

João Aristeu Seixas Cardoso

**Desenvolvimento de um robô móvel do tipo
Car-Like para realização de caminhos
pré-definidos**

Manaus - AM

2024

João Aristeu Seixas Cardoso

Desenvolvimento de um robô móvel do tipo *Car-Like* para realização de caminhos pré-definidos

Trabalho de Conclusão de Curso apresentado à banca avaliadora do curso de Engenharia de Controle e Automação da Escola Superior de Tecnologia da Universidade do Estado do Amazonas como pré-requisito para obtenção do título de Engenheiro de Controle e Automação.

Universidade do Estado do Amazonas – UEA

Escola Superior de Tecnologia – EST

Engenharia de Controle e Automação

Orientador: Rodrigo Farias Araújo

Manaus - AM

2024

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).
Sistema Integrado de Bibliotecas da Universidade do Estado do Amazonas.

J62dd Seixas Cardoso, João Aristeu
Desenvolvimento de um robô móvel do tipo Car-Like para realização de caminhos pré-definidos / João Aristeu Seixas Cardoso. Manaus : [s.n], 2024.
82 f.: color.; 297 cm.

TCC - Graduação em Engenharia de Controle e Automação; - Universidade do Estado do Amazonas, Manaus, 2024.
Inclui bibliografia
Orientador: Rodrigo Farias Araújo

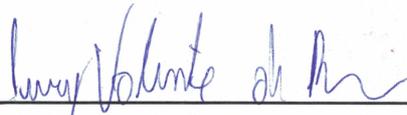
1. Robótica móvel. 2. Teoria de controle. 3. Rastreamento de caminho. 4. ROS. 5. Modelagem 3D.
I. Rodrigo Farias Araújo (Orient.). II. Universidade do Estado do Amazonas. III. Desenvolvimento de um robô móvel do tipo Car-Like para realização de caminhos pré definidos

João Aristeu Seixas Cardoso

Desenvolvimento de um robô móvel do tipo *Car-Like* para realização de caminhos pré-definidos

Trabalho de Conclusão de Curso apresentado à banca avaliadora do curso de Engenharia de Controle e Automação da Escola Superior de Tecnologia da Universidade do Estado do Amazonas como pré-requisito para obtenção do título de Engenheiro de Controle e Automação.

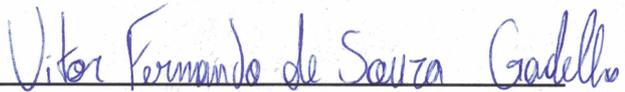
Trabalho aprovado. Manaus - AM, 27 de fevereiro de 2024:



Prof. Dr. Iury Valente de Bessa
Universidade Federal do Amazonas (UFAM)



**Prof. Dr. Luiz Alberto Queiroz
Cordovil Júnior**
Sidia/Samsung e Universidade do Estado do
Amazonas (UEA)



**Prof. Esp. Vitor Fernando de Souza
Gadelha**
Instituto Federal do Amazonas (IFAM)

Manaus - AM

2024

Agradecimentos

Primeiramente à Deus pelo dom da vida, por me guiar e me guardar em todos os momentos.

À minha família, e em especial aos meus pais, Ieda e Jucelino, expresso profunda gratidão pelo constante incentivo ao longo de toda a minha jornada na universidade. Agradeço também pelo amor incondicional e carinho que permeiam cada etapa da minha vida.

Aos amigos que fiz durante a graduação e aos que já conhecia antes. Essa caminhada foi de muito aprendizado e momentos marcantes graças a vocês.

Ao meu orientador, Prof. Dr. Rodrigo Farias Araújo, pelo conhecimento repassado nos tempos trabalhando juntos, nas matérias e durante a jornada de elaboração deste trabalho.

A todos os professores do curso de Engenharia de Controle e Automação que de alguma forma contribuíram para a minha formação.

Aos laboratórios e projetos dos quais fiz parte, pois possibilitaram que eu pudesse me desenvolver profissionalmente e pessoalmente.

Por fim, à Universidade do Estado do Amazonas, que foi minha segunda casa, por possibilitar a realização deste curso e de todas as experiências vividas.

Resumo

O desenvolvimento de robôs móveis do tipo *Car-Like* abrange a criação de veículos autônomos que emulam o movimento de carros convencionais. Embora as aplicações para esse tipo de robô sejam diversas, este trabalho concentra-se no desafio do rastreamento de caminho. O objetivo é posicionar o robô em um ponto inicial próximo de um caminho definido por uma função não dependente do tempo, $f(x)$. Através de um controle lateral, pretende-se que o robô se posicione sobre o caminho e o siga. O desenvolvimento do controlador para estabilizar o robô sobre o caminho adotará uma abordagem utilizando Desigualdades Matriciais Lineares (LMI). Além disso, o projeto visa desenvolver tanto o *hardware* quanto o *software* de um protótipo em escala reduzida de um robô móvel do tipo *Car-Like*. No que diz respeito ao *hardware*, o processo envolve modelagem 3D através de *software* CAD, impressão 3D das peças, e a concepção e fabricação de uma placa de circuito impresso para capturar dados dos sensores e acionar os atuadores do sistema. No âmbito do *software*, a implementação do controlador será realizada em um computador com o *Robot Operating System* (ROS). Dessa forma, o microcontrolador instalado no robô ficará responsável apenas pela comunicação com o computador via TCP/IP, execução da leitura dos sensores e comando dos atuadores.

Palavras-chave: modelagem 3D, teoria de controle, rastreamento de caminho, robótica móvel, ROS.

Abstract

The development of Car-Like mobile robots encompasses the creation of autonomous vehicles that emulate the motion of conventional cars. Although the applications for such robots are diverse, this work focuses on the challenge of path following. The goal is to position the robot near an initial point on a path defined by a time-independent function, $f(x)$. Through lateral control, the intention is for the robot to align itself with the path and follow it. The development of the controller to stabilize the robot on the path will adopt an approach using Linear Matrix Inequalities (LMI). Additionally, the project aims to develop both the hardware and software for a scaled-down prototype of a Car-Like mobile robot. Regarding the hardware, the process involves 3D modeling using CAD software, 3D printing of parts, and the design and manufacturing of a printed circuit board to capture sensors data and activate the system's actuators. In the realm of software, the controller's implementation will be on a computer with the Robot Operating System (ROS). Consequently, the microcontroller installed on the robot will be responsible only for communication with the computer via TCP/IP, sensors readings and actuators commands.

Keywords: 3D modelling, control theory, path following, mobile robotics, ROS.

Lista de ilustrações

Figura 1 – Rodas Omnidirecionais.	15
Figura 2 – Robô móvel omnidirecional (Palm-Pilot Robot Kit).	16
Figura 3 – Robô do tipo uniclo (Pioneer 2).	16
Figura 4 – Robô do tipo Car-Like (Berkeley Autonomous Race Car).	17
Figura 5 – Posição ideal do centro de giro de um veículo.	18
Figura 6 – Diferença de ângulos resultante da geometria trapezoidal.	18
Figura 7 – Esquemático de um robô <i>Car-Like</i>	19
Figura 8 – Diagrama esquemático de um modelo tipo bicicleta.	20
Figura 9 – Tarefas do sistema de controle de movimento.	22
Figura 10 – Diagrama esquemático de um modelo de rastreamento de caminho.	23
Figura 11 – Diagrama de um servomecanismo de tempo contínuo.	24
Figura 12 – Diagrama de blocos de um sistema amostrado de controle.	25
Figura 13 – Representação de sinal.	26
Figura 14 – Diferentes representações em espaço de estados do mesmo sistema.	27
Figura 15 – Politopo com cinco vértices.	30
Figura 16 – Processo de Modelagem por Fusão e Deposição (FDM).	34
Figura 17 – (a) Rolos de filamentos de cores diversas; e (b) Impressora 3D do tipo FDM modelo Original Prusa MK4.	34
Figura 18 – Exemplos de temperaturas para os materiais <i>PLA</i> , <i>ABS</i> , <i>TPU Flex</i> e <i>PETG</i>	35
Figura 19 – Esquema de corte a laser.	35
Figura 20 – <i>Encoder</i> óptico conectado ao eixo rotativo de um atuador.	38
Figura 21 – Sinais gerados por um <i>encoder</i> de quadratura.	38
Figura 22 – Funcionamento do <i>encoder</i> absoluto.	39
Figura 23 – Ilustração do princípio básico da segunda lei de <i>Newton</i> em duas dimensões, em que uma massa m está presa a molas num referencial.	40
Figura 24 – Ilustração do princípio básico de um giroscópio e como as rotações (Φ , Θ , Ψ) ocorrem ao redor do respectivo eixo.	42
Figura 25 – Circuito Ponte H. (a) Situação para o motor ficar parado; (b) Situação para o motor girar em um sentido; (c) Situação para o motor girar no sentido contrário ao de (b)	43
Figura 26 – Ponte H L298N.	43
Figura 27 – Sinais PWM (a) Sinal PWM; (b) Funcionamento do PWM em 50%; (c) Funcionamento do PWM em 30%	44
Figura 28 – Construção de um motor CC.	45
Figura 29 – Servomotor DS3218 PRO.	45

Figura 30 – Estrutura básica de um microcontrolador.	46
Figura 31 – Pinagem do ESP32.	47
Figura 32 – Robô móvel utilizado neste trabalho.	49
Figura 33 – Esboço do chassi e do sistema de direção.	50
Figura 34 – Sistema de direção desenvolvido.	51
Figura 35 – Haste de direção.	51
Figura 36 – Peças feitas com acrílico.	52
Figura 37 – Chassi genérico de um Car-Like.	53
Figura 38 – Componentes do diferencial mecânico.	53
Figura 39 – Sistema de tração com diferencial.	54
Figura 40 – Esquema elétrico do RMR <i>Car-Like</i>	55
Figura 41 – PCI projetada no <i>Altium Designer</i>	56
Figura 42 – PCI após o término da fabricação.	56
Figura 43 – Diagrama esquemático de um modelo tipo bicicleta com deslizamento.	59
Figura 44 – Conversão de um robô de quatro rodas no modelo triciclo.	61
Figura 45 – Funções de probabilidade de uma medição hipotética.	63
Figura 46 – Distribuição ROS Noetic.	64
Figura 47 – Diagrama de comunicação entre ESP32 e ROS.	64
Figura 48 – Diagrama dos tópicos utilizados pelo ESP32.	65
Figura 49 – Diagrama dos tópicos utilizados pelo código <code>control_node.cpp</code>	65
Figura 50 – Diagrama do algoritmo de controle lateral em <code>control_node.cpp</code>	66
Figura 51 – Diagrama do algoritmo implementado no ESP32.	66
Figura 52 – Resultado da simulação do teste prático I.	69
Figura 53 – Resultado da simulação do teste prático II.	69
Figura 54 – Resultado da simulação do teste prático III.	70
Figura 55 – Resultado da posição do robô e sinal de controle para o teste prático I.	71
Figura 56 – Resultado do erro de orientação do robô para o teste prático I.	71
Figura 57 – Resultado da velocidade das rodas durante o Teste I.	72
Figura 58 – Resultado da posição do robô e o sinal de controle para o teste prático II.	73
Figura 59 – Resultado do erro de orientação do robô para o teste prático II.	74
Figura 60 – Resultado da velocidade das rodas durante o Teste II.	74
Figura 61 – Resultado da posição do robô e o sinal de controle para o teste prático III.	75
Figura 62 – Resultado do erro de orientação do robô para o teste prático III.	76
Figura 63 – Resultado da velocidade das rodas durante o Teste III.	76

Sumário

1	INTRODUÇÃO	11
1.1	Objetivos	12
1.1.1	Objetivo geral	12
1.1.2	Objetivos específicos	12
1.2	Estrutura do Trabalho	13
2	REFERENCIAL TEÓRICO	14
2.1	Robótica Móvel	14
2.1.1	Tipos de estruturas de robôs móveis com rodas	15
2.1.1.1	Estrutura omnidirecional de três rodas	15
2.1.1.2	Estrutura uniciclo	15
2.1.1.3	Estrutura <i>Car-Like</i>	16
2.2	Modelagem Matemática de um Robô Móvel do tipo Car-Like	19
2.2.1	Movimento longitudinal	20
2.2.2	Movimento lateral	21
2.2.3	Rastreamento de caminho	21
2.3	Sistemas Digitais	24
2.3.1	Estabilidade de Lyapunov e Desigualdades Matriciais	28
2.4	Fabricação Digital	31
2.4.1	Manufatura Aditiva	32
2.4.1.1	Processo para impressão 3D	33
2.4.2	Corte a laser	35
2.5	Hardware	36
2.5.1	Sensores digitais	36
2.5.1.1	<i>Encoders</i>	37
2.5.1.2	Sensor IMU	39
2.5.2	Atuadores	41
2.5.2.1	Ponte H	41
2.5.2.2	<i>PWM</i>	42
2.5.2.3	Motor de corrente contínua	44
2.5.2.4	Servomotor	45
2.5.3	Microcontroladores	46
2.5.3.1	ESP32	46
3	METODOLOGIA	48
3.1	Projeto da Plataforma <i>Car-Like</i>	48

3.1.1	Estrutura mecânica	48
3.1.1.1	Chassi e Sistema de direção	48
3.1.1.2	Sistema de tração com diferencial	52
3.1.2	Estrutura eletrônica	54
3.2	Síntese de Controladores Robusto para Sistemas de Tempo Discreto	
	Incertos	56
3.3	Sistema de Controle Lateral	58
3.3.1	Odometria	60
3.3.2	Fusão sensorial	61
3.4	Implementação do Sistema de Controle Lateral	63
3.5	Descrição dos Testes	67
3.5.1	Testes Práticos	67
4	RESULTADOS	68
4.1	Simulações	68
4.2	Testes Práticos	70
4.2.1	Teste Prático I	70
4.2.2	Teste Prático II	72
4.2.3	Teste Prático III	73
5	CONCLUSÕES	77
5.1	Trabalhos Futuros	77
	REFERÊNCIAS	79

1 Introdução

Robôs móveis já fazem parte do imaginário das pessoas há muito tempo, principalmente, por conta de obras de ficção como os contos de Asimov que apresentavam a possibilidade de convivência com robôs móveis no dia-a-dia. Há algumas décadas esses cenários fictícios têm se tornando mais reais com os avanços da robótica móvel. Esta é uma área da robótica que trabalha com mecanismos capazes de se locomover pelo ambiente seja por meio terrestre (UGVs - *unmanned ground vehicles*), aéreo (UAVs - *unmanned aerial vehicles*) ou aquático (UVVs - *unmanned underwater vehicles*). Dentre os robôs móveis, destacam-se principalmente os robôs terrestres que podem se locomover no solo por meio de rodas, esteiras ou pernas (AGUIRRE, 2007b).

Há muitos estudos voltados a robôs móveis com rodas (RMRs) devido as suas propriedades e a sua utilidade em muitas aplicações. Destacam-se suas aplicações em ambientes industriais visando uma melhoria na produtividade de forma colaborativa ou não com seres humanos, mas também é possível citar aplicações domésticas como RMRs de entrega ou os aspiradores de pó. Em relação as suas propriedades, alguns RMRs possuem uma característica interessante que é a presença de restrições não-holonômicas que os impedem de executar movimentos normais à superfície do corpo de suas rodas (GUERRA, 2005). Além desta propriedade, incertezas estruturais e distúrbios externos impactam no movimento de RMRs quando se quer, por exemplo, realizar um simples deslocamento de um ponto a outro ou ao longo de uma trajetória.

De acordo com Rashad (2012), um dos módulos básicos de um robô móvel é o módulo de inteligência, o qual é responsável por executar algoritmos de planejamento de movimento. Pode-se tratar como uma saída deste módulo, a capacidade de o robô realizar uma trajetória segura durante uma tarefa. Devido às peculiaridades ocasionadas pelas restrições não-holonômicas, o desenvolvimento do algoritmo de controle executado neste módulo torna-se desafiador. Entre as técnicas de controle não-linear aplicadas a RMRs, pode-se citar: controlador não-linear baseado em *Lyapunov*, controle baseado em abordagem *backstepping*, controle preditivo baseado em modelo (MPC), controle baseado em planicidade e controle descontínuo (BESSAS; BENALIA; BOUDJEMA, 2016).

Na implementação real, é ideal contar com um controle inerentemente robusto, proporcionando uma convergência rápida e boas propriedades de robustez em relação à variação dos parâmetros e às perturbações externas. A maioria dos trabalhos foca em desenvolver controladores para RMRs considerando apenas o modelo cinemático do robô (SINGHAL; KUMAR; RANA, 2022). Essa suposição é adequada em sistemas mais simples, com menor complexidade estrutural e foi escolhida para este trabalho.

No grupo de RMRs há alguns tipos, mas vale destacar o tipo *Car-Like* que é caracterizado por possuir um sistema de tração traseiro e um sistema de direção na parte dianteira do robô. Estas características o tornam muito semelhante a um carro convencional e tornam seu estudo bem interessante, por isso foi o utilizado neste trabalho.

Portanto, este trabalho considera a hipótese de que um robô móvel com rodas do tipo *Car-Like* pode realizar o rastreamento de caminho de forma autônoma considerando apenas modelo cinemático com incertezas. Desta forma, o trabalho foca nos objetivos descritos a seguir.

1.1 Objetivos

1.1.1 Objetivo geral

Desenvolver um protótipo de um robô móvel do tipo *Car-Like* e projetar controladores que o tornem capaz de realizar rastreamento de caminho.

1.1.2 Objetivos específicos

1. Projetar a placa de circuito impresso PCB para controle do sistema;
2. Confeccionar a placa de circuito impresso PCB através de máquina de CNC;
3. Desenvolver o projeto mecânico do sistema em *software* de modelagem 3D;
4. Imprimir as peças modeladas através de impressora 3D;
5. Integrar o sistema mecânico e elétrico;
6. Desenvolver o algoritmo do microcontrolador para que o sistema receba os comandos de maneira correta;
7. Projetar o controlador lateral para rastreamento de caminho com base no modelo cinemático do robô considerando uma incerteza;
8. Validar o controlador implementado no MATLAB;
9. Implementar o controlador projetado em C++ e enviar o sinal de controle para o microcontrolador;
10. Analisar o desempenho do sistema real e compará-lo com os resultados simulados.

1.2 Estrutura do Trabalho

A estrutura deste estudo segue a seguinte organização. O Capítulo 2 aborda o referencial teórico adotado para esta pesquisa. No Capítulo 3, são detalhadas as metodologias a serem aplicadas neste projeto. Os resultados dos testes em campo são apresentados no Capítulo 4. Por último, o Capítulo 5 discute as considerações finais e a conclusão do trabalho.

2 Referencial Teórico

Este capítulo tem como objetivo explorar os principais fundamentos teóricos que serviram de alicerce para o desenvolvimento deste estudo. Serão abordados diversos tópicos, começando pela robótica móvel, abrangendo tipos de robôs móveis, modelagem matemática de robôs, resolução do problema de rastreamento de caminho, sistemas digitais, critério de estabilidade de *Lyapunov* e desigualdades matriciais. A exploração desses temas culminará na discussão sobre fabricação digital do protótipo, com destaque para as tecnologias empregadas, incluindo sensores digitais, atuadores e microcontroladores.

2.1 Robótica Móvel

A robótica móvel está passando por um estágio empolgante de desenvolvimento, impulsionado por avanços em sensores, inteligência artificial e tecnologia de locomoção. A robótica móvel é essencial em aplicações como veículos autônomos e robôs de serviços, desafiando os limites da automação inteligente. O futuro promete uma evolução contínua, com a robótica móvel desempenhando um papel central na transformação digital de diversos setores. Um típico sistema de robô móvel é constituído de quatro módulos: percepção, localização, inteligência e controle de movimento ([RASHAD, 2012](#)).

1. Percepção: Trata-se de um componente essencial de um sistema autônomo responsável por adquirir informações sobre o ambiente. Essa aquisição é realizada por meio de diversos sensores, dos quais são extraídas as informações relevantes a partir das medições realizadas.
2. Localização: Refere-se à capacidade do robô de conhecer com exatidão a sua localização no ambiente.
3. Inteligência: Engloba o processo de tomada de decisões e o planejamento da trajetória que um robô móvel emprega para atingir uma posição específica, fazendo uso de seus sensores e conhecimentos.
4. Controle de movimento: Este módulo desempenha a função de gerar comandos para o atuador, capacitando o robô a executar as tarefas de movimento estabelecidas no módulo de conhecimento.

Além disso, os robôs móveis podem ser classificados de diversas maneiras que incluem a forma de locomoção, os tipos de tarefas executadas e o seu grau de autonomia. De acordo com [Everett \(1995\)](#), um grande número de características influencia na escolha do sistema

de locomoção a ser utilizado em uma tarefa como: manobrabilidade, controlabilidade, estabilidade e tração. É possível representar os sistemas de locomoção dos robôs móveis por meio de modelos cinemáticos e dinâmicos. Apesar de o modelo dinâmico ser importante para o controle de alguns tipos de robôs, principalmente os robô aéreos e aquáticos, é comum a utilização apenas do modelo cinemático no desenvolvimento de sistemas de controle de movimento (AGUIRRE, 2007b). Nesse contexto, o presente trabalho busca focar em robôs móveis com rodas descritos apenas pelo modelo cinemático.

2.1.1 Tipos de estruturas de robôs móveis com rodas

Os RMRs podem assumir diferentes configurações e cada uma delas possui um modelo cinemático característico. A seguir serão apresentadas algumas das estruturas mais comuns para RMRs, bem como a abordada neste trabalho.

2.1.1.1 Estrutura omnidirecional de três rodas

Os robôs móveis omnidirecionais se destacam pela capacidade de realizar movimentos em qualquer direção no plano XY. Os robôs que possuem essa característica geralmente têm uma separação angular de 120° entre suas rodas. Esses robôs são frequentemente equipados com rodas especiais, ilustradas na [Figura 1](#), projetadas para facilitar os movimentos perpendiculares ao sentido de giro do robô. Vale mencionar que robôs móveis cuja configuração das rodas é omnidirecional não apresentam restrições não-holonômicas. A [Figura 2](#) ilustra um exemplo de robô omnidirecional.

