



**Programa de Pós-Graduação em Instrumentação, Controle e
Automação de Processos de Mineração (PROFICAM)
Escola de Minas, Universidade Federal de Ouro Preto (UFOP)
Associação Instituto Tecnológico Vale (ITV)**

Dissertação

**INSPEÇÃO ROBÓTICA DE TUBULAÇÕES: DISPOSITIVO DE BAIXO CUSTO E
SISTEMA DE NAVEGAÇÃO**

Alexandre Magno de S. Thiago Filho

**Ouro Preto
Minas Gerais, Brasil
2023**

Alexandre Magno de S. Thiago Filho

**INSPEÇÃO ROBÓTICA DE TUBULAÇÕES: DISPOSITIVO DE BAIXO CUSTO E
SISTEMA DE NAVEGAÇÃO**

Dissertação apresentada ao Programa de Pós-Graduação em Instrumentação, Controle e Automação de Processos de Mineração da Universidade Federal de Ouro Preto e do Instituto Tecnológico Vale, como parte dos requisitos para obtenção do título de Mestre em Engenharia de Controle e Automação.

Orientador: Prof. Gustavo Medeiros Freitas,
D.Sc.

Coorientador: Prof. Héctor Azpúrua, D.Sc.

Ouro Preto
2023

SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

T422i Thiago Filho, Alexandre Magno de S.

Inspeção robótica de tubulações [manuscrito]: dispositivo de baixo custo e sistema de navegação. / Alexandre Magno de S Thiago Filho. - 2023.

91 f.: il.: color., gráf., tab..

Orientador: Prof. Dr. Gustavo Medeiros Freitas.

Coorientador: Dr. Héctor Azpúrua.

Dissertação (Mestrado Profissional). Universidade Federal de Ouro Preto. Programa de Mestrado Profissional em Instrumentação, Controle e Automação de Processos de Mineração. Programa de Pós-Graduação em Instrumentação, Controle e Automação de Processos de Mineração.

Área de Concentração: Engenharia de Controle e Automação de Processos Minerais.

1. Robótica - Minas e mineração. 2. Tubulação - Inspeção. 3. Sistemas de navegação. I. Freitas, Gustavo Medeiros. II. Azpúrua, Héctor. III. Universidade Federal de Ouro Preto. IV. Título.

CDU 681.5:622.2

Bibliotecário(a) Responsável: Sione Galvão Rodrigues - CRB6 / 2526



FOLHA DE APROVAÇÃO

Alexandre Magno de S Thiago Filho

Inspeção robótica de tubulações: Dispositivo de baixo custo e sistema de navegação

Dissertação apresentada ao Programa de Pós-Graduação em Instrumentação, Controle e Automação de Processos de Mineração (PROFICAM), Convênio Universidade Federal de Ouro Preto/Associação Instituto Tecnológico Vale - UFOP/ITV, como requisito parcial para obtenção do título de Mestre em Engenharia de Controle e Automação na área de concentração em Instrumentação, Controle e Automação de Processos de Mineração

Aprovada em 20 de outubro de 2023

Membros da banca

Doutor - Gustavo Medeiros Freitas - Orientador - Universidade Federal de Minas Gerais
Doutor - Gustavo Pessin - Instituto Tecnológico Vale
Doutor - Marcelo Becker - Universidade de São Paulo
Doutor - Vivian Suzano Medeiros - Universidade de São Paulo

Gustavo Medeiros Freitas, orientador do trabalho, aprovou a versão final e autorizou seu depósito no Repositório Institucional da UFOP em 09/01/2023



Documento assinado eletronicamente por **Bruno Nazário Coelho, COORDENADOR(A) DE CURSO DE PÓS-GRADUAÇÃO EM INST. CONTROLE AUTOMAÇÃO DE PROCESSOS DE MINERAÇÃO**, em 15/01/2024, às 11:49, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **0650771** e o código CRC **06067AC8**.

Referência: Caso responda este documento, indicar expressamente o Processo nº 23109.000151/2024-31

SEI nº
0650771

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: (31)3552-7352 - www.ufop.br

Agradecimentos

Há dois anos, em meio a uma situação caótica no mundo, a pandemia, iniciei uma nova etapa da minha vida dedicada a adquirir mais conhecimentos sobre o mundo em que vivo. Conhecimentos estes que de nada serão úteis se forem motivo de orgulho, vaidade ou egoísmo. Portanto, espero utilizar tudo o que foi aprendido para fazer a diferença e transformar o mundo em um lugar melhor.

Agradeço, primeiramente, a Deus por todas as oportunidades, aos meus pais pelo apoio incondicional, ao meu irmão pela amizade, à minha família pelos exemplos e à Damaris por toda a ajuda e incentivo durante esse período.

Agradeço também aos meus orientadores Gustavo e Héctor por todo o conhecimento compartilhado, aos meus colegas André e Maurício por dividirem o peso das obrigações, e a todos do laboratório, em especial, Nilton, Jacó, Mário, Filipe, Saulo e Renato.

Através de palavras, demonstro essa gratidão citando algumas dessas pessoas que representam todas as outras que passaram pela minha trajetória. Contando com vocês, espero poder ser hoje um ser humano melhor do que fui ontem e amanhã, melhor do que fui hoje.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, Brasil (CAPES), Código de Financiamento 001; do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq); da Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG); e da Vale SA.

*“Ninguém pode voltar atrás e fazer
um novo começo. Mas qualquer
um pode recomeçar e fazer um
novo fim.” (Chico Xavier)*

Resumo

Resumo da Dissertação apresentada ao Programa de Pós Graduação em Instrumentação, Controle e Automação de Processos de Mineração como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

INSPEÇÃO ROBÓTICA DE TUBULAÇÕES: DISPOSITIVO DE BAIXO CUSTO E SISTEMA DE NAVEGAÇÃO

Alexandre Magno de S. Thiago Filho

Outubro/2023

Orientadores: Gustavo Medeiros Freitas

Héctor Azpúrua

A inspeção de tubulações de pequenos diâmetros é uma atividade comum na indústria e desempenha um papel fundamental na garantia do fluxo eficiente de materiais. Problemas como obstruções, rachaduras e descontinuidades podem prejudicar a capacidade de transporte, tornando necessário realizar inspeções regulares em sistemas industriais. O Instituto Tecnológico Vale (ITV) propõe uma plataforma robótica para a inspeção de tubulações com um diâmetro a partir de 14 polegadas. Este dispositivo é fabricado principalmente com materiais de baixo custo, incluindo peças de polímero produzidas por meio de impressão 3D. A locomoção do robô é garantida por esteiras ajustáveis, capazes de se adaptar a diferentes condições de tubulação. O robô utiliza câmeras e diversos sensores para identificar possíveis anomalias. O software do robô é desenvolvido com base no ROS, tornando mais fácil a incorporação de funcionalidades autônomas. Esta dissertação fornece detalhes sobre a criação da plataforma robótica e apresenta resultados que abordam o projeto mecânico, eletrônico e o software incorporado no robô. Além disso, o documento descreve a implementação de um sistema de navegação autônomo que utiliza máquinas de estados e controle com base em campos vetoriais artificiais, projetados para seguir paredes e navegar no centro de corredores. Por fim, simulações e experimentos de campo realizados com o dispositivo robótico demonstram um grande potencial de uso para a realização de inspeções.

Palavras-chave: Robótica de serviços, Inspeção de tubulações, Sistemas de navegação.

Macrotema: Usina; **Linha de Pesquisa:** Robótica Aplicada à Mineração; **Tema:** Inspeção Automática de Ativos;

Abstract

Abstract of Dissertation presented to the Graduate Program on Instrumentation, Control and Automation of Mining Process as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

ROBOTIC PIPE INSPECTION: LOW-COST DEVICE AND NAVIGATION SYSTEM

Alexandre Magno de S. Thiago Filho

October/2023

Advisors: Gustavo Medeiros Freitas

Héctor Azpúrua

The inspection of small-diameter pipelines is a common activity in the industry and plays a fundamental role in ensuring the efficient flow of materials. Issues such as blockages, cracks, and discontinuities can impair the transport capacity, making it necessary to conduct regular inspections in industrial systems. The Instituto Tecnológico Vale proposes a robotic platform for inspecting pipelines with a diameter starting from 14 inches. This device is primarily manufactured using low-cost materials, including polymer components produced through 3D printing. The robot's locomotion is ensured by adjustable tracks capable of adapting to various pipeline conditions. The robot utilizes cameras and various sensors to identify potential anomalies. The robot's software is developed based on ROS, making it easier to incorporate autonomous functionalities. This dissertation provides details about the creation of the robotic platform and presents results that encompass the mechanical, electronic, and embedded software aspects of the robot's design. Additionally, the document describes the implementation of an autonomous navigation system that utilizes state machines and control based on artificial vector fields, designed for wall following and navigating through corridor centers. Finally, simulations and field experiments conducted with the robotic device demonstrate significant potential for conducting inspections.

Keywords: Service robotics, Pipe inspection, Navigation system.

Macrotheme: Mining plant; **Research Line:** Robotics Applied to Mining; **Theme:** Automatic Asset Inspection;

Lista de Figuras

| | | |
|-------------|--|----|
| Figura 2.1 | Robô de Inspeção AGH University. Fonte: (CISZEWSKI et al. , 2017). | 18 |
| Figura 2.2 | Robô Kantaro. Fonte: (NASSIRAEI et al. , 2007). | 19 |
| Figura 2.3 | Robô Spider-Smart. Fonte: (QU et al. , 2018). | 19 |
| Figura 2.4 | Robô AP-02. Fonte: (AEROPIC, 2022). | 20 |
| Figura 2.5 | Robô Tritrax. Fonte: (EDDYFI, 2022). | 20 |
| Figura 2.6 | JettyRobot 6. Fonte: (JETTY ROBOT, 2023). | 21 |
| Figura 2.7 | Máquina de estados proposta para navegação autônoma do robô para inspeção de dutos. Fonte: (PEREIRA et al. , 2021). | 23 |
| Figura 2.8 | Máquina de estados para navegação de robôs por um ambiente por meio da identificação visual de padrões. Fonte: (SALES et al. , 2014). | 24 |
| Figura 2.9 | Estados da máquina de estados equivalentes a padrões reconhecidos no campo de cultivo. Fonte: (SALES et al. , 2014). | 24 |
| Figura 2.10 | Máquina de estados desenvolvida para o robô BoniRob. Fonte: (BIBER et al. , 2012). | 25 |
| Figura 2.11 | Características reais equivalentes às ações da máquina de estados. Fonte: (BIBER et al. , 2012). | 26 |
| Figura 2.12 | Exemplo de curva e campo vetorial proposto para navegação de robôs. Fonte: (REZENDE et al. , 2021). | 27 |
| Figura 2.13 | Visualização do campo vetorial resultante que possui como intuito o robô atingir o alvo. Fonte: (DOS SANTOS et al. , 2019). | 28 |
| Figura 2.14 | Caminhos percorridos pelo MAV com perturbações no ambiente. Fonte: (NELSON et al. , 2007). | 29 |
| Figura 3.1 | Modelo tridimensional do PipeBot, robô de inspeção de dutos. Fonte: O autor. | 30 |
| Figura 3.2 | Modelo tridimensional desenvolvido no <i>SolidWorks</i> do robô de inspeção de tubulações. Fonte: O autor. | 32 |
| Figura 3.3 | Módulos que constituem a estrutura do robô: suporte (1), base (2), locomoção (3), motor (4). Fonte: O autor. | 32 |
| Figura 3.4 | Corpo do robô em montagem inicial com módulos de teste. Fonte: O autor. | 33 |
| Figura 3.5 | Montagem do módulo do motor. Fonte: O autor. | 34 |
| Figura 3.6 | Montagem interna do módulo de locomoção. Fonte: O autor. | 35 |

| | | |
|-------------|---|----|
| Figura 3.7 | Componentes impressos em 3D do sistema mecânico do robô. Fonte: O autor. | 36 |
| Figura 3.8 | Montagem das peças impressas e usinadas. Fonte: O autor. | 37 |
| Figura 3.9 | Diagrama das conexões eletrônicas do PipeBot. Fonte: O autor. | 39 |
| Figura 3.10 | Montagem externa ao robô dos componentes eletrônicos embarcados. Fonte: O autor. | 43 |
| Figura 3.11 | Montagem da eletrônica na estrutura de integração proposta. Fonte: O autor. | 44 |
| Figura 3.12 | Diagrama dos pacotes ROS que constituem o <i>software</i> do robô. O diagrama com a cor preta indica os pacotes que são executados no robô e com a cor laranja os pacotes que são executados no computador base Fonte: O autor. | 46 |
| Figura 3.13 | Interface gráfica do simulador CoppeliaSim. Fonte: O autor. | 50 |
| Figura 3.14 | Modelo do robô criado no CoppeliaSim. Fonte: O autor. | 51 |
| Figura 3.15 | Ambientes de teste criado no CoppeliaSim. Fonte: O autor. | 52 |
| Figura 4.1 | Diagrama com exemplo básico de uma máquina de estados para um robô móvel. Fonte: O autor. | 56 |
| Figura 4.2 | Máquina de estados representando a estratégia de navegação para inspeção de tubulações. Fonte: Adaptada de (AMARAL et al. , 2022). | 57 |
| Figura 4.3 | Método de detecção de bifurcações. O círculo amarelo mostra o limite máximo de distância que o robô utiliza para dividir as medições em grupos. Fonte: Adaptada de (AMARAL et al. , 2022). | 58 |
| Figura 4.4 | Exemplo de campo vetorial cujas linhas integrais convergem para um cami- nho. Fonte: (FREITAS et al. , 2020). | 59 |
| Figura 4.5 | O campo vetorial (azul) é construído pela soma de uma componente con- vergente (vermelha) com uma componente tangente (verde). Fonte: Adaptada de (FREITAS et al. , 2020). | 60 |
| Figura 4.6 | Representação de um robô com cinemática diferencial. Fonte: Adaptada de (FREITAS et al. , 2020) | 61 |
| Figura 4.7 | Adição do $p_{feedback}$ na representação gráfica do modelo diferencial. Fonte: Adaptada de (FREITAS et al. , 2020). | 62 |
| Figura 4.8 | Representação visual do robô realizando navegação se mantendo no centro das paredes no simulador. Fonte: O autor. | 63 |
| Figura 4.9 | Representação visual do robô realizando navegação seguindo a parede da esquerda no simulador. Fonte: O autor. | 65 |
| Figura 5.1 | Teste de navegação autônoma pelo controle de campos vetoriais no ambiente virtual. Fonte: O autor. | 66 |
| Figura 5.2 | Caminho percorrido pelo robô utilizando a técnica de navegação com cam- pos vetoriais em conjunto com a máquina de estados no ambiente virtual. Fonte: O autor. | 67 |

| | | |
|------------|---|----|
| Figura 5.3 | PipeBot montado com todos os componentes mecânicos e eletrônicos. Fonte: O autor. | 68 |
| Figura 5.4 | Primeiros testes em ambiente externo. Fonte: O autor. | 69 |
| Figura 5.5 | PipeBot realizando inspeções em tubulações de forma teleoperada. Fonte: O autor. | 69 |
| Figura 5.6 | Visualização da nuvem de pontos 3D gerada com a câmera Realsense durante um experimento preliminar. Fonte: O autor. | 70 |
| Figura 5.7 | PipeBot realizando inspeções em galerias de forma teleoperada. Fonte: O autor. | 71 |
| Figura 5.8 | Teste conduzido em laboratório com o robô seguindo a parede utilizando o sistema de navegação proposto. Fonte: O autor. | 71 |

Lista de Tabelas

| | | |
|------------|--|----|
| Tabela 2.1 | Comparação de robôs de inspeção de dutos e galerias. | 22 |
| Tabela 3.1 | Especificações gerais do PipeBot. | 31 |

Sumário

| | | |
|----------|---|-----------|
| 1 | Introdução | 13 |
| 1.1 | Motivação | 14 |
| 1.2 | Objetivos | 15 |
| 1.3 | Contribuições | 16 |
| 1.4 | Estrutura da dissertação | 17 |
| 2 | Trabalhos relacionados | 18 |
| 2.1 | Plataformas robóticas para inspeção de ambientes confinados | 18 |
| 2.2 | Sistemas de navegação para robôs de inspeção | 22 |
| 2.2.1 | Sistemas de navegação baseados em máquina de estados | 23 |
| 2.2.2 | Controle de navegação de robôs por campos vetoriais artificiais | 26 |
| 3 | Descrição e desenvolvimento do robô | 30 |
| 3.1 | Sistema mecânico | 31 |
| 3.1.1 | Módulos do sistema mecânico | 32 |
| 3.1.2 | Desenvolvimento do sistema mecânico | 35 |
| 3.2 | Sistema eletrônico | 37 |
| 3.2.1 | Componentes do sistema eletrônico | 38 |
| 3.2.2 | Desenvolvimento do sistema eletrônico | 42 |
| 3.3 | Sistema de <i>software</i> | 44 |
| 3.3.1 | <i>Robot Operating System</i> | 44 |
| 3.3.2 | Módulos do <i>software</i> | 45 |
| 3.3.3 | Desenvolvimento do <i>software</i> | 48 |
| 3.4 | Simulação do robô | 49 |
| 4 | Sistema de navegação para inspeção robótica de tubulações | 53 |
| 4.1 | Máquina de estados | 55 |
| 4.1.1 | Máquinas de Estados Determinísticas Finitas | 55 |
| 4.1.2 | Máquina de estados para inspeção autônoma de tubulações | 56 |
| 4.2 | Campos Vetoriais artificiais | 59 |
| 4.2.1 | Modelagem por <i>feedback linearization</i> | 60 |

| | | |
|----------|---|-----------|
| 4.2.2 | Campos vetoriais para navegação em corredores | 62 |
| 4.2.3 | Campos vetoriais para seguimento de paredes | 64 |
| 5 | Experimentos e resultados | 66 |
| 5.1 | Resultados de simulação | 66 |
| 5.2 | Resultados experimentais | 68 |
| 5.3 | Lições aprendidas | 72 |
| 6 | Conclusão | 74 |
| 6.1 | Trabalhos futuros | 75 |
| | Referências bibliográficas | 76 |
| A | Projeto mecânico do PipeBot | 81 |
| B | Projeto eletrônico do PipeBot | 88 |

1. Introdução

Os sistemas de dutos são recorrentes em muitas atividades industriais da mineração, por exemplo, para descarregar rejeitos de mineração ou no transporte de polpa de minério. Devido à natureza do fluido, é comum ocorrer entupimentos, rachaduras ou descontinuidades, que impedem ou reduzem as capacidades de transporte nos dutos. Sendo que, a acumulação de resíduos, impactos mecânicos, corrosão e vibração são as principais razões para defeitos na tubulação. Nesse sentido, esses sistemas precisam passar por processos de inspeções constantes para identificar o local correto das anomalias e evitar falhas ou acidentes. Um dos principais desafios da manutenção é a realização de inspeções internas confiáveis das tubulações (SUN *et al.*, 2009).

Segundo Ogai e Bhattacharya (2018), alguns tipos de sensores ou dispositivos podem ser utilizados para realizar a inspeção de tubulações, como ultrassom ou radiografia. Todavia, estes sensores são limitados, pois nem sempre os dutos se encontram facilmente acessíveis por humanos, podendo estar enterrados ou passando por terreno de difícil acesso. Nesse contexto o desenvolvimento de robôs para realizar essa tarefa se torna uma alternativa altamente viável e eficiente.

Os robôs são máquinas que possuem capacidade de executar automaticamente uma série de ações de acordo com uma tarefa específica (SICILIANO e KHATIB, 2016). Dentre eles, os robôs móveis possuem, como característica essencial, a presença de uma base móvel que permita seu movimento pelo ambiente. Do ponto de vista mecânico, um robô móvel consiste em um ou mais corpos rígidos equipados com um sistema de locomoção (SICILIANO *et al.*, 2010).

A utilização de robôs móveis permite o aumento do grau de automação e a versatilidade de sistemas automatizados, facilitando a integração e o controle. Eles podem contribuir para uma melhora significativa da produtividade e eliminar a presença humana em ambientes potencialmente agressivos e perigosos para a saúde, como em indústrias de mineração, química, nuclear e militar (ROCHA, 2000). A utilização de baterias e comunicações locais permite a operação dos robôs móveis até mesmo em ambientes com pouco recurso de energia ou redes de comunicação.

Esta dissertação dá continuidade e desenvolve o projeto descrito em (DE BARROS *et al.*, 2021b), apresentando a concepção de um dispositivo robótico de baixo custo. O dispositivo, chamado de PipeBot, é caracterizado por peças mecânicas de fácil fabricação, eletrônica eficiente e um *software* modular. O objetivo principal é viabilizar a inspeção visual de dutos com diâmetro superior a 14 polegadas.

A estrutura mecânica é principalmente composta por polímeros e pode ser fabricada através da tecnologia de impressão 3D, enquanto o corpo do robô é construído a partir de um tubo de PVC. Os componentes eletrônicos têm a capacidade de monitorar internamente as tubulações de forma eficaz. O sistema de *software* foi subdividido em módulos, simplifi-

cando tanto o processo de desenvolvimento quanto de depuração. Além disso, um robô virtual foi criado no simulador CoppeliaSim, com o propósito de avaliar tanto o código desenvolvido quanto o desempenho do robô no ambiente.

Após a construção e validação do funcionamento do robô, foram incorporadas funcionalidades de navegação autônoma. Uma máquina de estados é empregada para auxiliar o robô na tomada de decisões em ambientes de tubulação. Isso permite que o robô tome ações apropriadas quando se depara, por exemplo, com bifurcações ou caminhos sem saída. Além disso, um sensor laser é utilizado para medir a distância entre o robô e as paredes ao seu redor, gerando informações de distância que são empregadas em um controle de campos vetoriais para ajustar o movimento do robô durante a exploração do ambiente. É relevante destacar que essas informações possibilitaram a implementação de duas técnicas: uma que permite a navegação seguindo uma parede lateral e outra que permite a navegação mantendo o robô no centro entre duas paredes.

A máquina de estados implementada foi validada por meio de simulações, onde em um ambiente de tubulações o robô foi capaz de interagir com o ambiente tomando ações necessárias, como voltar de pontos sem saída e decidir que direção tomar em bifurcações. O controle por campos vetoriais foi validado tanto em ambiente simulado quanto em ambiente reais controlado. Graças a natureza multidisciplinar do projeto, foi gerada uma lista de lições aprendidas durante o desenvolvimento do mesmo que podem ser de grande valia para a comunidade.

No apêndice desta dissertação foram anexados esquemáticos dos sistemas mecânico e eletrônico com intuito de facilitar e tornar possível a reprodução do trabalho desenvolvido. Cabe ressaltar que o projeto em questão possui um pedido de patente no Instituto Nacional de Propriedade Industrial (INPI) (DE BARROS *et al.*, 2021a).

1.1. Motivação

A atividade de inspeção em ambientes confinados apresenta riscos para os seres humanos, incluindo a presença de produtos químicos tóxicos, perigo de incêndio e explosão, a possibilidade de materiais ou objetos soltos no ambiente e até mesmo a presença de animais silvestres.

Com o objetivo de reduzir a exposição humana a situações de risco durante a inspeção em ambientes confinados, o Instituto Tecnológico Vale (ITV) desenvolveu o EspeleoRobô, um robô projetado para inspecionar áreas confinadas de médio a grande porte. No entanto, o tamanho do corpo mecânico deste robô é consideravelmente maior do que o necessário para inspecionar tubulações de pequeno porte. Uma alternativa seria adotar soluções comerciais de tamanho menor em comparação com o EspeleoRobô, mas essas opções comerciais têm custos significativamente elevados.

Portanto, esta dissertação tem como objetivo principal o desenvolvimento de um dispositivo móvel robótico capaz de coletar uma variedade de dados, incluindo imagens de alta

qualidade, que podem ser analisados por operadores humanos para identificar antecipadamente defeitos em tubulações com diâmetro a partir de 14 polegadas.

1.2. Objetivos

Os objetivos desta dissertação buscam atender às necessidades presentes no Instituto Tecnológico Vale (ITV) e, simultaneamente, criar tecnologias que garantam a segurança humana durante atividades de alto risco. Para isso, é proposto o desenvolvimento de um dispositivo robótico capaz de realizar inspeções visuais em tubulações a partir de 14 polegadas de diâmetro de forma autônoma. Esse dispositivo é confeccionado principalmente por meio da tecnologia de manufatura aditiva, utilizando impressão 3D de plástico para construir a maior parte de sua estrutura. O custo de componentes e fabricação deve ser o mais baixo possível, mantendo o compromisso entre preço e confiabilidade. Essa questão se deve ao tipo de ambiente que o robô é exposto, no qual há muitas possibilidades de se perder o dispositivo. O robô conta com um sistema de locomoção ajustável, permitindo uma adaptação mais eficaz aos diferentes diâmetros de tubulação a serem inspecionados. Por fim, é implementado um sistema de navegação autônomo por meio da utilização de uma máquina de estados e a implementação de um controle por campos vetoriais para controlar a movimentação e a tomada de decisões no ambiente a ser inspecionado.

De acordo com os objetivos citados acima, os objetivos específicos deste projeto são:

- A construção do modelo 3D do robô;
- A impressão 3D de todos os componentes do robô;
- A montagem da estrutura mecânica;
- A inserção de todos os componentes eletrônicos na estrutura mecânica;
- A criação de pacotes ROS para integração dos componentes eletrônicos;
- A adaptação de códigos já implementados para o Espeleorobô;
- A integração de pacotes ROS do robô com o simulador CoppeliaSim;
- A simulação do funcionamento teleoperado do robô;
- A realização de testes de teleoperação em ambientes reais;
- A implementação da máquina de estados para tomada de decisões;
- A implementação do controle por campos vetoriais;
- A simulação do comportamento da máquina de estados em tubulações;

- A realização de testes do controle de navegação com campos vetoriais em ambientes controlados.

1.3. Contribuições

O desenvolvimento deste projeto contribui diretamente para o aumento das possibilidades de inspeção que podem ser realizadas pelo Instituto Tecnológico da Vale, de modo a reduzir os riscos durante as atividades de inspeção e antecipar as possibilidades de vazamento dos fluídos transportados. Como resultado da pesquisa e o desenvolvimento desta plataforma robótica foi publicado o seguinte artigo:

- A. Filho, N. Neto, M. Delunardo, J. Domingues, L. Barros, H. Azpúrua, G. Pessin, G. Freitas. **“Dispositivo robótico para inspeção de tubulações com diâmetro reduzido”** em Congresso Brasileiro de Automática 2022 (CBA).

O artigo apresentado no CBA 2022 ficou entre os mais bem avaliados do congresso e, como resultado, foi convidado para submissão de uma versão estendida e em inglês na revista Journal of Control, Automation and Electrical Systems – JCAE, e está em processo de revisão.

- A. Filho, N. Neto, M. Delunardo, J. Domingues, L. Barros, H. Azpúrua, G. Pessin, G. Freitas, I. Amaral, A. Resende, C. Fany, . **“Robotic Pipe Inspection: Low-Cost Device and Navigation System”** em Journal of Control, Automation and Electrical Systems (JCAE).

Durante o período do mestrado, no mesmo contexto do desenvolvimento de robôs para inspeção, houveram contribuições no desenvolvimento do EspeleoRobô. A documentação desse projeto está contida no relatório técnico.

- G. Freitas; G. Cruz; I. Amaral; D. Marques; D. Coutinho; G. Pereira; C. Duarte; J. Baião; G. Machado; J. Domingues; L. Barros; A. Cid; A. Filho; M. Delunardo; N. Neto; R. Júnior; M. Oliveira; H. Azpúrua; G. Pessin. **“Relatório técnico final do projeto ‘Dispositivo Robótico de Inspeção de Ambientes Restritos e Confinados.’** (2023).

