924-2716. DOI: <https://doi.org/10.1016/j.isprsjprs.2013.04.011>.
- [38] A. Bienert, C. Hess, H. Maas e G. Von Oheimb, “A voxel-based technique to estimate the volume of trees from terrestrial laser scanner data,” *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2014.
- [39] *Neato Robotics XV-11 Tear-down*. URL: <https://www.sparkfun.com/news/490>, Acessado em: 2021-03-18.

- [40] *Hacking the Neato XV-11 (2015)*. URL: <https://xv11hacking.wikispaces.com/>, Acessado em: 2021-03-18.
- [41] D. Campos, J. Santos, J. Gonçalves e P. Costa, “Modeling and simulation of a hacked neato xv-11 laser scanner,” em *Robot 2015: Second Iberian Robotics Conference*, Springer, 2016, pp. 425–436.
- [42] J. Lima, J. Gonçalves e P. J. Costa, “Modeling of a low cost laser scanner sensor,” em *CONTROLO’2014–Proceedings of the 11th Portuguese Conference on Automatic Control*, Springer, 2015, pp. 697–705.
- [43] *ESP32 Pinout Reference*. URL: <https://randomnerdtutorials.com/esp32-pinout-reference-gpios/>, Acessado em: 2022-04-20.
- [44] E. Systems, *ESP32.WROOM.32 Datasheet*, Rep. Tech., 2022.
- [45] RepRap, *Motor nema 17 42BYGHW811*. URL: <https://www.reprap.pt/peças/motores-de-passos/motor-nema-17-42byghw811>, Acessado em: 2022-06-02.
- [46] DFRobot, *DRI0023*. URL: https://wiki.dfrobot.com/Stepper_Motor_Shield_For_Arduino_DRV8825__SKU_DRI0023, Acessado em: 2022-06-14.
- [47] S. de Álgebra e A. P. G. Pires, “Teorema da Mudança de Coordenadas,”
- [48] S. Liu, J. Zheng, X. Wang, Z. Zhang e R. Sun, “Target detection from 3D point-cloud using Gaussian function and CNN,” em *2019 34rd Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, IEEE, 2019, pp. 562–567.
- [49] T. da Silva Almeida, A. C. R. da Silva e S. R. Rossi, “MODELAGEM E ANÁLISE DO PADRÃO IEEE 1451 POR MEIO DE MÁQUINAS DE ESTADOS FINITOS,”
- [50] D. M. F. Campos, “Geração de código para Máquina de Estados/Grafcet em sistemas embebidos,” 2022.
- [51] P. Costa, J. Gonçalves, J. Lima e P. Malheirosa, “SimTwo Realistic Simulator: A Tool for the Development and Validation of Robot Software,” *Theory and Applications of Mathematics & Computer Science*, vol. 1, n.º 1, pp. 17–33, 2011.

- [52] G. Amaral e P. Costa, “SimTwo as a simulation environment for flight robot dynamics evaluation,” *U.Porto Journal of Engineering*, pp. 80–88, 2015.
- [53] A. F. de Domingues e Silva, “Ferramenta para Desenvolvimento de Inteligência em Jogos Simulados em ambiente SimTwo,” 2015.
- [54] A. D. M. Abreu, “Autopilot Simulator Prototype for Autonomous Driving based on SimTwo,” 2020.
- [55] J. L. Lima, J. A. Gonçalves, P. G. Costa e A. P. Moreira, “HUMANOID REALISTIC SIMULATOR: The servomotor joint modeling,” 2009.