Figura 1 – Rodas Omnidirecionais.

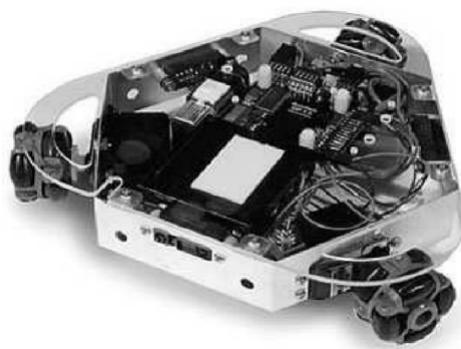


Fonte: (YADAV et al., 2022, p. 398)

2.1.1.2 Estrutura unicycle

Esses modelos de robôs móveis também são conhecidos como robôs móveis de tração diferencial. Nessa configuração, as rodas estão montadas em um mesmo eixo virtual, mas são acionadas independentemente. Essa configuração oferece a vantagem de permitir que o

Figura 2 – Robô móvel omnidirecional (Palm-Pilot Robot Kit).



Fonte: (BRANDAO, 2008, p. 18)

robô gire em torno de seu próprio eixo. Vale ressaltar que robôs com esta configuração não podem se deslocar lateralmente devido às restrições não-holonômicas. A Figura 3 apresenta o robô unicycle Pioneer 2 da empresa Mobile Robots.

Figura 3 – Robô do tipo unicycle (Pioneer 2).



Fonte: (SASAKI, 2012, p. 20)

2.1.1.3 Estrutura *Car-Like*

A estrutura *Car-Like* é a escolhida para estudo neste trabalho. Ela é caracterizada por uma semelhança significativa com a configuração de um automóvel convencional. Essa configuração é composta por um par de rodas montadas no mesmo eixo, tracionadas simultaneamente, geralmente posicionadas na parte traseira do robô. Além disso, há outro

par de rodas localizado na parte dianteira do robô conforme é visto na [Figura 4](#), que são responsáveis pela dirigibilidade do robô.

Figura 4 – Robô do tipo Car-Like (Berkeley Autonomous Race Car).



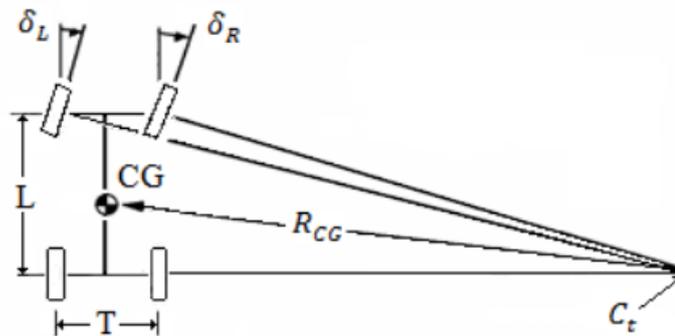
Fonte: ([GOLDENEYE, 2024](#))

A estrutura de RMRs do tipo *Car-Like* apresenta uma característica importante: o ângulo de esterçamento das rodas dianteiras deve obedecer à geometria de *Ackermann*. Esta geometria refere-se a uma configuração específica de direção de veículos, principalmente utilizada em veículos de direção de quatro rodas, como carros, caminhões e alguns veículos off-road. Ela é projetada para permitir um movimento preciso e eficiente das rodas dianteiras, proporcionando uma curva mais suave.

A geometria de Ackermann é caracterizada por um arranjo em que as linhas que passam pelos pontos de contato dos pneus dianteiros com o solo convergem em um ponto central imaginário conforme mostra a [Figura 5](#). Isso significa que, ao esterçar as rodas para realizar uma curva, cada roda percorre um raio de curva ligeiramente diferente com ângulos de esterçamento δ_L e δ_R distintos em relação ao centro de giro (C_t) ([DIXON, 2009](#)), mas o resultado é que todas as rodas seguem uma trajetória suave e curvam-se de maneira coordenada.

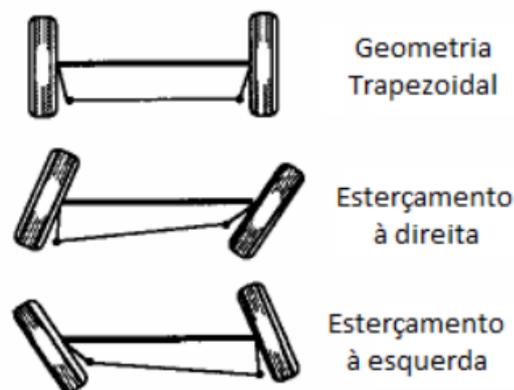
Segundo [Gillespie \(2021\)](#), é preciso que a geometria das barras de direção formem um trapézio ao invés de um retângulo para que a geometria de *Ackermann* seja atendida. Isto se ocorre pelo fato de que um padrão retangular ocasionaria uma orientação igual para a roda direita e esquerda, resultando no descumprimento da geometria de *Ackermann*. A diferença nos ângulos entre a roda direita e a roda esquerda resulta da assimetria que se manifesta no início do movimento de esterçamento da roda, amplificando-se à medida que o ângulo de esterçamento aumenta conforme mostra a [Figura 6](#).

Figura 5 – Posição ideal do centro de giro de um veículo.



Fonte: (ZANARDI, 1013, p. 20)

Figura 6 – Diferença de ângulos resultante da geometria trapezoidal.



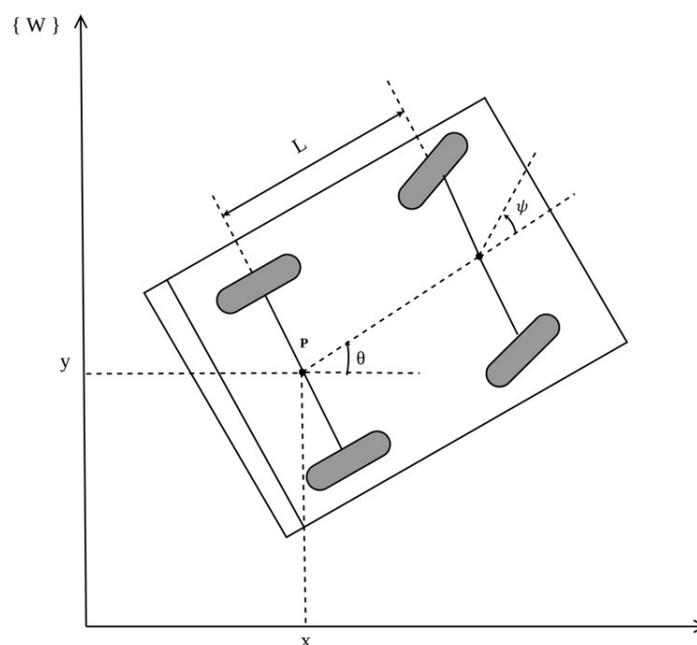
Fonte: (ZANARDI, 1013, p. 22)

Assumir que o sistema de direção do RMR atende à geometria de *Ackermann* possibilita substituir estas duas rodas por uma única roda no centro geométrico, reduzindo o sistema de quatro rodas para o que se chama de modelo de triciclo, que por sua vez, é equivalente a um modelo de bicicleta para fins de controle e localização (MARTINS et al., 2011). Os detalhes de controle e localização deste tipo de RMR serão explicados em próximas seções. Vale ressaltar que essa configuração possui restrições não-holonômicas que impedem o movimento lateral. Portanto, para alcançar uma orientação desejada, o robô precisa executar movimentos de translação e rotação, isto é, o robô não é capaz de girar em torno de seu próprio eixo.

2.2 Modelagem Matemática de um Robô Móvel do tipo Car-Like

Um robô móvel pode ser definido em seu espaço de trabalho através do vetor de configurações $q(t)$. A dimensão de $q(t)$ corresponde ao número de graus de liberdade do robô, representando a quantidade mínima de variáveis independentes necessárias, juntamente com sua geometria, para descrever completamente o robô. No caso de um robô móvel em forma de carro, há três graus de liberdade então o vetor de configuração pode ser escrito como $q(t) = [x(t), y(t), \theta(t)]^\top$, onde $x(t)$ e $y(t)$ especificam a posição cartesiana do robô em relação a um referencial fixo, $\{W\}$, no ambiente e $\theta(t)$ especifica a sua orientação conforme mostra a [Figura 7](#).

Figura 7 – Esquemático de um robô *Car-Like*.



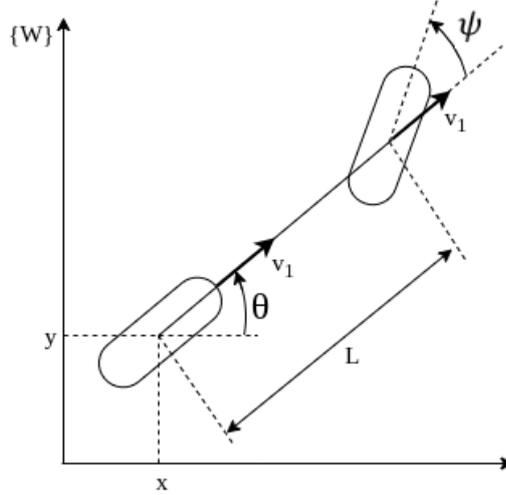
Fonte: Adaptado de [Arogeti e Berman \(2012\)](#).

Ao abordar o projeto de controle de movimento, é comum a utilização do modelo matemático do sistema representado por equações diferenciais. Para obter este modelo, é preciso assumir algumas características a cerca do robô ([RASHAD, 2012](#)), como:

1. Corpo rígido: O robô é modelado como um corpo rígido sobre rodas não-deformáveis.
2. Solo ideal: O robô se move em um plano horizontal e com coeficiente de atrito constante.
3. Rolagem pura: O ponto de contato entre as rodas e o solo satisfaz a condição de pura rolagem sem derrapagem.

Assumindo-se ainda que o robô *Car-Like* obedece à geometria de *Ackermann* e possui sistema de tração traseira, é possível utilizar um modelo cinemático tipo bicicleta, apresentado na [Figura 8](#) e descrito pela [Equação 2.1](#) a seguir,

Figura 8 – Diagrama esquemático de um modelo tipo bicicleta.



Fonte: Adaptado de [Arogeti e Berman \(2012\)](#).

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \\ \dot{\psi}(t) \end{bmatrix} = \begin{bmatrix} \cos \theta(t) \\ \sin \theta(t) \\ \frac{\tan \psi(t)}{L} \\ 0 \end{bmatrix} v_1(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} v_2(t), \quad (2.1)$$

em que $\psi(t)$ é o ângulo de esterçamento do robô, o qual é resultante de uma combinação dos ângulos de esterçamento das rodas ([MARTINS et al., 2011](#)), L é a distância entre o eixo dianteiro e traseiro, e $v_1(t)$ é a velocidade linear do robô e $v_2(t)$ é a taxa de variação do ângulo de esterçamento.

O modelo cinemático relaciona as velocidades dos atuadores, neste caso as rodas, às velocidades de um referencial fixo no robô, i.e., o ponto P indicado na [Figura 7](#). Partindo-se do modelo matemático (2.1), é possível desenvolver controladores de movimento que são divididos em longitudinal e lateral.

2.2.1 Movimento longitudinal

A abordagem longitudinal diz respeito à regulação da velocidade linear do robô móvel de forma adequada para a tarefa a ser realizada. [Dias \(2013\)](#) afirma que, ao dirigir um veículo, o ser humano executa o controle longitudinal com base em modelos mentais

previamente obtidos do veículo e do ato de dirigir. Neste caso, as pernas do condutor são os atuadores que agem sobre a planta (carro) e as decisões de controle são tomadas pelo cérebro de acordo com distúrbios como inclinação, tipo de terreno, atrito dinâmico de deslizamento da direção, dentre outros.

2.2.2 Movimento lateral

A abordagem lateral é responsável pelo esterçamento das rodas dianteiras que impactam na orientação do RMR em relação a um caminho ou trajetória geralmente (DIAS, 2013). O modelo cinemático lateral de um RMR *Car-Like* com tração traseira é aquele descrito pela Equação 2.1. Este modelo possui uma singularidade quando $\psi = \pm\pi/2$, que resulta numa descontinuidade em $\tan(\psi)/L$. Isto significa que as rodas dianteiras estão normais ao eixo longitudinal do robô, o que resultaria no travamento do movimento na direção deste eixo, mas não é um problema já que em aplicações práticas o ângulo de esterçamento (ψ) é limitado pela construção física do RMR e não chega a um valor tão alto quanto $\pm\pi/2$ (LUCA; ORIOLO; SAMSON, 2005). O limite do ângulo de esterçamento impacta diretamente na máxima curvatura, k_{max} , que o RMR conseguirá executar, portanto tem influência sobre os limites a serem considerados em problemas de rastreamento de caminho ou trajetória. O valor de curvatura máxima pode ser calculado conforme (2.2).

$$k_{max} = \frac{\tan \psi_{max}}{L} \quad (2.2)$$

Sabendo como é o comportamento do RMR *Car-Like* segundo seus modelos cinemáticos longitudinal e lateral, pode-se uni-los para resolver um problema que na robótica móvel é chamado de rastreamento de caminho.

2.2.3 Rastreamento de caminho

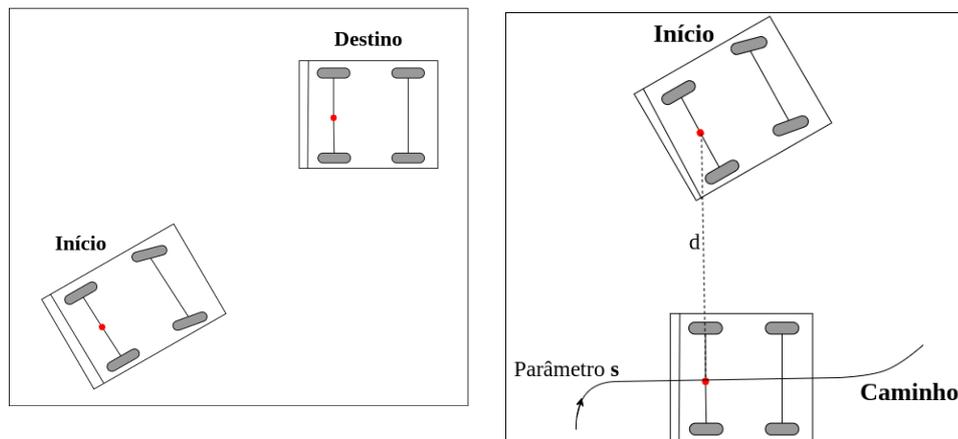
Para executar uma tarefa de controle de movimento, assume-se que não haverá colisão, uma vez que se parte da premissa de que o problema de desvio de obstáculos foi resolvido no estágio de planejamento de movimento. Segundo Rashad (2012), as tarefas de controle de movimento podem ser classificadas como:

1. Movimento ponto a ponto: A partir de uma configuração inicial particular, o objetivo do robô é alcançar uma configuração desejada. Considerado como um desafio de controle, a missão ponto a ponto refere-se à estabilização do robô no espaço de configuração, convergindo para um ponto de equilíbrio designado.
2. Rastreamento de caminho: O robô deve ser capaz de percorrer e acompanhar um caminho geométrico, seja começando sobre ou fora do caminho.

3. Rastreamento de trajetória: O robô deve atingir e seguir uma trajetória (ou seja, um caminho com uma lei de temporização associada), independentemente de iniciar dentro ou fora da trajetória.

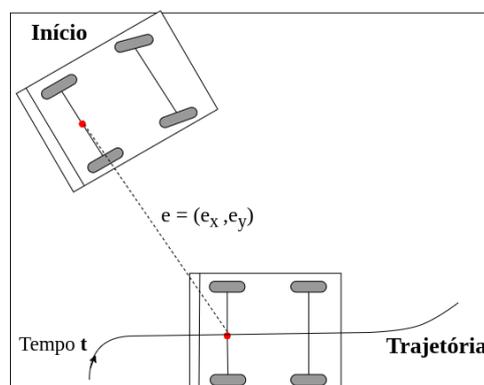
Vale ressaltar que no rastreamento de trajetórias, o caminho desejado para o robô é definido como uma função dependente do tempo, e em cada instante, apenas as características de um ponto ao longo desse caminho são transmitidas ao sistema de navegação como ponto de destino conforme mostrado em Figura 9(c). Se o robô for iniciado no meio do caminho desejado, ele irá voltar para o ponto de início do caminho ocasionando um fenômeno chamado de retrocesso. Este fenômeno não ocorre no rastreamento de caminho, uma vez que uma função independente do tempo descreve o caminho desejado. Nesse caso, o caminho é enviado ao sistema de navegação, que o processa para identificar o ponto mais próximo conforme mostrado em Figura 9(b). Posteriormente, o robô se direciona para esse ponto mais próximo do caminho, resultando na subsequente navegação ao longo da trajetória após alcançá-la (EMAM; FAKHARIAN, 2017a).

Figura 9 – Tarefas do sistema de controle de movimento.



(a) Movimento Ponto a Ponto;

(b) Rastreamento de caminho;



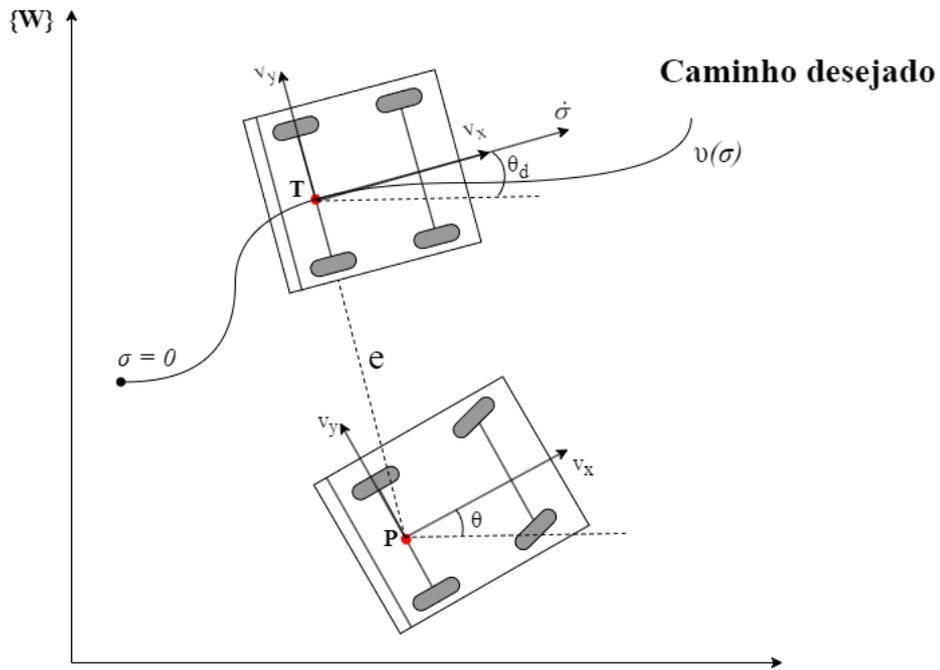
(c) Rastreamento de trajetória.

Fonte: Adaptado de Rashad (2012)

Em Wang et al. (2016), é apresentado um modelo para rastreamento de caminho conforme a Figura 10. Sabe-se que e representa a distância entre o ponto \mathbf{P} , situado no centro do eixo que liga as rodas traseiras, e o ponto mais próximo, \mathbf{T} , do caminho desejado. O erro de orientação, θ_e , é definido como a diferença entre a orientação do robô, θ , e a orientação desejada, θ_d , portanto $\theta_e = \theta - \theta_d$. v_x e v_y são velocidade longitudinal e lateral do robô, respectivamente. O comprimento de arco do ponto \mathbf{T} ao longo do caminho a partir de uma posição inicial pré-definida é representado por σ , enquanto se sabe que $\sigma \geq 0$. Por fim, $v(\sigma)$ representa a curvatura do caminho desejado no ponto \mathbf{T} . A coordenada curvilínea dos pontos \mathbf{T} ao longo do caminho σ pode ser expressa como:

$$\dot{\sigma} = \frac{1}{1 - e v(\sigma)} (v_x \cos \theta_e - v_y \sin \theta_e) \quad (2.3)$$

Figura 10 – Diagrama esquemático de um modelo de rastreamento de caminho.



Fonte: Adaptado de Arogeti e Berman (2012).

Neste trabalho é utilizado o modelo de rastreamento de caminho definido pela Equação 2.4, apresentado por Wang et al. (2016), porém, considerando o efeito do ângulo do deslizamento conforme será explicado na metodologia.

$$\begin{cases} \dot{e} = v_x \sin \theta_e + v_y \cos \theta_e \\ \dot{\theta}_e = \dot{\theta} - v(\sigma)v_x \end{cases} \quad (2.4)$$

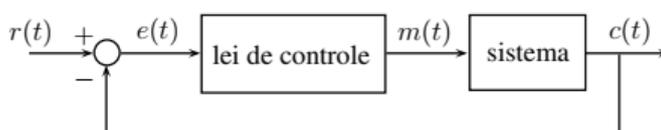
2.3 Sistemas Digitais

O termo digital tem origens na forma como os computadores executam operações, processando informações por meio de dígitos binários. Inicialmente restritas aos sistemas computacionais, as aplicações da eletrônica digital expandiram-se para diversas áreas. Atualmente, tecnologias digitais são empregadas em setores como televisão, sistemas de comunicação, radar, navegação, sistemas militares, instrumentação médica, controle de processos industriais e dispositivos eletrônicos de consumo. Ao longo dos anos, a tecnologia digital evoluiu desde circuitos com válvulas até circuitos integrados complexos, alguns contendo milhões de transistores, impulsionando avanços significativos em diversas áreas da sociedade (FLOYD, 2009).