Para alcançar os resultados desejados neste trabalho, foi crucial integrar todas as áreas de desenvolvimento do robô. Isso incluiu desde a produção das peças impressas até a montagem mecânica, do projeto eletrônico à fabricação das conexões, da implementação do controle à adaptação para operar no robô e da criação de máquinas de estado à utilização conforme as necessidades do robô. Assim, a principal contribuição deste trabalho concentrou-se na integração efetiva dessas áreas, com intuito de utilizar conhecimentos e tecnologias desenvolvidas para a criação de uma plataforma robótica para inspeção de tubulações. E por esse motivo a realização

da dissertação, conforme descrito neste documento, foi possível graças às contribuições essenciais de alguns pesquisadores: Nilton Neto na modelagem do sistema mecânico, Mario Delunardo na produção do circuito eletrônico, Israel Amaral na elaboração da máquina de estados do sistema de navegação, e Adriano Resende no desenvolvimento do controle por campos vetoriais.

1.4. Estrutura da dissertação

O Capítulo 1 introduz a robótica na inspeção de ambientes confinados, destacando sua relevância e fornecendo motivação para o estudo. Além disso, são apresentados os objetivos gerais e específicos da dissertação, seguido das contribuições deste trabalho.

O Capítulo 2 aborda sobre plataformas robóticas que tratam do problema em questão, e sistemas de navegação, com foco em máquina de estados e controle por campos vetoriais artificiais.

O Capítulo 3 descreve e detalha o desenvolvimento do sistema robótico proposto, abrangendo seu sistema mecânica, sistema eletrônico, *software* embarcado e simulação do robô.

O Capítulo 4 aborda a implementação do sistema de navegação, incluindo a discussão sobre a máquina de estados e o controle de campos vetoriais.

O Capítulo 5 apresenta simulações e resultados obtidos, abrangendo desde a validação, produção e desenvolvimento até os testes de navegação autônoma. Além disso, o Capítulo descreve as lições aprendidas durante a execução desse projeto.

O Capítulo 6 conclui o estudo do projeto desenvolvido na dissertação e aponta direções futuras, propondo melhorias observadas durante o desenvolvimento.

Ao final da dissertação foram anexados os projetos de engenharia do sistema mecânico e do sistema eletrônico do robô.

2. Trabalhos relacionados

Este capítulo tem como objetivo revisar os trabalhos relacionados à inspeção robótica em ambientes confinados e abordar temas relevantes para a solução do problema tratado nesta dissertação. Essa revisão da literatura é essencial para compreender o contexto da pesquisa e serve como base para o desenvolvimento da solução proposta.

2.1. Plataformas robóticas para inspeção de ambientes confinados

Para solucionar desafios relacionados à inspeção, diversas plataformas robóticas móveis foram desenvolvidas para inspeção de ambientes confinados, incluindo dutos e galerias. Nesta seção são apresentados estudos encontrados na literatura, os quais destacam os pontos cruciais incorporados em cada robô, fornecendo uma visão clara das abordagens empregadas.

O robô de inspeção desenvolvido pela Universidade de Ciência e Tecnologia AGH (CISZEWSKI *et al.*, 2017), mostrado na Figura 2.1, foi projetado para operar em tubulações com diâmetro a partir de 9.5 polegadas que contenham líquidos, como água, esgoto ou óleo. Para tal, suas conexões são seladas e os cabos utilizam conectores à prova d'água. A locomoção do robô se dá por meio de esteiras capazes de se posicionar em ângulos específicos com a utilização de atuadores mecânicos. Um ponto que vale ressaltar desse robô é que ele foi validado, inicialmente, usando simulações no *software* CoppeliaSim (ROHMER *et al.*, 2013).



Figura 2.1: Robô de Inspeção AGH University. Fonte: (CISZEWSKI *et al.*, 2017).

O robô KANTARO, apresentado na Figura 2.2, possui um *design* inovador voltado para inspeção de tubulações entre 7.8 e 11.8 polegadas (NASSIRAEI *et al.*, 2007). A mecânica desse robô facilita a superação de obstáculos, como juntas e curvas acentuadas. Sua operação visa ser totalmente autônoma, para realizar inspeções de baixo custo e alta eficiência. O dispositivo robótico possui comunicação sem fio para se locomover livremente por dentro das tubulações,

sendo equipado com um laser e uma câmera para a navegação autônoma e detecção de danos no interior das tubulações.

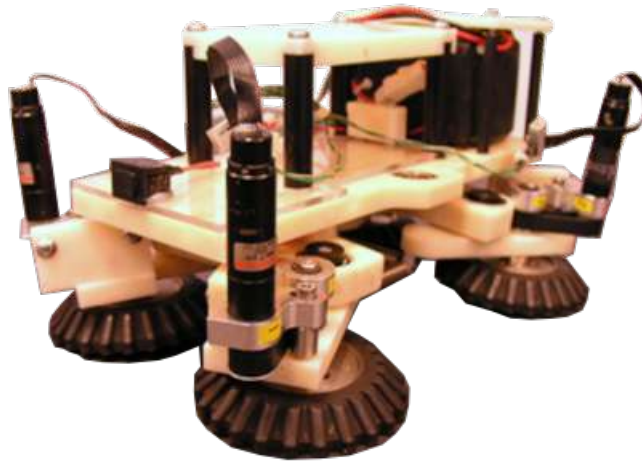


Figura 2.2: Robô Kantaro. Fonte: (NASSIRAEI **et al.**, 2007).

O Smart-Spider, mostrado na Figura 2.3, possui uma estrutura extensíveis que permite o acoplamento do robô em tubulações de 17.7 a 22.7 polegadas de diâmetro (QU **et al.**, 2018). A plataforma possui um corpo principal e um mecanismo flexível de rodas. O robô conta com duas câmeras, uma na parte frontal e outra na parte traseira, dentro de um domo de plástico transparente. O dispositivo é capaz de se locomover e executar algumas tarefas automaticamente, não dependendo de comunicação para realizar as inspeções. Todavia, quando necessário, o robô se comunica por meio de rede sem fio.



Figura 2.3: Robô Spider-Smart. Fonte: (QU **et al.**, 2018).

O Robô AP-02 (AEROPIC, 2022), mostrado na Figura 2.4, foi projetado para se movimentar em ambientes com dimensões a partir de 15.8 polegadas de diâmetro e em superfícies

adversas, por isso possui tração nas quatro rodas e braço mecânico traseiro que funciona como alavanca. O conjunto ótico do robô é composto por câmeras de alta definição e sensoramento remoto sem fio. A câmera frontal é posicionada sobre um *gimbal* iluminado, capaz de mudar sua orientação vertical (*tilt*) em até 75°. Um ponto importante, e que diferencia esse dispositivo de outros com a mesma finalidade, é que a comunicação entre a base e o robô é realizada mediante conexão sem fio. Este tipo de conexão geralmente diminui o alcance e qualidade do vídeo para operação.



Figura 2.4: Robô AP-02. Fonte: (AEROPIC, 2022).

O robô Tritrax (EDDYFI, 2022), mostrado na Figura 2.5) é uma solução robótica de inspeção para avaliação de condições de integridade de tubulações de 8 a 16 polegadas de diâmetro. O Tritrax é um dispositivo comercial que custa em torno de US\$ 100 mil, e possui um sistema de locomoção flexível e proteção à submersão até 60 metros. O sistema de locomoção é composto por três esteiras motorizadas e um chassi com três pernas expansíveis, o que permite realizar inspeções em uma grande variedade de diâmetros de dutos em qualquer orientação, inclusive subidas e descidas verticais. A parte frontal do robô contém iluminação LED e uma câmera que pode ser direcionada de acordo com a necessidade. Além disso, o robô possui comunicação via cabo para teleoperação.



Figura 2.5: Robô Tritrax. Fonte: (EDDYFI, 2022).

O JettyRobot 6, representado na Figura 2.6, é uma solução robótica versátil projetada especificamente para operações em tubulações com diâmetro na faixa de 13.8 a 28 polegadas. Sua ampla gama de funcionalidades o torna uma escolha destacada para várias tarefas, incluindo inspeção, limpeza e manutenção. O JettyRobot 6 é um robô comercial que custa por volta de US\$ 150 mil, e está equipado com ferramentas especializadas, como jateamento de ar ou água, escovas e capacidades de sucção, permitindo soluções altamente eficientes e eficazes. Seu *design* modular permite intervenções adaptáveis e precisas em sistemas de tubulações em diversas indústrias. Esse robô disponível comercialmente, fabricado pela JettyRobot, é controlado por teleoperação usando comunicação por cabo. O mesmo cabo é responsável por transmitir dados de comunicação e energia (JETTY ROBOT, 2023).



Figura 2.6: JettyRobot 6. Fonte: (JETTY ROBOT, 2023).

A definição das premissas para o desenvolvimento do robô PipeBot, abordado nesta dissertação, resultou de uma análise das características dos robôs apresentados em trabalhos relacionados e das plataformas criadas no Instituto Tecnológico da Vale. O foco na inspeção de tubulações com diâmetros acima de 14 polegadas foi o principal critério que influenciou o tamanho e o método de locomoção do PipeBot. Cada robô mencionado nesta seção contribuiu para avaliar as vantagens e desvantagens de suas características. Aspectos como o método de locomoção, sensores de inspeção, atuadores mecânicos e formatos dos robôs influenciaram na escolha da plataforma final desenvolvida.

Uma comparação entre os diferentes robôs de inspeção é apresentada na Tabela 2.1. Esta tabela destaca as principais características e capacidades de cada robô, incluindo seus respectivos ambientes, sistemas de locomoção, métodos de comunicação e funcionalidades.

Tabela 2.1: Comparação de robôs de inspeção de dutos e galerias.

| <i>Robô</i> | <i>Ambiente</i> | <i>Locomoção</i> | <i>Operação</i> | <i>Característica-chave</i> | <i>Diâmetro mínimo</i> |
|----------------------------|-----------------------|-------------------------|-----------------------------------|------------------------------------|------------------------|
| AGH (2017) | Tubulações | Esteiras | Teleoperação por cabo | À prova d'água | A partir de 9.5" |
| KANTARO (2007) | Tubulações | <i>Design</i> inovador | Autônomo | Laser | De 7.8" a 11.8" |
| Smart-Spider (2018) | Tubulações | Mecanismo flexível | Autônomo | Duas câmeras | De 17.7" a 22.7" |
| AP-02 (2022) | Terrenos desafiadores | Tração nas quatro rodas | Teleoperação sem fio | Braço como alavanca | A partir de 15.8" |
| Tritrax (2022) | Tubulações | Pernas expansíveis | Teleoperação por cabo | Inspeções verticais | De 8" a 16" |
| JettyRobot (2023) | Tubulações | Pernas expansíveis | Teleoperação por cabo | Jateamento e escovas | De 13.8" a 28" |
| PipeBot (Proposta) | Tubulações | Esteiras | Autônomo ou teleoperação por cabo | Mecânica e <i>software</i> modular | A partir de 14" |

Fonte: O autor.

2.2. Sistemas de navegação para robôs de inspeção

A navegação autônoma de robôs móveis envolve a aplicação de técnicas que garantem um deslocamento seguro do robô de um ponto a outro em seu ambiente. Essas técnicas abrangem a localização precisa do robô, a criação de mapas do ambiente, o planejamento de trajetórias e o controle do sistema robótico. Portanto, um dos principais desafios da robótica móvel autônoma é encontrar uma rota que leve o robô de sua posição atual ao seu destino desejado, fazendo uso da informação de localização e garantindo um controle eficaz da movimentação do robô para alcançar seu objetivo final (RODRIGUES, 2010).

Diversas soluções têm sido propostas para sistemas de navegação de robôs de inspeção, incluindo o emprego de máquinas de estados e controle de navegação por campos vetoriais artificiais. A abordagem por máquinas de estados é escolhida devido ao seu potencial para conferir ao robô um comportamento modular, hierárquico e previsível, fundamentado em informações do ambiente. A seleção da abordagem de controle por campos vetoriais artificiais é motivada por sua capacidade de possibilitar a navegação suave e eficaz em ambientes complexos, bem como pela habilidade de reagir de forma mais natural às mudanças e perturbações que ocorrem em volta do robô. Outro fator determinante para escolha dessas abordagens foi o conhecimento prévio adquirido nas implementações dessas técnicas em outros robôs do Instituto Tecnológico

Vale. Nas próximas subseções, serão descritos exemplos e aplicações dessas abordagens.

2.2.1. Sistemas de navegação baseados em máquina de estados

Uma máquina de estados é um modelo computacional utilizado para representar o comportamento sequencial de sistemas complexos, onde o sistema assume diferentes estados em resposta a eventos ou condições específicas. Esses estados são definidos como conjuntos distintos de ações, transições e condições, e a máquina de estados controla a transição entre eles. Cada estado representa um comportamento ou modo de funcionamento do sistema, e as transições ocorrem com base em eventos, entradas do usuário ou condições pré-definidas. Esse modelo oferece uma abordagem estruturada e organizada para a implementação de lógica de controle em sistemas, tornando-os mais compreensíveis e gerenciáveis (DESAI *et al.*, 2017).

No estudo realizado por Pereira *et al.* (2021), é demonstrada a aplicação de uma máquina de estados, apresentada na Figura 2.7, uma abordagem para solucionar o problema de navegação de um robô em dutos. Este estudo foi realizado com o EspeleoRobô e tem como objetivo a inspeção de sistemas de barragens de rejeitos, os quais frequentemente apresentam pontos de entrada e saída estrategicamente posicionados, simplificando o acesso a áreas que tendem a sofrer um desgaste mais acentuado. Nesse contexto, a proposta orienta o movimento do robô em direção ao centro do tubo e faz uso de um algoritmo de tomada de decisões projetado para enfrentar esses desafios, demonstrando resultados satisfatórios mesmo diante de bifurcações e becos sem saída. A lógica operacional para a navegação é definida por meio de uma máquina de estados, a qual é ativada de acordo com as informações adquiridas pelo sensor LiDAR.

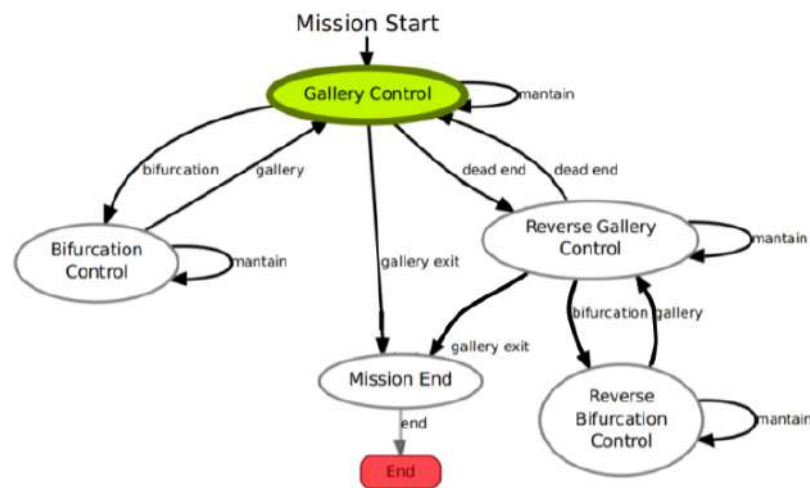


Figura 2.7: Máquina de estados proposta para navegação autônoma do robô para inspeção de dutos. Fonte: (PEREIRA *et al.*, 2021).

O artigo conduzido por Sales *et al.* (2014), demonstra a aplicação de uma máquina de estados no contexto de sistemas de navegação. A abordagem integra a localização e a navegação usando um mapa topológico baseado em uma técnica chamada Máquina de Estados Finita

Adaptativa. Nesse método, o ambiente é mapeado como um grafo, e cada caminho possível é representado por uma sequência de estados controlados por uma Máquina de Estados Finita (FSM), que pode ser visualizada na Figura 2.8.

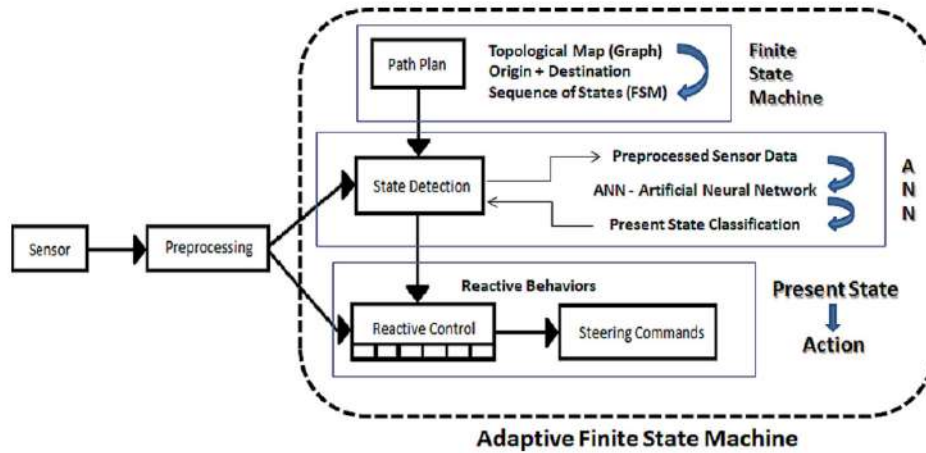


Figura 2.8: Máquina de estados para navegação de robôs por um ambiente por meio da identificação visual de padrões. Fonte: (SALES et al., 2014).

No artigo, uma Rede Neural Artificial é treinada para reconhecer padrões, como os da Figura 2.9, nos dados de entrada, que representam o contexto atual da FSM. Quando um novo padrão é reconhecido, a FSM muda de estado e executa ação associada. Isso permite que um robô móvel siga uma sequência de estados/comportamentos para atingir um destino, se mantendo adaptado a diferentes situações. O método foi testado com sucesso em ambientes internos e externos, se mostrando promissor para o controle e navegação de robôs autônomos.

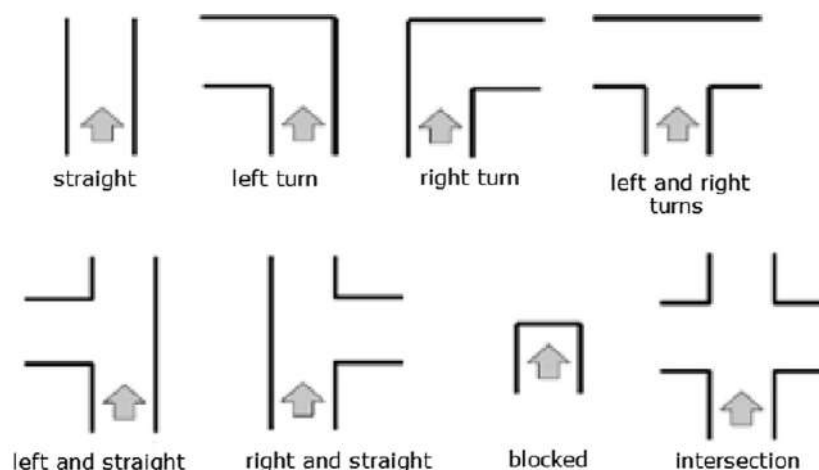


Figura 2.9: Estados da máquina de estados equivalentes a padrões reconhecidos no campo de cultivo. Fonte: (SALES et al., 2014).

O desenvolvimento mencionado no estudo de Biber et al. (2012) resultou na criação de um robô chamado BoniRob. Este robô é projetado para se mover autonomamente em campos

de cultivo organizados em filas e pode ser configurado para trabalhar com diversos tipos de culturas, como milho ou trigo, em diferentes espaçamentos de filas e variadas quantidades de filas. Uma aplicação interessante para um robô como esse é atuar como um “inspetor de culturas”, realizando tarefas repetitivas de análise de plantas para auxiliar os agricultores.

Para que o robô possa realizar essas tarefas, foi criado um conjunto de características, por meio de uma máquina de estados, que descrevem todas as condições de trabalho permitidas, a Figura 2.10 mostra a estrutura desta máquina de estados. Se o robô não conseguir identificar uma situação específica usando essas características, ele entra em um estado de erro, o que significa que algo não está conforme o esperado no ambiente. Nesse caso, o robô deve parar imediatamente e aguardar até que a situação seja resolvida, possivelmente com a intervenção de um ser humano.

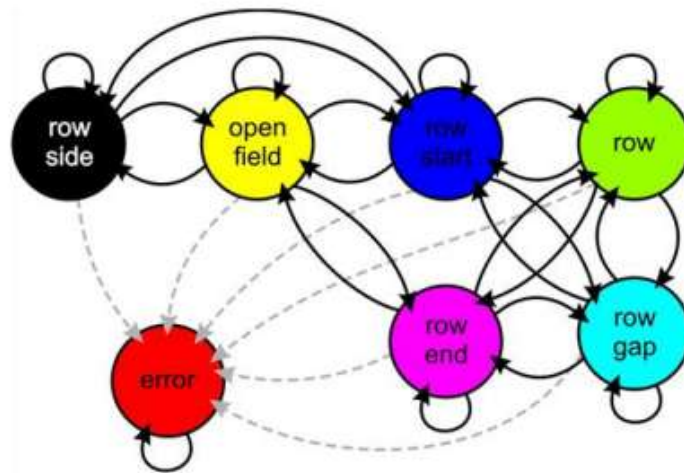


Figura 2.10: Máquina de estados desenvolvida para o robô BoniRob. Fonte: (BIBER **et al.**, 2012).

No campo, o robô espera encontrar diferentes tipos de características, como áreas abertas (*open field*), laterais da fila (*row side*), filas (*row*), espaços entre as filas (*row gap*), início das filas (*row start*) e fim das filas (*row end*), conforme ilustrado na Figura 2.11. Portanto, o robô é capaz de realizar seu trabalho de forma contínua, tomando decisões com base nas informações que coleta por meio de sensores e seguindo um conjunto de ações definidas pela máquina de estados.





























| | open field | row | row start | row end | row gap | row side |
|----------|---|---|---|---|--|---|
| one row |  |  |  |  |  |  |
| | |  |  |  |  |  |
| | |  |  |  |  |  |
| | | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| two rows | |  |  |  |  | |
| | |  |  |  |  | |
| | |  |  |  |  | |
| | | ⋮ | ⋮ | ⋮ | ⋮ | |

Figura 2.11: Características reais equivalentes às ações da máquina de estados. Fonte: (BIBER et al., 2012).

2.2.2. Controle de navegação de robôs por campos vetoriais artificiais

O controle por campos vetoriais artificiais é uma abordagem que pode ser utilizada na robótica móvel, a fim de simplificar a navegação autônoma de robôs em ambientes complexos. Esse método se baseia na representação do movimento do robô por meio de vetores que indicam direções preferenciais de deslocamento. Utilizando informações, como dados de sensores de distância, campos vetoriais artificiais são gerados para guiar o robô em direção ao seu destino, podendo também ser adaptados para evitar obstáculos presentes no ambiente. Essa técnica é amplamente aplicada em robótica móvel, incluindo veículos e drones, permitindo que os robôs tomem decisões de movimento baseadas nas mudanças do ambiente, tornando a navegação mais intuitiva e eficiente (KAZEM et al., 2010).

No artigo apresentado por Rezende et al. (2021), foi demonstrado um exemplo de aplicação das técnicas de campos vetoriais artificiais. A metodologia proposta visa calcular um campo vetorial artificial, como o ilustrado na Figura 2.12, que varia no tempo e é definido em um espaço de n dimensões, com o objetivo de gerar caminhos convergentes para uma curva predefinida desejada. Para a construção desse campo, é utilizada a função de distância euclidiana, a qual é facilmente calculada a partir de uma representação paramétrica da curva em questão. O cálculo do termo *feedforward* é utilizado para compensar a dependência temporal do campo, assegurando que sua magnitude seja limitada pela velocidade máxima da curva. Isso possibilita a normalização do campo vetorial variável no tempo, mantendo uma norma constante, sem afetar negativamente a convergência. O estudo também oferece evidências da convergência do

campo vetorial variável no tempo normalizado proposto e demonstra a existência de limites finais, mesmo quando perturbações limitadas estão presentes. Por fim, o artigo inclui várias simulações e experimentos com um quadricóptero real para validar a metodologia proposta, demonstrando a aplicabilidade prática e eficácia do campo vetorial no contexto de controle de trajetórias.

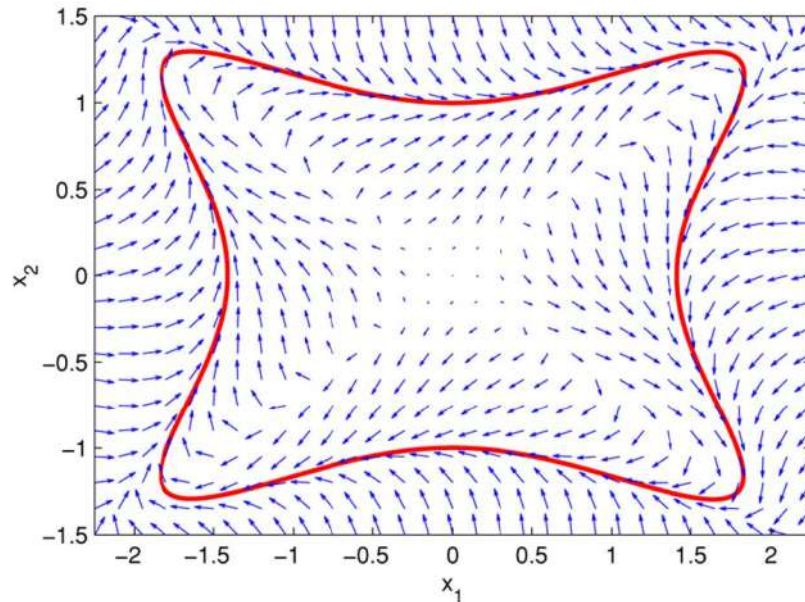


Figura 2.12: Exemplo de curva e campo vetorial proposto para navegação de robôs. Fonte: (REZENDE **et al.**, 2021).

No estudo realizado por dos Santos **et al.** (2019), foi apresentado um sistema de robôs que jogam futebol, semelhante ao jogo humano. O objetivo do jogo é marcar o máximo de gols possível contra o time adversário. Para solucionar esse desafio, o artigo sugere o uso de um método de controle baseado em campos vetoriais artificiais, sendo esse método uma combinação de campos: uns que atraem os robôs na direção desejada e outros que os repelem de obstáculos. Os resultados do estudo mostraram que essa técnica melhorou a eficiência dos robôs durante o jogo, pois permitiu que eles se movessem em direção à bola de acordo com a estratégia de jogo, ao mesmo tempo em que evitavam colisões com obstáculos, economizando tempo, a Figura 2.13 mostra o simulador dos robôs de futebol durante uma partida.

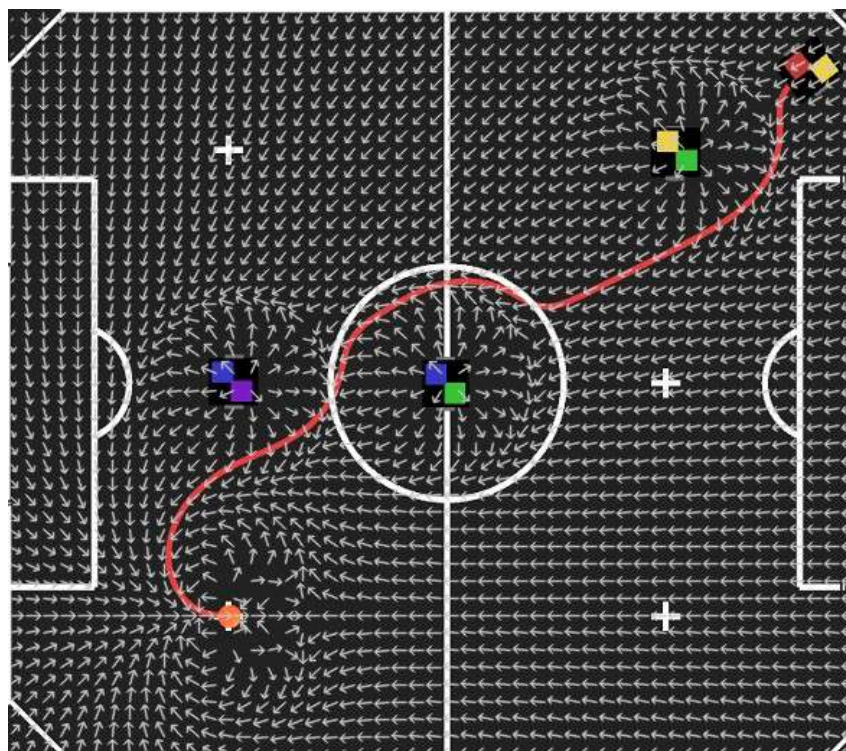
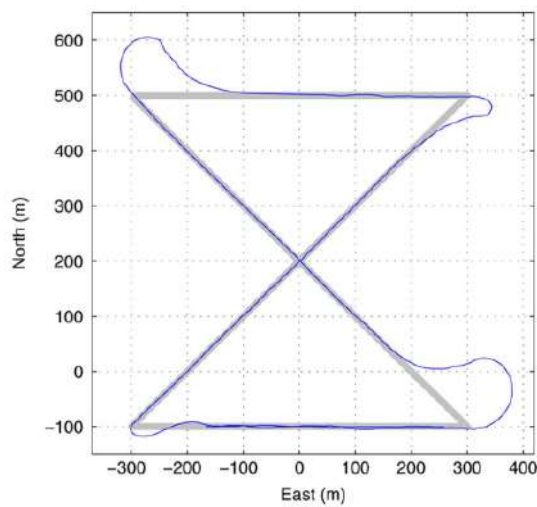
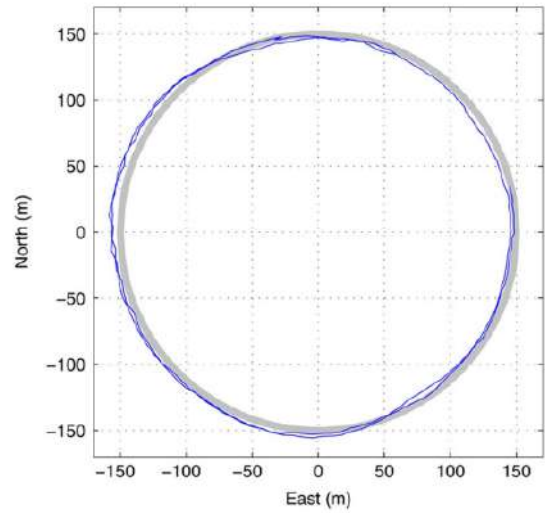


Figura 2.13: Visualização do campo vetorial resultante que possui como intuito o robô atingir o alvo. Fonte: (DOS SANTOS **et al.**, 2019).

O artigo de Nelson **et al.** (2007) desenvolve um método para que veículos aéreos miniaturizados (MAV) sigam com precisão um caminho específico. Esse método se baseia na ideia de campos vetoriais artificiais, que são usados para criar instruções de voo desejadas para o controle do MAV. Foram criadas leis de controle para seguir caminhos em linha reta e para trajetórias circulares. Usando critérios de estabilidade de Lyapunov, foi demonstrada que a abordagem de campos vetoriais artificiais permite um acompanhamento assintótico para trajetórias em linha reta e circulares na presença de perturbações de vento constantes. Testes de voo experimentais mostraram que os erros médios de seguir caminhos foram menores que o comprimento de uma asa para trajetórias em linha reta, e menores que três comprimentos de asa para trajetórias com mudanças frequentes de direção. Alguns dos resultados obtidos podem ser visualizados nas Figuras 2.14a e 2.14b.



(a) Seguimento em linha reta em condições de vento forte.



(b) Seguimento em órbita em condições de vento forte.

Figura 2.14: Caminhos percorridos pelo MAV com perturbações no ambiente. Fonte: (NELSON et al., 2007).

Outro exemplo da aplicação de campos vetoriais na navegação de robôs móveis foi apresentado por Jie et al. (2006). Neste trabalho, é proposto um método de navegação que combina a abordagem do ciclo limite com a técnica de campo vetorial, visando orientar um robô móvel de forma a evitar obstáculos inesperados em um ambiente dinâmico. O método do ciclo limite é aplicado com o propósito de evitar obstáculos posicionados diretamente à frente do robô, enquanto o método do campo vetorial é empregado para lidar com obstáculos localizados nas proximidades laterais do robô. A eficácia dessa abordagem foi avaliada por meio de testes realizados com um robô móvel Pioneer 2-DX. Tanto simulações quanto experimentos práticos foram conduzidos, demonstrando a capacidade dos métodos propostos para a navegação de um robô móvel em ambientes dinâmicos e repletos de desafios.

3. Descrição e desenvolvimento do robô

O dispositivo robótico apresentado nesta dissertação, denominado PipeBot e ilustrado na Figura 3.1, foi desenvolvido com o propósito de atender às demandas de inspeções de dutos com um diâmetro mínimo de 14 polegadas. A transformação do projeto digital em um protótipo funcional proporcionou diversos desafios, como a correção de modelos 3D, ajustes de componentes eletrônicos e a implementação de novas funções no *software*.

O método de locomoção escolhido foi o uso de esteiras, devido a comparações com outras plataformas e simulações do robô. Esse tipo de locomoção demonstrou ser capaz de superar obstáculos com facilidade, além de proporcionar uma tração mais eficiente devido a maior área de contato com o solo. A estrutura das esteiras é acoplada ao corpo do robô para evitar o aumento das dimensões gerais, além disso, é necessário apenas um motor por esteira para mover o sistema.

Inicialmente, o modo de operação escolhido foi o teleoperado via cabo. Essa escolha se deve à robustez dessa forma de comunicação, que não é suscetível a interferências do ambiente e tem capacidade para transmitir grande volume de dados. No entanto, à medida que o desenvolvimento avançou, foi implementada uma capacidade de navegação autônoma para permitir que o robô se deslocasse de forma livre no ambiente. Essa mudança foi motivada pelos problemas associados ao uso do cabo, como a possibilidade do robô ficar preso em partes do ambiente, por causa de um agarramento do cabo, e também por conta do aumento do peso que o robô deve arrastar à medida que se desloca por distâncias maiores (MARTZ *et al.*, 2020).

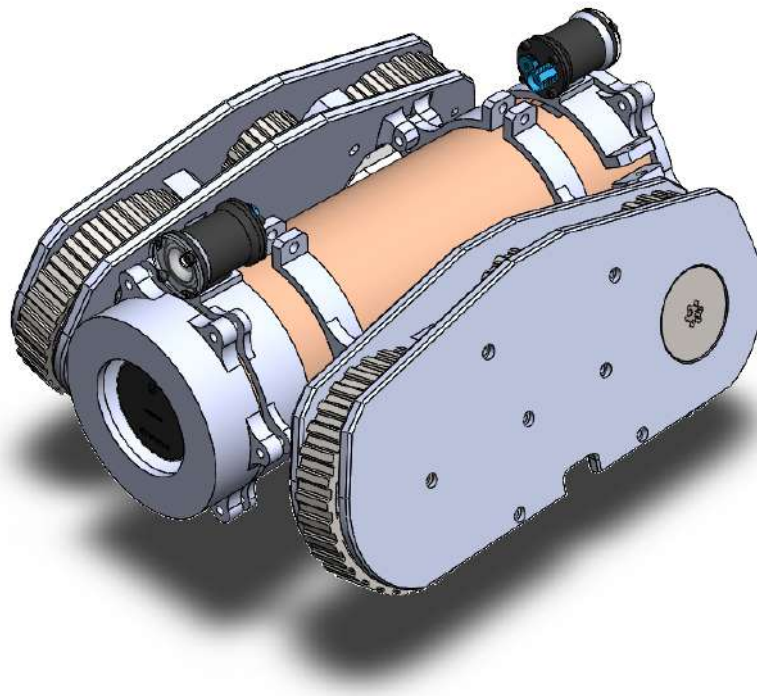


Figura 3.1: Modelo tridimensional do PipeBot, robô de inspeção de dutos. Fonte: O autor.

A característica chave desse robô é sua modularidade, tanto em termos mecânicos,

eletrônicos e de *software*. Essa abordagem foi adotada pois facilita a manutenção, correções e melhorias no dispositivo, gerando um sistema altamente flexível e adaptável. As especificações gerais do robô podem ser visualizadas na Tabela 3.1.

Tabela 3.1: Especificações gerais do PipeBot.

| Altura | Comprimento | Largura | Peso |
|--------|-------------|---------|--------|
| 23 cm | 36 cm | 27 cm | 4,5 kg |

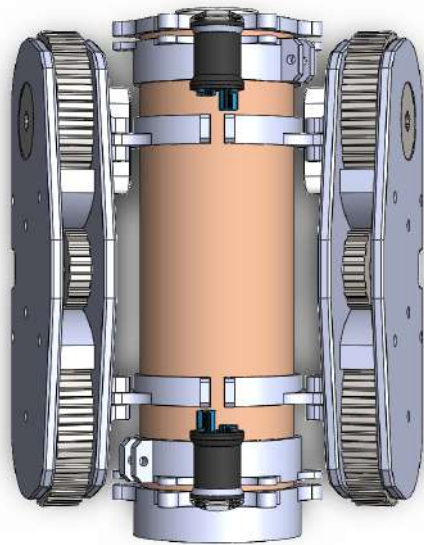
Fonte: O autor.

3.1. Sistema mecânico

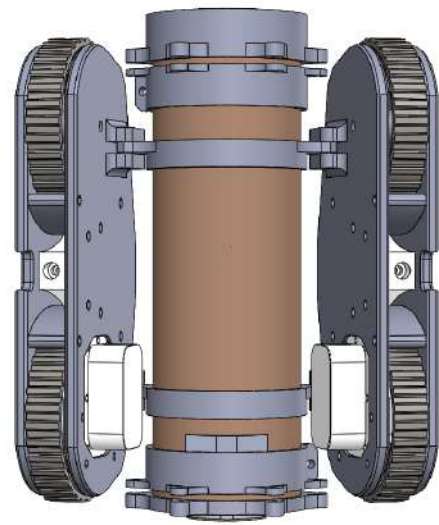
O projeto mecânico do robô seguiu um conjunto de premissas, conforme detalhado no relatório técnico de Teotônio (2021). As principais premissas que orientaram o desenvolvimento do dispositivo nessa dissertação foram:

1. **Fabricação por manufatura aditiva:** A maioria das partes do dispositivo foram projetadas para serem fabricadas por manufatura aditiva, ou seja, utilizando a tecnologia de impressão 3D. As exceções foram peças estruturais, como polias e eixos, componentes comerciais e partes que necessitam atender tolerâncias dimensionais específicas para montagem.
2. **Construção compacta:** O robô foi projetado com uma construção mecânica mínima e compacta, possibilitando inspeções em dutos com um diâmetro mínimo de 14 polegadas. Essa característica é fundamental para garantir a acessibilidade em espaços confinados.
3. **Eficiência de movimento:** O dispositivo foi desenvolvido para se movimentar de forma eficiente dentro dos dutos por meio de esteiras, permitindo a inspeção contínua e ágil. Além disso, foi projetado para superar obstáculos encontrados durante a operação.

O projeto mecânico foi desenvolvido integralmente utilizando o *software* de modelagem *SolidWorks*, como também ilustrado na Figura 3.2. Esse ambiente de modelagem proporcionou as ferramentas necessárias para a criação detalhada e precisa das peças que compõem o PipeBot.



(a) Vista superior modelo 3D do robô.



(b) Vista inferior modelo 3D do robô.

Figura 3.2: Modelo tridimensional desenvolvido no *SolidWorks* do robô de inspeção de tubulações. Fonte: O autor.

3.1.1. Módulos do sistema mecânico

As partes mecânicas do robô foram agrupadas e organizadas em diferentes módulos, conforme apresentado na Figura 3.3. O propósito desta divisão foi de simplificar tanto a montagem quanto a produção do PipeBot. A seguir é descrito a função e a utilização de um cada desses módulos.

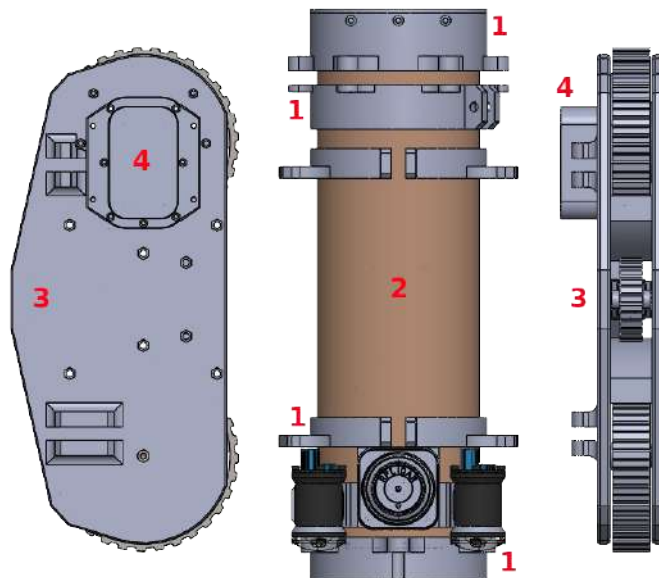


Figura 3.3: Módulos que constituem a estrutura do robô: suporte (1), base (2), locomoção (3), motor (4). Fonte: O autor.

O **módulo base** do robô é composto por um tubo de PVC com um diâmetro de 110 mm e uma espessura de 6.1 mm. Esse corpo desempenha o papel de resguardar a eletrônica embarcada no interior do robô, mantendo-a protegida contra elementos externos. Nas extremidades do tubo são fixadas as tampas traseira e dianteira, enquanto ao longo do corpo são posicionados os suportes necessários para garantir a estabilidade e a funcionalidade do conjunto. Esse *design* do corpo proporciona uma estrutura sólida e confiável para todo o sistema. A Figura 3.4 mostra as características e formato do corpo do robô.

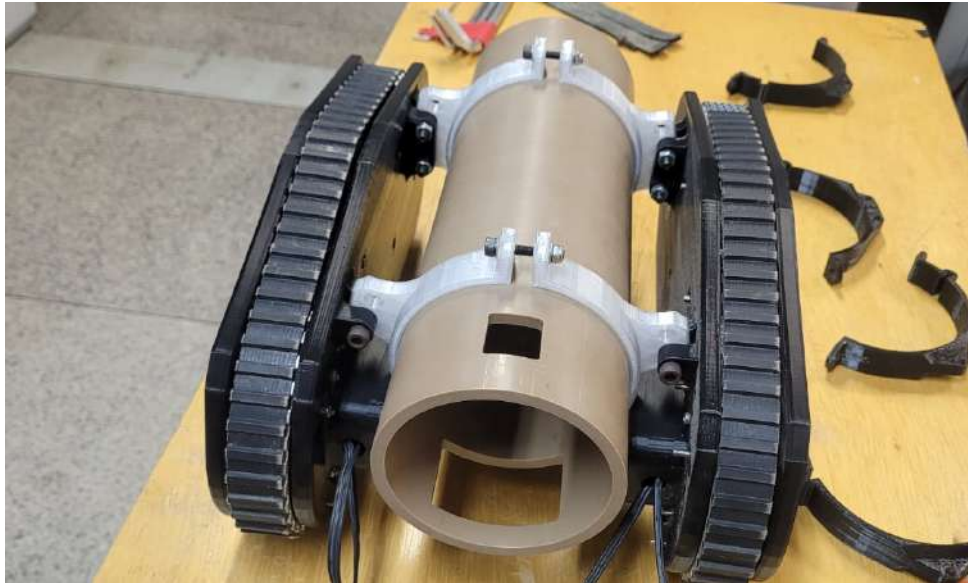


Figura 3.4: Corpo do robô em montagem inicial com módulos de teste. Fonte: O autor.

O **módulo de motor** foi desenvolvido de forma que o motor trabalhe encapsulado. A construção do conjunto é feita utilizando uma placa usinada e uma placa impressa em 3D. A placa impressa pode ser projetada sob medida para acomodar o motor de forma precisa. Isso elimina a necessidade de peças padronizadas que poderiam não se adaptar ao projeto de maneira ideal.

Os eixos de transmissão de potência são usinados para atender as tolerâncias dimensionais necessárias para o encaixe entre as peças. Posteriormente este módulo é acoplado no módulo de locomoção para realizar o controle de movimentação do robô. A Figura 3.5 mostra os componentes que fazem parte deste módulo.



Figura 3.5: Montagem do módulo do motor. Fonte: O autor.

O módulo de locomoção é construído a partir de duas placas impressas em 3D que são unidas por parafusos e porcas. Dentro desse módulo, o motor é fixado na placa interna, e é responsável por gerar a tração nas correias por meio de uma polia que está conectada ao eixo do motor. Esse eixo é suportado por um rolamento na placa impressa. As polias frontal e traseira são peças comerciais selecionadas a partir de catálogos nacionais.

A correia escolhida possui um perfil com dentes tanto na parte interna quanto na externa, o que permite que as polias girem enquanto a parte externa da correia faz atrito com a tubulação para movimentar o robô. Para manter a tensão adequada nas correias e garantir o funcionamento correto, foi implementado um mecanismo tensionador simples. Quando um parafuso é rosqueado, ele pressiona uma polia menor, que possui o mesmo passo e um diâmetro menor do que as polias de tração, contra a parte interna da correia na parte superior do módulo de movimentação. O mecanismo tensionador consiste em uma base impressa que suporta a força de esticamento da correia. O parafuso que conecta as duas partes age como um eixo e passa pelo interior de dois rolamentos que sustentam a polia. Essa base impressa é fixada na parte interna da placa lateral do módulo de locomoção. Todos os componentes podem ser visualizados na Figura 3.6.



Figura 3.6: Montagem interna do módulo de locomoção. Fonte: O autor.

O módulo de suporte consiste em um conjunto de peças que são usadas para prender os módulos de esteira, a iluminação e sensores ao corpo do robô. Esses suportes são essenciais para permitir que cada módulo seja anexado de forma independente. No total, 6 peças com formato de “aro” compõe o conjunto de suportes, e cada uma delas tem uma função específica, conforme ilustrado na modelagem da Figura 3.3.

Para fixar o módulo das esteiras no corpo do robô, é utilizado um par de suportes com formato de abraçadeira, que possuem abas nas extremidades, com furos para que possam ser unidos ao corpo do robô por meio de parafusos. Esses suportes abraçam o corpo feito de PVC do robô e o mantêm fixo. Além disso, eles possuem um ajuste de angulação que permite inclinar as esteiras conforme necessário.

As tampas frontal e traseira do módulo de suporte são fixadas por mais dois suportes, que são anexados ao corpo do robô de maneira semelhante aos mencionados anteriormente, ou seja, por meio de uma espécie de abraçadeira que é apertada contra o corpo usando parafusos. Após essa fixação, as tampas são presas a esses suportes também por meio de parafusos.

O suporte que segura a tampa frontal também desempenha um papel estrutural ao fixar os LEDs de iluminação e o sensor laser. Esses dispositivos estão localizados na parte superior do robô e são presos parafusados no suporte, garantindo que não se soltem durante as atividades de inspeção.

Ao fim desta dissertação é possível visualizar o projeto mecânico do robô por meio de desenhos e visões explodidas dos componentes no apêndice A.

3.1.2. Desenvolvimento do sistema mecânico

A produção do sistema mecânico teve início com a criação de modelos 3D de todos os componentes. Essa abordagem virtual permitiu a detecção antecipada de possíveis erros e a identificação de oportunidades de aprimoramento, antes mesmo da fabricação das peças.

Além disso, essa modelagem serviu como base para simulações que validaram o funcionamento do projeto. Durante essa etapa, várias melhorias foram implementadas, incluindo o ajuste de encaixes entre as peças e decisões relacionadas ao posicionamento dos sensores.

Posteriormente, utilizando a tecnologia de manufatura aditiva, foi iniciado o processo de impressão 3D dos componentes modelados. A facilidade oferecida por essa tecnologia possibilitou a rápida produção de diversos protótipos das peças de plástico. Esses protótipos foram necessários para aperfeiçoar as dimensões dos componentes, ajustar furos de parafusos e otimizar as folgas entre peças com movimento, além disso, permitiram a realização de testes de resistência mecânica, resultando em um projeto mais robusto e peças mais duráveis. A Figura 3.7 apresenta alguns exemplos dos componentes da estrutura mecânica produzidos por uma impressora 3D.

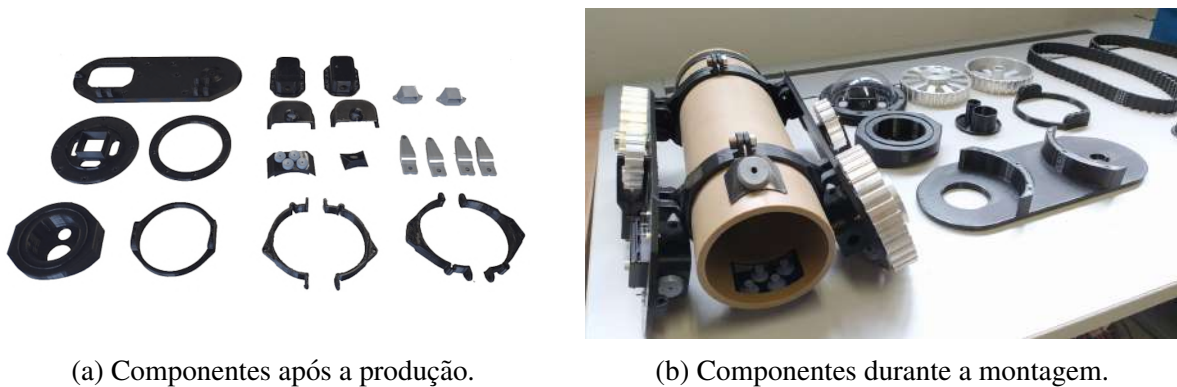


Figura 3.7: Componentes impressos em 3D do sistema mecânico do robô. Fonte: O autor.

Após a produção das peças impressas foi iniciado a montagem. O projeto modular do dispositivo simplificou este processo. Isso se deve ao fato de que um conjunto de peças pode ser montado de maneira independente, e depois acoplado a outros módulos.

O processo de montagem foi dividido em três etapas distintas. Inicialmente, foi realizada a montagem dos dois módulos da esteira, cada um com suas respectivas peças. Em seguida, o corpo do robô foi equipado com os suportes impressos em 3D, desempenhando o papel de conexão entre os grupos de peças e os periféricos. Por último, o módulo de locomoção foi acoplado aos conjuntos previamente montados, concluindo a montagem do sistema mecânico.

A Figura 3.8 oferece uma visão do processo de montagem, destacando uma das laterais do dispositivo com suas peças impressas e usinadas. Nessa representação, é possível observar não apenas as peças individuais, mas também a interligação de todos os módulos, que já foram conectados entre si, formando um sistema mecânico funcional. É possível visualizar o progresso da montagem, ilustrando como as partes se encaixam e se complementam no conjunto final.

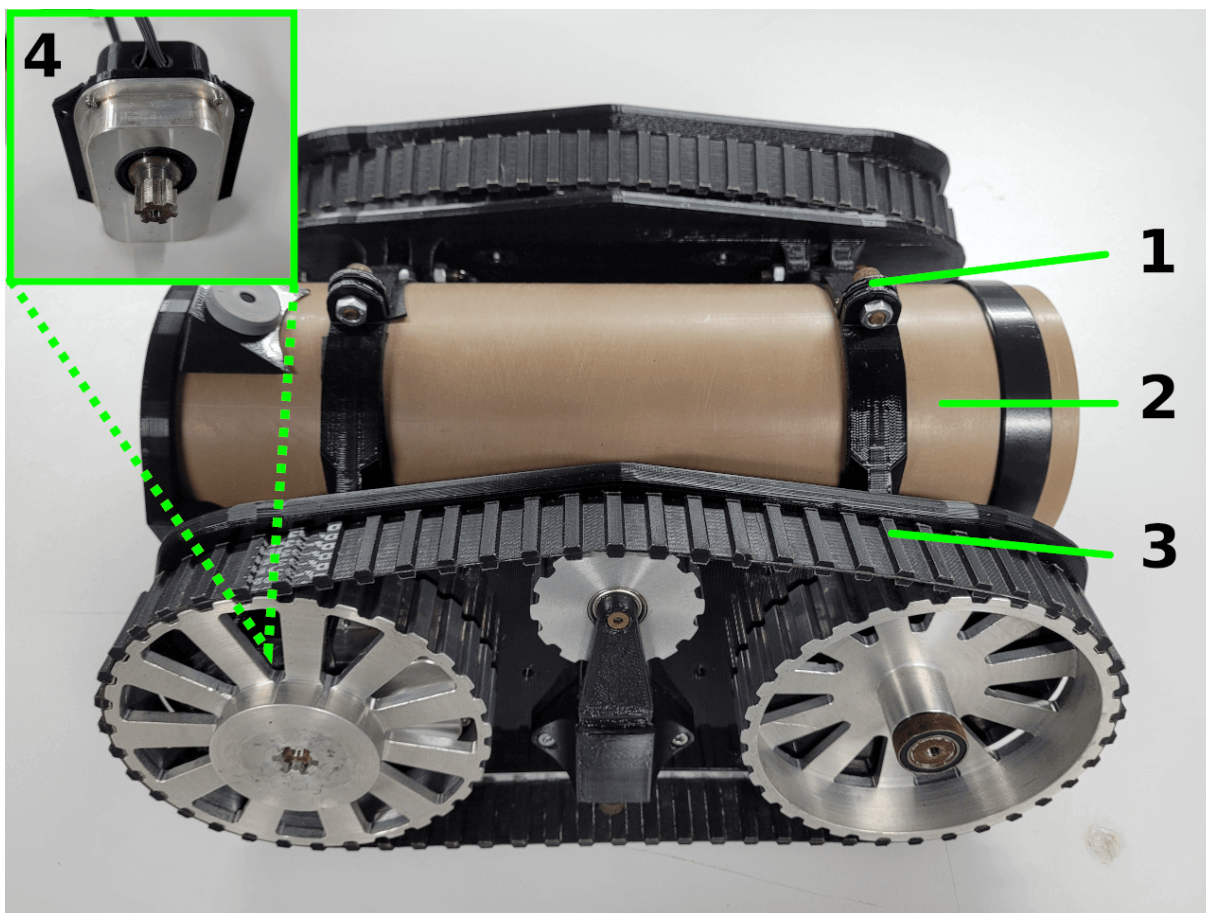


Figura 3.8: Montagem das peças impressas e usinadas. Fonte: O autor.

3.2. Sistema eletrônico

A eletrônica do robô foi projetada de forma a minimizar a quantidade de componentes necessários, visando a criação de um equipamento de baixo custo e montagem simplificada. O dispositivo é alimentado por uma bateria Li-Po, e dois conversores DC-DC são utilizados para adequar a alimentação às especificações de cada componente. O componente principal deste sistema é a placa *Raspberry PI*, que desempenha todo o processamento, incluindo o envio de sinais aos periféricos, o armazenamento de dados e, quando necessário, a comunicação com o operador.

O robô é capaz de sensorar o ambiente a ser inspecionado por meio de um laser, uma câmera e LEDs de iluminação que melhoram a visibilidade dentro dos dutos. Além disso, o dispositivo utiliza dois motores elétricos de alto torque, permitindo que ele movimente com facilidade o seu próprio peso, o cabo de comunicação e supere eventuais obstáculos encontrados durante as inspeções nos dutos.