Sistemas digitais são, por natureza, sujeitos a amostragem, mas há diversos outros sistemas, mesmo analógicos, que também passam por esse processo. Um exemplo dessa categoria inclui sistemas de radar e sonar nos quais novas informações tornam-se disponíveis a cada varredura realizada. Somente quando a informação amostrada passa por uma conversão analógico-digital (AD) é que o resultado é *digital* (AGUIRRE, 2023).

Na Figura 11, é apresentado o diagrama de blocos de um sistema de controle com realimentação negativa. O princípio de funcionamento deste sistema é baseado na contínua comparação da saída $c(t)$ com a referência $r(t)$. Essa diferença é chamada de erro $e(t)$ que existe para todo e qualquer valor de t (AGUIRRE, 2023). A lei de controle, gerada a partir do sinal de erro $e(t)$, determina $m(t)$, chamada ação de controle.

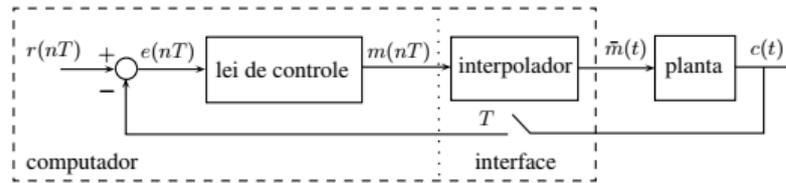
Figura 11 – Diagrama de um servomecanismo de tempo contínuo.



Fonte: (AGUIRRE, 2023)

Considerando que a variável de saída $c(t)$ existe de maneira contínua, no entanto, só está disponível em instantes regularmente espaçados. O sistema de controle assume a forma mostrada na Figura 12, onde o intervalo entre uma medida e outra da saída é chamado de período de amostragem (T). É importante ressaltar que o valor da saída só pode ser comparado com a referência a cada período de amostragem, então se tem uma sequência possivelmente infinita mas *contável*, de valores de erro $e(T)$, $e(2T)$, $e(3T)$, ... ao contrário do conjunto infinito e *não contável* de valores que correspondem à variável contínua $e(t)$ considerada dentro do intervalo de tempo $t \in [t_0, t_f]$.

Figura 12 – Diagrama de blocos de um sistema amostrado de controle.



Fonte: (AGUIRRE, 2023)

Observa-se na Figura 12 que a parte do sistema indicada dentro da linha tracejada, opera com sequências discretas, enquanto que a outra parte é de tempo contínuo. Estas são características de um sistema de dados amostrados e quando o sistema é de controle, utiliza-se o termo *sistemas amostrados de controle*. Finalmente, se a lei de controle é implementada em computador digital, usa-se a denominação: *controle digital* segundo Aguirre (2023). Neste tipo de sistema, a presença de partes discreta e contínua impõe desafios na representação, análise e controle. A necessidade de aplicar ação de controle discreta a um sistema predominantemente contínuo requer o uso da interpolação. O interpolador é essencial para a conversão digital-analógica, tornando imperativo incorporar métodos de interpolação em tais sistemas.

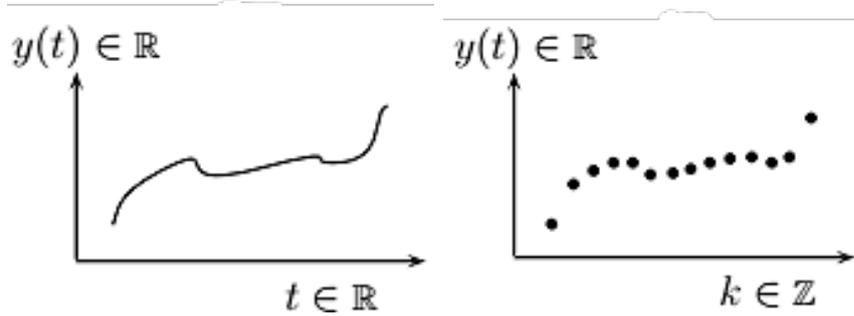
Na Figura 13 são mostradas as representações para um sinal. Considerando em Figura 13(a) o sinal $y(t)$ como uma função do tempo, é possível discretizar apenas o seu domínio a partir de uma amostragem ideal, obtendo-se o sinal $y(k)$ mostrado na Figura 13(b). Nota-se que os valores deste sinal pertencem ao eixo real, pois podem assumir qualquer valor já que este sinal é discreto no tempo, mas analógico. Por fim, representando-se o valor de $y(k)$ em código binário com um número limitado de bits (Figura 13(c)), valores (analógicos) próximos poderão ter a mesma representação binária. Isto é um efeito da *quantização* onde a quantidade de bits do conversor AD influencia nos possíveis valores do domínio digital \mathbf{D} .

Dentre as formas de representar um sistema dinâmico, este trabalho optou pela representação no espaço de estados por ser mais adequada para lidar com sistemas multivariáveis, e pela possibilidade de representar sistemas não lineares (AGUIRRE, 2023). Um sistema em tempo discreto não linear pode ser escrito como

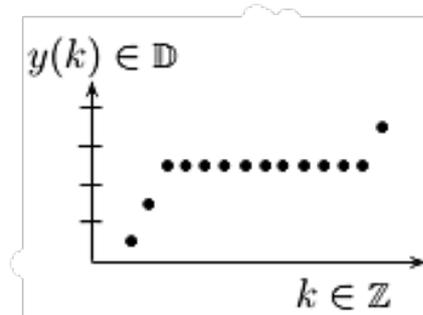
$$\begin{aligned} \mathbf{x}(k+1) &= f[\mathbf{x}(k), \mathbf{u}(k)] \\ \mathbf{y}(k) &= h[\mathbf{x}(k), \mathbf{u}(k)] \end{aligned} \quad (2.5)$$

em que $\mathbf{x} \in \mathbb{R}^n$ representa o vetor de variáveis de estado ou, de forma mais simples, o vetor de estado, $\mathbf{u} \in \mathbb{R}^r$ é o vetor de entradas, e $\mathbf{y} \in \mathbb{R}^p$ é o vetor de saídas. No cenário onde temos apenas uma entrada e uma saída $r = p = 1$. A primeira equação de (2.5)

Figura 13 – Representação de sinal.



(a) Tempo contínuo e analógico. (b) Tempo discreto e analógico.



(c) Tempo discreto e digital.

Fonte: (AGUIRRE, 2023)

descreve a dinâmica do sistema, ou seja, como a entrada $\mathbf{u}(k)$ e o vetor de estado $\mathbf{x}(k)$ atuais determinam o vetor de estado no próximo instante $\mathbf{x}(k+1)$, então é chamada de *equação dinâmica*. A segunda equação de (2.5) descreve a saída em função do vetor de estados e, em certos casos, em função da entrada também, então é chamada de *equação de saída*.

No caso linear e variante no tempo, a representação em espaço de estados do sistema é dada por

$$\begin{aligned}\mathbf{x}(k+1) &= A(k)\mathbf{x}(k) + B(k)\mathbf{u}(k) \\ \mathbf{y}(k) &= C(k)\mathbf{x}(k) + D(k)\mathbf{u}(k)\end{aligned}\tag{2.6}$$

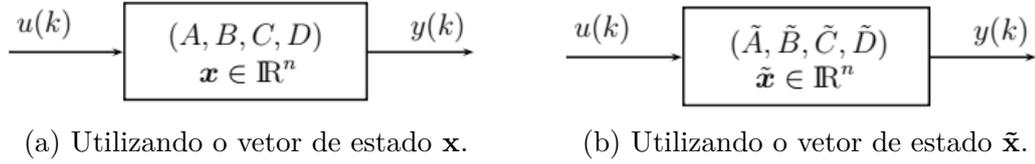
em que $A(k) \in \mathbb{R}^{n \times n}$, $B(k) \in \mathbb{R}^{n \times r}$, $C(k) \in \mathbb{R}^{p \times n}$ e $D(k) \in \mathbb{R}^{p \times r}$ são matrizes que variam com o *tempo discreto*. Para o caso em que o sistema é invariante no tempo, estas matrizes são constantes e a representação torna-se

$$\begin{aligned}\mathbf{x}(k+1) &= A\mathbf{x}(k) + B\mathbf{u}(k) \\ \mathbf{y}(k) &= C\mathbf{x}(k) + D\mathbf{u}(k)\end{aligned}\tag{2.7}$$

Esta representação será a utilizada neste trabalho. A partir da [Equação 2.7](#) sabe-se que o vetor de estado futuro, $\mathbf{x}(k+1)$, e o vetor da saída atual, $\mathbf{y}(k)$, são obtidos por combinações lineares do estado e do vetor de entrada atuais. [Aguirre \(2023\)](#) cita que um mesmo sistema pode ser representado por vetores de estados distintos, o que resulta

em um diferente conjunto de matrizes. Vale ressaltar que apenas a representação muda, enquanto o sistema e os sinais de entrada e saída permanecem os mesmos como ilustrado na [Figura 14](#).

Figura 14 – Diferentes representações em espaço de estados do mesmo sistema.



Fonte: ([AGUIRRE, 2023](#))

Em ([AGUIRRE, 2023](#)) é apresentado o conceito de transformação de similaridade que será utilizado nas próximas seções. Considerando uma matriz não singular $P \in \mathbb{R}^{n \times n}$ de forma que

$$\mathbf{x}(k) = P\tilde{\mathbf{x}}(k), \quad (2.8)$$

a matriz P relaciona o vetor de estado \mathbf{x} a um outro vetor de estado $\tilde{\mathbf{x}}$. Substituindo (2.8) em (2.7), obtém-se

$$\begin{aligned} P\tilde{\mathbf{x}}(k+1) &= AP\tilde{\mathbf{x}}(k) + B\mathbf{u}(k) \\ \mathbf{y}(k) &= CP\tilde{\mathbf{x}}(k) + D\mathbf{u}(k) \end{aligned} \quad (2.9)$$

e como, por hipótese, P é não singular, chega-se a

$$\begin{aligned} \tilde{\mathbf{x}}(k+1) &= P^{-1}AP\tilde{\mathbf{x}}(k) + P^{-1}B\mathbf{u}(k) \\ \mathbf{y}(k) &= CP\tilde{\mathbf{x}}(k) + D\mathbf{u}(k) \end{aligned} \quad (2.10)$$

ou ainda

$$\begin{aligned} \tilde{\mathbf{x}}(k+1) &= \tilde{A}\tilde{\mathbf{x}}(k) + \tilde{B}\mathbf{u}(k) \\ \mathbf{y}(k) &= \tilde{C}\tilde{\mathbf{x}}(k) + \tilde{D}\mathbf{u}(k) \end{aligned} \quad (2.11)$$

em que

$$\tilde{A} = P^{-1}AP, \tilde{B} = P^{-1}B, \tilde{C} = CP, \tilde{D} = D \quad (2.12)$$

Em termos simples, a matriz P define uma transformação de similaridade, resultando na representação do sistema como $(\tilde{A}, \tilde{B}, \tilde{C}, \tilde{D})$, conforme ilustrado na [Figura 14](#). Essa transformação de similaridade transfere o estado do sistema de um conjunto de coordenadas para outro. Importante notar que o sistema em si não sofre alterações, apenas as coordenadas em que é representado. Dessa forma, a representação do sistema é modificada, mas suas propriedades intrínsecas, como a função de transferência e a equação característica, permanecem inalteradas.

2.3.1 Estabilidade de Lyapunov e Desigualdades Matriciais

A utilização de Desigualdades Matriciais Lineares (LMIs, do inglês *Linear Matrix Inequalities*) na análise de sistemas dinâmicos tem uma história que ultrapassa os 100 anos. Essa trajetória tem início por volta de 1890, quando *Lyapunov* publicou sua obra fundamental introduzindo o que atualmente denomina-se teoria de *Lyapunov*. O próximo marco significativo ocorreu na década de 1940. *Lur'e*, *Postnikov* e outros na União Soviética aplicaram os métodos de *Lyapunov* a problemas práticos específicos em engenharia de controle, notadamente o desafio da estabilidade de um sistema de controle com uma não-linearidade no atuador. Mesmo sem formular desigualdades matriciais de maneira explícita, seus critérios de estabilidade assumem a forma de LMIs. Essas desigualdades foram convertidas em desigualdades polinomiais, as quais eram então verificadas manualmente (principalmente para sistemas pequenos). Contudo, a empolgação justificada surgiu da perspectiva de aplicar a teoria de *Lyapunov* a problemas práticos e complexos em engenharia de controle ([BOYD et al., 1994](#)).

Basicamente, *Lur'e* e outros foram os primeiros a aplicar os métodos de *Lyapunov* a problemas práticos de engenharia de controle. No entanto, somente em meados da década de 1960, outros trabalhos foram publicados para reconhecer o papel importante das LMIs na teoria de controle. Então os estudos foram evoluindo até o estado atual, conforme mencionado por [Boyd et al. \(1994\)](#).

Uma LMI tem a forma

$$F(x) \triangleq F_0 + \sum_{i=1}^m x_i F_i > 0 \quad (2.13)$$

em que $x \in \mathbb{R}^m$ é a variável e as matrizes simétricas $F_i = F_i^T \in \mathbb{R}^{n \times n}$, $i = 0, \dots, m$, são dadas. O símbolo de desigualdade em (2.13) significa que $F(x)$ é definida positiva, ou seja, $u^T F(x) u > 0$ para todo $u \in \mathbb{R}^n$ não nulo. A LMI (2.13) é uma restrição convexa em x , ou seja, o conjunto $\{x \in \mathbb{R}^m \mid F(x) > 0\}$ é convexo. Segundo [Aguirre \(2007a\)](#), esta propriedade é fundamental para que se possa formular problemas de controle robusto tal qual problemas de otimização convexa que minimizam uma função objetivo linear de um vetor de variáveis de decisão $x \in \mathbb{R}^m$, sujeito a restrições do tipo LMIs:

$$\begin{cases} \min_x & c^T x \\ \text{sujeito a:} & F(x) > 0 \end{cases} \quad (2.14)$$

Embora a LMI (2.13) possa parecer ter uma forma especializada, ela pode representar uma ampla variedade de restrições convexas em x . Em particular, desigualdades lineares, desigualdades quadráticas (convexas), desigualdades de normas de matriz e restrições que surgem na teoria de controle, como as desigualdades matriciais quadráticas convexas de *Lyapunov* ([BOYD et al., 1994](#)).

Considere o sistema dinâmico linear autônomo da forma:

$$\Delta[x] = Ax \quad (2.15)$$

sendo que $x \in \mathbb{R}^n$ e $A \in \mathbb{R}^{n \times n}$. Utilizar-se-á uma notação especial representada pelo operador $\Delta[\bullet]$ em (2.15) que pode indicar: $\dot{x}(t)$ para sistemas de tempo contínuo ou $x(k+1)$ para sistemas de tempo discreto. O sistema definido pela Equação 2.15 possui um ponto de equilíbrio quando $\Delta[x(t)] \equiv 0$, ou seja, não há variação.

No caso contínuo, ter todas as derivadas nulas significa que os estados não estão variando no tempo, então isto significa que os estados estão no ponto de equilíbrio. Caso A seja não singular, então o único ponto de equilíbrio, tal que $\dot{x}(t) \equiv 0$, é a origem $x(t) = 0$ (AGUIRRE, 2007a). O critério de estabilidade de *Lyapunov* é obtido a partir da definição de uma função quadrática positiva como:

$$V(x(t)) = x^T(t)Px(t) > 0, \text{ sendo } P = P^T > 0, P \in \mathbb{R}^{n \times n}$$

e avaliando a derivada (caso contínuo) ao longo das trajetórias $x(t)$, ou seja:

$$\dot{V}(x(t)) = x^T(t)(A^T P + PA)x(t)$$

Já que $V(x(t))$ foi adotada como sendo positiva, para se obter estabilidade assintótica ($x(t) \equiv 0$ é o ponto de equilíbrio) é necessário que $\dot{V}(x)$ seja negativa, ou seja, $\dot{V}(x) < 0$, ou seja

$$A^T P + PA < 0, \text{ com } P > 0$$

Para o caso a tempo discreto é aplicada a mesma estratégia. Define-se a mesma função de *Lyapunov* $V(x(k))$ e avalia-se a diferença ao longo das trajetórias $x(k)$ tal que:

$$\begin{aligned} \Delta V(x(k)) &\triangleq V(x(k+1)) - V(x(k)) \\ &= x^T(k)(A^T P A - P)x(k) \end{aligned}$$

Para a estabilidade do sistema autônomo a tempo discreto basta que

$$A^T P A - P < 0, \text{ com } P > 0$$

Dessa forma, o Teorema de *Lyapunov* garante que o sistema (2.15) é assintoticamente estável se e somente se existir uma matriz $P = P^T > 0$ satisfazendo a LMI

1. $A^T P + PA < 0$, para tempo contínuo, ou
2. $A^T P A - P < 0$, para tempo discreto.

Conforme visto em [Aguirre \(2007a\)](#), a maneira como o critério de estabilidade de *Lyapunov* é expresso através de LMIs proporciona uma discussão significativa no âmbito da teoria de controle robusto. Isso envolve explorar métodos para verificar a estabilidade não apenas de sistemas individuais, mas também de famílias de sistemas ou sistemas que possuem incertezas. Motivado por esta questão define-se:

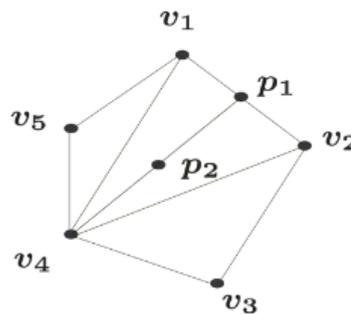
- **Conjunto Poliedral** - Intersecção de um número finito de subespaços fechados. Os conjuntos poliedrais são convexos e fechados.
- **Politopo** - É um conjunto poliedral *limitado*. Em outras palavras, um politopo é uma casca convexa de um conjunto finito de vértices, onde todo elemento no politopo pode ser gerado pela combinação convexa dos seus vértices.

Considerando um politopo descrito por cinco vértices, $\rho = \text{co}\{v_1, v_2, \dots, v_5\}$ (co - denota a casca convexa), conforme ilustrada na [Figura 15](#). É possível escrever qualquer ponto $p \in \rho$ por meio de um combinação convexa dos vértices:

$$p = \sum_{i=1}^5 \gamma_i v_i, \quad \gamma_i \geq 0, \quad \sum_{i=1}^5 \gamma_i = 1$$

Por exemplo, o ponto $p_1 = \frac{1}{2}v_1 + \frac{1}{2}v_2 + 0v_3 + 0v_4 + 0v_5$ e $p_2 = \frac{1}{3}v_1 + \frac{1}{3}v_2 + 0v_3 + \frac{1}{3}v_4 + 0v_5$. De forma análoga, supõe-se que cada vértice é uma matriz A de um conjunto de sistemas autônomos.

Figura 15 – Politopo com cinco vértices.



Fonte: ([AGUIRRE, 2007a](#))

Essa característica de convexidade é adequada para abordar a estabilidade de um sistema incerto ao formular o problema, examinando a estabilidade em um politopo e aplicando a teoria de *Lyapunov* ([AGUIRRE, 2007a](#)). Considerando um sistema dinâmico com κ vértices:

$$\Delta[x(t)] = Ax(t), \quad A \in \rho \triangleq \left\{ A \mid A = \sum_{i=1}^{\kappa} \gamma_i A_i, \gamma_i \geq 0, \sum_{i=1}^{\kappa} \gamma_i = 1 \right\} \quad (2.16)$$

O sistema incerto (2.16) é considerado quadraticamente estável se,

1. para sistemas de tempo contínuo, existe uma matriz $P = P^T > 0$, tal que

$$A^T P + P A < 0, \quad \forall A \in \rho$$

2. para sistemas de tempo discreto, existe uma matriz $P = P^T > 0$, tal que

$$A^T P A - P < 0, \quad \forall A \in \rho$$

Observe que a determinação da estabilidade quadrática envolve a escolha de uma única matriz de *Lyapunov* que atenda simultaneamente a todas as diferentes descrições de sistemas contidas no politopo (AGUIRRE, 2007a). Devido à natureza da formulação da estabilidade quadrática, em vez de verificar a condição para os inúmeros sistemas pertencentes ao politopo, é suficiente avaliar a condição apenas nos seus κ vértices. Portanto, para determinar se o sistema é quadraticamente estável, é necessário verificar a existência de uma matriz $P = P^T > 0$ que satisfaça o conjunto finito correspondente às κ LMIs, como exemplificado abaixo:

$$\begin{aligned} A_i^T P + P A_i < 0, \quad \forall i = 1, \dots, \kappa & \quad (\text{tempo contínuo}) \\ A_i^T P A_i - P < 0, \quad \forall i = 1, \dots, \kappa & \quad (\text{tempo discreto}) \end{aligned}$$

2.4 Fabricação Digital

A fabricação digital engloba o emprego de métodos para criar produtos, objetos e protótipos com base em modelos virtuais. As vantagens desse método incluem a rapidez e exatidão na criação das características físicas e geométricas dos modelos virtuais, além de permitir a fabricação em larga escala de produtos personalizados (ORCIUOLI (2009)).

De modo geral, a obtenção de produtos por meio de processos de fabricação digital envolve a utilização de sistemas de manufatura aditiva (como a impressão 3D) ou de manufatura subtrativa (como a usinagem CNC). Os sistemas de manufatura aditiva operam adicionando camadas de um ou mais materiais para formar o produto final, enquanto a manufatura subtrativa resulta na obtenção do produto ao remover gradualmente material. Em ambos os métodos mencionados, o primeiro passo consiste na criação de um modelo virtual, desenvolvido por meio de softwares de CAD (*Computer Aided Design*), que possibilitam a geração e manipulação tridimensional de formas. Esses programas permitem a atribuição de diversos elementos às peças, como materiais, texturas, volumes e formas distintas, o que permite explorar aspectos como o encaixe entre as peças, a montagem e desmontagem do produto, entre outros (Espinoza e Schaeffer (2004)).