É relevante destacar que, ao longo do processo de desenvolvimento desse robô, foram conduzidos testes utilizando dois tipos distintos de câmeras: uma equipada com um laser integrado e outra que faz uso da tecnologia RGBD. Como resultado dessas variações nas

configurações, é possível observar imagens do robô capturadas com diferentes conjuntos de câmeras, às vezes com ou sem a presença da montagem do laser planar.

Essa diversidade de configurações e a possibilidade de alternar entre elas durante os testes demonstram a abordagem modular adotada no projeto do robô. Essa flexibilidade permite a exploração de diferentes opções e a adaptação do sistema às necessidades específicas de cada situação de inspeção.

3.2.1. Componentes do sistema eletrônico

O diagrama na Figura 3.9 representa as conexões elétricas e a comunicação entre os periféricos do sistema, que serão descritos a seguir. É importante destacar que a escolha dos equipamentos foi pautada em critérios como custo reduzido, baixo consumo de energia e dimensões compactas.

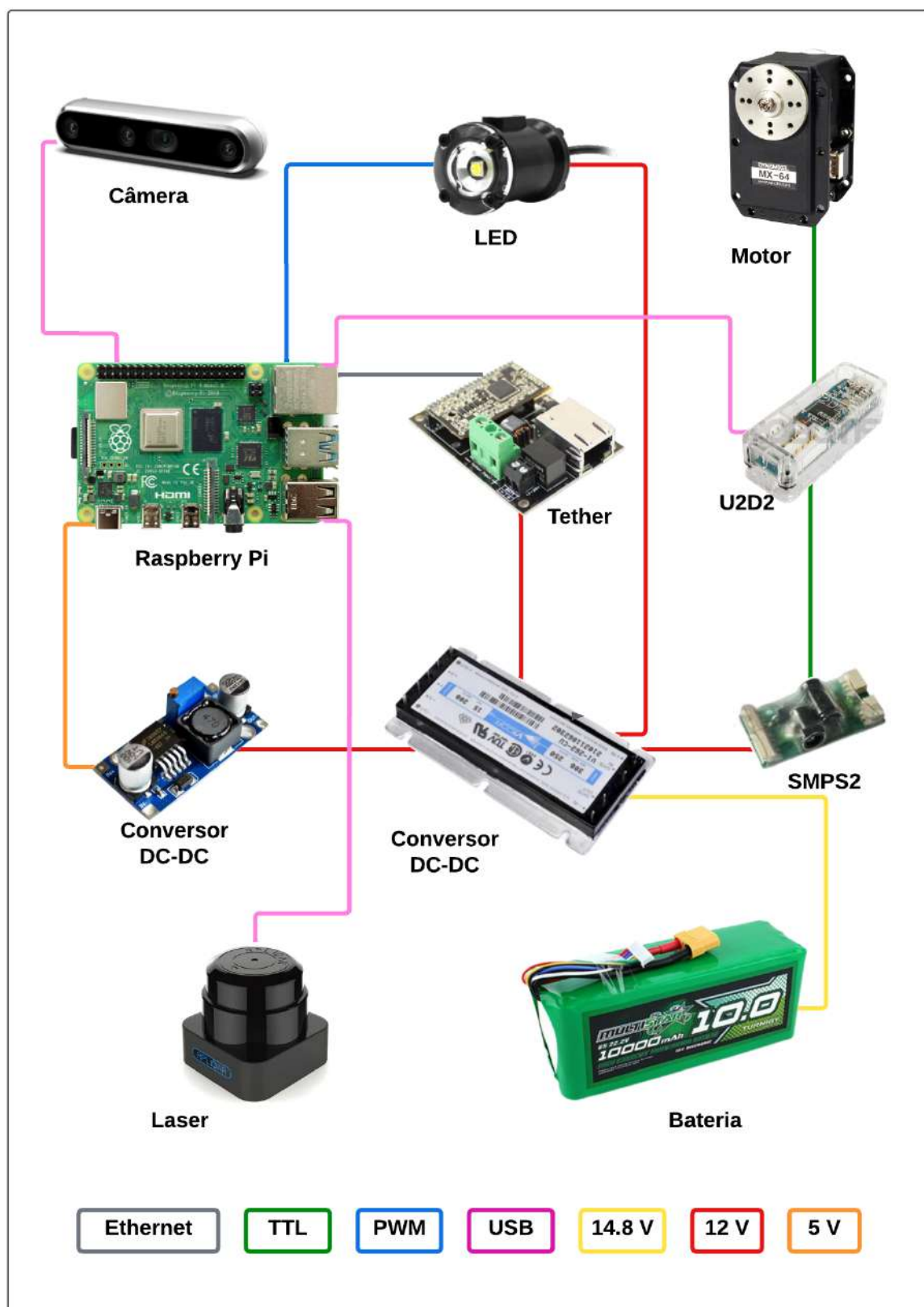


Figura 3.9: Diagrama das conexões eletrônicas do PipeBot. Fonte: O autor.

Raspberry PI: A placa controladora utilizada é uma Raspberry PI 4 Model B com 4 GB

de memória RAM. Este controlador possui uma boa capacidade de processamento e tamanho físico reduzido, e consegue rodar o sistema operacional Linux, possibilitando o controle remoto de toda a locomoção, iluminação e aquisição de dados do robô via ROS. Valor: US\$ 130; *Datasheet*: <https://datasheets.raspberrypi.com/rpi4/raspberry-pi-4-datasheet.pdf>

Controlador U2D2: Conversor de comunicação que permite controlar e operar os motores Dynamixel por meio de uma porta USB. Valor: US\$ 32; *Datasheet*: <https://emanual.robotis.com/docs/en/parts/interface/u2d2/>

Motor Dynamixel: O motor escolhido para o robô é um Dynamixel MX-64AT. Ele é um servo motor com controladores próprios garantindo acionamento robusto, economia de espaço e alta densidade de potência. O conceito descentralizado desses motores inteligentes minimiza o uso de controladores externos, diminuindo o espaço que esses circuitos ocupam. Os motores possuem um sistema de comunicação e alimentação baseado no protocolo TTL, em que são necessários apenas 3 fios para alimentação e conexão entre os componentes da rede. Além disso, os motores selecionados podem ser controlados via ROS, permitindo a alteração de parâmetros como velocidade, aceleração, torque, limite de temperatura de trabalho, entre outros itens. A comunicação entre motores e a Raspberry PI é feita por meio do controlador U2D2, o que facilita sua utilização. Suas principais especificações técnicas são: tensão de trabalho de 10 a 14.8 V, corrente máxima de 4.1 A, torque máximo de 6 Nm, velocidade máxima 63 RPM e temperatura de trabalho máxima de 80 °C. Valor US\$ 369; *Datasheet*: <https://emanual.robotis.com/docs/en/dxl/mx/mx-64/>

Conversor SMPS2: Este dispositivo é conectado a bateria e é responsável por fornecer energia aos motores Dynamixel, ele funciona como filtro e regulador de tensão. Valor: US\$ 7; *Datasheet*: <https://www.robotis.us/smps2dynamixel/>

Bateria: A escolha da bateria leva em consideração principalmente seu volume e peso, uma vez que o robô não pode ser grande e busca aumentar a autonomia energética de operação. Neste aspecto, as baterias do tipo Li-Po foram escolhidas para o projeto devido a alta densidade energética, além de não apresentarem diminuição da capacidade de carga após um número elevado de recargas. Outro aspecto importante é que baterias Li-Po, em comparação com as de Li-Ion, são menos propensas à ocorrência de vazamentos do seu material interno diminuindo a chance de incidentes. Para a implementação do protótipo, foi utilizada uma bateria Turnigy 12000mAh 4S1P 12C 14.8V. Valor: US\$ 150.

Câmera Intel Realsense D435i: A câmera possui tecnologia RGBD, o que significa que ela é capaz de fornecer tanto informações de profundidade (D) quanto de cor (RGB). Para isso, a câmera usa visão estereoscópica para calcular a profundidade. Ela consiste em um capturador esquerdo, um capturador direito e um projetor infravermelho. O projetor infravermelho melhora a precisão da profundidade em cenários com baixa textura. Os capturadores esquerdo e direito capturam a cena e enviam os dados para um processador de imagem de profundidade, que calcula valores de profundidade correlacionando pontos nas imagens e através do deslocamento

entre elas. Os valores de profundidade são processados para criar um quadro, e quadros subsequentes formam um vídeo de profundidade. As principais especificações desse dispositivo são: faixa de trabalho de 0.3 a 3 m, resolução máxima da imagem de profundidade de 1280×720 pixels, resolução máxima da imagem RGB de 1920×1080 pixels, campo de visão de imagem de profundidade $87^\circ \times 58^\circ$ e campo de visão de imagem RGB $69^\circ \times 42^\circ$. *Datasheet*: Valor: US\$ 334; <https://dev.intelrealsense.com/docs/intel-realsense-d400-series-product-family-datasheet>

Câmera Intel Realsense LiDAR L515: Essa câmera é um dispositivo capaz de fornecer além de imagens de alta definição, informações de distâncias do ambiente inspecionado. A câmera utiliza um laser infravermelho, um componente MEMS (Micro-Eletromecânico), um fotodiodo infravermelho, uma câmera colorida, um controlador MEMS e um chip de processamento de imagem. O MEMS é usado para mover o feixe de laser infravermelho por toda a área de visão da câmera. O chip de processamento de imagem L515 processa os dados refletidos pelo feixe de laser capturados pelo fotodiodo e gera um ponto de profundidade que representa a distância precisa de um ponto específico na cena em relação à câmera. A combinação desses pontos de profundidade forma um conjunto de dados de profundidade em formato de nuvem de pontos que representa toda a cena. As principais especificações desse dispositivo são: faixa de trabalho de 0.25 a 9 m, resolução máxima da imagem de profundidade de 1024×728 pixels, resolução máxima da imagem RGB de 1920×1080 pixels, campo de visão de imagem de profundidade $70^\circ \times 55^\circ$ e campo de visão de imagem RGB $70^\circ \times 40^\circ$. Valor: US\$ 589; *Datasheet*: <https://dev.intelrealsense.com/docs/lidar-camera-l515-datasheet>

As câmeras da Intel possuem um sensor IMU (*Inertial measurement unit*) integrado, que fornece informações sobre a orientação e o movimento do dispositivo, e também não necessitam de uma alimentação externa, pois utiliza apenas uma porta USB 3.0 para receber energia e realizar a comunicação. A combinação desses recursos, juntamente com as imagens de alta definição obtidas pelas câmeras, resulta em um considerável aumento nas funcionalidades do robô. Essas características abrem perspectivas promissoras para futuras aplicações, como o mapeamento tridimensional do ambiente.

Tether: O Fathom-x Tether é um cabo umbilical capaz de reproduzir uma comunicação *ethernet* com apenas um par de fios (padrão IEEE-1901), garantindo alta velocidade de transmissão com robustez. Esse dispositivo permite transmitir *streams* de vídeo e coleta de dados em cabos com mais de 300 metros de comprimento. A comunicação via cabo torna a transmissão de dados estável, pois não sofre qualquer tipo de interferência nos ambientes de inspeção. Ela garante transmissão de vídeos sem perda de qualidade e comunicação em tempo integral com o robô, tornando a teleoperação mais responsiva. O ponto negativo dessa tecnologia é exigir torques mais elevados dos motores, pois o robô deve vencer um atrito crescente do cabo de comunicação conforme inspeciona o ambiente. Além disso, essa solução aumenta a possibilidade de enroscar o cabo de comunicação em alguma parte da tubulação e gerar problemas de mobilidade. Para amenizar o problema, é possível utilizar soluções semelhantes às apresenta-

das em (SHAPOVALOV e PEREIRA, 2020), que propõe um planejador de caminhos que busca reduzir as chances de enroscamento do cabo durante a operação. Valor: US\$ 240; *Datasheet*: <https://bluerobotics.com/store/comm-control-power/tether-interface/fathom-x-tether-interface-board-set-copy/>

Iluminação: A iluminação do robô é feita utilizando LEDs de alta potência da marca BlueRobotics e modelo Lumen Subsea, com alto fluxo luminoso e índice de reprodução de cor constante, que podem ser facilmente controlados via PWM. O LED possui intensidade luminosa máxima de 1500 lumens e pode ser alimentado em um intervalo de 7 a 48 volts. Valor: US\$ 300; *Datasheet*: <https://bluerobotics.com/store/thrusters/lights/lumen-r2-rp/>

Conversor DC-DC: Para manter a tensão constante mesmo com o descarregamento da bateria, um conversor 28V/12V (V28A12C200BL) foi adicionado para a alimentação do sistema embarcado, desde os motores até os LEDs para iluminação, câmeras, switch e Tether. Valor: US\$ 411; *Datasheet*: <https://www.vicorpower.com/products?productType=cfg&productKey=V28A12C200BL>

Conversor DC-DC: Este dispositivo reduz a tensão disponibilizada pelo conversor DC-DC de 12 V para 5 V, que corresponde à tensão de alimentação da Raspberry PI, placa controladora do robô. Valor: US\$ 10;

Laser: O RPLIDAR S1, desenvolvido pela SLAMTEC, é um scanner laser 2D de custo acessível. Ele possui velocidade de rotação capaz de capturar até 9200 amostras por segundo. Com um alcance de até 40 metros, este dispositivo realiza varreduras 2D completas de 360 graus, gerando dados de nuvem de pontos que são essenciais para fins de mapeamento e localização. A frequência de varredura pode ser ajustada no intervalo de 8-15Hz, proporcionando uma resolução angular precisa de 0,391°. Além disso, sua versatilidade permite a utilização tanto em ambientes internos quanto externos (SLAMTEC, 2023). Valor: US\$ 560; *Datasheet*: <https://www.slamtec.com/en/Lidar/S1>

3.2.2. Desenvolvimento do sistema eletrônico

A montagem da eletrônica começou com a verificação das tensões necessárias para cada componente. Isso foi importante porque a bateria fornece uma tensão maior do que alguns periféricos podem suportar. Para resolver isso, foi utilizado um conversor de tensão DC-DC, que não apenas mantém a tensão de entrada correta, mas também garante uma tensão constante durante o descarregamento, desde que haja carga mínima na bateria. Com base nessas especificações de potência, fabricamos os cabos de energia que conectam o conversor aos periféricos. Os outros cabos necessários para conectar os componentes já haviam sido adquiridos de forma comercial.

Antes da montagem dos componentes eletrônicos no robô, foram realizados testes de bancada de forma externa ao robô, conforme ilustrado na Figura 3.10. Nesse momento todos os periféricos foram acionados via comandos, incluindo por exemplo a validação dos LEDs se deu

por meio de testes de intensidade luminosa em um ambiente escuro, no qual foram alterados os níveis de PWM e avaliada a nitidez da imagem gerada pela câmera.



Figura 3.10: Montagem externa ao robô dos componentes eletrônicos embarcados. Fonte: O autor.

A operação remota do robô é feita por meio do uso do Tether, e sua montagem é muito simples, basta conectar um cabo de rede entre o primeiro Tether e a Raspberry PI, um par de fios que liga o primeiro Tether ao segundo Tether, e, por fim, um cabo de rede que conecta o segundo Tether ao computador base.

A Raspberry PI é alimentada por um conversor de tensão específico, uma vez que o conversor DC-DC utilizado fornece uma tensão maior do que a suportada. Os periféricos são conectados à Raspberry PI para a integração com o *software* que foi desenvolvido para o robô.

Durante o processo de montagem, foi verificado que a eletrônica estava solta demais dentro do corpo do robô. Isso poderia resultar em danos, como cabos soltos ou curtos-circuitos, devido aos movimentos do dispositivo. Portanto, foi criada uma estrutura interna para acomodar e fixar todos os componentes eletrônicos de forma mais segura. A Figura 3.11 mostra tanto o modelo 3D, quanto a montagem da estrutura projetada e impressa em 3D com todos os componentes fixados. Essa estrutura tem o mesmo diâmetro interno do corpo do robô e permite uma fixação contra as paredes internas. Uma vantagem desse aprimoramento no projeto é a possibilidade de montar toda a eletrônica externamente ao robô, verificar o funcionamento dos componentes agrupados e, em seguida, simplesmente encaixar essa unidade no corpo cilíndrico

do robô.



Figura 3.11: Montagem da eletrônica na estrutura de integração proposta. Fonte: O autor.

O projeto eletrônico pode ser encontrado no Apêndice B, oferecendo uma visão de todos os componentes e das conexões necessárias para o correto funcionamento do robô, por meio de diagramas técnicos ilustrativos.

3.3. Sistema de *software*

O sistema desenvolvido com o uso do ROS (Robot Operating System) foi projetado com uma abordagem modular, resultando na criação de diversos pacotes ROS. Cada um desses pacotes representa programas com funções específicas e bem definidas. A ênfase na modularidade do sistema oferece diversas vantagens, incluindo a facilidade de desenvolvimento colaborativo e a simplificação da depuração para identificar e corrigir falhas.

3.3.1. *Robot Operating System*

O ROS (Robot Operating System) é um sistema operacional que concentra todas as funcionalidades necessárias para a criação de sistemas robóticos em uma única plataforma. Ele se destaca por várias características (SCHOLLMEIER, 2001), (QUIGLEY et al., 2009):

1. **Arquitetura *Peer-to-Peer*:** O ROS utiliza uma arquitetura de redes *peer-to-peer*, em que cada dispositivo na rede pode atuar como cliente ou servidor. Isso possibilita o compartilhamento de serviços e dados sem depender de um servidor central, tornando a comunicação eficiente e distribuída.
2. **Multi-Linguagem:** O ROS oferece suporte a diversas linguagens de programação, permitindo aos desenvolvedores escolher a que melhor se adequa às suas necessidades. Essa versatilidade facilita a colaboração e a integração de diferentes componentes.

3. **Abordagem *Tools-Based*:** O ROS adota um *design* de *microkernel*, onde muitas pequenas ferramentas são utilizadas para criar e executar vários componentes do sistema. Isso torna o sistema altamente modular, ajudando a gerenciar a complexidade do desenvolvimento de sistemas robóticos.
4. **Código Aberto (*Open-Source*):** O ROS é um projeto de código aberto, o que significa que seu código-fonte está disponível para todos. Isso acelera o desenvolvimento, pois muitos níveis de *software* podem ser projetados e depurados em paralelo. O ROS é distribuído sob os termos da licença BSD, o que permite seu uso em projetos comerciais e não comerciais.

O ideal *open-source* do ROS incentivou a criação de uma ampla variedade de pacotes desenvolvidos pela comunidade de usuários. O ecossistema do ROS abriga milhares de pacotes, abrangendo desde implementações de novos algoritmos até recursos de qualidade industrial. Isso cria uma infraestrutura comum que oferece acesso a *drivers* de *hardware*, recursos genéricos de robôs, ferramentas de desenvolvimento e bibliotecas externas úteis (ROS, 2019). A versatilidade do ROS e o suporte da comunidade o tornam uma escolha sólida para o desenvolvimento de sistemas robóticos, especialmente quando se trata de projetos que demandam um equilíbrio entre custo e eficiência.

3.3.2. Módulos do *software*

Para um projeto como o desta dissertação, que busca criar um robô de baixo custo, mas que ainda seja eficiente em suas operações, o ROS oferece várias vantagens. Ele permite aproveitar a vasta base de recursos e pacotes disponíveis, economizando tempo e recursos no desenvolvimento. Além disso, a versatilidade do ROS em termos de linguagem de programação facilita a integração de diferentes componentes e a adaptação às necessidades específicas do projeto.

Os pacotes ou nós do ROS são pequenos programas independentes que podem ser executados em vários sistemas ao mesmo tempo. Eles se comunicam uns com os outros através do ROS Master, que tem a função de permitir que esses nós se encontrem. Uma vez que eles se localizam, podem se comunicar diretamente, ponto a ponto. Esses nós também podem ser executados em máquinas diferentes, que são reconhecidas como clientes. Quando esses clientes se conectam ao ROS Master, a rede é compartilhada e todos têm acesso aos programas do sistema.

No caso do PipeBot, foi escolhido a Raspberry PI como o ROS Master, e o computador de operação remota se conecta a ela como cliente. A grande vantagem de centralizar as funcionalidades desse modo é que diferentes bases podem se conectar ao robô, permitindo que diversos operadores tenham acesso às informações de diferentes partes do sistema ao mesmo tempo.

É importante destacar que durante a operação de navegação autônoma, não é necessária uma conexão ativa entre a Raspberry PI e o computador de teleoperação. Isso ocorre porque

o objetivo é permitir que o robô alcance seu objetivo de forma independente. Portanto, todas as informações que estão sendo visualizadas e processadas pelo robô são armazenadas durante este período.

A Figura 3.12 ilustra a estrutura do *software*, onde esses diversos pacotes ROS interagem para cumprir as tarefas necessárias no funcionamento do sistema como um todo. Essa abordagem organizada contribui para eficiência do sistema, permitindo uma manutenção mais fácil e aprimoramentos contínuos.

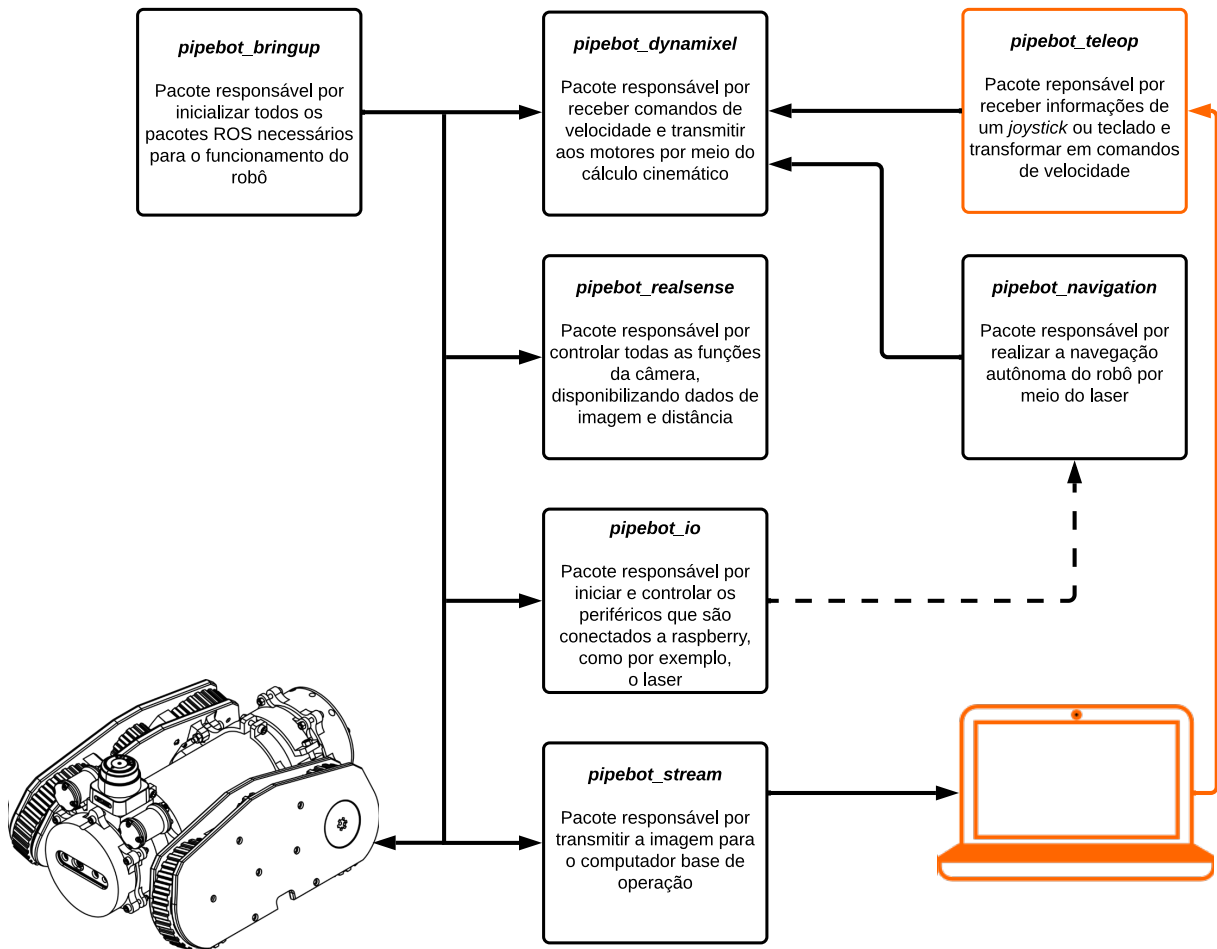


Figura 3.12: Diagrama dos pacotes ROS que constituem o *software* do robô. O diagrama com a cor preta indica os pacotes que são executados no robô e com a cor laranja os pacotes que são executados no computador base. Fonte: O autor.

O **Módulo de Inicialização do Robô** (*pipebot_bringup*) tem como principal função preparar o robô para o uso, garantindo a inicialização de todos os componentes essenciais. Ele é acionado de forma automática assim que o robô é ligado, assegurando que todo o procedimento necessário para iniciar uma inspeção seja realizado após a energia ser fornecida ao sistema. É importante destacar que esse pacote foi desenvolvido de forma específica para o PipeBot.

O **Módulo de Integração dos Motores com ROS** (*pipebot_dynamixel*) tem a tarefa de estabelecer uma comunicação eficiente entre o computador embarcado e os motores do robô. Esse módulo utiliza o pacote ROS chamado *dynamixel_workbench*, desenvolvido pela Dynami-

xel, a empresa fabricante dos motores, como base para essa comunicação. O pacote *pipebot_dynamixel* foi criado com o propósito de ajustar parâmetros físicos específicos do robô, utilizando o pacote original como ponto de partida. Uma das principais modificações realizadas no pacote original foi o ajuste da cinemática diferencial para controlar o robô, visto que esse pacote recebe comandos de velocidade.

O **Módulo de Controle da Câmera** (*pipebot_realsense*) tem a responsabilidade de gerenciar a câmera Realsense D435i ou a Realsense L515, fazendo uso dos pacotes ROS fornecidos pela Intel, a fabricante da câmera. Foi desenvolvido um pacote chamado *pipebot_realsense* no qual foram criados arquivos de configurações personalizadas para atender às necessidades específicas do projeto, utilizando como base o pacote original da Intel. Exemplos dessas modificações nos parâmetros foram a resolução da imagem, taxas de quadros por segundo, exposição de luz automática e configurações nos parâmetros de informações de profundidade.

O **Módulo de Transmissão de Imagem** (*pipebot_stream*) possibilita a gravação e a transmissão de fluxos de vídeo por meio do protocolo *Ethernet*. Isso possibilita a transmissão das imagens capturadas pela câmera, permitindo análise ou monitoramento remoto. Este pacote realiza configurações relacionadas aos fluxos de vídeo, incluindo qualidade da imagem, largura de banda, taxa de quadros por segundo e resolução. Além disso, o módulo utiliza o *software* FFMpeg como base para os controles do vídeo. No modo de navegação autônoma ele também é utilizado para realizar a gravação das imagens na memória interna do robô.

O **Módulo de Controle de Periféricos** (*pipebot_io*) lida com os dispositivos periféricos incorporados no robô, transformando informações de sensores e atuadores em mensagens e comandos executáveis. Esse módulo tem a capacidade de controlar a intensidade luminosa dos LEDs usando a técnica de modulação por largura de pulso (PWM) e de ler o nível de bateria do robô. No *script* de inicialização deste pacote, também são inicializados todos os sensores que estão integrados ao robô, como é o caso do laser planar, que possui seu próprio pacote ROS e requer apenas a inicialização para disponibilizar informações de distância.