Os protótipos podem ser empregados para solucionar questões de design e aprimorar a compreensão do produto em desenvolvimento (Hallgrímsson (2012)). Portanto, técnicas de fabricação digital serão empregadas no projeto do robô móvel do tipo *Car-like* desenvolvido neste trabalho.

2.4.1 Manufatura Aditiva

O projeto mecânico do robô móvel desta pesquisa será fabricado por meio de impressão 3D, que pode ser definida como o processo de realizar um objeto físico a partir de uma representação geométrica computadorizada através do depósito sucessivo e ordenado de materiais (SHHRUBUDIN; LEE; RAMLAN, 2019). O processo de manufatura aditiva é o contrário de processos subtrativos, que extraem material de um objeto para se obter o produto desejado.

Essa diferença entre adicionar ou subtrair materiais para a manufatura de uma peça tem implicações profundas no processo produtivo industrial (CAMPBELL et al., 2011), onde pode ser citado:

1. A utilização da manufatura aditiva pode simplificar ou mesmo eliminar as linhas de produção e os elos das cadeias de abastecimento. Uma vez que a manufatura aditiva é capaz de criar peças complexas em um único processo, ao contrário da fabricação convencional que requer a produção de múltiplas peças em diferentes etapas que posteriormente são montadas. Estas peças frequentemente são obtidas de locais distantes e variados ao redor do globo. Dessa maneira, a manufatura aditiva tem o potencial de centralizar a produção destas diversas peças em um único local, resultando em uma redução das complexidades da cadeia de abastecimento;
2. Assim como a Internet revolucionou a transferência de informações globais sem a exigência de deslocamento físico, a manufatura aditiva, notadamente a Impressão 3D, viabiliza a rápida transmissão de projetos, em vez de produtos, através da rede, eliminando a necessidade de movimentar objetos físicos. Esses projetos podem ser posteriormente materializados por uma impressora 3D compatível;
3. Uma empresa teria a capacidade de fabricar uma variedade de peças distintas e personalizadas, sem encargos adicionais, da mesma forma que não precisaria manter um inventário de mercadorias;
4. Uma vez que as peças podem ser produzidas em proximidade ao cliente final ou pelo próprio, a manufatura aditiva reduz os custos associados ao transporte e à logística, ao mesmo tempo em que estimula a economia local ao transferir a fase de produção de nações estrangeiras;

2.4.1.1 Processo para impressão 3D

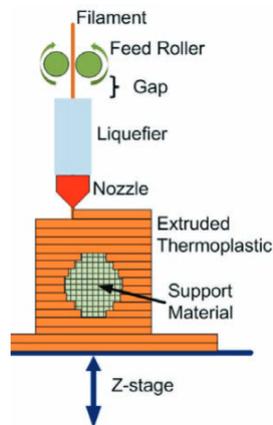
A maneira mais comum de manufatura aditiva é a tecnologia de Impressão 3D. O procedimento para criar um objeto utilizando Impressão 3D pode ser resumido pelos seguintes passos:

1. Projetar um modelo em *software* de CAD (*Computer-Assisted Design*) ou baixar em algum site como *Thingiverse*, *3D Cults*, *Printables* ou *MyMiniFactory*;
2. Obter um arquivo capaz de ser lido por software de fatiamento, normalmente no formato *.STL* ou *.3MF*;
3. Utilizar software de fatiamento para transformar o modelo digital de um objeto num arquivo especial de formato *.Gcode* compatível com a impressora a ser utilizado. É neste passo que se escolhe a resolução e o preenchimento da peça, além de informar o material para fabricação também;
4. Enviar o arquivo à impressora 3D para que possa começar o processo de criação do objeto, camada por camada.

O processo mais amplamente empregado para realizar a impressão é conhecido como Modelagem por Fusão e Deposição (FDM), que envolve a extrusão de materiais termoplásticos ou cera (CAMPBELL et al., 2011) por meio de um bico aquecido. Este bico deposita o material em camadas sucessivas, conforme ilustrado na Figura 16. Quando o plástico é aquecido à temperatura de fusão, ele se torna líquido e flui pelo bico do extrusor, resfriando e solidificando assim que é colocado sobre a peça. Após a conclusão de uma camada, a impressora move-se verticalmente em 1 unidade da resolução definida no software de fatiamento ao longo do eixo Z e começa a depositar material para a camada subsequente, seguindo esse processo até que o objeto esteja finalizado. O plástico é fornecido ao extrusor na forma de um filamento estreito, impulsionado por roletes. Na Figura 17 é possível ver alguns exemplos de filamentos e há um exemplo de impressora do tipo FDM também.

Outras técnicas também existem para a realização da impressão 3D, tal como a estereolitografia, pioneira na área, capaz de criar objetos poliméricos e metálicos por síntese (CAMPBELL et al., 2011). Na impressão 3D, especialmente no método FDM, os materiais mais frequentemente empregados são polímeros como o *ABS* (acrilonitrila butadieno estireno), o *PLA* (ácido polilático), o *TPU Flex* (Poliuretano Termoplástico Flexível) e o *PETG* (Polietileno Tereftalato de Etileno Glicol). Cada um destes materiais tem propriedades diferentes que influenciam no passo 3 do processo de impressão 3D. Uma característica que vale ressaltar são as temperaturas do extrusor e da mesa de impressão utilizadas com os materiais citados anteriormente. Na Figura 18, há alguns exemplos de

Figura 16 – Processo de Modelagem por Fusão e Deposição (FDM).



Fonte: (CAMPBELL et al., 2011, p. 4)

Figura 17 – (a) Rolos de filamentos de cores diversas; e (b) Impressora 3D do tipo FDM modelo Original Prusa MK4.



(a)

(b)

Fonte: (PRUSA, 2024)

temperaturas que se pode utilizar, mas não há um valor exato, então, cabe ao responsável pela impressão, definir as temperaturas adequadas para cada projeto.

As vantagens da manufatura aditiva através da impressão 3D para esta pesquisa se concentram, principalmente, na habilidade de rapidamente prototipar modelos e na facilidade de substituir componentes. Consequentemente, será viável examinar e aprimorar o modelo a ser empregado no robô móvel do tipo *Car-Like* até que ele atenda às necessidades da pesquisa. Além disso, peças danificadas podem ser prontamente substituídas, contribuindo para a continuidade do experimento.

Figura 18 – Exemplos de temperaturas para os materiais *PLA*, *ABS*, *TPU Flex* e *PETG*.

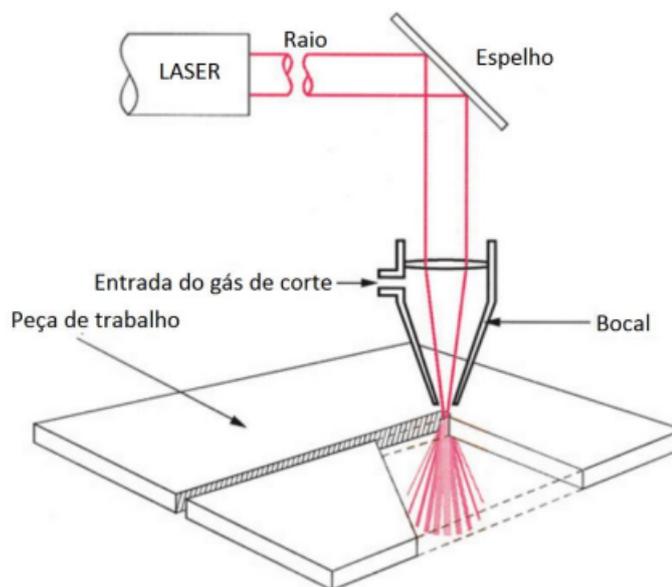
Materiais	Temperatura (°C)	
	Bico	Mesa
PLA	185	55
ABS	235	90
TPU Flex	210	55
PETG	240	80

Fonte: Adaptado de [Silva et al. \(2021\)](#)

2.4.2 Corte a laser

Segundo [Dias \(2015\)](#), um dos equipamentos que constituem um laboratório de fabricação digital é a máquina de corte a laser. Esses equipamentos oferecem amplas oportunidades para a criação de protótipos e podem ser empregados de maneira eficaz para apoiar métodos de ensino-aprendizagem baseados em projetos nos cursos de engenharia. O esquema deste tipo de máquina pode ser visto na [Figura 19](#).

Figura 19 – Esquema de corte a laser.



Fonte: ([DIAS, 2015](#))

Pode-se destacar algumas vantagens do uso de máquinas de corte a laser ([DIAS, 2015](#)) como:

- Elevada precisão;

- Excelente qualidade da superfície cortada;
- Baixos níveis de deformação
- Zona termicamente afetada (ZTA) reduzida;
- Velocidade de corte elevada;
- Versatilidade em função da variedade de materiais;
- Corte de figuras geométricas complexas;

Porém, há algumas desvantagens como:

- Alto custo de investimento;
- Liberação de produtos tóxicos;
- Formação de óxido;
- Reque grande espaço físico;

Foi utilizada uma máquina de corte a laser para fabricar algumas peças do RMR que seriam mais demoradas caso fossem feitas em impressão 3D, mas é importante ressaltar que estas peças poderiam ser impressas também.

2.5 Hardware

Para adquirir as informações de entrada do sistema e implementar um sinal de controle, este estudo empregará dispositivos eletrônicos: sensores e atuadores, ao mesmo tempo em que incorporará os algoritmos e rotinas em um microprocessador.

2.5.1 Sensores digitais

Um sensor é um dispositivo que consegue identificar mudanças de magnitude em uma determinada propriedade física e transforma essa alteração em um sinal utilizável por sistemas de medição ou controle (ALCIATORE; HISTAND, 2012). O resultado emitido por sensores é tipicamente convertido por um transdutor em formas pneumáticas ou elétricas. Esses sinais são de natureza analógica, portanto, para que possam ser processados por sistemas digitais, é necessário empregar um conversor analógico-digital (ADC).

Dessa forma, sensores digitais são caracterizados como dispositivos cuja saída consiste em uma propriedade de valor discreto e frequência de amostragem fixa, geralmente expressa como tensão ou corrente elétrica. Esses valores são estabelecidos em intervalos

predeterminados pela resolução do ADC e variam dentro de um intervalo entre um valor mínimo e máximo. Isso oferece uma notável vantagem em relação aos sensores analógicos, visto que são menos vulneráveis a interferências de sinal.

Na seleção de um sensor apropriado, usualmente se levam em consideração algumas de suas características, como faixa de medição, tipo de sinal de saída, frequência máxima de amostragem, requisitos de alimentação e a necessidade de circuitos complementares.

Os sensores desempenham um papel fundamental em sistemas de controle, pois possibilitam a avaliação da discrepância entre a saída atual do sistema e seu ponto de referência, que é identificado como o erro do sistema. Esses dispositivos podem ser incorporados em diferentes pontos da configuração de controle, mas normalmente são inseridos no circuito de realimentação em sistemas de malha fechada.

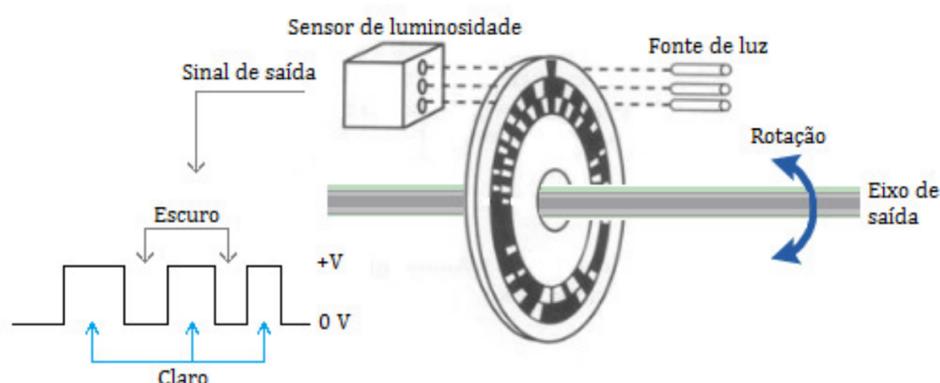
2.5.1.1 Encoders

No mercado, existem diversos tipos de *encoders*, que podem ser categorizados em duas principais categorias: *encoders* incrementais e *encoders* absolutos. Como os próprios nomes sugerem, os *encoders* incrementais mensuram a taxa de rotação e podem inferir a posição relativa. Por outro lado, os *encoders* absolutos mensuram diretamente tanto a posição quanto a velocidade angular absoluta. Ao comparar esses dois tipos, observa-se que os *encoders* incrementais são mais simples de usar e possuem um custo mais acessível, conforme apontado por [Everett \(1995\)](#).

Dentre os tipos mais comuns de *encoders* incrementais, destacam-se os ópticos e os indutivos. Um *encoder* óptico é comumente construído em um disco transparente feito de material plástico, o qual apresenta áreas que permitem a passagem de luz e outras que a bloqueiam. A [Figura 20](#) representa um exemplo de um *encoder* óptico conectado ao eixo de um atuador, sendo utilizado para mensurar a velocidade desse atuador.

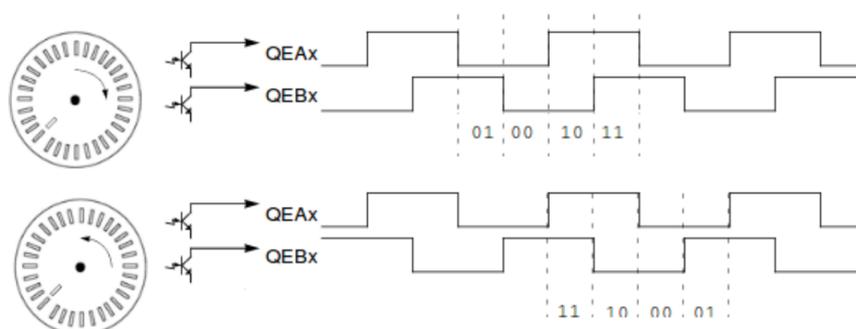
Nesse arranjo, o sensor é deslocado pelo eixo de saída de um atuador. Uma extremidade do *encoder* abriga uma fonte luminosa, enquanto na extremidade oposta, de maneira simétrica, se encontra um sensor de luminosidade. Devido à rotação da porção móvel do dispositivo, a luz emitida pela fonte luminosa passa através das áreas transparentes da estrutura, mas é bloqueada quando incide sobre as regiões opacas do disco. O sensor de luminosidade registra tanto o bloqueio quanto a passagem da luz. Quando a luminosidade é detectada, um sinal de nível lógico alto é gerado, enquanto a ausência de detecção resulta em um sinal de nível lógico baixo. Consequentemente, o sensor produz um conjunto de pulsos digitais formados pelos dígitos '0' e '1', como exemplificado na [Figura 20](#). Através desse sinal, é possível estabelecer uma correspondência entre a quantidade de pulsos contados e a velocidade de rotação do rotor.

Os encoders podem ser categorizados como: simples, de quadratura e absoluto.

Figura 20 – *Encoder* óptico conectado ao eixo rotativo de um atuador.

Fonte: (RODRIGUES, 2017, p. 30)

Um *encoder* simples gera somente um sinal, o qual possibilita calcular o deslocamento relativo do rotor a partir de contagens de pulsos, tendo como referência um ponto inicial. Por outro lado, um *encoder* de quadratura emite dois sinais com uma defasagem de 90 graus entre eles, o que justifica a própria denominação dada ao dispositivo. Esses sinais consistem em sequências de pulsos originados de dois canais, geralmente nomeados como canal A e canal B. Ao contrário do *encoder* simples, que apenas com um canal permite calcular a posição relativa e a velocidade do servo motor através da contagem de pulsos do sensor, o *encoder* de quadratura, com a presença do segundo canal, também possibilita a determinação do sentido de rotação do atuador. A Figura 21 apresenta uma representação dos sinais produzidos pelo *encoder* de quadratura.

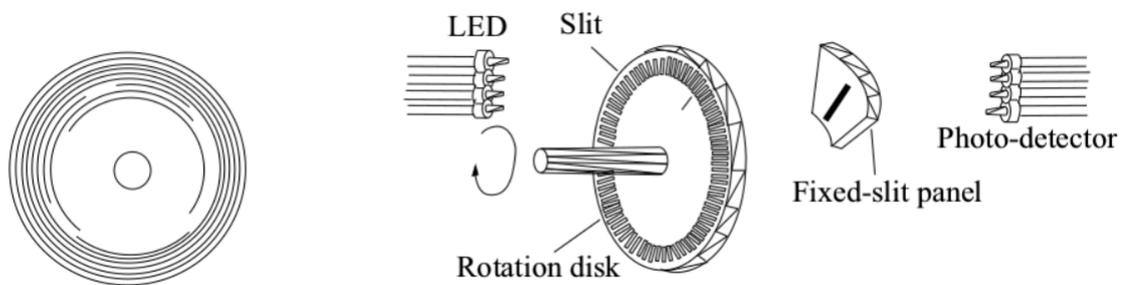
Figura 21 – Sinais gerados por um *encoder* de quadratura.

Fonte: (RODRIGUES, 2017, p. 31)

O princípio óptico é empregado na criação de uma variante mais precisa de *encoder* conhecida como *encoder* absoluto. Esse tipo de *encoder* absoluto inclui múltiplas faixas

contendo segmentos transparentes e opacos. Cada faixa é associada a um sensor de luminosidade específico, o que resulta em uma codificação digital para cada posição. Ainda que forneça um valor absoluto de posição, o custo significativo associado a esse tipo de sensor restringe sua aplicação, tornando os *encoders* incrementais mais amplamente utilizados. A [Figura 22](#) oferece uma representação da disposição dos componentes para o funcionamento de um *encoder* absoluto.

Figura 22 – Funcionamento do *encoder* absoluto.



Fonte: ([RODRIGUES, 2017](#), p. 31)

Um *encoder* incremental, como o caso dos *encoders* simples e os *encoders* em quadratura, pode ser criado usando também os princípios da indução eletromagnética. Sensores dessa categoria são conhecidos como *encoders* indutivos. Nesses aparelhos, o princípio da indução é aplicado para calcular a posição relativa entre um rotor e um estator. No entanto, os sinais elétricos que eles produzem se assemelham àqueles gerados por um *encoder* óptico. A [Equação 2.17](#) define o ângulo de rotação, indicado por $\Delta\theta$. Nessa equação, a variável n_R representa o número de pulsos por rotação do *encoder*. Essa informação é normalmente fornecida nas especificações técnicas do dispositivo, pois depende das características da fabricação do sensor. A variável n_L se refere ao número de pulsos capturados pelo sensor.

$$\Delta\theta = 360^\circ \frac{n_L}{n_R} \quad (2.17)$$

A [Equação 2.18](#) fornece a velocidade angular $\Delta\omega$ do disco do sensor. Dessa forma, deve-se definir um intervalo de tempo Δt , o qual corresponde ao intervalo entre as leituras realizadas pelo sensor infravermelho no caso de *encoders* do tipo óptico. Então:

$$\Delta\omega = \frac{\Delta\theta}{\Delta t} \quad (2.18)$$

2.5.1.2 Sensor IMU

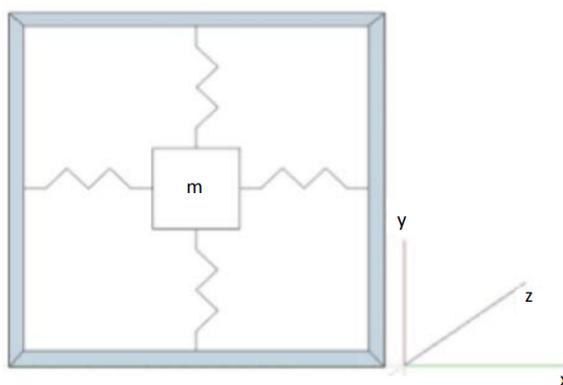
Os sensores IMU têm normalmente um sensor de acelerômetro e um giroscópio ([FILIPPESCHI et al., 2017](#)). Segundo [Faisal, Purboyo e Ansori \(2019\)](#), o sensor IMU é

normalmente utilizado em manobras de aeronaves, incluindo veículos aéreos não tripulados e aeronaves espaciais, incluindo *landers* e satélites. O acelerômetro é um instrumento automático para medir a aceleração, detectar e medir a vibração e medir a aceleração devida ao corpo (inclinação). O acelerômetro pode ser utilizado para medir as vibrações em automóveis, motores, edifícios e instalações de segurança. Os giroscópios são dispositivos que se montam numa estrutura e podem e pode detectar uma velocidade angular se essa estrutura estiver rodando. Existem algumas classes de giroscópios, consoante o funcionamento físico e da tecnologia envolvida.

Um acelerômetro também tem a capacidade de realizar medições em dispositivos eletrônicos, como jogos tridimensionais, mouses de computador e smartphones, além de ser empregado na detecção de atividade sísmica. Em robôs móveis, por exemplo, com a informação de aceleração obtida por meio de um sensor IMU, pode-se obter a velocidade do robô calculando a integral dos valores de aceleração e até mesmo a posição integrando-se a velocidade.

Segundo [Faisal, Purboyo e Ansori \(2019\)](#), um acelerômetro MEMS (*Micro-Electro-Mechanical Systems*) numa dimensão pode ser visto basicamente como um processo de funcionamento da segunda lei de *Newton* em que uma massa está ligada a uma mola num quadro de referência conforme a [Figura 23](#).

Figura 23 – Ilustração do princípio básico da segunda lei de *Newton* em duas dimensões, em que uma massa m está presa a molas num referencial.



Fonte: ([FAISAL; PURBOYO; ANSORI, 2019](#), p. 2)

[Faisal, Purboyo e Ansori \(2019\)](#) afirmam ainda que as seguintes características devem ser consideradas quando se for escolher um acelerômetro para alguma aplicação:

- **Largura de banda (Hz):** indica o *range* de vibrações de frequência que é respondida pelo acelerômetro ou, por vezes, a melhor leitura que pode ser feita.