No **Módulo de Teleoperação**, foi criado o pacote *pipebot_teleop* baseado em um pacote do EspeleoRobô chamado *espeleo_teleop* (AZPÚRUA et al., 2021). Esse módulo é responsável por converter as entradas do usuário em mensagens de comando de velocidade. As entradas podem ser realizadas de duas maneiras: por meio de um *joystick* ou diretamente pelo teclado do notebook. Para implementar essa funcionalidade, foi utilizado o pacote nativo do ROS chamado *teleop_twist_joy*. Foram configurados dois parâmetros essenciais desse pacote para controlar o EspeleoRobô. O primeiro é a “zona morta” do *joystick*, que representa um limite inferior do valor do analógico que é desconsiderado devido ao ruído. O segundo parâmetro se refere à taxa de atualização de mensagens do *joystick* quando ele não está recebendo nenhuma ação do operador. Essas configurações permitem um controle preciso e responsivo do robô durante a teleoperação.

O **Módulo de Navegação Autônoma** (*pipebot_navigation*) utiliza como base dois pacotes. Um é o *vectorfield_stack* (https://github.com/adrianomcr/vectorfield_stack)

que é uma implementação de controle baseado em estratégia de campos vetoriais artificiais. E o outro é *duct_explorer* (https://github.com/israelfi/duct_explorer) que implementa uma máquina de estados que confere ao robô a capacidade de realizar navegação autônoma dentro de tubulações. Detalhes adicionais e explicações sobre a navegação autônoma serão apresentados no Capítulo 4.

3.3.3. Desenvolvimento do *software*

O processo de configuração foi iniciado com a instalação padrão do sistema operacional Ubuntu versão 20.04 na Raspberry PI. A escolha desse sistema se baseou principalmente em sua compatibilidade com o ROS e na sua baixa demanda de recursos computacionais, tornando-o ideal para o projeto. Após a instalação a comunicação entre a Raspberry PI e o computador de teleoperação foi configurada e testada, garantindo uma conexão estável.

Foram realizados os testes de cada pacote ROS do robô. Primeiramente, o pacote de teleoperação desenvolvido especificamente para o EspeleoRobô foi integrado ao novo sistema e submetidos a testes iniciais. Uma das principais mudanças realizada nesse pacote foi sua atualização para a linguagem de programação Python 3. Esse processo foi necessário pois a linguagem citada é uma dependência do ROS Noetic, o qual foi utilizado no sistema do PipeBot.

Em seguida, componentes como os motores Dynamixel e as câmeras Realsense foram integrados utilizando os pacotes ROS fornecidos pelos próprios fabricantes, com algumas alterações, como mencionado anteriormente.

No entanto, ao prosseguir com a inicialização simultânea de todos os pacotes necessários para o funcionamento do robô, houve a sobrecarga de processamento na Raspberry PI, o que resultou em travamentos durante a operação e tornou inviável a realização da tarefa de inspeção. Por conta disso, uma série de otimizações foram tomadas, afim de solucionar a questão do alto custo computacional do sistema.

A primeira otimização consistiu na reinstalação do sistema operacional Linux Ubuntu, porém, sem interface gráfica, o que resultou em um ambiente mais leve para a Raspberry PI. Além disso, uma série de programas nativos, como editores de texto, gerenciadores de arquivos e reprodutores de mídia, foram desinstalados para simplificar o sistema. Essas medidas foram cruciais para liberar recursos de processamento e memória da Raspberry PI, permitindo que ela executasse as tarefas de forma mais eficaz.

Outra melhoria foi a atualização do *firmware* da câmera Realsense. Isso permitiu a configuração interna de parâmetros como resolução máxima, taxa de quadros por segundo, tipo de compressão e largura de banda de comunicação na câmera. No que se refere à transmissão de imagens, foi utilizado o *software* FFmpeg para fazer o *streaming* do conteúdo para o computador de operação remota, aproveitando um *hardware* específico na Raspberry PI. Isso liberou a CPU da Raspberry PI para outras tarefas.

O *hardware* de codificação de vídeo é uma parte especializada do processamento da

Raspberry PI dedicada a lidar com tarefas relacionadas ao tratamento de vídeos de maneira mais eficiente. Em vez de depender principalmente do poder de processamento geral da CPU, a codificação por *hardware* utiliza circuitos otimizados para acelerar a compressão de vídeo, tornando-a mais rápida e econômica em termos de energia. Isso é particularmente útil em aplicações como transmissões ao vivo, onde a eficiência e a qualidade são essenciais.

O comando a seguir ilustra como é executado a transmissão de vídeo por meio do *hardware* de vídeo na Raspberry PI:

```
./ffmpeg -i /dev/video6 -vcodec h264_v4l2m2m -b:v 8M -f  
rtsp -rtsp_transport tcp rtsp://10.42.0.1:8888/live.sdp
```

Este comando utiliza o *software* FFmpeg para capturar o vídeo do dispositivo /dev/video6, codificá-lo com o codec h264_v4l2m2m, configurar uma taxa de bits de vídeo de 8 Mbps e transmitir o resultado via RTSP usando o transporte TCP para o endereço rtsp://10.42.0.1:8888/

Os motores também tiveram seus *firmwares* atualizados para utilizar uma função que recebe valores de corrente ao invés de valores de velocidade. Isso foi feito porque, em situações em que ocorriam problemas como o travamento da esteira, o motor não conseguia atingir a velocidade desejada e entrava em falha, exigindo um reinício manual. Com comandos de corrente, a meta foi manter uma corrente específica, independentemente de a roda estar girando ou não, eliminando assim esse problema de falha. Vale ressaltar que foi criada uma função para converter o comando de velocidade do *joystick* para valores de corrente.

Por fim, após a implementação das otimizações tanto na Raspberry Pi quanto nos *softwares* do sistema, foi possível alcançar um desempenho eficiente do robô. Durante a execução de suas tarefas, o consumo computacional passou de 100% para uma média de 70%. Isso demonstra que as melhorias implementadas foram eficazes em garantir o funcionamento estável do robô, possibilitando que ele cumprisse suas atividades dentro das capacidades de processamento disponíveis.

3.4. Simulação do robô

A evolução do poder de processamento dos computadores, aliada à disponibilidade de uma ampla gama de *softwares* e *hardwares* com licenças de uso livre, trouxe uma mudança drástica no cenário da simulação robótica em 3D. Essa evolução possibilitou a realização de simulações com sistemas de *feedback*, que mantêm uma comunicação constante e cíclica com as variações do ambiente simulado (ROHMER *et al.*, 2013).

Nesse contexto, o simulador de robótica CoppeliaSim surge como resultado de um esforço para conciliar todos esses requisitos em uma estrutura de simulação versátil e escalável. O CoppeliaSim oferece um ambiente de desenvolvimento integrado e é baseado em uma arquitetura de controle distribuído. Isso significa que cada objeto ou modelo dentro da simulação pode ser controlado de forma independente por meio de diversos métodos, como um *script*

embutido, um *plugin*, um nó ROS, um cliente API remoto ou até mesmo uma solução personalizada. Por conta disso o CoppeliaSim é altamente versátil e adequado para aplicações que envolvem múltiplos robôs. Além disso, os controladores podem ser escritos em várias linguagens, incluindo C, C++, Python, Java, Lua, Matlab ou Octave (COPPELIASIM, 2023).

A decisão de utilizar o CoppeliaSim neste projeto trouxe como vantagem a capacidade de realizar testes com o modelo 3D do robô antes mesmo de sua fabricação. Essa abordagem teve um impacto significativo, pois permitiu a identificação antecipada de possíveis problemas que, de outra forma, só seriam descobertos após a construção física do robô. Isso resultou em economia de tempo e recursos, além de contribuir para um processo de desenvolvimento mais eficiente e livre de erros.

O simulador, que pode ser observado na Figura 3.13, oferece a versatilidade de definir dados físicos e geométricos reais do robô e do ambiente, tornando a simulação mais próxima da realidade. A integração do CoppeliaSim com o ROS é realizada por meio de uma API (*Application Programming Interface*), possibilitando uma comunicação direta com o *software* do robô. Isso significa que os mesmos códigos de comando desenvolvidos para o protótipo físico podem ser usados para controlar o dispositivo virtual.

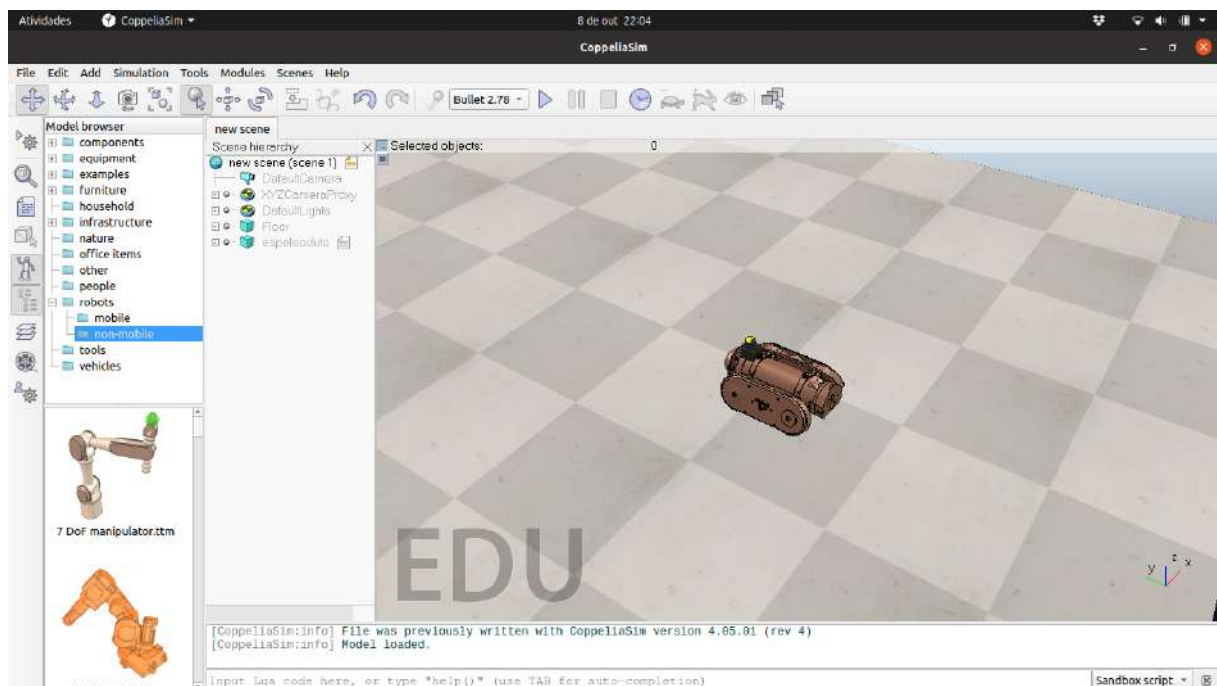


Figura 3.13: Interface gráfica do simulador CoppeliaSim. Fonte: O autor.

A Figura 3.14 exibe o modelo desenvolvido no ambiente de simulação CoppeliaSim. Neste modelo, alguns sensores foram incorporados ao robô, com o propósito de verificar as capacidades de percepção e interação do sistema com seu ambiente. Esses sensores abrangem uma variedade de tipos, incluindo câmeras para visão computacional, acelerômetros para detecção de aceleração, giroscópios para medição de orientação e sensores de rotação dos motores, conhecidos como *encoders*.

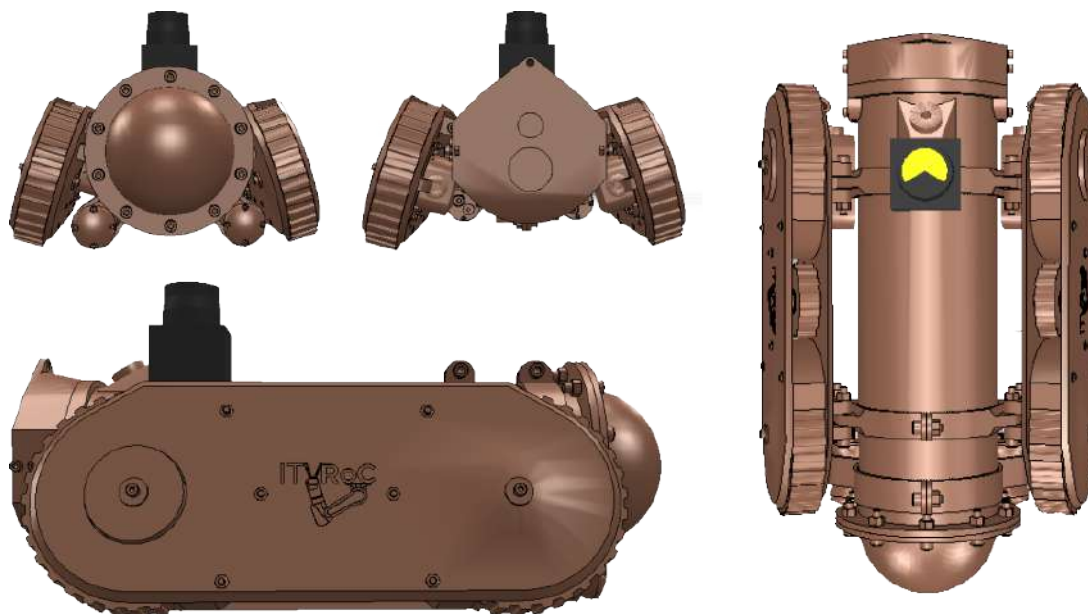


Figura 3.14: Modelo do robô criado no CoppeliaSim. Fonte: O autor.

A adição desses componentes permite que o robô adquira informações sobre o ambiente simulado e tome decisões com base nessas informações. Além disso, a implementação de *scripts* que estabelecem a integração entre a simulação no CoppeliaSim e o ambiente de desenvolvimento ROS (*Robot Operating System*) possibilita que testes e experimentos realizados no ambiente de simulação possam ser diretamente transferidos e replicados no robô real.

Para criar o robô no CoppeliaSim, foi adotada uma abordagem de modelagem que envolveu o uso de formas geométricas primitivas, como cilindros, para sua representação na simulação. Em seguida, foi incorporada uma modelagem criada no *SolidWorks*, conforme ilustrado na Figura 3.14. Vale ressaltar que essa modelagem adicionada possui apenas um caráter visual e não exerce influência direta sobre a dinâmica da simulação. A razão para essa escolha é influenciada pela necessidade de evitar a complexidade excessiva que modelos 3D possuem em suas malhas, pois nessas malhas poderiam resultar em uma grande quantidade de triângulos, tornando a simulação computacionalmente pesada para execução.

Por fim, na Figura 3.15, é possível observar que o CoppeliaSim oferece a capacidade de criar ambientes virtuais. Por exemplo, foram modeladas tubulações e galerias no ambiente de simulação, permitindo a realização de testes que se assemelham o máximo possível à realidade. Essa funcionalidade de criar ambientes virtuais adiciona um alto grau de realismo as simulações, contribuindo para uma validação mais precisa e completa do sistema antes de sua implementação física.

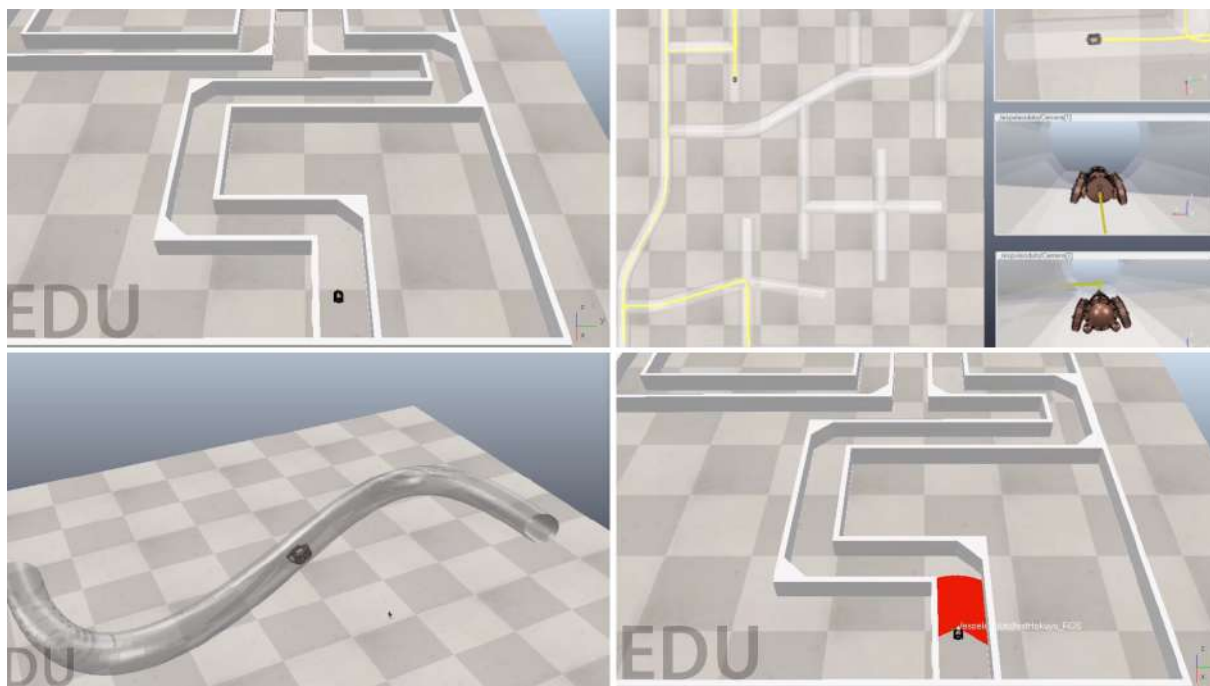


Figura 3.15: Ambientes de teste criado no Coppeliasim. Fonte: O autor.

4. Sistema de navegação para inspeção robótica de tubulações

Devido ao propósito deste projeto, que é desenvolver um robô de baixo custo, foram enfrentadas limitações significativas, especialmente em relação aos recursos de *hardware*. Essas limitações resultam em restrições de capacidade de processamento para o robô em desenvolvimento. Isso implica que técnicas de SLAM que dependem da utilização de imagens, nuvem de pontos ou cálculos *online* se tornem inviáveis devido ao alto custo computacional. Portanto, foi considerada uma alternativa de solução de navegação que demanda uma quantidade reduzida de dados e requer menos poder de processamento, com o objetivo de ser implementada no robô abordado nesta dissertação.

Para a locomoção autônoma do robô dentro dos dutos, é essencial aplicar uma estratégia de movimentação que assegure sua capacidade de percorrer com sucesso o caminho a ser inspecionado. A abordagem de navegação baseada em sensores envolve a integração das informações sensoriais, que refletem o estado atual do ambiente, diretamente no processo de locomoção do robô. Isso contrasta com as técnicas de navegação clássicas, nas quais é necessário ter um conhecimento completo da geometria do ambiente antes de iniciar o planejamento. Existem várias vantagens notáveis na navegação baseada em sensores em comparação com os métodos clássicos (RODRIGUES, 2010):

1. **Falta de conhecimento prévio:** Em muitos cenários, o robô não possui informações antecipadas sobre o ambiente em que vai operar. A navegação baseada em sensores permite que ele tome decisões adaptativas com base nas informações sensoriais disponíveis, tornando-o apto a lidar com ambientes desconhecidos.
2. **Conhecimento parcial:** Devido às limitações de memória e às restrições de *hardware*, o robô frequentemente possui apenas um conhecimento parcial do ambiente. A navegação baseada em sensores permite que ele navegue de forma eficiente, mesmo com informações limitadas.
3. **Modelos de ambiente não precisos:** Muitas vezes, os modelos do ambiente não são precisos o suficiente para uma navegação confiável. A abordagem baseada em sensores permite que o robô ajuste seu comportamento com base nas informações sensoriais, se adaptando a variações e imprecisões no ambiente.
4. **Mudanças rápidas e situações inesperadas:** Ambientes dinâmicos podem apresentar situações inesperadas ou mudanças rápidas. A navegação baseada em sensores permite que o robô reaja a essas mudanças de maneira mais ágil e eficaz do que os métodos clássicos, que dependem de informações estáticas.

Em resumo, a navegação baseada em sensores é uma abordagem altamente vantajosa, especialmente em contextos em que o ambiente é desconhecido, as informações são limitadas,

os modelos são imprecisos ou podem ocorrer mudanças inesperadas. Essa estratégia oferece uma maior adaptabilidade e independência ao robô durante sua locomoção autônoma.

Uma das abordagens que exploram essa técnica foi apresentada por Pereira **et al.** (2021). A proposta desse método envolve a movimentação do robô com uma velocidade linear fixa, ao mesmo tempo em que calcula uma velocidade angular para manter a plataforma móvel posicionada no centro da tubulação que está sendo inspecionada. Esse cálculo da velocidade angular se baseia na medição das distâncias entre o robô e as paredes laterais da tubulação. O algoritmo foi projetado com o objetivo de manter essas distâncias iguais nos lados esquerdo e direito do robô.

Além disso, a técnica permite a utilização de dados inerciais para controlar a orientação do robô e mantê-lo constantemente alinhado. Para alcançar esses objetivos, o algoritmo implementa um controle PID (Proporcional Integral Derivativo). Esse controle é responsável por calcular a velocidade angular necessária para manter as distâncias laterais iguais e, ao mesmo tempo, reduzir qualquer inclinação indesejada da plataforma robótica.

Com base nos estudos mencionados e em virtude das particularidades desta dissertação, foi adotada uma estratégia de navegação em sistemas de tubulação utilizando um método que se baseia em máquina de estados e controle de campos vetoriais artificiais. Isso se deve, principalmente, à menor demanda de recursos de processamento necessária por essa abordagem.

A máquina de estados utilizada foi proposta por Amaral **et al.** (2022) com o propósito de resolver o desafio da exploração robótica em tubulações e galerias. Já o controle por campos vetoriais artificiais foi desenvolvido para ser aplicado no EspeleoRobô (REZENDE **et al.**, 2021) e, nesta dissertação, foi adaptado para ser empregado no PipeBot.

4.1. Máquina de estados

As máquinas de estados desempenham um papel fundamental na modelagem do comportamento do robô, permitindo a divisão do processo de navegação em diferentes estados. Essa abordagem capacita o robô a tomar decisões que levam em consideração o contexto e as condições do ambiente em que se encontra (FOUKARAKIS et al., 2014).

4.1.1. Máquinas de Estados Determinísticas Finitas

As máquinas de estados são modelo matemático utilizados para representar programas de computadores, circuitos lógicos e até mesmo sistemas. Basicamente, uma máquina de estados consiste em um conjunto de estados distintos, onde a máquina opera em um estado por vez, conhecido como estado atual, armazenando informações sobre o histórico das mudanças ocorridas desde o início do sistema. As transições entre estados são acionadas por condições específicas, e a máquina pode executar ações definidas em momentos determinados. Essa versatilidade torna as máquinas de estados um conceito essencial para a modelagem e compreensão de uma ampla gama de sistemas e processos complexos (BÖRGER, 2010).

Máquinas de estados são particularmente adequada para sistemas que operam *online*, nos quais é crucial cumprir prazos rígidos, onde o tempo é um fator crítico. Além disso, essa abordagem permite que a tolerância a falhas seja tratada separadamente de outros aspectos do *design* do sistema. Isso dá aos programadores a possibilidade de escolher a abstração de programação mais apropriada para sua aplicação e, ao mesmo tempo, utilizar a máquina de estados para implementar a tolerância a falhas nessa abstração, o que torna o sistema mais robusto e confiável.

Uma máquina de estados determinística é um conceito que envolve variáveis que representam o estado atual do sistema e comandos que alteram esse estado. Cada comando é executado de forma determinística, o que significa que sua execução é previsível e não depende de fatores aleatórios. Essa execução de comando é indivisível em relação a outros comandos, ou seja, ocorre de forma isolada e não pode ser interrompida no meio. O resultado da execução do comando pode ser direcionado para diferentes destinos, como por exemplo, atuadores em um sistema de controle de processo (SCHNEIDER, 2005).

Assim, é possível observar que uma máquina de estados simplifica a análise do sistema ao estabelecer estados finitos e pré-definidos. Isso se mostra importante no desenvolvimento de sistemas que trabalham *online*, pois esses estados oferecem uma representação clara do comportamento dinâmico do sistema.

Um diagrama de um autômato finito determinístico (Deterministic finite automata - DFA) constitui uma representação visual desse conceito, sendo caracterizado por:

$$M = (\Sigma, Q, \sigma, q_0, F)$$

- Σ é o alfabeto de símbolos de entrada.

- Q é o conjunto finito de estados possíveis.
- σ é a função de transição.
- q_0 é o estado inicial tal que $q_0 \in Q$.
- F é o conjunto de estados finais tal que $F \subseteq Q$.

Segundo Zenkevich **et al.** (2019), na robótica móvel, uma máquina de estados é frequentemente utilizada para controlar o comportamento do robô. O robô pode passar por diferentes estados, como “navegação”, “desvio de obstáculos”, “busca de alvo” e assim por diante, um exemplo é apresentado na Figura 4.1. Cada estado define as ações que o robô deve executar e as condições de transição para o próximo estado. Por exemplo, quando um sensor detecta um obstáculo à frente, o robô pode mudar do estado de “navegação” para o estado de “desvio de obstáculo” e iniciar ações para evitar a colisão. Isso permite que os robôs móveis sejam programados de forma modular e adaptativa, respondendo de maneira eficiente às mudanças em seu ambiente, tornando-os ideais para tarefas de navegação autônoma em ambientes dinâmicos.

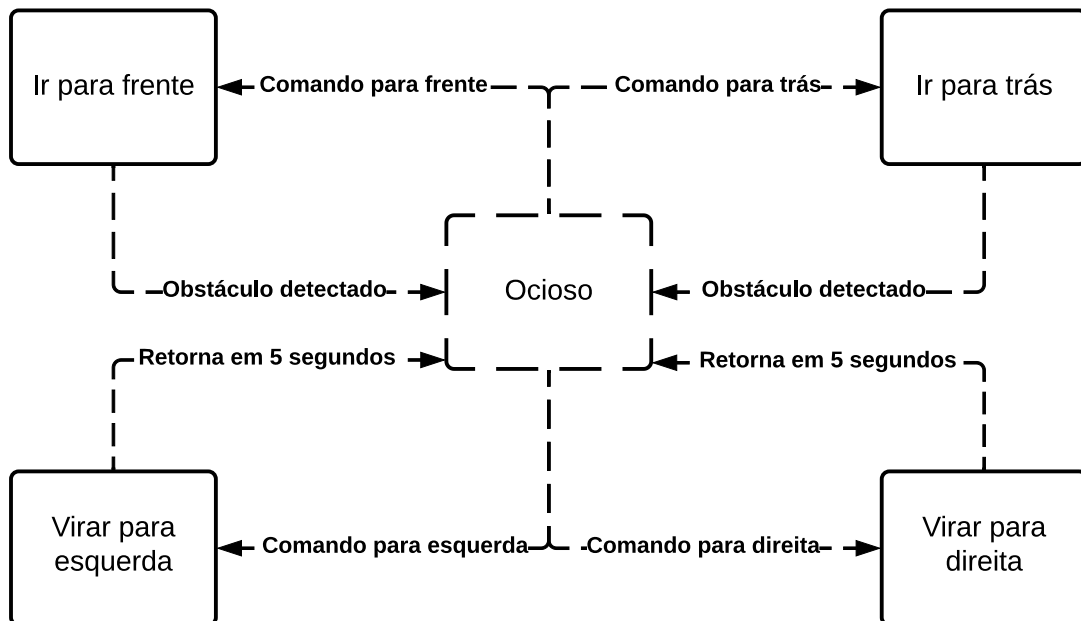


Figura 4.1: Diagrama com exemplo básico de uma máquina de estados para um robô móvel.
Fonte: O autor.

4.1.2. Máquina de estados para inspeção autônoma de tubulações

Normalmente, sistemas de tubulação possuem pontos de entrada e saída em locais estratégicos, facilitando o acesso às seções que geralmente sofrem desgaste significativo. Nesse

sentido, o objetivo da navegação é guiar o robô ao longo da linha central da tubulação e utilizar um algoritmo de tomada de decisão com base na identificação de bifurcações e pontos sem saída.

Assim, a máquina de estados apresentada em (AMARAL et al., 2022) foi utilizada. Esse sistema de navegação utiliza uma lógica de estados e transições. Cada estado implementa um método de navegação adequado para uma situação específica, enquanto cada transição identifica o método de navegação mais apropriado para a posição atual do robô. A Figura 4.2 ilustra a máquina de estados utilizada.

O funcionamento geral da máquina de estados ocorre da seguinte forma: enquanto o robô não detecta bifurcações ou pontos sem saída, ele permanece no estado de controle de corredor, navegando para frente pelo centro da tubulação. No entanto, quando identifica uma bifurcação ou um ponto sem saída, ele muda para o estado de controle de bifurcação e passa a seguir a parede esquerda da tubulação. A opção de seguir a parede esquerda pode limitar a exploração de ambientes com múltiplas saídas ou ambientes que contenham *loops* internos. No entanto, essa abordagem é escolhida devido à sua simplicidade e eficiência na tomada de decisões dentro das tubulações.

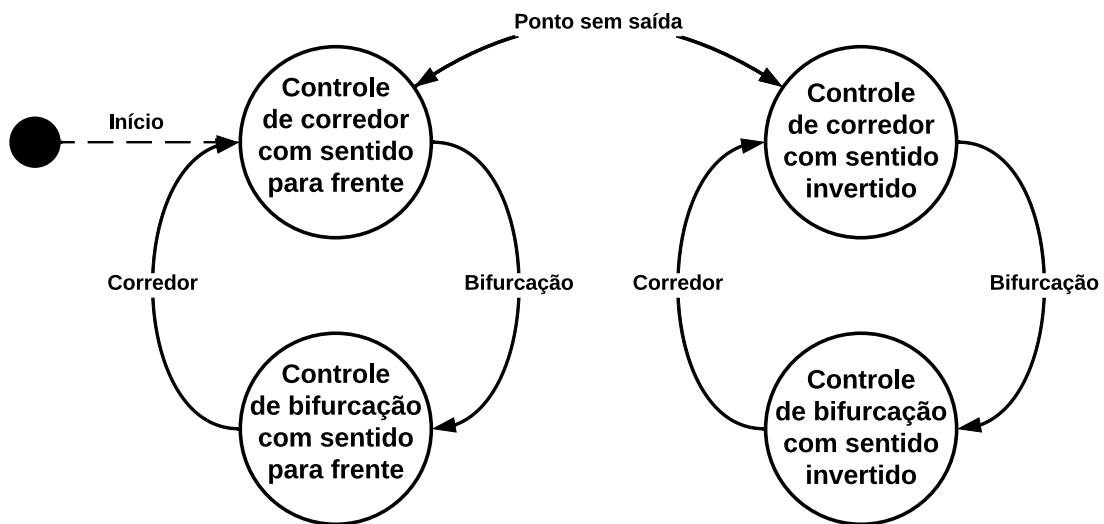


Figura 4.2: Máquina de estados representando a estratégia de navegação para inspeção de tubulações. Fonte: Adaptada de (AMARAL et al., 2022).

As transições entre estados são descritas da seguinte forma:

- *Transição 1 - início:* Esta transição ocorre incondicionalmente assim que o robô é ligado, iniciando a inspeção;
- *Transição 2 - bifurcação:* Disparada quando o algoritmo detecta um desvio à frente do robô. A identificação de bifurcações é baseada na premissa de que cada caminho possível é representado por um grupo de feixes com valores de distância maiores que um limite predefinido, como mostrado na Figura 4.3. Quando mais de um grupo de feixes é iden-

tificado na frente do robô, o sistema considera que há uma bifurcação, e a transição de estado é acionada;

- *Transição 3 - corredor*: Acionada quando não há detecção de bifurcação, indicando que há apenas um caminho possível para o robô percorrer, caracterizando um corredor;
- *Transição 4 - ponto sem saída*: O sensor de distância identifica um obstáculo no caminho à frente do robô, indicando que a única rota possível é retornar no sentido inverso. Essa identificação ocorre quando a média das leituras do sensor a laser é menor que um parâmetro predefinido.

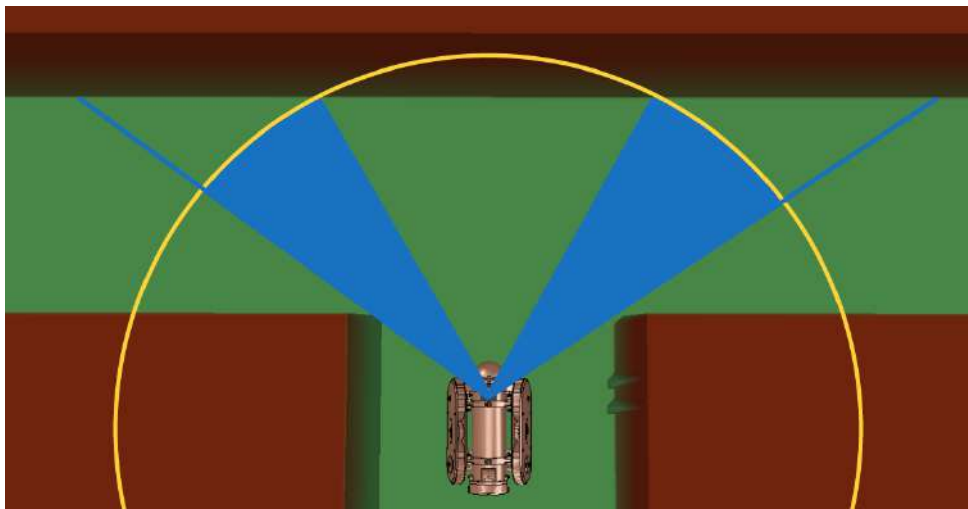


Figura 4.3: Método de detecção de bifurcações. O círculo amarelo mostra o limite máximo de distância que o robô utiliza para dividir as medições em grupos. Fonte: Adaptada de (AMARAL et al., 2022).

Os estados são detalhados a seguir:

- *Estado 1 - Controle de corredor com sentido para frente*: Neste estado, o robô busca se manter na linha central do corredor, navegando para frente usando as leis de controle apresentadas na Seção 4.2.2;
- *Estado 2 - Controle de Bifurcação com sentido para frente*: Se uma bifurcação for detectada, o estado operacional do robô muda para seguir o lado esquerdo, a fim de permanecer no lado esquerdo do desvio, usando as leis de controle apresentadas na Seção 4.2.3;
- *Estado 3 - Controle de Corredor em sentido invertido*: O robô busca se manter na linha central da tubulação usando as leis de controle apresentadas na Seção 4.2.2, semelhante ao estado Estado 1, mas navegando no sentido inverso;
- *Estado 4 - Controle de Bifurcação em sentido invertido*: Semelhante ao estado Estado 2, o controle é alterado para seguir o lado esquerdo, mas agora com comandos de velocidade

negativos, usando as leis de controle apresentadas na Seção 4.2.3, semelhante ao estado Estado 2, mas navegando no sentido inverso.

4.2. Campos Vetoriais artificiais

Campos vetoriais artificiais consistem em uma estratégia de controle que pode ser aplicada para o comando de robôs móveis, permitindo que eles realizem tarefas complexas, como geração e rastreamento de padrões no espaço. Essencialmente, um campo vetorial é uma função matemática que atribui vetores a pontos em um domínio n-dimensional. Em robótica, esse domínio representa o espaço de configuração do robô, enquanto os vetores indicam direções e magnitudes que o robô deve seguir.

Esses campos podem ser descritos como funções $f(p_R) : \mathbb{R}^n \rightarrow \mathbb{R}^n$, onde p_R representa as coordenadas no espaço \mathbb{R}^n . No contexto da navegação de robôs, a posição do robô geralmente pertence a \mathbb{R}^2 , e inicialmente, é assumido que a taxa de mudança das coordenadas seja $\dot{p}_R = v_u$. Essa técnica é aplicada para orientar um robô em direção a um determinado percurso, usando uma lei de controle $v_u = f(p_R)$, que depende do estado atual do robô. Uma representação gráfica desses campos está ilustrada na Figura 4.4.

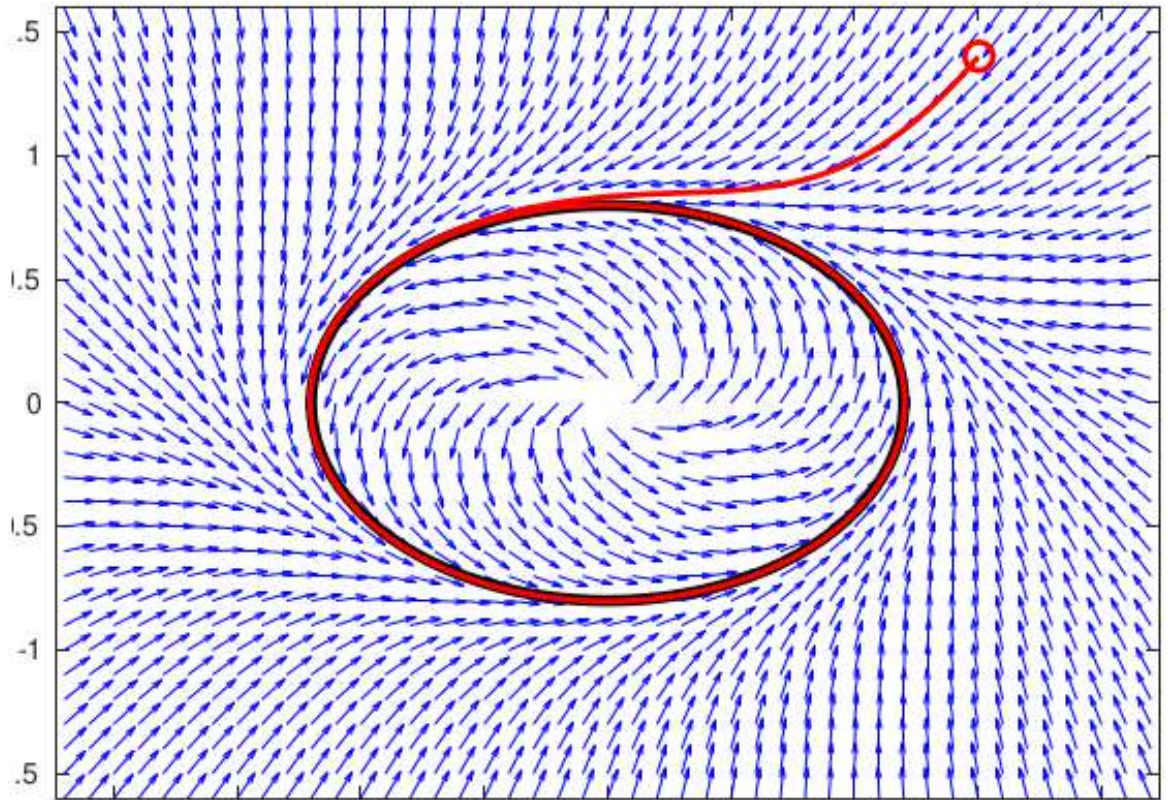


Figura 4.4: Exemplo de campo vetorial cujas linhas integrais convergem para um caminho. Fonte: (FREITAS et al., 2020).

O controle por campos vetoriais utiliza uma abordagem fundamentada na determinação de uma velocidade de referência composta por duas partes distintas: (i) a componente convergente; e (ii) a componente tangente. A componente convergente tem a responsabilidade de direcionar o robô em direção à trajetória de referência, enquanto a componente tangente orienta o robô ao longo do percurso. A Figura 4.5 ilustra quatro pontos distintos em uma função de campo vetorial. Os vetores em vermelho representam a componente convergente, e os vetores em verde representam a componente tangente.

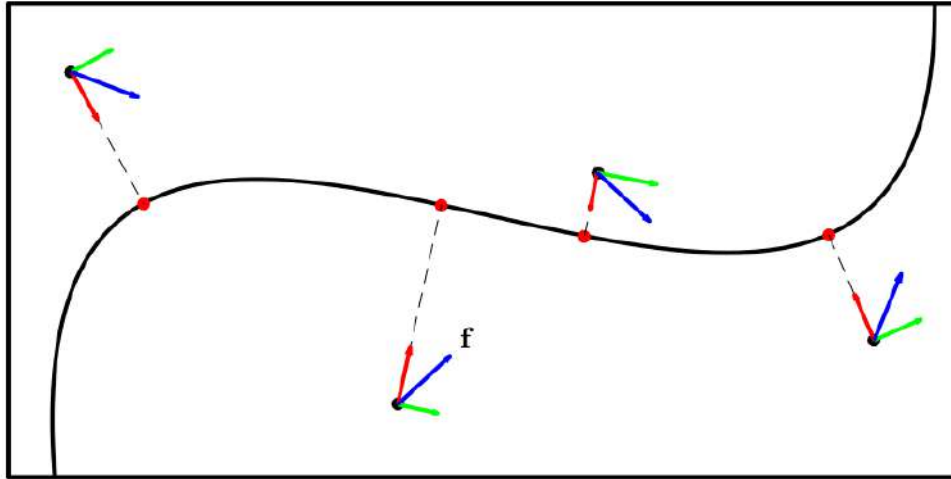


Figura 4.5: O campo vetorial (azul) é construído pela soma de uma componente convergente (vermelha) com uma componente tangente (verde). Fonte: Adaptada de (FREITAS et al., 2020).

4.2.1. Modelagem por *feedback linearization*

Para implementar o controle de navegação de um robô, é fundamental contar com um modelo matemático que represente suas características essenciais. Esse modelo precisa ser completo o suficiente para descrever propriedades básicas do robô, mas ao mesmo tempo, não deve ser excessivamente complexo, a fim de facilitar o projeto das leis de controle. O modelo mais simples que se pode escolher é o do integrador $\dot{p} = v$, isto é, as velocidades em x e y do robô podem ser diretamente impostas por meio de entradas de controle v_x e v_y .

O modelo cinemático considerado para o robô é apresentado na Equação 4.1, onde x e y são as coordenadas de posição, θ é o ângulo de guinada, e as entradas de controle v e ω são as velocidades linear e angular, respectivamente. As velocidades v e ω , entradas do modelo, podem ser convertidas para velocidades das rodas por meio de uma transformação linear que depende de parâmetros geométricos do robô.

$$\begin{aligned}\dot{p}_x &= v \cos(\theta), \\ \dot{p}_y &= v \sin(\theta), \\ \dot{\theta} &= \omega.\end{aligned}\tag{4.1}$$

O modelo 4.1 desconsidera qualquer deslizamento das rodas com o solo, neste caso, a

restrição não-holonômica é descrita pela Equação 4.2

$$\dot{P}_x \sin(\theta) - \dot{P}_y \cos(\theta) = 0, \quad (4.2)$$

e significa que o robô não possui velocidade na direção ortogonal ao vetor x_b , do sistema de coordenadas local, de acordo com o indicado na Figura 4.6.

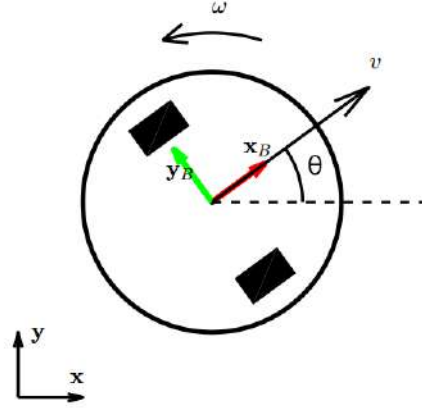


Figura 4.6: Representação de um robô com cinemática diferencial. Fonte: Adaptada de (FREITAS et al., 2020)

Uma estratégia viável para lidar com o problema da não-holonomia é a abordagem de controle por *feedback linearization*, utilizando um ponto auxiliar chamado $p_{feedback}$, conforme descrito por Siciliano et al. (2009). Esse ponto é fixo em relação ao robô e está localizado a uma distância $d > 0$ do centro do robô na direção x_b . Enquanto o centro do robô (que é o ponto médio entre as rodas) não pode se mover na direção y_b , o ponto $p_{feedback}$ não possui essa restrição.

A técnica descrita em (SICILIANO et al., 2009) envolve a determinação de uma matriz de transformação que calcula um par (v, ω) de forma que a velocidade do ponto $p_{feedback}$ seja v_u . Essa transformação é definida da seguinte forma:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -d^{-1} \sin(\theta) & d^{-1} \cos(\theta) \end{bmatrix} \begin{bmatrix} v_{u,x} \\ v_{u,y} \end{bmatrix} \quad (4.3)$$

Com a Equação 4.3 é possível aplicar uma velocidade v_u ao ponto $p_{feedback}$ (conforme a Figura 4.7). Se for assumido que a posição p do robô é o ponto $p_{feedback}$, este converge para a referência.

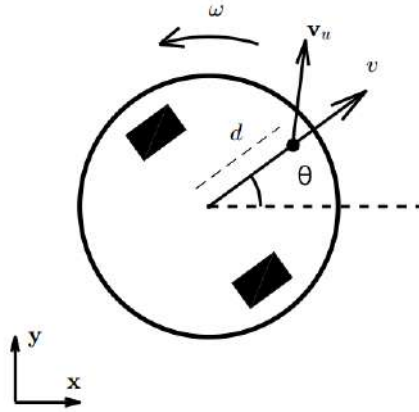


Figura 4.7: Adição do $p_{feedback}$ na representação gráfica do modelo diferencial. Fonte: Adaptada de (FREITAS et al., 2020).

4.2.2. Campos vetoriais para navegação em corredores

A primeira lei de controle utilizada nessa dissertação foi descrita em (REZENDE et al., 2021), com o propósito de guiar o robô pelo meio do duto, utilizando as duas paredes laterais para centralizar o robô. O campo vetorial F que converge para uma curva localizada no centro de duas paredes é dado por:

$$\mathbf{F}^b = G \frac{\mathbf{D}^b}{\|\mathbf{D}^b\|} + H \frac{R\mathbf{D}^b}{\|R\mathbf{D}^b\|}, \quad (4.4)$$

Aqui, \mathbf{D}^b representa o vetor que se estende do ponto mais próximo à trajetória de referência (localizado no centro da galeria) até a posição atual do robô. O sobrescrito ^b indica que as variáveis estão sendo calculadas no referencial do corpo do robô. Além disso, $R \in SO(2)$ é uma matriz de rotação de 90° , definida como:

$$R = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}. \quad (4.5)$$

Assim, o vetor $R\mathbf{D}^b$ assume uma direção tangente à curva. As funções G e H são definidas da seguinte maneira:

$$\begin{aligned} G &= -\frac{2}{\pi} \operatorname{atan}\left(k_f \cdot \|\mathbf{D}^b\|\right), \quad k_f > 0, \\ H &= \sqrt{1 - G^2}, \end{aligned} \quad (4.6)$$

sendo $k_f > 0$ o ganho de convergência.

O ângulo φ_E (no referencial do robô) corresponde ao ângulo no qual o ponto mais próximo à esquerda foi detectado, enquanto L_E representa a distância medida até esse ponto. De maneira equivalente, para o lado direito, temos φ_D e L_D . É importante observar que, em geral, φ_E é maior que zero, e φ_D é menor que zero, uma vez que esses ângulos são definidos

com relação ao eixo x do corpo do robô. A Figura 4.8 ilustra as definições utilizadas no controle do seguidor de corredor.

O vetor \mathbf{D}^b pode ser calculado como:

$$\mathbf{D}^b = \frac{L_E - L_D}{2} \begin{bmatrix} \cos(\varphi_D) \\ \sin(\varphi_D) \end{bmatrix}. \quad (4.7)$$

Ao calcular o vetor \mathbf{D}^b no referencial do corpo, o campo vetorial \mathbf{F}^b também é obtido no mesmo referencial do corpo. Portanto, o cálculo de \mathbf{F}^b é realizado sem depender de informações globais.

Devido às restrições não holonômicas do robô em uso, foi adotada a abordagem de *feedback linearization*, conforme mencionado na Seção 4.2.1, para transformar o vetor \mathbf{F}^b em velocidades linear (v) e angular (ω) do robô. Essa estratégia envolve a atribuição da velocidade de referência \mathbf{F}^b a um ponto virtual localizado a uma distância d à frente do ponto central entre as rodas do robô. A simplificação desta estratégia é possível devido ao fato de a velocidade de referência \mathbf{F}^b estar definida no referencial do corpo. Portanto, podemos expressá-la da seguinte maneira:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{d} \end{bmatrix} \begin{bmatrix} F_x^b \\ F_y^b \end{bmatrix}. \quad (4.8)$$

onde $[F_x^b, F_y^b]^T = \mathbf{F}^b$.

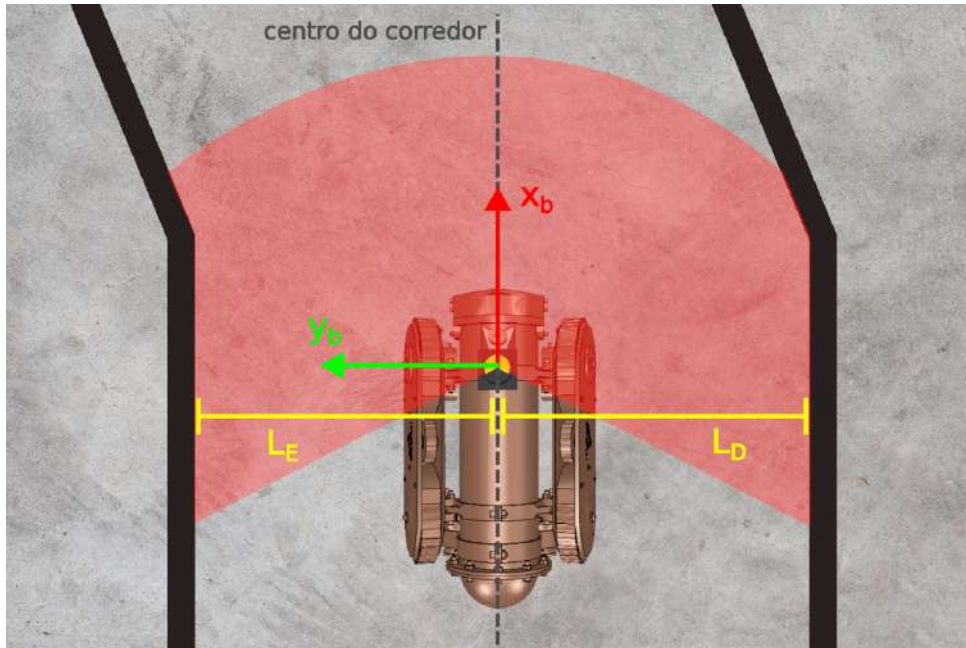


Figura 4.8: Representação visual do robô realizando navegação se mantendo no centro das paredes no simulador. Fonte: O autor.

4.2.3. Campos vetoriais para seguimento de paredes

A segunda lei de controle utilizada nessa dissertação tem como objetivo direcionar o robô para seguir um caminho imaginário a uma distância constante de uma parede. Essa abordagem é aplicada quando o robô se depara com uma bifurcação e pode ser calculada de forma análoga a primeira lei. Realizando a alteração apenas do campo vetorial F que converge para uma curva localizada a uma distância ε da parede. Esse campo é dado por:

$$\mathbf{F}^b = G \frac{\mathbf{E}^b}{\|\mathbf{E}^b\|} + H \frac{R\mathbf{D}^b}{\|\mathbf{D}^b\|}, \quad (4.9)$$

Onde \mathbf{D}^b é o vetor mais curto da parede para o robô, e \mathbf{E}^b é o vetor mais curto do caminho desejado para o robô. O ângulo φ_m (no referencial do robô) corresponde ao ângulo no qual o ponto mais próximo a parede foi detectado, enquanto L_m representa a distância medida até esse ponto, o ε é a distância desejada que robô deve permanecer da parede durante o seu movimento. Por definição o robô segue a parede da esquerda e portanto a medida mais próxima considerada fica no lado esquerdo do robô. A Figura 4.8 ilustra as definições utilizadas no controle do seguidor de corredor.

$$\mathbf{E}^b = \mathbf{D}^b - \varepsilon \frac{\mathbf{D}^b}{\|\mathbf{D}^b\|}. \quad (4.10)$$

$$\mathbf{D}^b = -L_m \begin{bmatrix} \cos(\varphi_m) \\ \sin(\varphi_m) \end{bmatrix}. \quad (4.11)$$

Utilizando novamente a abordagem do *feedback linearization* mencionado na Seção 4.2.1 e o novo campo gerado é possível definir a lei de controle que permite ao robô navegar tangencialmente a uma parede com base apenas na distância do feixe de laser mais próximo.

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{d} \end{bmatrix} \begin{bmatrix} F_x^b \\ F_y^b \end{bmatrix}. \quad (4.12)$$

onde $[F_x^b, F_y^b]^T = \mathbf{F}^b$.

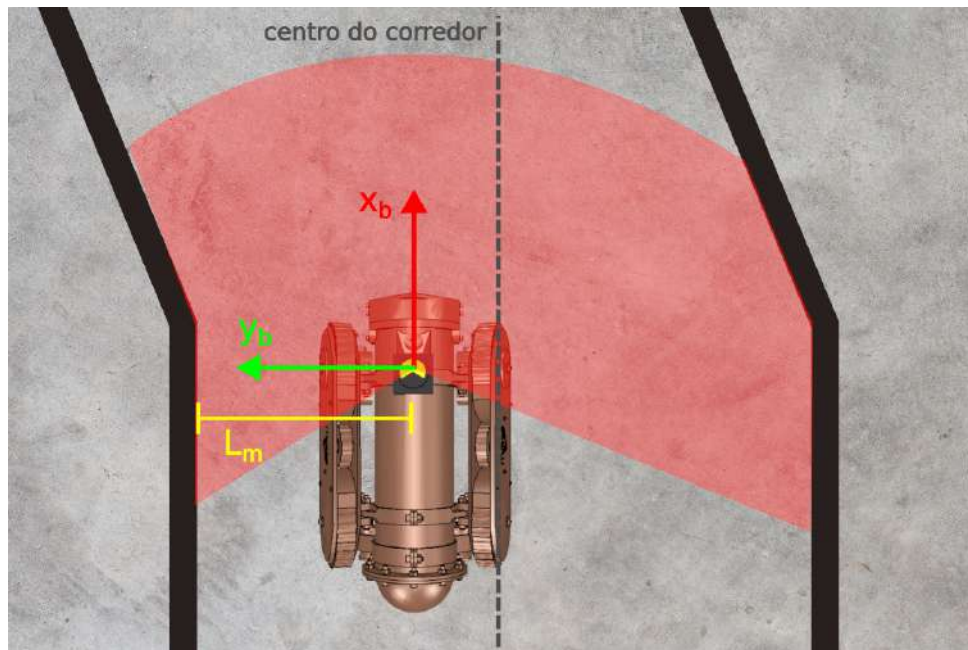


Figura 4.9: Representação visual do robô realizando navegação seguindo a parede da esquerda no simulador. Fonte: O autor.