- **Sensibilidade** ($LSBg^{-1}, mVg^{-1}$): tem a função de quantificar o menor sinal que pode ser detectado ou a mudança de um sinal elétrico de saída por mudança de um sinal elétrico de entrada. É válida para um frequência.
- **Solidez do ruído de tensão** ($\mu g/SQRTHz$): é a mudança com raiz quadrada que são o inverso da largura de banda. Quanto mais rápido se lê a variação do acelerômetro, pior pode ficar a leitura. O ruído tem maior influência no desempenho do acelerômetro.
- **Tensão Zero-G**: é o termo especificado para o *range* de frequências que se pode esperar que sejam emitidos abaixo de acelerações de $0g$.
- **Frequência de resposta (Hz)**: é especificada por uma tolerância de banda ($\pm 5\%$, etc.) que o sensor irá perceber um movimento e indicar o resultado correto. Uma tolerância de banda permite ao consumidor contar a sensibilidade dos dispositivos a partir de referências de frequência em qualquer gama de frequências.
- **Range dinâmico (g)**: é o intervalo entre a menor amplitude detectável que pode ser medida pelo acelerômetro e a maior amplitude antes da distorção ou resultado cortado do sinal.

Diversas categorias de giroscópios estão disponíveis, variando conforme os princípios físicos e tecnologias aplicados. O giroscópio pode ser empregado de forma independente ou integrado a sistemas complexos, tais como a Unidade de Medida Inercial (IMU), bússola giroscópica, sistema de referência de direção e atitude, além de sistemas de navegação.

O princípio de funcionamento de um giroscópio é a rotação em torno dos respectivos eixos, ilustrado na [Figura 24](#). O giroscópio introduz alterações de capacitância para detectar estas deslocamentos. Com base neste fato, a velocidade angular da IMU pode ser medida e, integrando o sinal, pode-se obter a orientação.

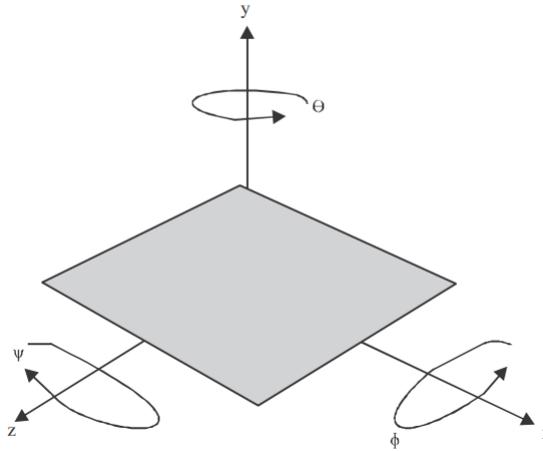
2.5.2 Atuadores

Atuadores são os componentes de um sistema mecatrônico que possuem a capacidade de intervir em uma variável, seja por meio de um movimento físico, como o ato de abrir e fechar uma válvula, ou ao ajustar a saída de um sinal, como tensão ou corrente elétrica ([ALCIATORE; HISTAND, 2012](#)).

2.5.2.1 Ponte H

É um circuito necessário para efetuar o controle de motores DC através de um microcontrolador, pois, em geral, os microcontroladores não conseguem fornecer a potência elétrica adequada para o funcionamento do motor ([Tavares \(2022\)](#)).

Figura 24 – Ilustração do princípio básico de um giroscópio e como as rotações (Φ , Θ , Ψ) ocorrem ao redor do respectivo eixo.



Fonte: (FAISAL; PURBOYO; ANSORI, 2019, p. 3)

Na configuração mais fundamental de montagem, excluindo componentes adicionais que assegurariam uma proteção adicional tanto ao microcontrolador quanto ao motor, o circuito é construído empregando quatro chaves. Essas chaves podem abranger vários elementos eletrônicos, como transistores ou relés, que apresentam um comportamento de ligar e desligar.

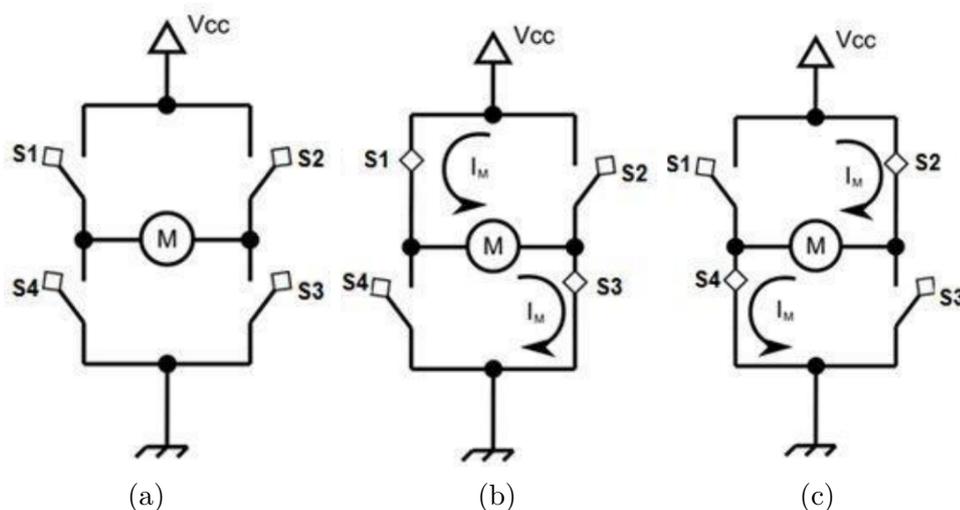
Considerando que as chaves sejam identificadas como S1, S2, S3 e S4, conforme ilustrado na Figura 25, o ato de acionar duas delas em sequência alternada (ou seja, S1 e S3, ou S2 e S4) através do microcontrolador, define a direção em que o motor irá rodar. Isso possibilita que o robô realize movimentos para frente ou para trás ao executar manobras.

Neste projeto, está previsto o emprego da Ponte H L298N que está na Figura 26. Esta solução é amplamente adotada para fins de gerenciamento de motores, e uma das benefícios de sua aplicação é a ocupação reduzida de espaço, a simplicidade do circuito e a capacidade de possuir dois circuitos H, permitindo a regulação de dois motores simultaneamente.

2.5.2.2 PWM

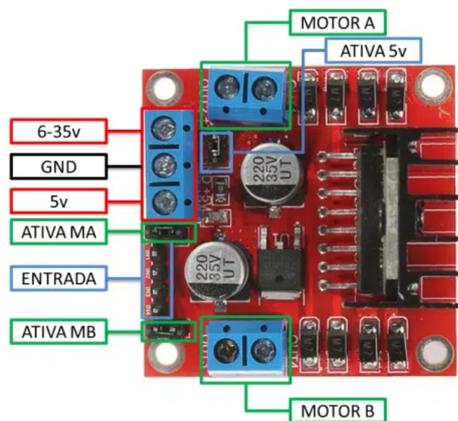
Na técnica de modulação por largura de pulso (*PWM*, do inglês, *Pulse-Width Modulation*), uma fonte de energia de corrente contínua é rapidamente alternada entre dois níveis (como LIGADO e DESLIGADO) a uma frequência fixa f . Essa frequência geralmente excede 1 kHz. O estado “LIGADO” é mantido por um período de tempo variável t_1 dentro de um período constante T , onde $T = 1/f$. A forma de onda resultante, assimétrica, exibe um ciclo de trabalho que representa a proporção entre o tempo em que

Figura 25 – Circuito Ponte H. (a) Situação para o motor ficar parado; (b) Situação para o motor girar em um sentido; (c) Situação para o motor girar no sentido contrário ao de (b)



Fonte: (TAVARES, 2022, p. 94)

Figura 26 – Ponte H L298N.



Fonte: (THOMSEN, 2013)

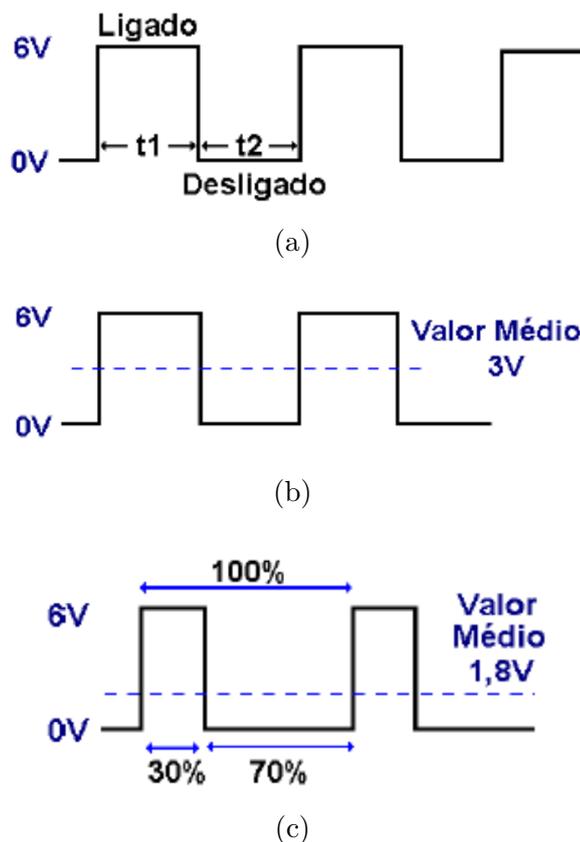
está “LIGADO” e o período completo da forma de onda. À medida que o controlador ajusta o ciclo de trabalho, a corrente média que passa pelo motor é alterada, o que por sua vez provoca modificações na velocidade e no torque de saída (Tavares (2022)).

Na Figura 27(b), está indicada uma situação onde os tempos de estado “LIGADO” e “DESLIGADO”, t_1 e t_2 , são iguais. Dessa forma, obtêm-se ao fim do ciclo o sinal equivalente a média dos sinais aplicados, ou seja, 3 V.

Na Figura 27(c), está indicada uma situação onde os tempos de estado “LIGADO”

e “DESLIGADO”, t_1 e t_2 , estão em proporções diferentes. Nesta situação, $t_1 = 30\%$ e $t_2 = 70\%$, logo o estado “LIGADO” tem a duração t_1 e o estado “DESLIGADO” tem a duração t_2 . Dessa forma, obtêm-se ao fim do ciclo o sinal equivalente a 1,8 V que é obtido por meio da multiplicação simples entre o tempo de estado “LIGADO” e o valor máximo de tensão que é 6V.

Figura 27 – Sinais PWM (a) Sinal PWM; (b) Funcionamento do PWM em 50%; (c) Funcionamento do PWM em 30%



Fonte: (TAVARES, 2022, p. 95)

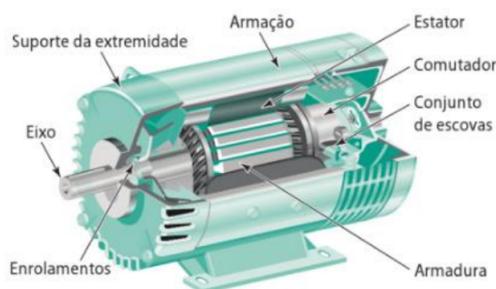
Para efetuar o ajuste de velocidade do motor CC e de ângulo do servomotor do robô móvel proposto nesta pesquisa, será utilizada esta técnica.

2.5.2.3 Motor de corrente contínua

Segundo [Castro \(2016\)](#), os motores de corrente contínua (CC) são dispositivos elétricos que funcionam utilizando um fluxo constante de corrente, conforme implicitamente indicado pelo nome. Ao receber uma tensão contínua nos enrolamentos do motor, é gerada uma força magnetomotriz, que desempenha um papel fundamental na conversão da energia elétrica em energia cinética. Esses motores são frequentemente empregados em situações que demandam um controle preciso da velocidade, especialmente quando é necessário um torque crítico.

A estrutura de um motor de CC é composta por elementos como a armadura, o estator, o comutador, o conjunto de escovas, o eixo e os enrolamentos, como ilustrado na [Figura 28](#). Dependendo da configuração da construção, os motores CC podem ser categorizados em tipos como motores de ímã permanente, motores *shunt*, motores série e motores compostos, como discutido por [Alciatore e Histan \(2012\)](#). O ajuste da velocidade dos motores CC é alcançado pela modificação da voltagem aplicada ao dispositivo.

Figura 28 – Construção de um motor CC.



Fonte: ([CASTRO, 2016](#), p. 14)

2.5.2.4 Servomotor

Um dos tipos mais utilizados de motor elétrico é o servomotor, pois ele permite controle preciso sobre o ângulo de rotação do eixo ([Bishop \(2017\)](#)). Para esta pesquisa, será utilizado como um dos atuadores o servomotor *DS3218 PRO*, exibido na [Figura 29](#).

Figura 29 – Servomotor DS3218 PRO.

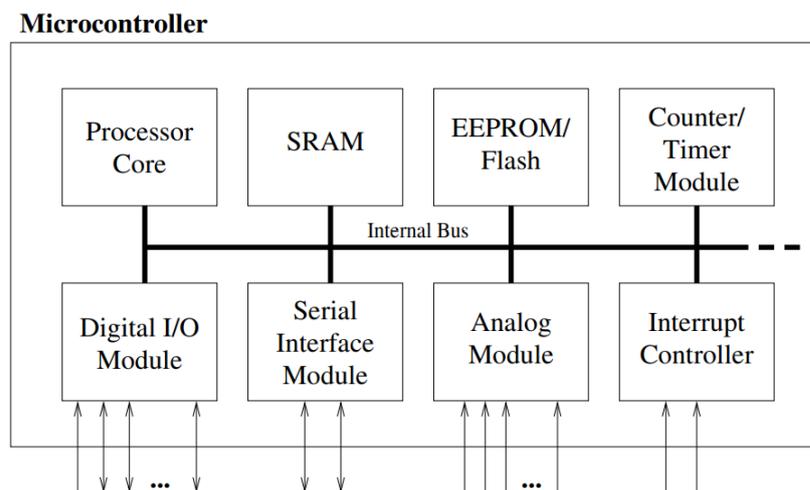


Este servomotor opera ao receber pulsos de modulação de largura de pulso (*PWM*, do inglês *Pulse-Width Modulation*), utilizados para controlar o ângulo de seu eixo. A faixa padrão de amplitude de tensão abrange até 3,3 V ou 5 V. O suprimento de energia situa-se entre 4,8 V e 6,8 V, permitindo que o eixo realize rotações de até 180°.

2.5.3 Microcontroladores

Um microcontrolador é um dispositivo projetado para operar de maneira independente, contendo todos os elementos necessários para suas funções de monitoramento e/ou controle. Além do processador central, ele incorpora memória, diversos controladores de interface, um ou mais dispositivos temporizadores, um controlador de interrupções e, igualmente importante, pinos de Entrada/Saída de uso geral que facilitam a conexão direta com seu entorno conforme mostra a [Figura 30](#). Adicionalmente, os microcontroladores apresentam operações de nível de bits que possibilitam a modificação de um único bit em um conjunto de bits sem afetar os demais ([GRIDLING; WEISS, 2007](#)).

Figura 30 – Estrutura básica de um microcontrolador.



Fonte: ([GRIDLING; WEISS, 2007](#), p. 5)

Nesta pesquisa, deseja-se implementar os controladores do robô móvel em um microcontrolador ESP32 que também será o responsável pela leitura dos sensores e acionamento dos atuadores.

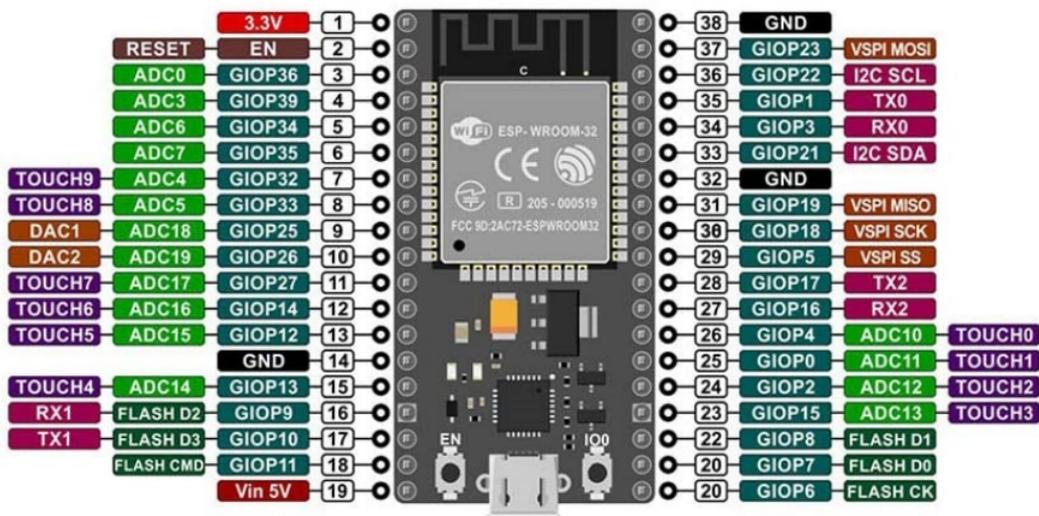
2.5.3.1 ESP32

O microcontrolador ESP32, desenvolvido pela empresa *Espressif Systems*, foi introduzido no mercado em 2016. Ele é conhecido por suas especificações técnicas distintas, incluindo alta capacidade de processamento, facilidade de acesso e capacidades de conectividade. Como resultado, tem sido cada vez mais adotado por programadores e profissionais da área ([SANTOS; JUNIOR, 2019](#)).

O ESP32 é composto por um processador robusto, que pode ser de um único núcleo ou dual-core de 32 bits (com dois núcleos físicos de processamento), e se destaca em várias categorias quando comparado a outros microcontroladores. Além disso, inclui

um limitador de tensão de 5 V para proteção em casos de sobrecarga de alimentação elétrica. Também incorpora um Relógio de Tempo Real (RTC), que mantém a data e hora atualizadas em situações de queda de energia, junto com outros periféricos. A configuração detalhada de seus pinos está representada na [Figura 31](#), juntamente com algumas de suas funcionalidades associadas a esses pinos.

Figura 31 – Pinagem do ESP32.



Fonte: (SILVA, 2021, p. 20)

Devido ao seu tamanho compacto, é conveniente integrar o ESP32 em placas de circuito impresso e outros dispositivos eletrônicos. A programação do ESP32 é realizada utilizando a linguagem *C/C++*, e além de ser possível programá-lo dentro do ambiente próprio chamado *Software Development Kit* (SDK), também é viável programar o microcontrolador através da interface da *Arduino IDE* conforme [Silva \(2021\)](#).

3 Metodologia

Neste capítulo, serão apresentadas as etapas detalhadas para desenvolver o sistema de controle e rastreamento de caminho para o robô móvel *Car-Like*. Isso inclui desde a prototipagem 3D e desenvolvimento do hardware até a implementação do software. Cada fase será explicada com ênfase nas técnicas e ferramentas utilizadas. Estratégias para validação experimental também serão discutidas para garantir a eficácia e confiabilidade do controle proposto. Essa abordagem busca proporcionar uma visão clara das práticas adotadas para atingir os objetivos do estudo.

3.1 Projeto da Plataforma *Car-Like*

Nesta seção, será descrito como foi realizado o projeto mecânico do RMR *Car-Like* utilizado neste trabalho. A abordagem incluirá principalmente a prototipagem mecânica do sistema de direção, que segue a geometria de *Ackermann*, e do sistema de tração diferencial, responsável por reduzir ou eliminar o escorregamento lateral do veículo, além de prevenir o capotamento em curvas em alta velocidade. Além disso, mostrar-se-á o circuito elétrico do sistema, bem como a placa de circuito impresso (PCI) desenvolvida.

3.1.1 Estrutura mecânica

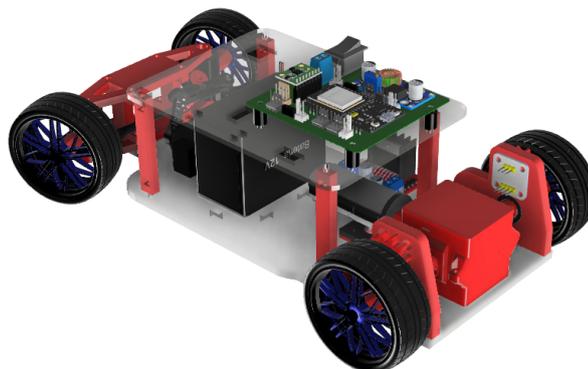
A prototipagem da estrutura mecânica foi feita pensando em fabricar a maioria das peças utilizando métodos de fabricação digital como impressão 3D do tipo FDM e corte a laser, visando uma fácil reposição em caso de desgaste, falha ou modificação no desenho. Foi utilizado o *software Fusion 360* da empresa *Autodesk* para a prototipagem 3D, mas os arquivos *.STL* para impressão 3D, *.PDF* para corte a laser e *.STEP* para qualquer modificação do protótipo 3D estão disponíveis no [repositório do projeto](#).

Na [Figura 32](#) está o protótipo 3D final que foi produzido e utilizado no trabalho. A prototipagem dos principais mecanismos serão explicados em detalhes nas próximas seções.