5. Experimentos e resultados

Este capítulo aborda tanto as simulações realizadas com o robô virtual quanto os experimentos conduzidos com o robô real. Tais simulações e experimentos foram possibilitaram a identificação de áreas de aprimoramento, ao mesmo tempo em que validaram as funcionalidades propostas para o robô. Os resultados obtidos evidenciaram uma característica fundamental do desenvolvimento: sua adaptabilidade a uma variedade de cenários. Isso se traduz na facilidade de incorporar novos sensores, integrar componentes mecânicos ou eletrônicos adicionais e utilizar um software modular capaz de integrar novas soluções de maneira rápida.

5.1. Resultados de simulação

A validação da plataforma robótica e das técnicas de navegação ocorreu inicialmente por meio de simulações utilizando um modelo virtual do robô no CoppeliaSim. A integração deste ambiente com o ROS permitiu a realização de testes dos mesmos pacotes implementados para o protótipo real. Esse processo de validação possibilitou a verificação das funcionalidades do sistema antes mesmo da implementação no ambiente físico.

Os principais testes realizados no simulador incluíram o controle do robô simulado por meio de um *joystick*, a locomoção do robô dentro de tubulações, a capacidade do sistema de esteira em transpor pequenos obstáculos e a validação do modelo cinemático do robô.

O simulador desempenhou um papel fundamental na validação tanto da máquina de estados quanto do controle por campos vetoriais. Ele permitiu a simulação de um sensor laser planar e a realização de ajustes em todos os parâmetros antes dos testes no robô real. Um exemplo dessas simulações é ilustrado na Figura 5.1. Também é possível visualizar algumas das simulações no repositório em <https://github.com/alexandremstf/roboduto>.

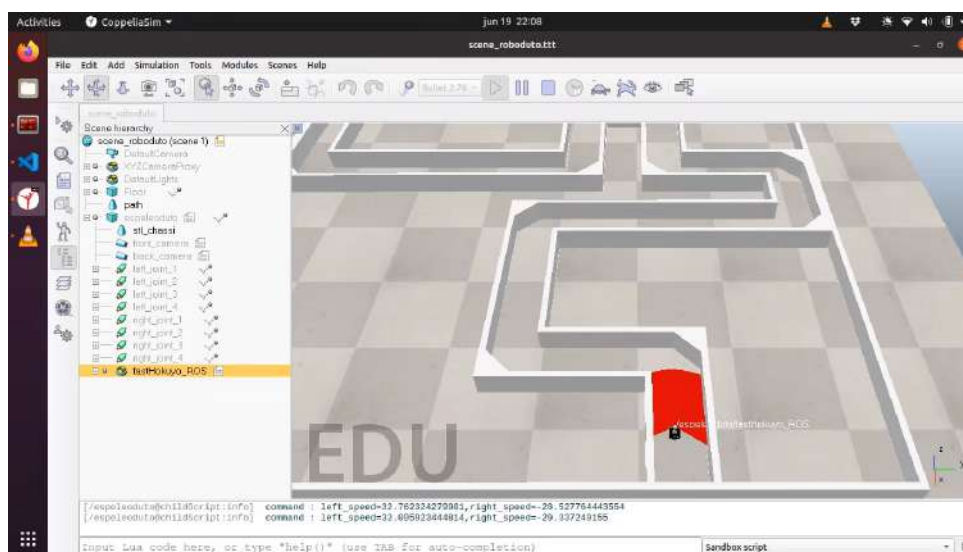


Figura 5.1: Teste de navegação autônoma pelo controle de campos vetoriais no ambiente virtual. Fonte: O autor.

Os testes da máquina de estados e o controle de navegação por campos vetoriais foram realizados em diversos cenários. A Figura 5.2 representa um mapa de tubulações criado não apenas para testar o controle, mas também para avaliar a estratégia da máquina de estados, que tomou decisões em todos os momentos em que o robô encontrava bifurcações e pontos sem saída. Para isso, no simulador, foi criado um sistema de tubulações com uma área de $30m \times 30m$, que inclui múltiplas bifurcações e caminhos sem saída. O caminho destacado em amarelo na Figura 5.2 ilustra a capacidade do robô de explorar o ambiente utilizando em conjunto as duas técnicas. Vídeos desse experimento também podem ser encontrados no repositório mencionado anteriormente.

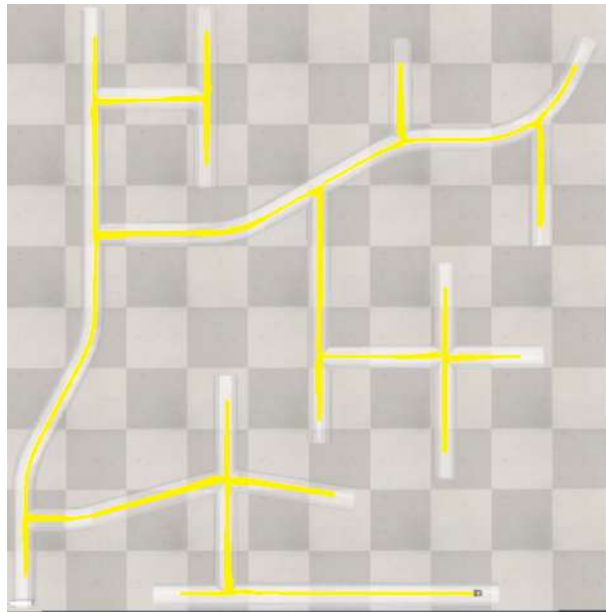


Figura 5.2: Caminho percorrido pelo robô utilizando a técnica de navegação com campos vetoriais em conjunto com a máquina de estados no ambiente virtual. Fonte: O autor.

Os resultados da simulação foram promissores, demonstrando que a estratégia proposta é capaz de identificar corretamente todos os pontos de interesse ao longo do sistema de tubulação, por meio da máquina de estados. Além disso, o robô navega consistentemente no centro dos corredores ou a uma distância fixa da parede, o que demonstra a eficácia do controle por campos vetoriais artificiais para solucionar este problema. Esses resultados apontam para o potencial dessa abordagem de navegação autônoma em aplicações práticas em ambientes reais.

5.2. Resultados experimentais

A produção de todas as peças impressas em 3D foi concluída com êxito. Desafios como a transferência do modelo 3D para a impressora e ajustes finos foram realizados com sucesso, resultando na produção de peças estruturais totalmente funcionais. Após a fabricação, o processo de montagem, tanto mecânica quanto eletrônica, foi iniciado. Isso envolveu a incorporação dos componentes na estrutura mecânica, bem como a criação de pacotes que permitiram um controle eficiente de cada periférico, tornando o robô funcional. A criação dos pacotes possibilitou a gravação de imagens das câmeras, a ativação da iluminação por meio dos LEDs embarcados, a comunicação com o operador remoto quando necessário e o controle dos motores para a movimentação do robô. Os testes realizados com o protótipo produzido foram bem sucedidos. A Figura 5.3 ilustra o robô totalmente montado, e vídeos desses testes iniciais estão disponíveis em <https://github.com/alexandremstf/pipebot>.



Figura 5.3: PipeBot montado com todos os componentes mecânicos e eletrônicos. Fonte: O autor.

• Testes teleoperados em terrenos acidentados

Após uma série de testes realizados no laboratório com o robô operado remotamente, ele foi submetido a testes de campo, incluindo terrenos acidentados. Nessa etapa, o controle do robô foi realizado por meio de comunicação sem fio, permitindo a aquisição de informações do ambiente que foram transmitidas para um computador central de forma remota. Os testes externos confirmaram a funcionalidade da plataforma para ser teleoperada em ambientes externos, como pode ser observado na Figura 5.4.



Figura 5.4: Primeiros testes em ambiente externo. Fonte: O autor.

- **Testes teleoperados em tubulações e galerias**

Na Figura 5.5, é apresentado o primeiro teste realizado dentro de uma tubulação, no qual o robô demonstrou a capacidade de superar obstáculos com facilidade, mantendo uma comunicação constante com a base de teleoperação. Vale ressaltar que no teste a comunicação sem fio foi eficiente devido a curta distância que o robô estava do computador base. As esteiras funcionaram de maneira satisfatória mesmo em terrenos irregulares, o que ressalta a eficácia do sistema em enfrentar desafios no ambiente de inspeção. É importante destacar que a inspeção da tubulação deve ocorrer quando esta estiver paralisada ou desativada, uma vez que o robô não é à prova d'água.



Figura 5.5: PipeBot realizando inspeções em tubulações de forma teleoperada. Fonte: O autor.

Outro teste conduzido na tubulação com o objetivo de analisar maneiras de gerar mapas 3D envolveu a utilização da Realsense L515, que permite a criação de uma representação em

nuvem de pontos do ambiente a ser inspecionado, como ilustrado na Figura 5.6. Entretanto, essa é uma tarefa que demanda elevado poder de processamento e, portanto, precisa de uma análise mais aprofundada antes de ser incorporada ao robô abordado nesta dissertação. Por essa razão, esse tópico será considerado na seção de Trabalhos Futuros.

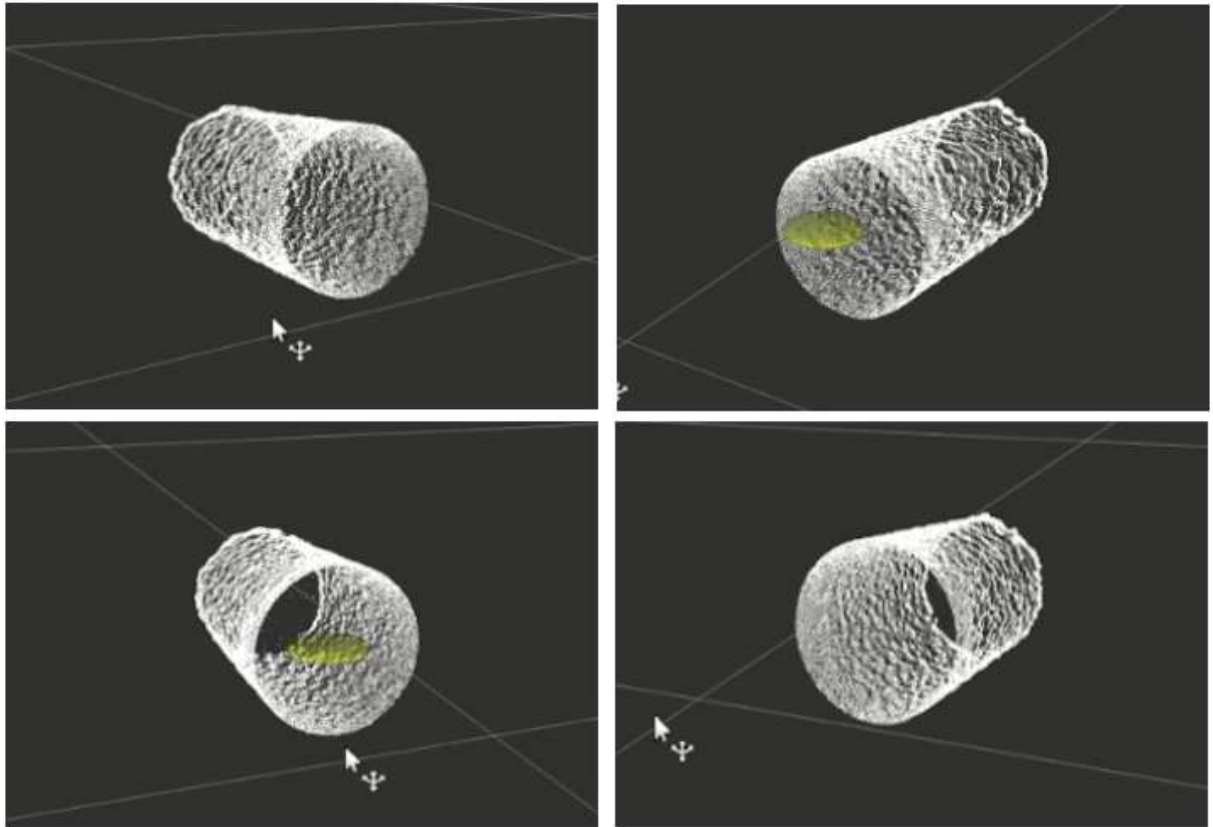


Figura 5.6: Visualização da nuvem de pontos 3D gerada com a câmera Realsense durante um experimento preliminar. Fonte: O autor.

Mesmo que o objetivo principal do robô seja a realização de inspeções em tubulações, é importante destacar que em algumas áreas industriais existem galerias estreitas que também requerem inspeções constantes. A Figura 5.7 ilustra o PipeBot realizando uma inspeção em uma área operacional da Vale no Maranhão. Nesse cenário, o robô foi teleoperado e sua comunicação foi realizada via cabo, demonstrando sua versatilidade e adaptabilidade para diferentes ambientes e condições de inspeção.



Figura 5.7: PipeBot realizando inspeções em galerias de forma teleoperada. Fonte: O autor.

- **Teste do controle de navegação em laboratório**

Com a plataforma validada em operações teleoperadas, foi possível a implementação de soluções mais complexas que dependiam do funcionamento eficaz do robô como um todo. Um exemplo foi a navegação autônoma, uma vez que os testes anteriores revelaram que a comunicação por cabo gerava limitações às capacidades do robô. Essas limitações se tornavam evidentes quando o robô ingressava em ambientes de inspeção, devido ao atrito do cabo com o solo, o que dificultava sua movimentação e aumentava o risco de enrolamento do cabo em superfícies irregulares.

A implementação do sistema de navegação, que utiliza campos vetoriais, produziu resultados satisfatórios nos testes de laboratório. O robô conseguiu seguir com sucesso uma parede lateral designada a uma distância definida, completando autonomamente todo o contorno de um corredor.

O teste realizado no laboratório com o sistema de navegação proposto está ilustrado na Figura 5.8, e os vídeos desses testes estão disponíveis no repositório mencionado anteriormente. Esse avanço representa um passo significativo em direção à capacidade autônoma do robô em ambientes reais. Vale ressaltar que nesse momento não houve atuação da máquina de estados conforme nos experimentos de simulação, no protótipo real foi validada a navegação pelo controle de campos vetoriais.



Figura 5.8: Teste conduzido em laboratório com o robô seguindo a parede utilizando o sistema de navegação proposto. Fonte: O autor.

5.3. Lições aprendidas

O desenvolvimento do dispositivo robótico resultou em uma série de lições aprendidas. Durante o processo de produção do robô, ocorreu interação entre diferentes áreas da robótica, como mecânica, eletrônica, software e controle. Essas lições estão detalhadas nesta seção e serão úteis para auxiliar futuros projetos do Instituto Tecnológico Vale.

A utilização da tecnologia de impressão 3D para fabricar todas as peças do robô resultou em um desenvolvimento rápido e econômico do protótipo. Todas as melhorias propostas puderam ser prontamente validadas no robô, eliminando a necessidade de aguardar longos períodos para a produção de novas peças. Um exemplo foi a produção do módulo de suporte, que foi produzido cinco vezes até alcançar o modelo final. Isso se deu tanto aos ajustes necessários quanto aos testes de esforços mecânicos que indicaram pontos de melhorias.

Em relação ao *software*, a reutilização de pacotes ROS criados por outros autores para utilização no robô deste projeto foi outro ponto que acelerou o processo de criação do protótipo, como por exemplo o pacote de controle por campos vetoriais proposto por Rezende *et al.* (2021). Outro fato que vale ser destacado é que a experiência gerada pelo desenvolvimento do EspeleoRobô, auxiliou em tomadas de decisões que evitaram alguns problemas, como por exemplo a modularização do *software* para controlar separadamente um conjunto de componentes.

A integração dos componentes eletrônicos inicialmente foi realizada externamente ao robô, o que facilitou a criação de cabos e conexões. Além disso, por esses componentes não estarem encapsulados, foi possível realizar testes de tensão em cada ponto do circuito. Um exemplo que destaca a importância de testar até mesmo componentes consolidados ocorreu com a porta USB. Durante testes em ambientes externos, houve ocorrência de erros nos motores do robô em momentos de impacto. Esses erros eram causados por problemas de mau contato entre o cabo USB de comunicação entre o controlador do motor e a Raspberry Pi.

A realização de testes fora de um ambiente controlado, como o laboratório, é essencial para identificar possíveis problemas que o robô poderia enfrentar ao ser exposto a um ambiente de produção. Outro exemplo ocorreu em relação às esteiras, que não estavam em contato suficiente com o chão para gerar tração no robô. Isso aconteceu devido a componentes impressos na estrutura da esteira que estavam arrastando no chão. No entanto, esse problema passou despercebido no laboratório.

Durante o teste de campo teleoperado de inspeção nas galerias da Vale no Maranhão, foi observada uma quantidade significativa de areia no ambiente, o que levou à sua entrada no sistema de locomoção e resultou em um esforço excessivo nos motores. Esse esforço foi tão intenso que chegou a impedir o funcionamento do motor. Foi identificado que a areia entrou no módulo de locomoção devido ao movimento das esteiras, mas não conseguiu sair devido ao sistema ser fechado. Nesse contexto, para evitar o acúmulo excessivo de areia, seria possível aprimorar o sistema de locomoção, aumentando o espaço entre as placas e a esteira, ou incor-

porando pontos vazados na placa de apoio. Outra alternativa viável seria considerar o uso de outros mecanismos de locomoção, como rodas.

Outro exemplo de problema que não ocorreu em um ambiente controlado foi o superaquecimento da Raspberry Pi, que atingiu sua temperatura máxima. Isso levou o processador a entrar em um estado de “throttling”, limitando sua capacidade de processamento e causando travamentos no sistema do robô, impossibilitando a execução da tarefa de inspeção. Portanto, a adição de um dissipador de calor ajudou a resolver esse problema.

Durante a fase final do desenvolvimento do robô, houve uma tentativa de realizar o mapeamento da área inspecionada. Uma das abordagens foi a utilização do pacote ROS GMapping, no entanto, fazer isso de forma online se mostrou inviável devido à capacidade de processamento limitada da Raspberry Pi, que não suportava a execução dessa tarefa em conjunto com o sistema do robô. Isso ressalta ainda mais a importância de um sistema de navegação que não dependa nem do mapa nem da localização do robô.

Uma tentativa foi salvar os dados da inspeção e realizar a geração do mapa de maneira offline em um computador mais potente. Essa metodologia se mostrou viável até o momento em que se tornou evidente a necessidade de estimar a posição do robô não apenas pelo laser, mas também por meio de uma estimativa de odometria do robô, a qual ainda não estava desenvolvida. Portanto, devido à necessidade de implementar uma fusão de sensores para gerar dados de odometria e à falta de capacidade de processamento, até o momento não foi possível realizar a geração online dos mapas nos locais inspecionados. Isso ressalta ainda mais a importância de um sistema de navegação que não dependa nem do mapa nem da localização do robô.

6. Conclusão

O desenvolvimento descrito nesta dissertação resultou no desenvolvimento de um robô funcional capaz de realizar inspeções em dutos com diâmetro de 14 polegadas ou mais. A produção desse protótipo envolveu a montagem da estrutura mecânica, a conexão eletrônica de todos os componentes de *hardware* e a implementação do *software* embarcado.

Além disso, esta dissertação apresentou uma estratégia de navegação autônoma em sistemas de dutos comuns na indústria de mineração, a qual foi validada por meio de simulações e testes em laboratório. A solução desenvolvida se mostrou altamente adaptável para diferentes cenários, pois permite a integração de novas funcionalidades eletrônicas, mecânicas e de software, economicamente viável, de fácil implementação e não necessita de localização precisa ou informações prévias sobre o ambiente. O uso de máquina de estados e campos vetoriais permitiu que o robô navegasse de forma funcional.

Foram conduzidos experimentos com o robô real em ambientes controlados, e os resultados obtidos demonstraram ser promissores, confirmando o sucesso do protótipo. Uma conquista significativa é que a premissa do projeto de desenvolver um robô de baixo custo foi alcançada, com um custo de produção estimado em aproximadamente US\$ 5 mil dólares. Isso é especialmente relevante quando consideramos que plataformas robóticas comerciais podem ultrapassar a marca dos US\$ 100 mil dólares em custos.

6.1. Trabalhos futuros

A consolidação do robô como uma plataforma de inspeção móvel confiável é o principal objetivo desta qualificação de dissertação. As seções a seguir apresentam propostas do que pode ser melhorado e alcançado com a continuação deste projeto.

Começando pelas melhorias mecânicas, é possível aperfeiçoar o módulo de locomoção adicionando um rolamento de apoio na polia motora para garantir uma tração mais uniforme na correia. Além disso, é necessário repensar o projeto para tornar o robô à prova d'água, especialmente na junção entre as partes impressas em 3D e os componentes usinados. Uma extensão no comprimento do corpo do robô também pode ser considerada para acomodar melhor os componentes eletrônicos.

No que diz respeito à parte eletrônica, é viável desenvolver uma placa de circuito impressa para consolidar os componentes eletrônicos e reduzir a quantidade de fios no robô. A utilização de cabos USB do tipo C com pontas em 90 graus pode otimizar o espaço interno, especialmente onde a câmera Realsense está fixada. Além disso, a pesquisa por componentes de menor tamanho, como o conversor DC-DC, pode facilitar o processo de montagem.

A integração de algoritmos de SLAM (*Simultaneous Localization and Mapping*) é fundamental para aprimorar a capacidade do robô de gerar mapas dos locais inspecionados e facilitar a detecção de problemas nas tubulações. O Gmapping é uma opção para criar mapas 2D, enquanto o RTAB-Map trabalha com mapas 3D, embora com maior demanda computacional. É importante pensar em estratégias de como executar esses algoritmos já que a capacidade de processamento da Raspberry Pi não é muito grande. Alternativas como a gravação dos dados para processamento posterior ou otimização de processos na Raspberry podem ser uma solução.

No contexto da utilização de algoritmos SLAM, é uma estratégia que auxilia no processo de identificação de anomalias durante a inspeção visual. Essa estratégia de mapeamento e localização simultâneos, oferece a capacidade de reconstruir um mapa tridimensional detalhado do ambiente, podendo destacar as posições aproximadas das anomalias detectadas. A integração do RTABMAP, por exemplo, pode permitir uma recuperação precisa dessas posições, além de associar informações visuais, temporais e espaciais. Isso capacita operadores ou sistemas automatizados a realizar análises mais abrangentes.

Além disso, com o processamento da Raspberry sendo limitado, nem todas as técnicas de navegação podem ser utilizadas no robô de inspeção. Nesse contexto, testar outros algoritmos simples e eficiente, além do que foi feito nessa dissertação, ou até mesmo testar outros *hardware* de baixo custo, pode gerar a possibilidade de inserir novas funcionalidades autônomas.

Esses trabalhos futuros visam aperfeiçoar diversos aspectos do robô de inspeção, desde suas capacidades mecânicas e eletrônicas até a integração de algoritmos avançados de SLAM e controle, proporcionando um robô mais eficaz e autônomo para inspeções em tubulações.

Referências bibliográficas

- AEROPIC. “**Robô de inspeção Ap-02**”. <https://www.aeropic.com.br/robotics>, 2022. Acesso em: 2022-10-05.
- AMARAL, I., FANY, C., PEREIRA, G., et al.. “**Estratégias de Navegação para a Inspeção Autônoma de Galerias**”. Em: **Anais do Congresso Brasileiro de Automática 2022**. sbabra, dez. 2022.
- AZPÚRUA, H., REZENDE, A., POTJE, G., et al.. “**Towards semi-autonomous robotic inspection and mapping in confined spaces with the EspeleoRobô**”, **Journal of Intelligent & Robotic Systems**, v. 101, n. 4, pp. 1–27, 2021.
- BIBER, P., WEISS, U., DORNA, M., et al.. “**Navigation system of the autonomous agricultural robot BoniRob**”. Em: **Workshop on Agricultural Robotics: Enabling Safe, Efficient, and Affordable Robots for Food Production (Collocated with IROS 2012)**, Vilamoura, Portugal, 2012.
- BÖRGER, E. **The abstract state machines method for high-level system design and analysis**. Springer, 2010.
- CISZEWSKI, M., MITKA, Ł., BURATOWSKI, T., et al.. “**Modeling and simulation of a tracked mobile inspection robot in MATLAB and V-REP software**”, **Journal of Automation, Mobile Robotics and Intelligent Systems**, pp. 5–11, 2017.
- COPPELIASIM. “**Robot Simulator CoppeliaSim**”. <https://www.coppeliarobotics.com/>, 2023. Acesso em: 2023-08-01.
- DE BARROS, L. G., DE OLIVEIRA TEOTÔNIO, R., ALVES, P. H., et al.. “**Dispositivo robótico para inspeção de tubos**”. Jun 2021a. **Patent INPI, BR 10 2021 010685 9**, Jun. 2021.
- DE BARROS, L. G. D., TEOTÔNIO, R., ALVES, P., et al.. “**A robotic device for inspecting small diameter pipes**”. Em: **Proceedings of the 26th International Congress of Mechanical Engineering**. ABCM, 2021b.

- DESAI, A., SAHA, I., YANG, J., et al.. **“DRONA: a framework for safe distributed mobile robotics”**. Em: **Proceedings of the 8th International Conference on Cyber-Physical Systems**, pp. 239–248, 2017.
- DOS SANTOS, P. H., FILHO, A. M. S. T., DE REZENDE FREITAS, E. J. **“Desenvolvimento de robôs diferenciais futebolistas: uma aplicação de campos vetoriais no planejamento de caminho”**. Em: **ANAIS DO 14º SIMPÓSIO BRASILEIRO DE AUTOMAÇÃO INTELIGENTE**, Ouro Preto, 2019. Galoá. Disponível em: <<https://proceedings.science/sbai-2019/trabalhos/desenvolvimento-de-robos-diferenciais-futebolistas-uma-aplicacao-de-campos-vetor?lang=pt-br>>. Acesso em: 01 out. 2023.
- EDDYFI. **“VTI Vertical Crawler”**. <https://eddyfi.com/en/product/tritrax-vertical-pipe-inspection-crawler>, 2022. Acesso em: 2022-10-05.
- FOUKARAKIS, M., LEONIDIS, A., ANTONA, M., et al.. **“Combining Finite State Machine and Decision-Making Tools for Adaptable Robot Behavior”**. Em: **Universal Access in Human-Computer Interaction. Aging and Assistive Environments**, Springer International Publishing, pp. 625–635, Cham, 2014. doi: 10.1007/978-3-319-07446-7_60. Disponível em: <https://doi.org/10.1007/978-3-319-07446-7_60>.
- FREITAS, G. M., MARTINS, A., MIRANDA, V. R. F., et al.. **Segundo relatório técnico de acompanhamento do projeto ‘Dispositivo Robótico de Inspeção de Ambientes Restritos e Confinados’**. Relatório técnico, Universidade Federal de Minas Gerais e Instituto Tecnológico Vale, 2020.
- JETTY ROBOT. **“JettyRobot 6”**. <https://www.jettyrobot.com/our-robots/jettyrobot-6>, 2023. Acesso em: 25 de junho de 2023.
- JIE, M. S., BAEK, J. H., HONG, Y. S., et al.. **“Real time obstacle avoidance for mobile robot using limit-cycle and vector field method”**. Em: **Knowledge-Based Intelligent Information and Engineering Systems: 10th International Conference, KES 2006, Bournemouth, UK, October 9-11, 2006. Proceedings, Part I 10**, pp. 866–873. Springer, 2006.
- KAZEM, B. I., HAMAD, A. H., MOZAEI, M. M. **“Modified vector field histogram with a neural network learning model for mobile robot path planning and obstacle avoidance.”** *Int. J. Adv. Comp. Techn.*, v. 2, n. 5, pp. 166–173, 2010.
- MARTZ, J., AL-SABBAN, W., SMITH, R. N. **“Survey of unmanned subterranean exploration, navigation, and localisation”**, *IET Cyber-Systems and Robotics*, v. 2, n. 1, pp. 1–13, 2020.