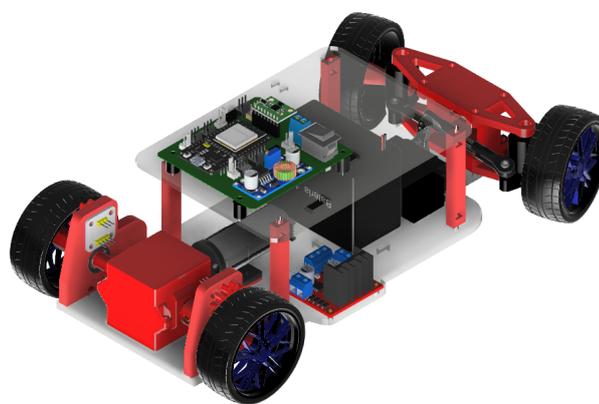
3.1.1.1 Chassi e Sistema de direção

Antes de definir as dimensões do chassi foi desenhado o esboço da geometria do sistema de direção. Conforme mostra a [Figura 33\(a\)](#), percebe-se que as barras de direção, indicadas pelas linhas tracejadas azuis, descrevem uma geometria trapezoidal para atender à geometria de *Ackermann* como mencionado na [subseção 2.1.1.3](#). O próximo passo foi definir a distância entre o eixo frontal e traseira das rodas, a qual mede 248 mm. Além

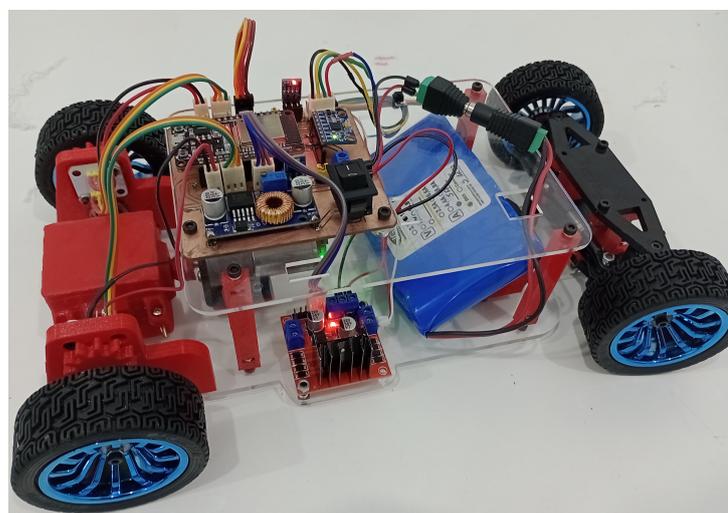
Figura 32 – Robô móvel utilizado neste trabalho.



(a) Perspectiva à esquerda.



(b) Perspectiva à direita.

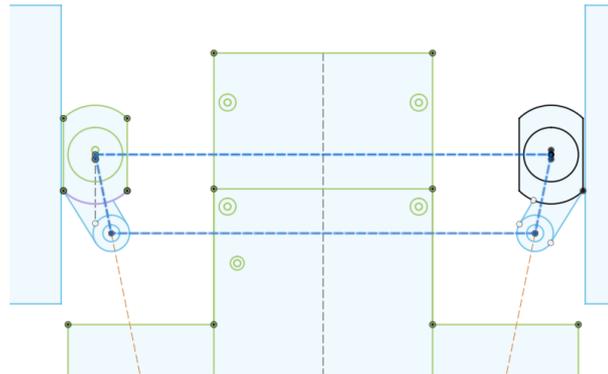


(c) Robô real.

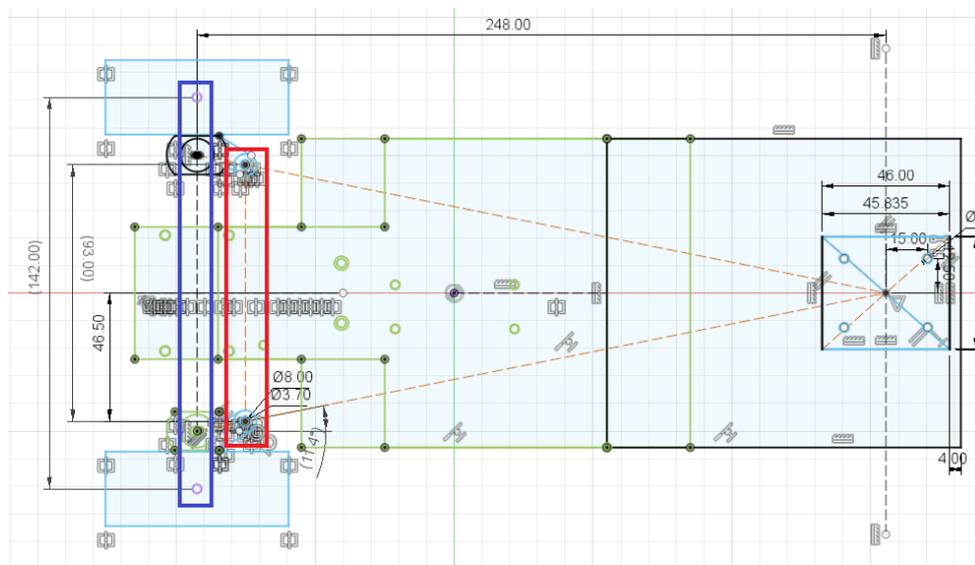
disso, conforme mostra a [Figura 33\(b\)](#), foi possível definir a distância lateral entre as rodas de 142 mm indicada pelo retângulo em azul e o comprimento da barra estabilizadora de 93 mm indicada pelo retângulo em vermelho.

Os esboços criados serviram como referência para prototipar as peças que compõem o sistema de direção conforme mostra a [Figura 34\(a\)](#), que ilustra como a geometria

Figura 33 – Esboço do chassi e do sistema de direção.



(a) Geometria das barras de direção.



(b) Esboço do chassi.

trapezoidal das barras de direção continua sendo atendida. Foi escolhido um servomotor, como o da [Figura 29](#), para atuar neste sistema de direção, e na [Figura 34\(b\)](#) é possível ver como o mecanismo funciona. Na [Figura 34\(c\)](#) é possível ver como ficou a versão real deste mecanismo. Vale ressaltar que todas as peças que compõem a parte mecânica deste sistema de direção, com exceção das hastes de direção, ilustradas na [Figura 35](#), foram impressas em 3D com material PETG para as peças em vermelho e ABS para as que são pretas.

Ao finalizar o sistema de direção, modelou-se o chassi de acordo com os componentes que eram inseridos como a caixa do diferencial, neste caso o motor, a ponte H, os sensores e a bateria. Na [Figura 36\(a\)](#) está o modelo final do chassi. Este foi fabricado por corte a laser em uma chapa de acrílico cristal de 5 mm assim como a peça da [Figura 36\(b\)](#), pois era mais prático e rápido em comparação com impressão 3D, no entanto, é possível imprimi-las e, caso seja feito, sugere-se a utilização de um material resistente como ABS

Figura 34 – Sistema de direção desenvolvido.



(a) Vista superior do mecanismos de direção.

(b) Mecanismo de direção atuado com um servomotor.



(c) Mecanismo de direção real.

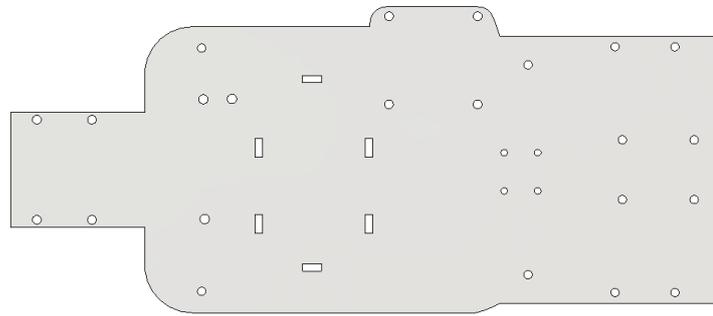
Figura 35 – Haste de direção.



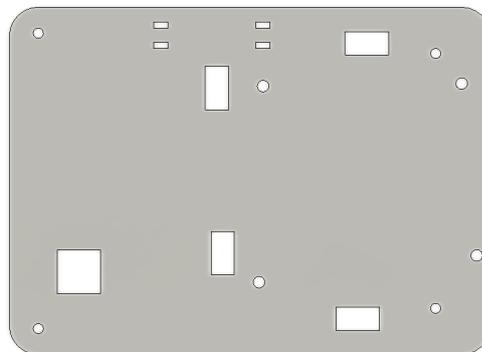
Fonte: ([AMAZON, 2024](#))

ou PETG.

Figura 36 – Peças feitas com acrílico.



(a) Vista superior do Chassi.



(b) Vista superior da base de fixação para a PCI.

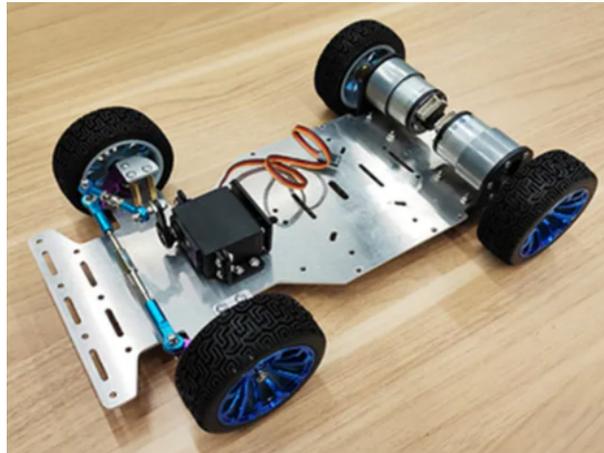
3.1.1.2 Sistema de tração com diferencial

É comum a utilização de dois motores independentes conectados às rodas traseiras de um RMR *Car-Like* no sistema de tração como o mostrado na [Figura 37](#), pois é preciso que as rodas traseiras girem em velocidades diferentes para que o robô consiga executar curvas sem sofrer deslizamento ([COSTA, 2019](#)). Neste tipo de configuração é utilizado um diferencial eletrônico, i.e. via *software*, para que os motores girem em velocidades diferentes como em ([MARTINS et al., 2011](#)).

No caso do RMR utilizado neste trabalho, optou-se por desenvolver um diferencial mecânico que juntamente com apenas um motor CC, compõem o sistema de tração. O diferencial é responsável por distribuir por igual o torque gerado pelo motor aos semieixos de transmissão através de engrenagens ([COSTA, 2019](#)). Foi modelado um diferencial aberto que é o tipo mais comum e está presente em grande parte dos carros de passeio ([COSTA, 2019](#)).

Na [Figura 38\(a\)](#) estão indicados alguns dos componentes de um diferencial. O eixo do motor é conectado diretamente à engrenagem pinhão que transmite o movimento para a coroa, que por sua vez transmite o movimento para os semi-eixos, onde se conectam as

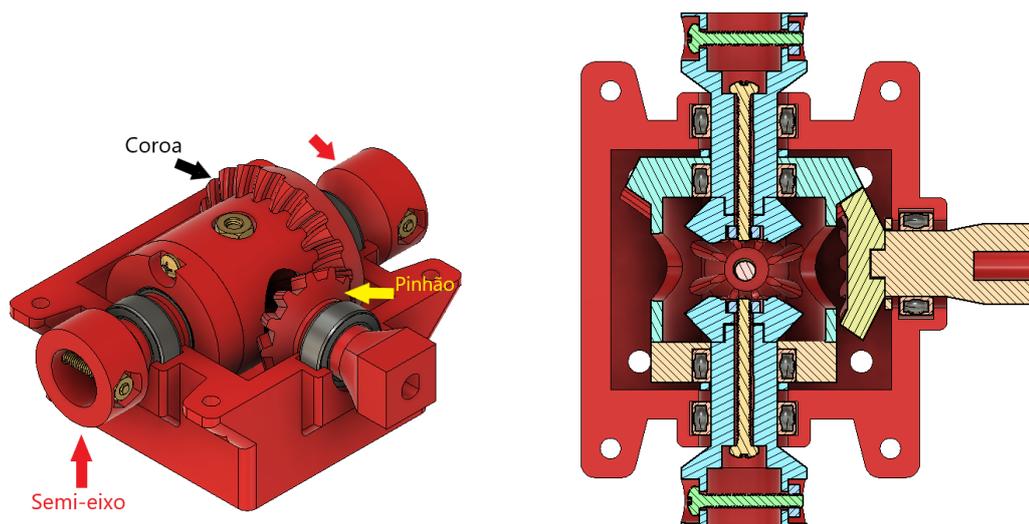
Figura 37 – Chassi genérico de um Car-Like.



Fonte: ([ALIEXPRESS, 2024](#)).

rodas. É importante mencionar que há um conjunto de duas engrenagens planetárias e duas satélites dentro do encapsulamento conforme mostra a [Figura 38\(b\)](#). As satélites são conectadas aos semi-eixos e as planetárias giram em torno de um eixo fixo que no caso é um parafuso. Optou-se pela utilização de parafusos e porcas para montar as engrenagens satélites, ao invés de colar já que são impressas em partes, pois é mais prático de substituir em caso de modificações ou peças danificadas.

Figura 38 – Componentes do diferencial mecânico.



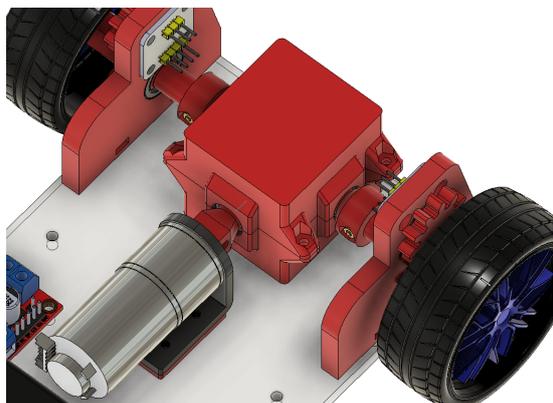
(a) Indicação de alguns dos componentes de um diferencial mecânico.

(b) Visão do corte superior do diferencial.

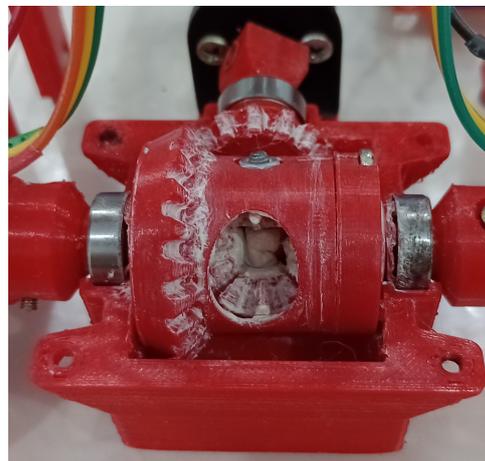
O sistema de tração com diferencial finalizado está ilustrado na [Figura 39\(a\)](#) e, na [Figura 39\(b\)](#) se vê a versão real do diferencial sem a tampa superior para que seja possível uma melhor visualização do mecanismo. Destaca-se que todos os componentes,

com exceção dos parafusos, porcas e rolamentos; foram impressos em 3D utilizando o material PETG. Em relação à estrutura mecânica, o sistema de direção e o de tração são os principais, como o restante dos componentes são simples então não serão destacados.¹

Figura 39 – Sistema de tração com diferencial.



(a) Versão final do sistema de tração com diferencial.



(b) Versão real do diferencial modelado sem a tampa superior.

3.1.2 Estrutura eletrônica

O sistema eletrônico desta plataforma *Car-Like* é constituído, principalmente, por um motor CC com *encoder* incremental, um servomotor, um sensor IMU, dois *encoders* absolutos, uma ponte H, conversores de tensão e um microcontrolador. A partir disto foi feito um esquema elétrico com a ligação de todos estes componentes conforme mostra a [Figura 40](#).

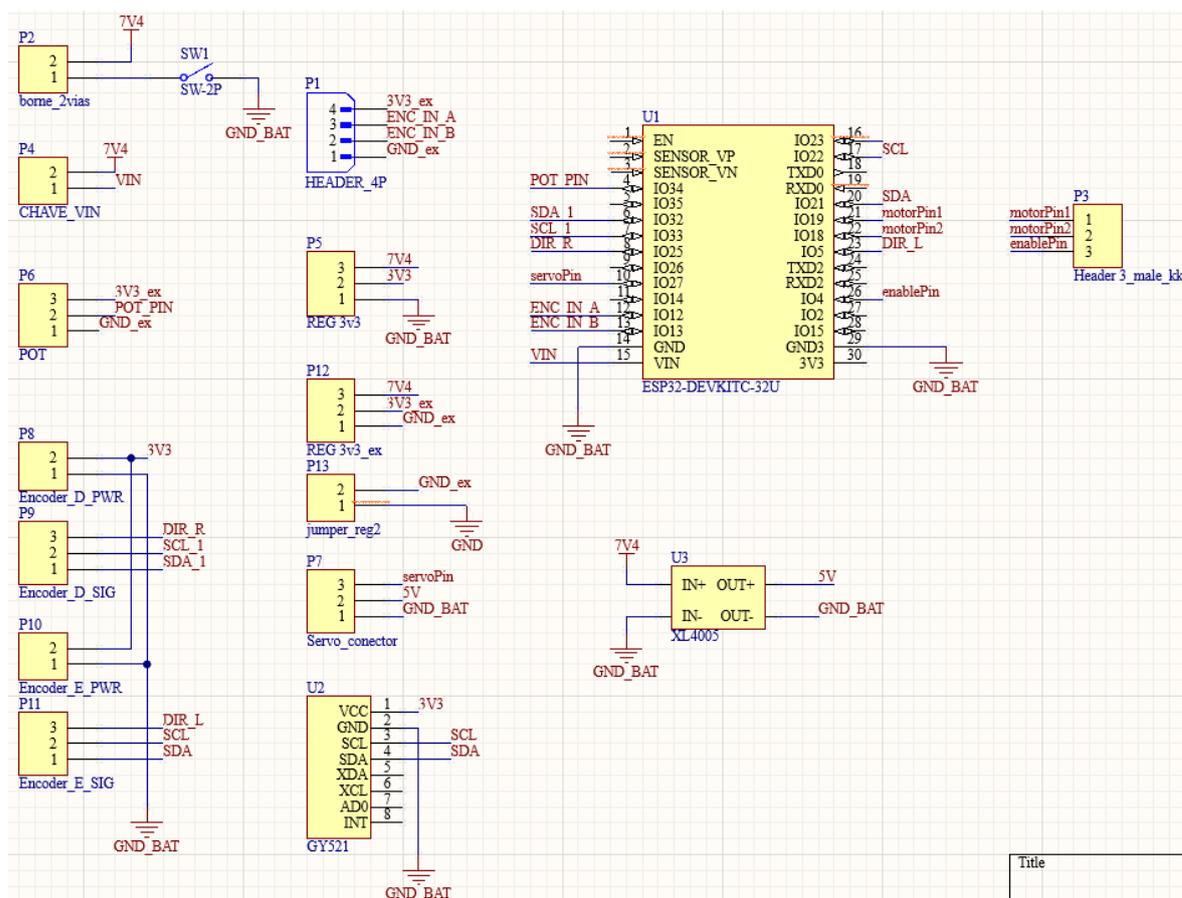
Foi utilizado o *software Altium Designer* para a montagem deste esquema elétrico e posteriormente para o projeto da PCI. Neste esquema é importante citar a forma como foram ligados os *encoders* absolutos (AS5600) e o sensor IMU (MPU6050), visto que ambos utilizam comunicação I2C. Como o microcontrolador (ESP32) utilizado possui 2 barramentos I2C, foi possível ligar um AS5600 em cada barramento já que eles possuem o mesmo endereço então não poderiam estar no mesmo barramento, enquanto que o MPU6050 compartilha o barramento I2C com o AS5600 instalado do lado esquerdo da traseira do RMR.

Para o acionamento do motor e ajuste de velocidade por PWM, foi utilizada a ponte H L298N, da qual foi necessário utilizar apenas um canal já que há somente um motor. O Servomotor utilizado é do modelo DS3218 PRO que possui uma tensão de alimentação entre d 4,8 V a 6,8 V, enquanto que o motor CC é de 6 V, os encoders e o sensor IMU opera a 3,3 V. Vale mencionar que o sistema todo é alimentado por uma bateria de lítio de 12 V, então foi necessário utilizar reguladores de tensão *Ams1117* para 3,3 V e um módulo

¹ A lista dos materiais utilizados pode ser encontrada no [repositório do projeto](#).

regulador de tensão *Step-Down* XL4005 ajustado para 5 V alimentando o servomotor. A alimentação do motor CC é feita por meio da ponte H que é alimentada diretamente pela bateria, então por segurança, a tensão de saída que vai para o motor foi ajustada limitando o valor de PWM no código do microcontrolador.

Figura 40 – Esquema elétrico do RMR *Car-Like*.

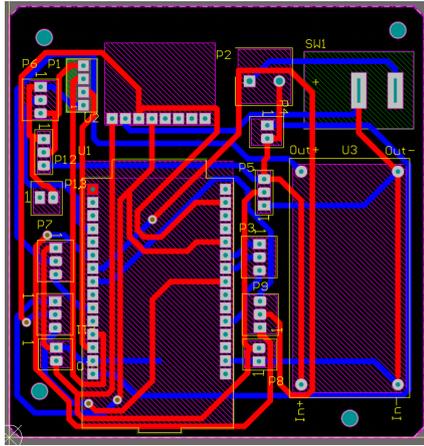


O esquema elétrico da Figura 40 foi montado em uma *protoboard* e alguns testes foram realizados para validação. Após este período de teste, o circuito foi aprovado então projetou-se uma PCI para organizar e agrupar todos os componentes de forma mais organizada e até modular de certa forma como mostra a Figura 41(b). Note que a PCI projetada é de dupla face conforme mostrado na Figura 41(a).

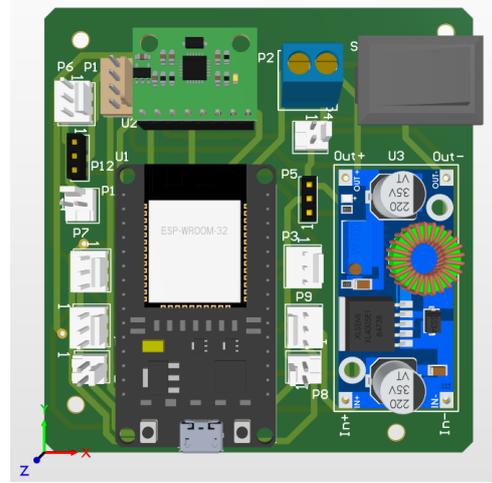
Por fim, fabricou-se a PCI apresentada na Figura 42(a) antes de os componentes serem soldados. O passo seguinte foi a confecção dos conectores e soldagem dos componentes para se obter a PCI da Figura 42(b).²

² O projeto do *Altium Designer* com as bibliotecas utilizadas e os arquivos *Gerber* para fabricação também estão disponíveis no [repositório do projeto](#).

Figura 41 – PCI projetada no *Altium Designer*.

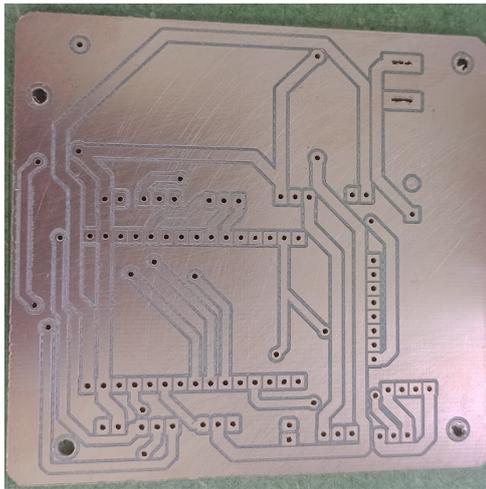


(a) Trilhas do *bottom* e *top layer* da PCI projetada.

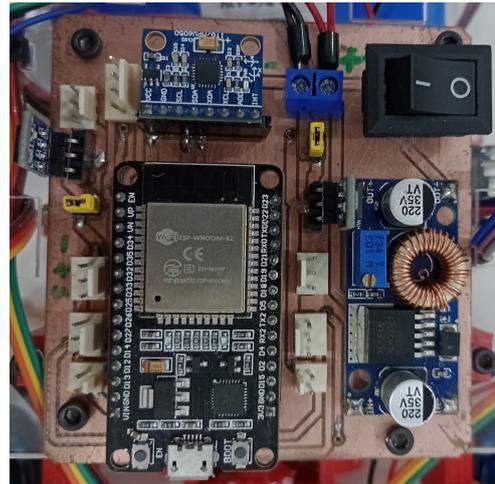


(b) Modelo 3D da PCI projetada.

Figura 42 – PCI após o término da fabricação.



(a) Trilhas do *bottom* e *top layer* da PCI projetada.



(b) PCI finalizada.

3.2 Síntese de Controladores Robusto para Sistemas de Tempo Discreto Incertos

Partindo do critério de estabilidade de *Lyapunov* visto na [subseção 2.3.1](#), pode-se obter controladores que estabilizem um sistema com a presença de incertezas. Considerando um sistema dinâmico linear incerto na forma:

$$x(k + 1) = Ax(k) + Bu(k) \tag{3.1}$$

em que $x(k) \in \mathbb{R}^n$ é o vetor de estados, $u(k) \in \mathbb{R}^p$ é o vetor de entradas de controle, e

$$(A, B) \in \rho = \{(A, B) \mid A = \sum_{i=1}^{\kappa} \gamma_i A_i, \quad B = \sum_{i=1}^{\kappa} \gamma_i B_i\}$$

com $\gamma_i \geq 0$, $\sum_{i=1}^k \gamma_i = 1$, $A_i \in \mathbb{R}^{n \times n}$, $B_i \in \mathbb{R}^{n \times p}$. Considere uma lei de controle por realimentação de estados como:

$$u(k) = Kx(k), \quad K \in \mathbb{R}^{p \times n} \quad (3.2)$$

De acordo com [Dullerud e Paganini \(2013\)](#), pode-se escrever o sistema incerto em malha fechada substituindo-se (3.2) em (3.1) para obter:

$$x(k+1) = (A + BK)x(k) \quad (3.3)$$

Observe que o sistema em malha fechada se trata de um sistema autônomo, em que a matriz em malha fechada é dada por:

$$A_f = A + BK$$

Considerando que o sistema é controlável, pode-se encontrar um ganho K que estabiliza o sistema (3.3) por meio de uma LMI. Sabendo que o sistema incerto é estável se as LMIs do item 2 forem factíveis, pode-se substituir A_f nas desigualdades para obter:

$$\begin{cases} P = P^T > 0 \\ (A + BK)^T P (A + BK) - P < 0, \quad \forall (A, B) \in \rho \end{cases} \quad (3.4)$$

Assim, a estabilização do sistema incerto por meio do controlador de retroalimentação de estados é viável quando as desigualdades em (3.4) são atendidas. No entanto, é notável que a matriz de realimentação K também é uma variável no problema, o que confere uma natureza não-linear à segunda desigualdade em (3.4). Como solução para este problema, utiliza-se o complemento de Schur, que é um artifício para converter uma desigualdade, convexa, não-linear em uma LMI ([BOYD et al., 1994](#)).

Aplicando o complemento de Schur na segunda desigualdade de (3.4), obtém-se:

$$\begin{bmatrix} P & (A + BK)^T \\ A + BK & P^{-1} \end{bmatrix} > 0$$

Aplicando uma transformação de similaridade, i.e. multiplicando à direita por $\text{diag}\{P^{-1}, I\}$ e à esquerda pela sua transposta, temos que

$$\begin{bmatrix} P^{-1} & 0 \\ 0 & I \end{bmatrix}^T \begin{bmatrix} P & (A + BK)^T \\ A + BK & P^{-1} \end{bmatrix} \begin{bmatrix} P^{-1} & 0 \\ 0 & I \end{bmatrix} > 0$$

então

$$\begin{bmatrix} P^{-1} P P^{-1} & P^{-1} (A + BK)^T \\ (A + BK) P^{-1} & P^{-1} \end{bmatrix} > 0$$

Finalmente fazendo-se o uso das variáveis linearizantes, substituindo as não-linearidades por uma única variável, para obter a solução do problema de forma indireta. Então, $Y \triangleq P^{-1}$ e $Z \triangleq KP^{-1} = KY$, tem-se que:

$$\begin{bmatrix} Y & YA^T + Z^T B^T \\ AY + BZ & Y \end{bmatrix} > 0 \quad (3.5)$$

Assim, o sistema linear incerto (3.1) é estabilizável pela lei de controle por realimentação de estados se existirem a matriz Y e Z que satisfaça seguinte LMI:

$$\begin{bmatrix} Y & YA^T + Z^T B^T \\ AY + BZ & Y \end{bmatrix} > 0, \quad \forall (A, B) \in \rho \quad (3.6)$$

Note que a primeira restrição de (3.4) já é garantida pela restrição (3.6), uma vez que $Y > 0 \implies P > 0$.

A matriz de realimentação pode ser recuperada fazendo $K = ZP$ onde $P = Y^{-1}$. Esta seção demonstrou o procedimento para criar um controlador robusto utilizando LMIs para um sistema dinâmico com incertezas. Da mesma forma que na subseção 2.3.1, a estabilidade é garantida se a LMI (3.6) é factível para todos os vértices do politopo ρ .

3.3 Sistema de Controle Lateral

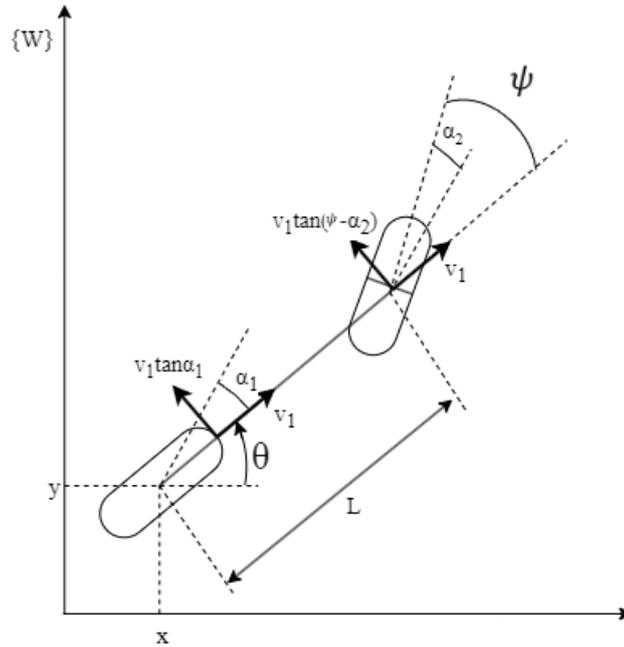
Conforme mencionado na subseção 2.2.2, o controle lateral é responsável principalmente por atuar sobre o esterçamento das rodas dianteiras em um problema de rastreamento de caminho. Vale ressaltar que há trabalhos que levam em consideração também o controle longitudinal neste tipo de problema para que o RMR possua uma velocidade adequada para cada ponto desejado do caminho tal qual é apresentado em Rashad (2012). No entanto neste trabalho, focou-se somente no controle lateral.

O modelo cinemático do RMR utilizado foi baseado no apresentado na Equação 2.1, mas considerando o efeito dos ângulos de deslizamento das rodas dianteiras e traseiras. Este efeito é um dos responsáveis por fazer com que o robô não consiga realizar movimentos laterais precisos. Considerando então que o RMR pode ser representado por um modelo do tipo bicicleta, mas agora considerando os efeitos de deslizamento como visto na Figura 43, pode-se reescrever o modelo cinemático 2.1, (AROGETI; BERMAN, 2012), como:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{y}(t) \\ \dot{\theta}(t) \\ \dot{\psi}(t) \end{bmatrix} = \begin{bmatrix} \cos \theta(t) - \tan \alpha_1 \sin \theta(t) \\ \sin \theta(t) + \tan \alpha_1 \cos \theta(t) \\ \frac{\tan(\psi(t) - \alpha_2) - \tan \alpha_1}{L} \\ 0 \end{bmatrix} v_1(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} v_2(t), \quad (3.7)$$

em que $x(t)$ e $y(t)$ descrevem a posição do robô, $\theta(t)$ é a orientação e $\psi(t)$ é o esterçamento do RMR, $v_1(t)$ é a velocidade linear do robô e $v_2(t)$ é a taxa de variação do ângulo de esterçamento, além disso há os ângulos de deslizamento das rodas dianteiras, α_1 , e das rodas traseiras, α_2 , que são as incertezas do modelo.

Figura 43 – Diagrama esquemático de um modelo tipo bicicleta com deslizamento.



Fonte: Adaptado de [Arogeti e Berman \(2012\)](#).

Além do modelo cinemático do robô, o efeito do ângulo de deslizamento também foi considerado no modelo 2.4 sabendo que $v_y = v_x \tan \alpha_1$, obteve-se:

$$\begin{cases} \dot{e} = v_x \sin \theta_e + v_x \tan \alpha_1 \cos \theta_e \\ \dot{\theta}_e = v_x \frac{\tan(\psi - \alpha_2) - \tan \alpha_1}{L} - v(\sigma) v_x \end{cases} \quad (3.8)$$

Dessa forma considerando a [Equação 3.8](#), o problema de seguimento de caminho pode ser resolvido como um problema de regulação a partir de um controlador que estabilize os estados e e θ_e . No entanto, note que o modelo é não-linear, sendo necessária sua linearização.

Embora seja comum desconsiderar todas as interferências externas ao buscar o ponto de equilíbrio de um sistema durante a linearização, é viável explorar a estabilização do sistema em um ponto de equilíbrio alternativo na presença de perturbações externas. Esse fenômeno ocorre em sistemas que possuem múltiplos pontos de equilíbrio, como é o caso do sistema de seguir caminho de um robô *Car-Like* ([EMAM; FAKHARIAN, 2017b](#)).

Para encontrar o ponto de equilíbrio $X_{eq} = [e_{eq} \ \beta_{eq}]$, em que $\beta = \theta_e$ apenas para facilitar a nomenclatura do ponto de equilíbrio, define-se $\dot{X} = 0$ então:

$$\begin{aligned} \dot{e} &= 0 \\ v_x \sin \beta_{eq} + v_x \tan \alpha_1 \cos \beta_{eq} &= 0 \\ \frac{\sin \beta_{eq}}{\cos \beta_{eq}} &= -\tan \alpha_1 \\ \tan \beta_{eq} &= -\tan \alpha_1 \\ \beta_{eq} &= -\alpha_1 \end{aligned}$$

e como α_1 e α_2 serão considerados iguais como em (EMAM; FAKHARIAN, 2017b)

$$\begin{aligned} v_x \frac{\tan(\psi - \alpha_2) - \tan \alpha_1}{L} - v(\sigma) &= 0 \\ \dot{\theta}_e &= 0 \end{aligned}$$

Como não há restrições para no valor de e_{eq} , pode-se assumir qualquer valor real. Portanto, o ponto de equilíbrio do modelo do robô *Car-Like* com a presença de perturbações externas é definido como $[e_{eq} \ \beta_{eq}] = [0 \ -\alpha_1]$. Assim o modelo 3.8 linearizado é:

$$\begin{cases} \dot{e} = v_x \cos \alpha_1 + v_x \tan \alpha_1 \sin \alpha_1 \\ \dot{\theta}_e = \frac{v_x}{L} \psi \end{cases} \quad (3.9)$$

A representação em espaço de estados é:

$$\begin{bmatrix} \dot{e} \\ \dot{\theta}_e \end{bmatrix} = \begin{bmatrix} 0 & v_x \cos \alpha_1 + v_x \tan \alpha_1 \sin \alpha_1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} e \\ \theta_e \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{v_x}{L} \end{bmatrix} \psi \quad (3.10)$$

em que v_x foi definido em malha aberta resultando de um valor constante de *PWM* enviado para o motor do robô.

3.3.1 Odometria

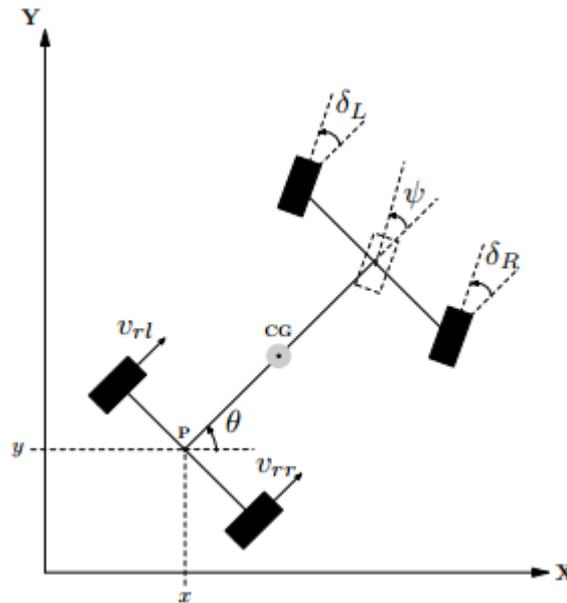
Na robótica móvel, conhecer a localização precisa do robô, ou estimar adequadamente a imprecisão, é primordial para o rastreamento de caminho. Determinar a posição e orientação do RMR é uma tarefa que oferece diversas opções de resolução. Essas opções variam desde a utilização de sensores proprioceptivos básicos, como *encoders* nas rodas e direção, sensores inerciais (*Inertial Measurement Unit* - IMU) e sensores absolutos (GPS), até a incorporação de sensores exteroceptivos, como sonares, câmeras e *lasers*. Além disso, uma abordagem eficaz pode envolver a combinação estratégica dessas diferentes fontes sensoriais para obter uma estimativa mais precisa e robusta da posição e orientação do RMR (MARTINS et al., 2011).

Neste trabalho optou-se por utilizar a odometria diferencial que considera a utilização um *encoder* em cada roda traseira do robô para determinar sua posição e orientação em relação a um ponto de referência fixo. A evolução da posição e orientação do ponto P ilustrado na Figura 44 pode ser representada na forma discreta por:

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{odom,k+1} \end{bmatrix} = \begin{bmatrix} x_k + v \cos(\theta_{odom,k}) \\ y_k + v \sin(\theta_{odom,k}) \\ \theta_{odom,k} + \Psi \times T_s \end{bmatrix} \quad (3.11)$$

em que Ψ é a taxa de variação do ângulo de esterçamento do robô, T_s é a taxa de amostragem e $v = \frac{v_{rl} + v_{rr}}{2}$, onde v_{rl} e v_{rr} são as velocidades da roda traseira esquerda e direita respectivamente.

Figura 44 – Conversão de um robô de quatro rodas no modelo triciclo.



Fonte: Martins et al. (2011).

Martins et al. (2011) apresentam um detalhe importante sobre RMR *Car-Like* que atendem à geometria de *Ackermann* que é o fato de poder representar o modelo tal qual de um triciclo como ilustra a Figura 44, onde o ângulo de esterçamento, ψ , é obtido a partir de uma combinação dos ângulos de esterçamento das rodas, δ_L e δ_R , com a Equação 3.12.

$$\psi = \arctan \left\{ \frac{2 \tan(\delta_L) \tan(\delta_R)}{\tan(\delta_L) + \tan(\delta_R)} \right\} \quad (3.12)$$

3.3.2 Fusão sensorial

A utilização de odometria pressupõe um movimento planar, além de que imprecisões no dimensionamento, deslizamentos, falhas na superfície de contato entre pneu e solo (erros

sistemáticos) e comportamentos singulares (eventuais) contribuem para a considerável imprecisão da posição e orientação após percorrer alguns metros. Isso é especialmente evidente no que se refere à orientação do robô.

Devido a esse erro, principalmente, na estimativa da orientação, optou-se por utilizar uma fusão sensorial já que o RMR deste trabalho também possui um IMU, do qual pode-se extrair a informação de orientação.

Conforme visto em [Aguirre \(2004\)](#), considerando-se que é desejado obter a posição, \mathbf{x} , de um objeto a partir de um instrumento de medição, tem-se uma medida \mathbf{y} . Como no contexto de aplicações práticas, as medidas dos sensores possuem um desvio padrão, caracteriza-se a medida \mathbf{y} de \mathbf{x} por uma função de densidade de probabilidade condicional $f(x|y)$. Supondo-se então que haja duas medições da posição do objeto, y_1 e y_2 , obtidas por instrumentos com características diferentes, faz sentido utilizar estas duas medidas para obter a localização do objeto. Em termos de função de densidade de probabilidade, define-se $f(x|y_1y_2)$, a qual é a função de densidade de probabilidade condicional de \mathbf{x} dadas as medições de y_1 e y_2 .

Assumindo-se que as medições y_1 e y_2 tem distribuição gaussiana, a função densidade de probabilidade de cada uma é descrito pela média e desvio padrão, \bar{z} e σ . Para exemplificar, obteve-se a medida y_1 com seu desvio padrão σ_1 . Sua função de densidade de probabilidade $f_1(x|y_1)$ com média $y_1 = 0,3$ e $\sigma_1 = 0,5$ está ilustrada na [Figura 45](#). Em seguida, obteve-se uma medida y_2 de média $y_2 = 1$ com outro instrumento mais preciso que o primeiro, i.e., $\sigma_2 < \sigma_1$, representada por $f_2(x|y_2)$. Para obter uma medida que combine as duas anteriores, usa-se:

$$\mu = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2}y_1 + \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2}y_2 \quad (3.13)$$

$$\sigma^2 = \left(\frac{1}{\sigma_1^2} + \frac{1}{\sigma_2^2} \right)^{-1} \quad (3.14)$$

A partir disto obteve-se a função de densidade de probabilidade $f_3(x|y_1, y_2)$, em que $\sigma_3 < \sigma_1$ e $\sigma_3 < \sigma_2$ conforme ilustrado na [Figura 45](#), e $y_3 = 0,74$. Pode-se reescrever [\(3.13\)](#) como:

$$\mu = y_1 + \mathbf{G}(y_2 - y_1) \quad (3.15)$$

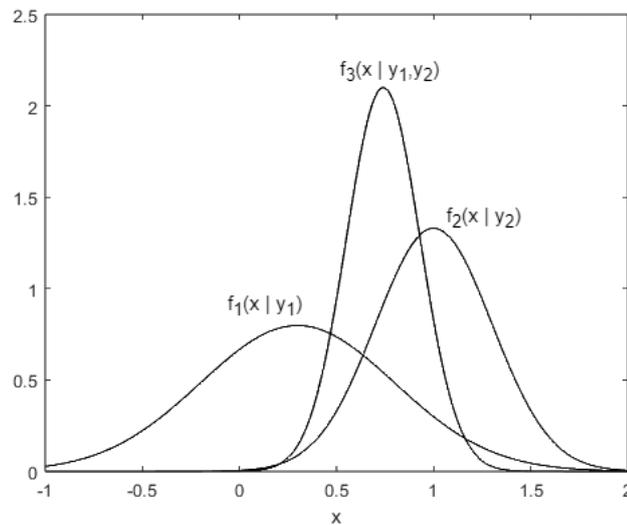
em que

$$\mathbf{G} = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} \quad e \quad \sigma_2^2 = \sigma_1^2 - \mathbf{G}\sigma_1^2 \quad (3.16)$$

Dessa forma, escreveu-se o ângulo de orientação do robô, θ , como a combinação da medida obtida pela odometria e pela leitura da IMU, θ_{odom} e θ_{IMU} respectivamente. Foram realizadas 100 medições de θ_{odom} e θ_{IMU} para descobrir os desvios padrão, obtendo-se $\sigma_{odom} = 0,00289$ e $\sigma_{IMU} = 0,00109$. Então obtém-se θ com:

$$\theta = \theta_{IMU} + 0,1245(\theta_{odom} - \theta_{IMU}) \quad (3.17)$$

Figura 45 – Funções de probabilidade de uma medição hipotética.



Fonte: Adaptado de [Aguirre \(2004\)](#).

onde este foi o ângulo de orientação utilizado, então a [Equação 3.11](#) foi reescrita em função de θ :

$$\begin{bmatrix} x_{k+1} \\ y_{k+1} \\ \theta_{k+1} \end{bmatrix} = \begin{bmatrix} x_k + v \cos(\theta) \\ y_k + v \sin(\theta) \\ \theta_k + \Delta\theta \end{bmatrix} \quad (3.18)$$

3.4 Implementação do Sistema de Controle Lateral

Utilizou-se ROS, *Robot Operating System*, que é uma plataforma de código aberto usada para facilitar o desenvolvimento de software para robôs. Ele oferece bibliotecas e ferramentas para simplificar tarefas de programação e é amplamente utilizado em projetos acadêmicos e industriais relacionados à robótica. Foi escolhida a distribuição *ROS Noetic* em um computador com Ubuntu 20.04 LTS conforme ilustra a [Figura 46](#).