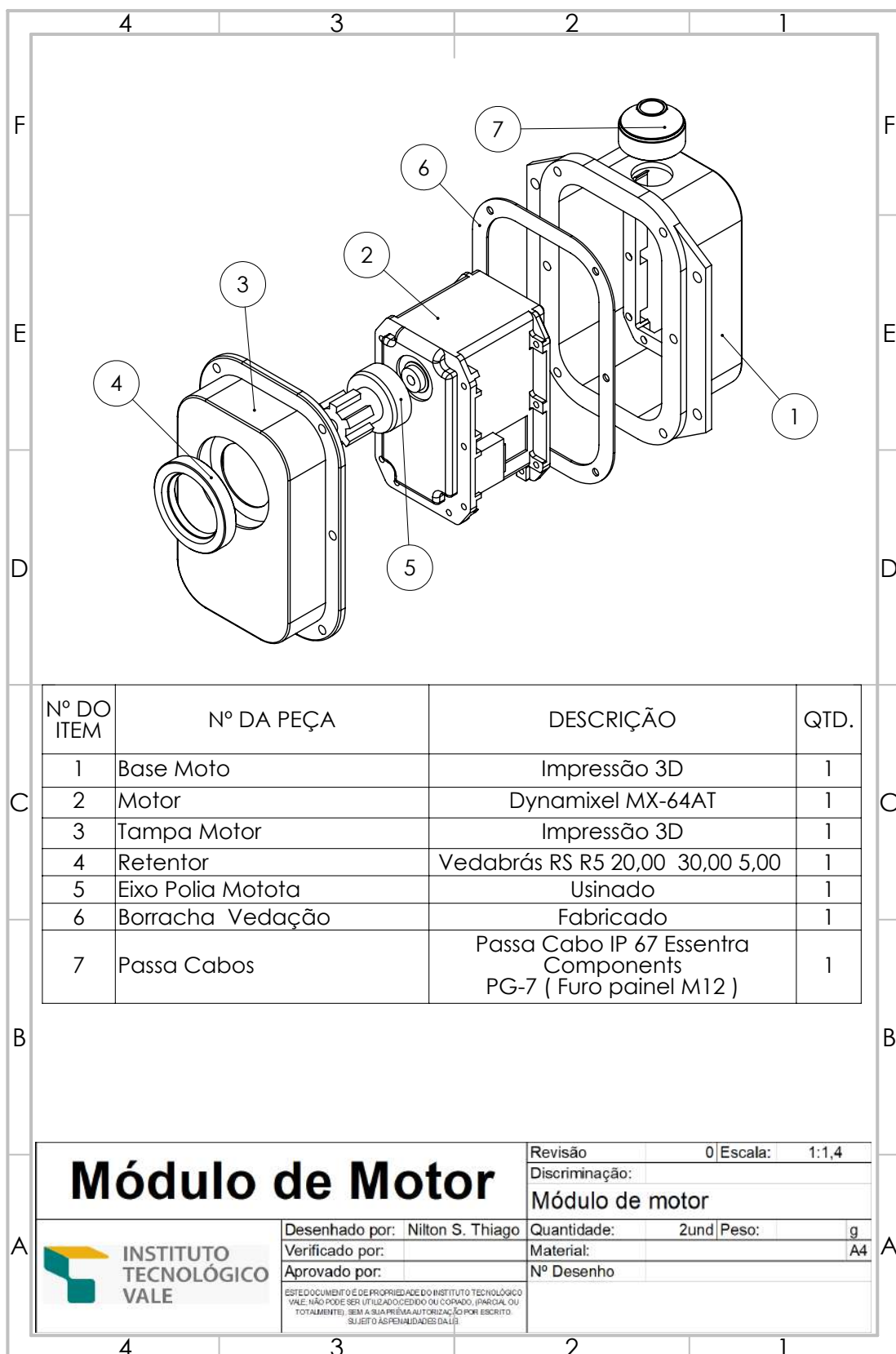
- NASSIRAEI, A. A., KAWAMURA, Y., AHRARY, A., et al.. **“Concept and design of a fully autonomous sewer pipe inspection mobile robot”kantaró”**. Em: **Proceedings 2007 IEEE International Conference on Robotics and Automation**, pp. 136–143. IEEE, 2007.
- NELSON, D. R., BARBER, D. B., MCLAIN, T. W., et al.. **“Vector field path following for miniature air vehicles”**, **IEEE Transactions on Robotics**, v. 23, n. 3, pp. 519–529, 2007.
- OGAI, H., BHATTACHARYA, B. **Pipe inspection robots for structural health and condition monitoring**. Springer, 2018.
- PEREIRA, G., DUARTE, C., MARQUES, D., et al.. **“Towards a Simple Navigation Strategy for Autonomous Inspection of Ducts and Galleries”**. Em: **2021 Latin American Robotics Symposium (LARS), 2021 Brazilian Symposium on Robotics (SBR), and 2021 Workshop on Robotics in Education (WRE)**. IEEE, out. 2021.
- QU, Y., DURDEVIC, P., YANG, Z. **“Smart-spider: Autonomous self-driven in-line robot for versatile pipeline inspection”**, **Ifac-papersonline**, v. 51, n. 8, pp. 251–256, 2018.
- QUIGLEY, M., CONLEY, K., GERKEY, B., et al.. **“ROS: an open-source Robot Operating System”**. Em: **ICRA workshop on open source software**, v. 3, p. 5. Kobe, Japan, 2009.
- REZENDE, A. M. C., GONCALVES, V. M., PIMENTA, L. C. A. **“Constructive Time-Varying Vector Fields for Robot Navigation”**, **IEEE Transactions on Robotics**, v. 38, n. 2, pp. 852–867, abr. 2021.
- ROCHA, R. P. **“Estado da Arte da Robótica móvel em Portugal”**, Instituto de Sistemas e Robótica, Coimbra, Portugal. URL: <http://www.dee.uc.pt/~rprocha>, 2000.
- RODRIGUES, D. P. **“Técnicas de Navegação”**. 2010.
- ROHMER, E., SINGH, S. P. N., FREESE, M. **“CoppeliaSim (formerly V-REP): a Versatile and Scalable Robot Simulation Framework”**. Em: **Proc. of The International Conference on Intelligent Robots and Systems (IROS)**, 2013. www.coppeliarobotics.com.
- ROS. **“About ROS”**. <https://www.ros.org/about-ros/>, 2019. Acesso em: 2022-10-05.
- SALES, D. O., CORREA, D. O., FERNANDES, L. C., et al.. **“Adaptive finite state machine based visual autonomous navigation system”**, **Engineering Applications of Artificial Intelligence**, v. 29, pp. 152–162, 2014.

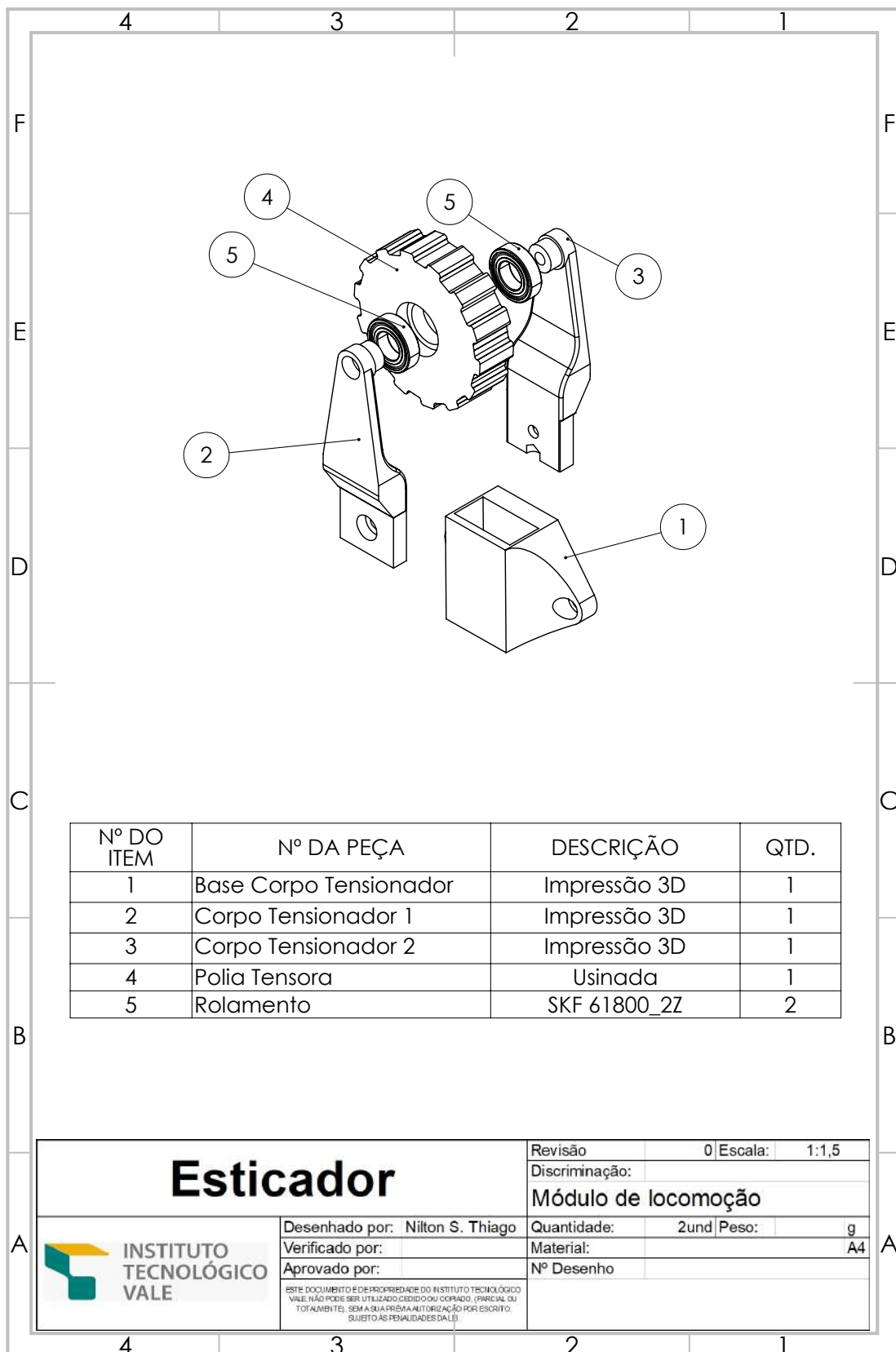
- SCHNEIDER, F. B. **“The state machine approach: A tutorial”**, **Fault-tolerant distributed computing**, pp. 18–41, 2005.
- SCHOLLMEIER, R. **“A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications”**. Em: **Proceedings First International Conference on Peer-to-Peer Computing**, pp. 101–102. IEEE, 2001.
- SHAPOVALOV, D., PEREIRA, G. A. S. **“Exploration of unknown environments with a tethered mobile robot”**. Em: **2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)**, pp. 6826–6831, 2020.
- SICILIANO, B., KHATIB, O. **Springer handbook of robotics**. Springer, 2016.
- SICILIANO, B., SCIAVICCO, L., VILLANI, L., et al.. **Force control**. Springer, 2009.
- SICILIANO, B., SCIAVICCO, L., VILLANI, L., et al.. **Robotics: modelling, planning and control**. Springer Science & Business Media, 2010.
- SLAMTEC. **“RPLidar S1”**. <https://www.slamtec.com/en/Lidar/S1>, 2023. Acesso em: 30 de junho de 2023.
- SUN, Y., MA, L., MORRIS, J. **“A practical approach for reliability prediction of pipeline systems”**, **European Journal of Operational Research**, v. 198, n. 1, pp. 210–214, 2009.
- TEOTÔNIO, R. **Projeto de um dispositivo robótico para inspeção de tubulações de diâmetros reduzidos**. Relatório técnico, Instituto Tecnológico Vale, 2021.
- ZENKEVICH, S. L., NAZAROVA, A. V., ZHU, H. **“Logical control a group of mobile robots”**, **Smart Electromechanical Systems: Group Interaction**, pp. 31–43, 2019.

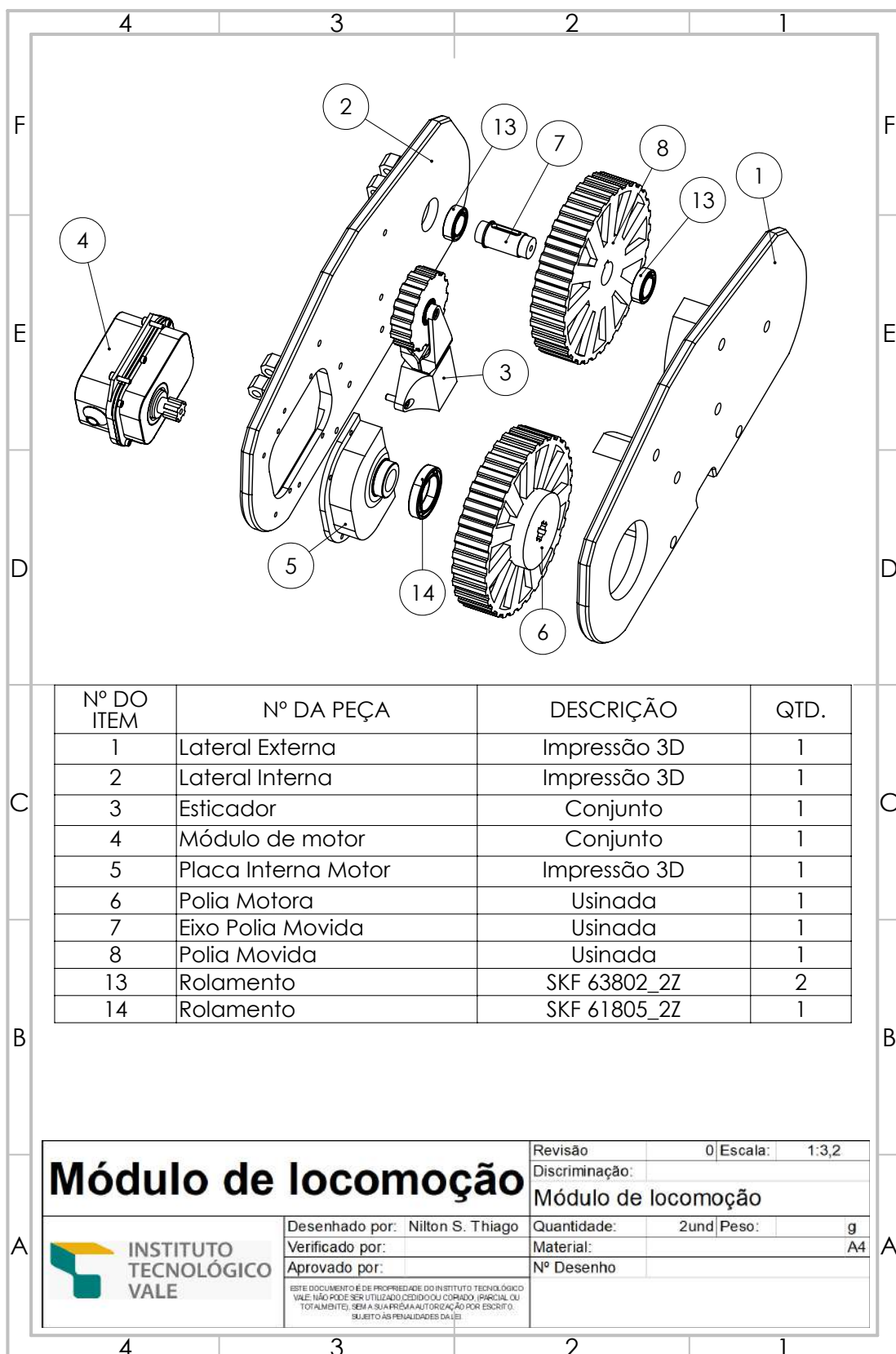
Apêndices

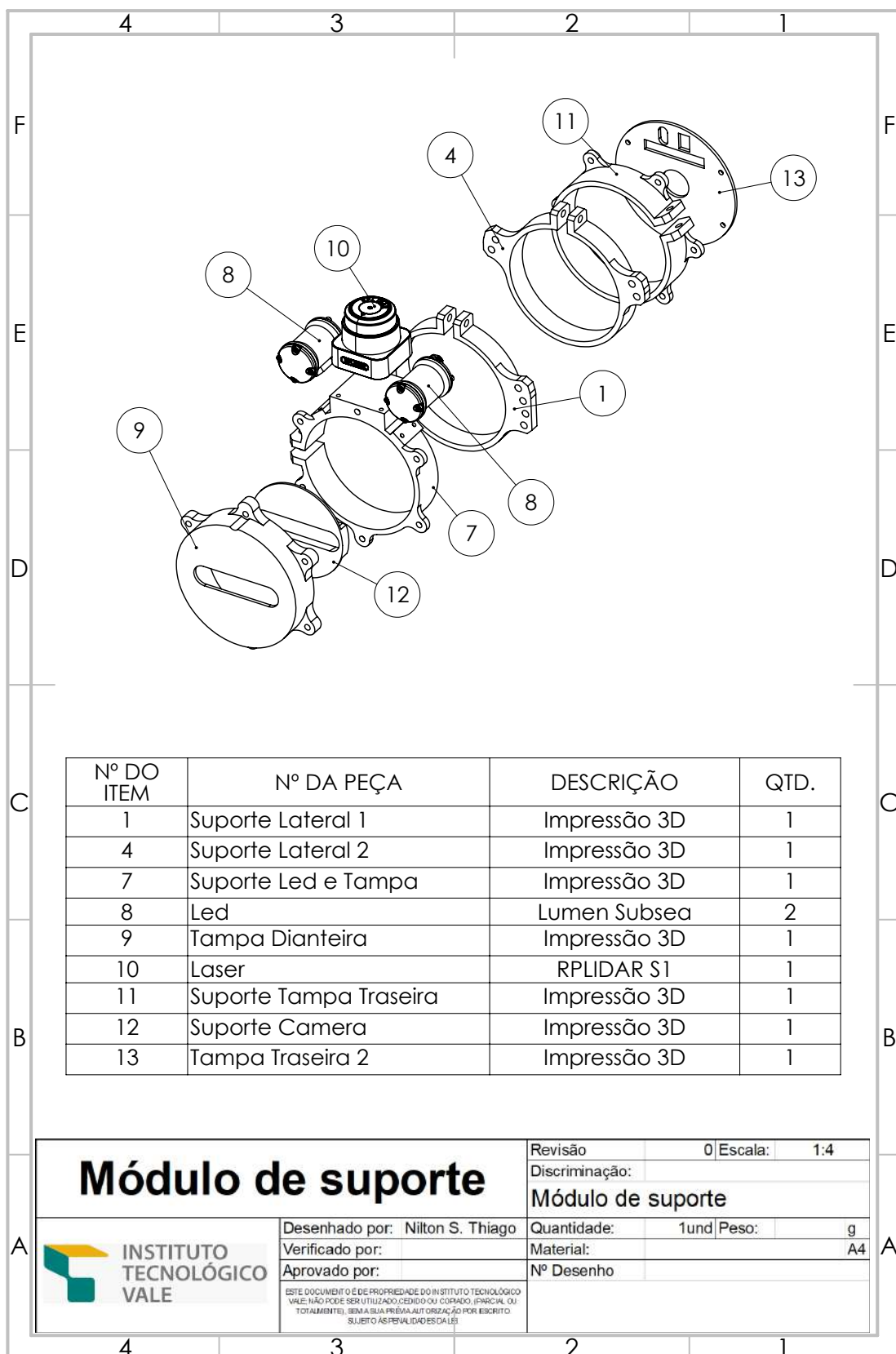
A. Projeto mecânico do PipeBot

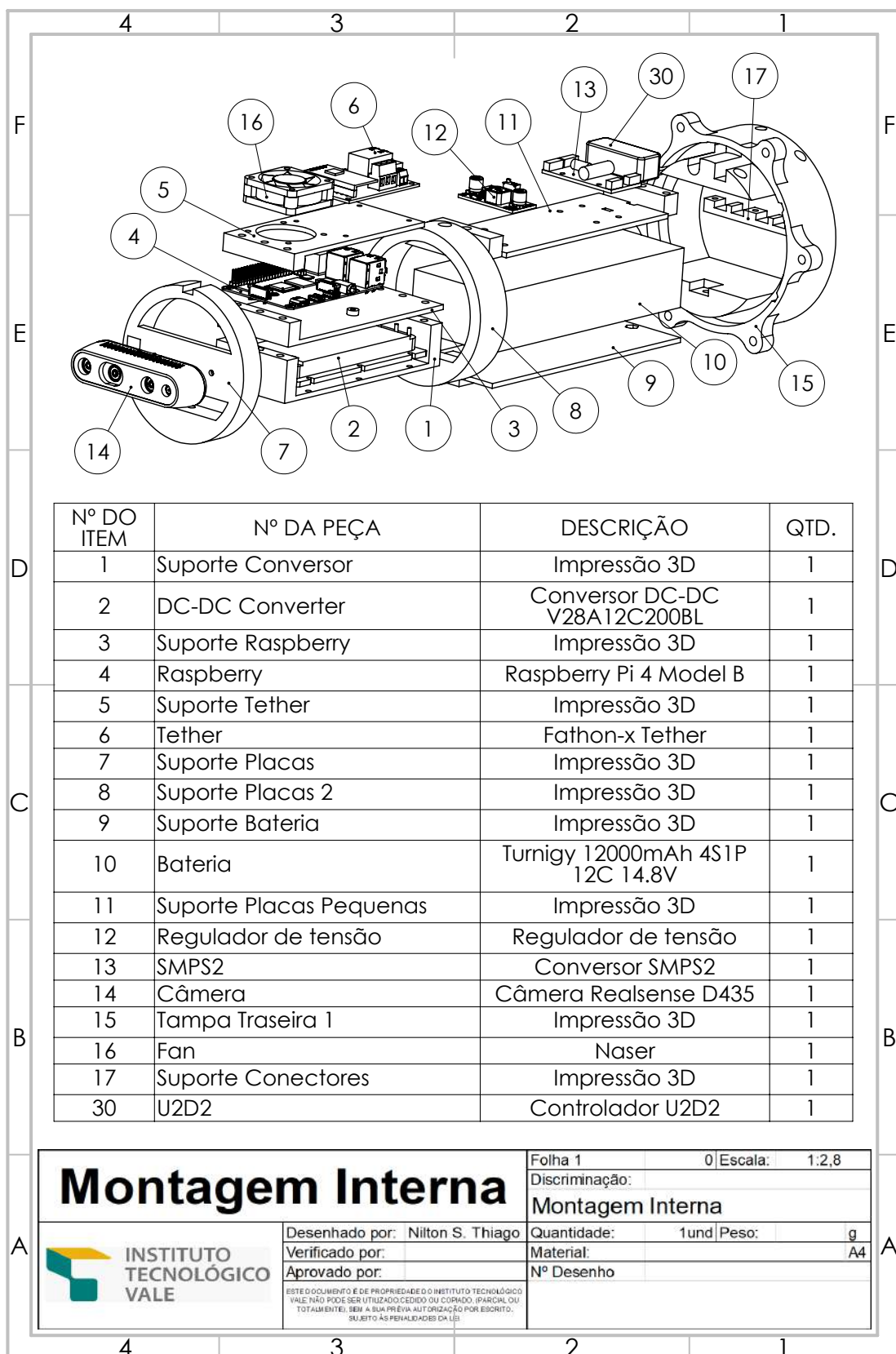
Este apêndice contém o projeto mecânico mais recente, que representa uma atualização da modelagem 3D do robô. Durante o desenvolvimento desta dissertação, foram implementadas diversas melhorias e correções, resultando em significativas alterações em relação ao que foi originalmente descrito no relatório técnico (TEOTÔNIO, 2021).

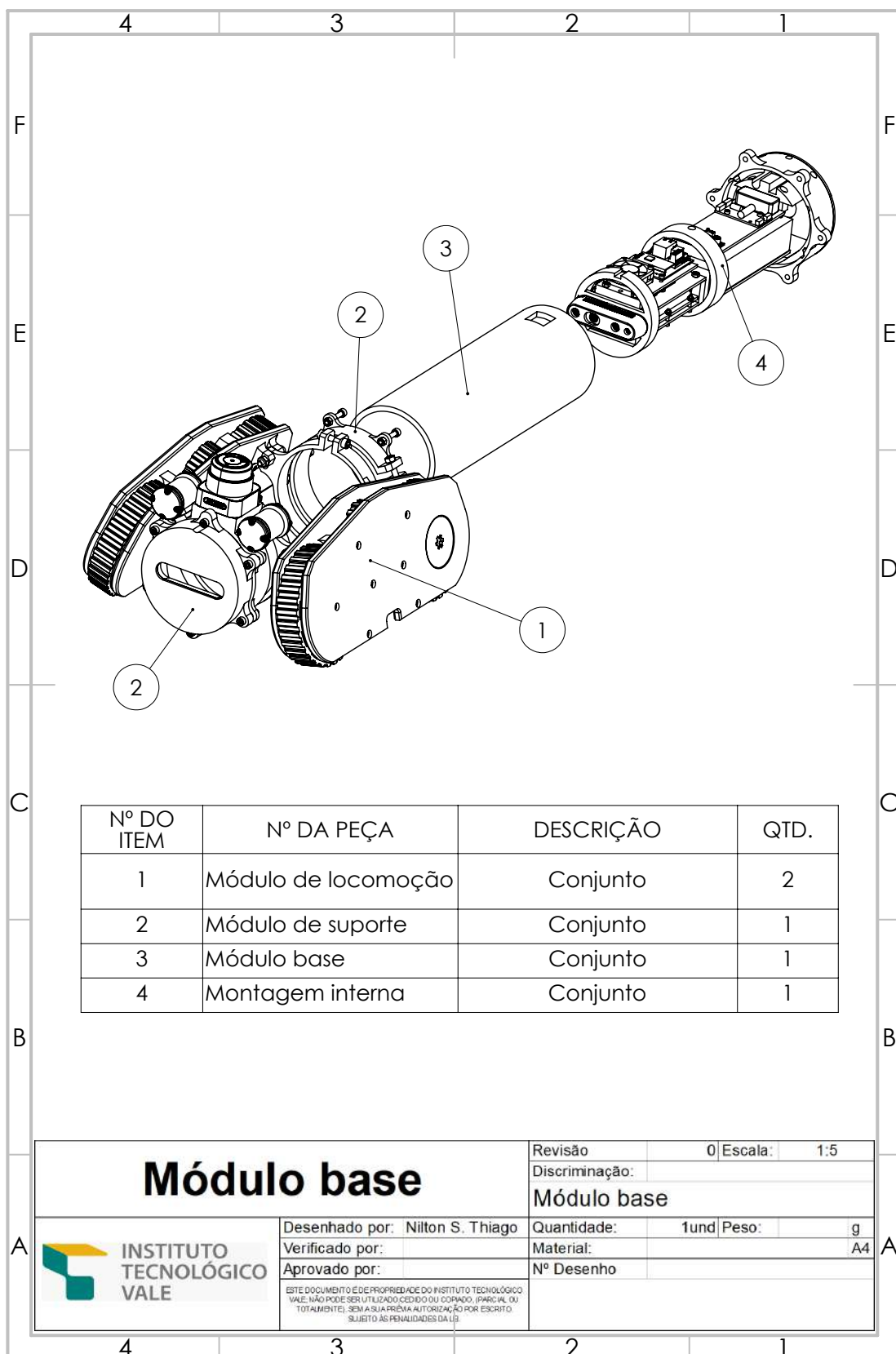












B. Projeto eletrônico do PipeBot

Neste apêndice, está disponível o projeto eletrônico mais atual, o qual constitui uma revisão de alguns componentes do robô. No decorrer da elaboração desta dissertação, foram realizadas algumas modificações em comparação com o que foi previamente definido no documento técnico (TEOTÔNIO, 2021).