Como mencionado na [subseção 2.5.3.1](#), optou-se por utilizar o microcontrolador ESP32, programado em C/C++ com o *framework Arduino*, em conjunto com ROS por meio do protocolo TCP/IP conforme ilustrado pela [Figura 47](#), onde o microcontrolador é responsável por:

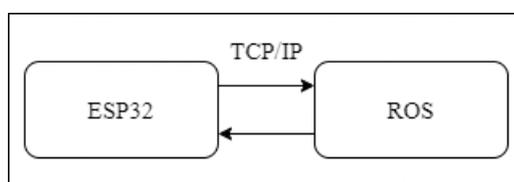
1. Ler os sensores.
2. Acionar o motor DC e o Servomotor.

Figura 46 – Distribuição ROS Noetic.



Fonte: ROS (2024).

Figura 47 – Diagrama de comunicação entre ESP32 e ROS.



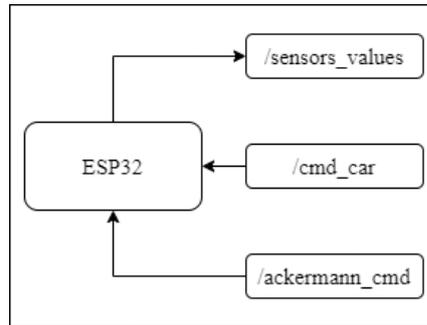
O diagrama da Figura 48, mostra que os valores de velocidades de cada roda traseira, do motor DC e da velocidade angular medida pela *IMU* são publicados em um tópico chamado `/sensor_values` utilizando o formato de mensagem personalizado. Já os comandos para o motor DC e para o Servomotor são recebidos através do tópico `/cmd_car`. Vale mencionar que há também o tópico `/ackermann_cmd`, por onde é possível controlar o robô manualmente através do teclado do computador.³

Em um computador com ROS instalado, foi desenvolvido o arquivo `control_node.cpp`⁴ em C++. Este código é o responsável por receber os dados dos sensores do robô, por implementar o cálculo de odometria considerando a fusão sensorial para estimar a localização do robô e por implementar o controlador lateral para o rastreamento de caminho. O sinal de controle é enviado para o tópico `/cmd_car`.

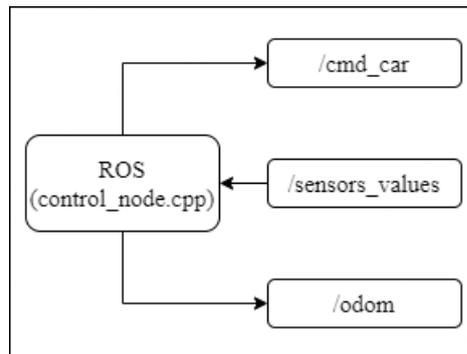
³ Todas mensagens personalizadas criadas, assim como o *workspace* do projeto podem ser encontrados em [repositório do projeto](#).

⁴ Código disponível no [repositório do projeto](#).

Figura 48 – Diagrama dos tópicos utilizados pelo ESP32.



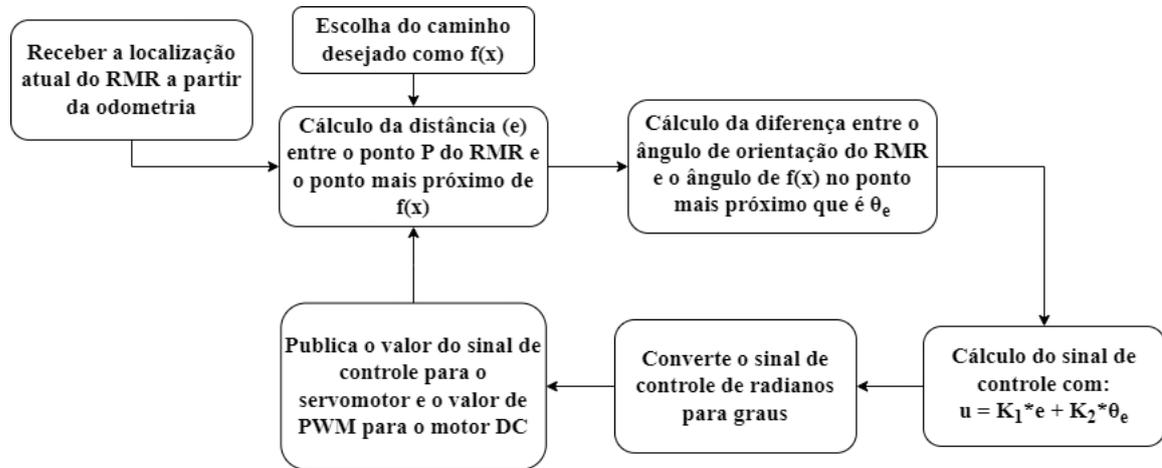
Para o cálculo de odometria era preciso utilizar as velocidades das rodas traseiras a partir do tópico `/sensor_values` e o resultado da odometria era utilizado pelo controle lateral, mas também era publicado no tópico `/odom`. O diagrama da [Figura 49](#) apresenta esses tópicos.

Figura 49 – Diagrama dos tópicos utilizados pelo código `control_node.cpp`.

Neste trabalho, optou-se por desenvolver um controlador robusto como o da [seção 3.2](#) considerando os ângulos de deslizamento das rodas dianteiras e traseiras, α_2 e α_1 respectivamente, como incertezas. Como a taxa de amostragem utilizada pelo sistema era 10 Hz, fez-se a discretização do modelo (3.9) para enfim aplicar o método descrito na [seção 3.2](#).

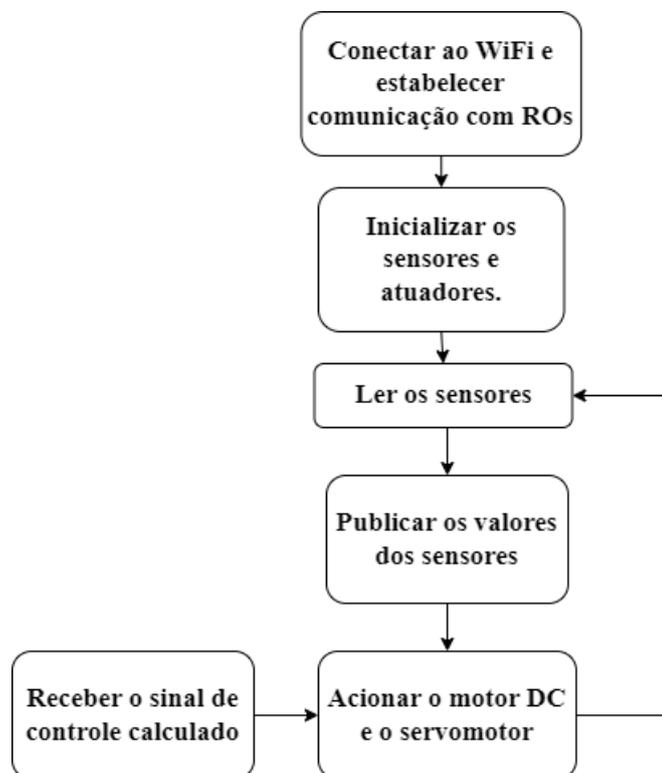
O algoritmo para implementação do controle lateral no código `control_node.cpp` que é executado no computador com ROS é ilustrado no diagrama da [Figura 50](#). Este diagrama ilustra apenas o método `controller()`, o restante dos métodos também estão disponíveis no [repositório do projeto](#). Vale ressaltar que e e θ_e são os estados do modelo (3.10), que se deseja estabilizar.

Na [Figura 51](#) é ilustrado o algoritmo que é executado no ESP32 a cada taxa de amostragem que foi definida como 10 Hz pela limitação dos *encoders* utilizados, pois se observou que em frequências mais altas, o valor lido apresentava muita variação. É

Figura 50 – Diagrama do algoritmo de controle lateral em `control_node.cpp`.

importante observar que todo o cálculo “pesado” é feito fora do ESP32, então este consegue operar na taxa de amostragem desejada.

Figura 51 – Diagrama do algoritmo implementado no ESP32.



3.5 Descrição dos Testes

Para a realização dos testes, considerou-se um caminho do tipo $f(x) = 0,577x$. Outra definição importante foi o ângulo de deslizamento das rodas dianteiras e traseiras, então para isso foi definido o máximo ângulo de esterçamento, ψ_{\max} , como $\pm 27^\circ$. Substituindo este valor de ψ_{\max} na [Equação 2.2](#), obtém-se $k_{\max} = 2,0545$ que resulta em um raio de curvatura igual a 0,4867 m. No entanto foram feitos alguns testes para verificar se esse raio era obtido durante as curvas do RMR com o máximo ângulo de esterçamento e constatou-se que o raio real era igual a 0,5843 m. Este erro se dá justamente pelo efeito do deslizamento das rodas que foram definidos então como $\alpha_1 = \alpha_2 = 4^\circ$. Em todos os testes escolheu-se o valor de PWM = 180 que resulta em uma velocidade longitudinal $v_x = 0,3$ m/s.

3.5.1 Testes Práticos

Foram realizados três testes, onde tempo de duração de cada teste foi definido baseado no tempo de estabilização obtido na simulação do sistema feita no *MATLAB*. As condições iniciais do robô para cada teste está indicada a seguir:

1. Teste prático I

$$\begin{bmatrix} x & y & \theta \end{bmatrix}^\top = \begin{bmatrix} 0 & 0,5 & \frac{\pi}{4} \end{bmatrix}^\top$$

2. Teste prático II

$$\begin{bmatrix} x & y & \theta \end{bmatrix}^\top = \begin{bmatrix} 1 & 0 & \frac{\pi}{2} \end{bmatrix}^\top$$

3. Teste prático III

$$\begin{bmatrix} x & y & \theta \end{bmatrix}^\top = \begin{bmatrix} 0,4 & 1 & -\frac{\pi}{4} \end{bmatrix}^\top$$

Os resultados obtidos nestes testes, assim como a comparação com as simulações serão apresentados no próximo capítulo.

4 Resultados

Neste capítulo, serão apresentados os resultados dos testes executados e descritos na [seção 3.5](#). O controlador utilizado para que o robô executasse o rastreamento de caminho, foi desenvolvido com base no que foi descrito na [seção 3.2](#). Considerando o modelo da [Equação 3.10](#), foi criado um politopo com os vértices sendo a incerteza $\alpha_1 = [0, 4^\circ]$, além disso, discretizou-se este modelo por meio do método de discretização de *Euler* com a taxa de amostragem 10 Hz.

Utilizando a [Equação 3.6](#) foram encontrados os ganhos do controlador lateral no *MATLAB* considerando uma velocidade longitudinal constante e igual a 0,3 m/s resultando em:

$$K = \begin{bmatrix} -4,095 & -7,8182 \end{bmatrix}$$

4.1 Simulações

Foram realizadas simulações de todos os testes práticos descritos em [3.5.1](#) considerando o caminho $f(x) = 0,577x$.

A [Figura 52](#) mostra o resultado obtido pela simulação das condições propostas no teste I. Observa-se que a distância, representada por e , não alcança o valor zero e fica em torno de 0,0575 m. Contudo, o ângulo de esterçamento do robô, denotado por ψ , permanece constante, embora não seja nulo, mas sim igual a 10° . Vale ressaltar que a distância e o sinal de controle não são zerados mesmo após a estabilização do sistema, pois a simulação foi feita utilizando o valor de α_1 constante, porém este é um parâmetro variante no tempo que aumenta conforme o ângulo de esterçamento. Dessa forma, os outros resultados também apresentarão valores de distância e sinal de controle diferente de zero, mas na prática o resultado pode ser diferente.

O resultado da simulação do teste II está ilustrado na [Figura 53](#). Esta mostra que a distância permanece diferente de zero e fica em torno de 0,0563 m. Já o ângulo de esterçamento estabiliza em aproximadamente 8° .

Na simulação do teste III a distância permanece diferente de zero com um valor em torno de 0,0605 m. Já o ângulo de esterçamento estabiliza em aproximadamente 8° conforme ilustrado na [Figura 54](#).

Figura 52 – Resultado da simulação do teste prático I.

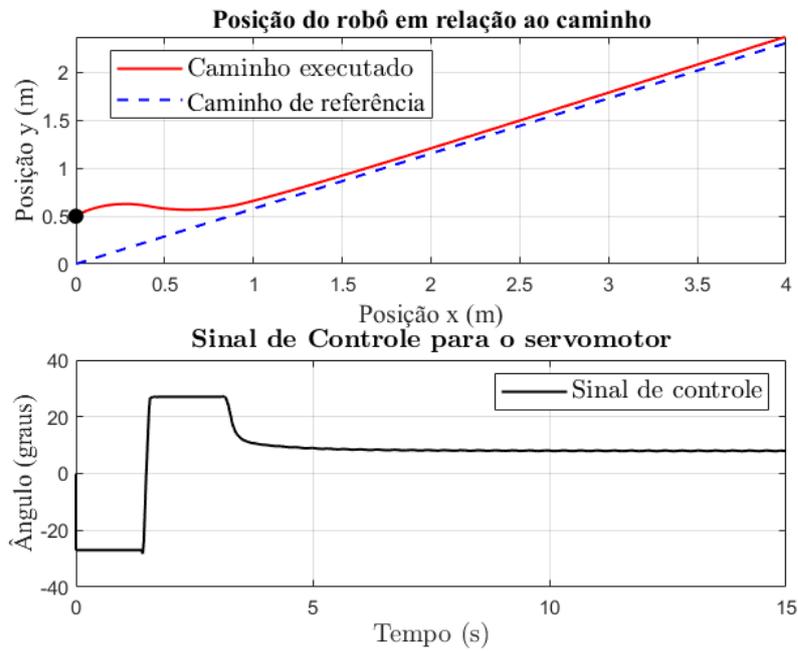


Figura 53 – Resultado da simulação do teste prático II.

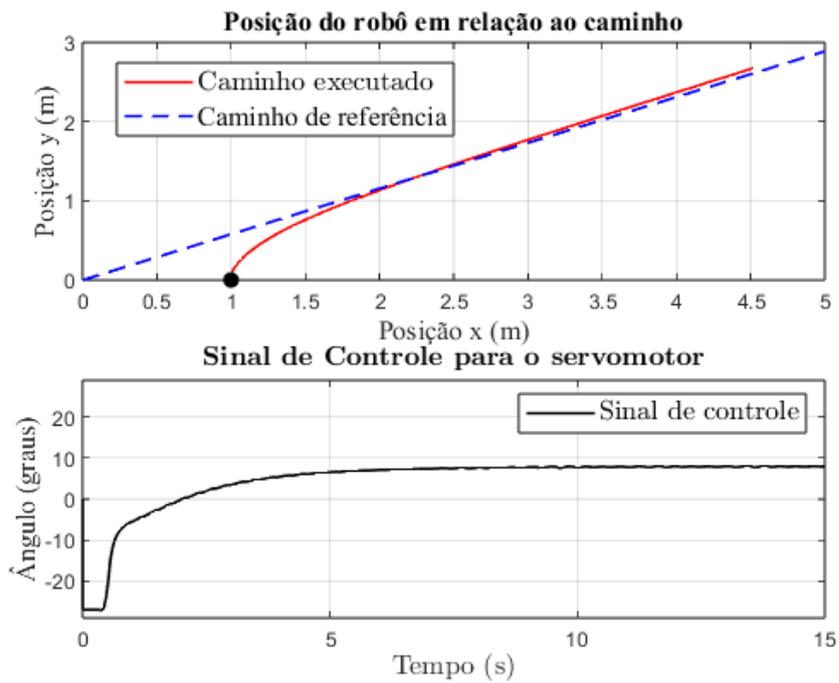
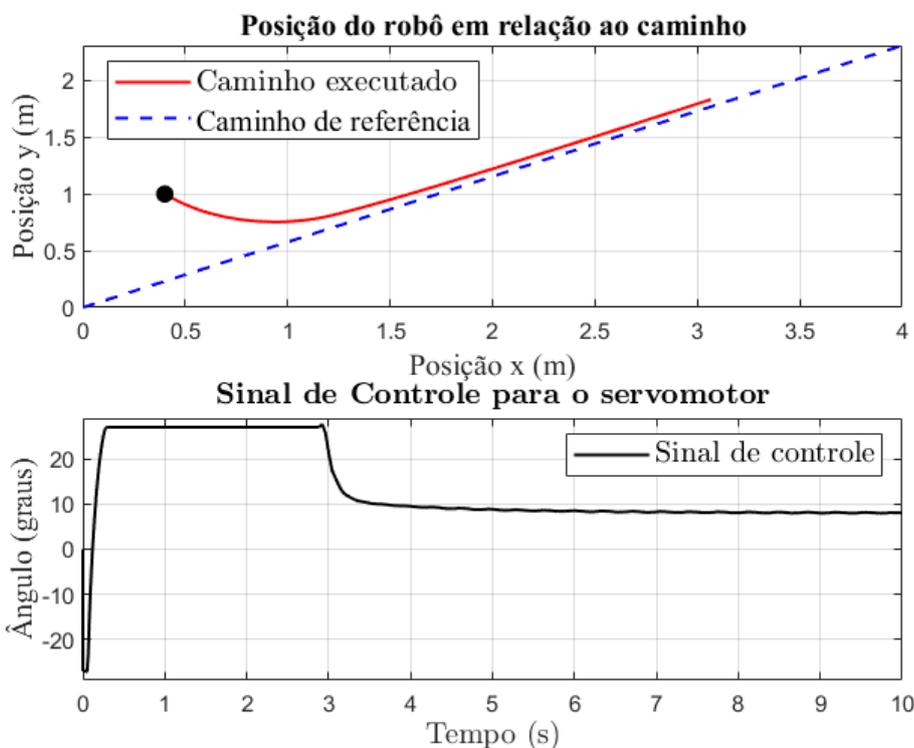


Figura 54 – Resultado da simulação do teste prático III.



4.2 Testes Práticos

4.2.1 Teste Prático I

O Teste I foi realizado e gerou o resultado de posição do robô em relação ao caminho desejado conforme ilustrado pela [Figura 55](#) onde é possível notar um sobressinal que é explicado pela falta de controlador longitudinal. Nesta figura também está ilustrado o sinal de controle. Diferente do sinal gerado pela simulação, este apresenta uma oscilação e isso ocorreu devido à zona morta do servomotor presente no sistema de direção. A zona morta é uma não-linearidade que não foi levada em consideração na concepção do controlador. No entanto, mesmo com essa oscilação no sinal de controle é possível notar que devido o pequeno tempo que o servo teria para mudar sua angulação, não impactou muito sobre a orientação do robô. Isso é notado pelo pequeno erro de orientação mostrado na [Figura 56](#). Na [Figura 56](#) está ilustrado que a distância do robô para o caminho desejado tendeu a zero, mas o seu valor estabilizou em $\pm 0,012$ m. Já a orientação do robô em relação ao ângulo do caminho, ficou variando com baixas amplitudes de $\pm 1,14^\circ$ em torno de zero. Em comparação com os valores obtidos na simulação do Teste I, a distância foi menor e o robô permaneceu sobre o caminho. Esta diferença se dá pelo fato de α_1 não ser um parâmetro constante como considerado na simulação. Outro resultado obtido foi em relação ao funcionamento do diferencial desenvolvido. [Figura 57](#) é ilustrado que nos instantes

Figura 55 – Resultado da posição do robô e sinal de controle para o teste prático I.

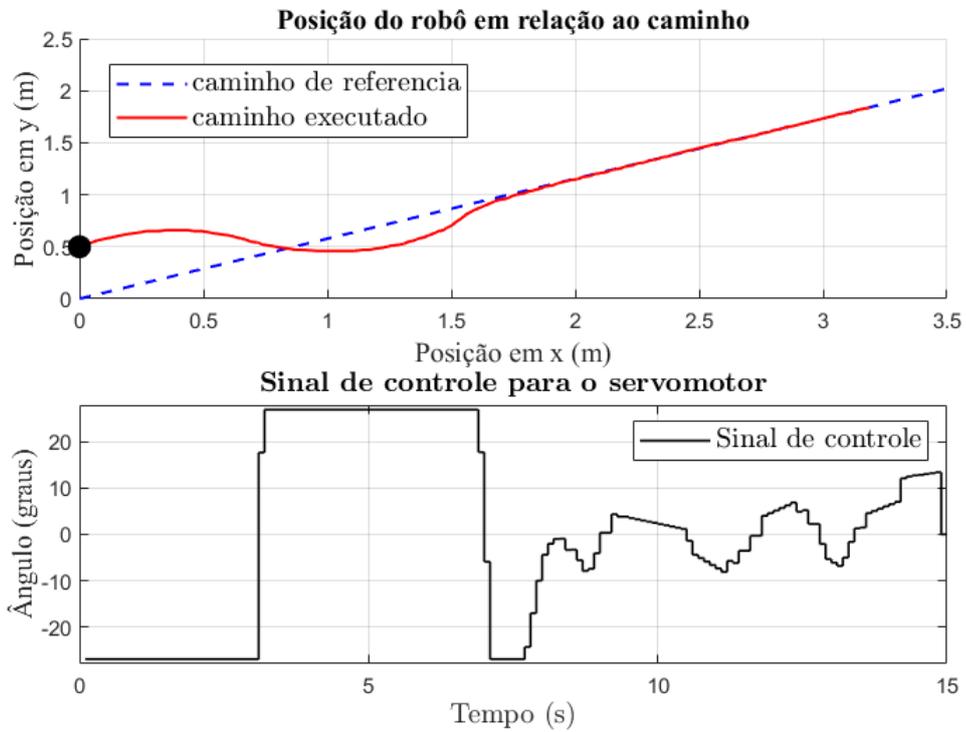
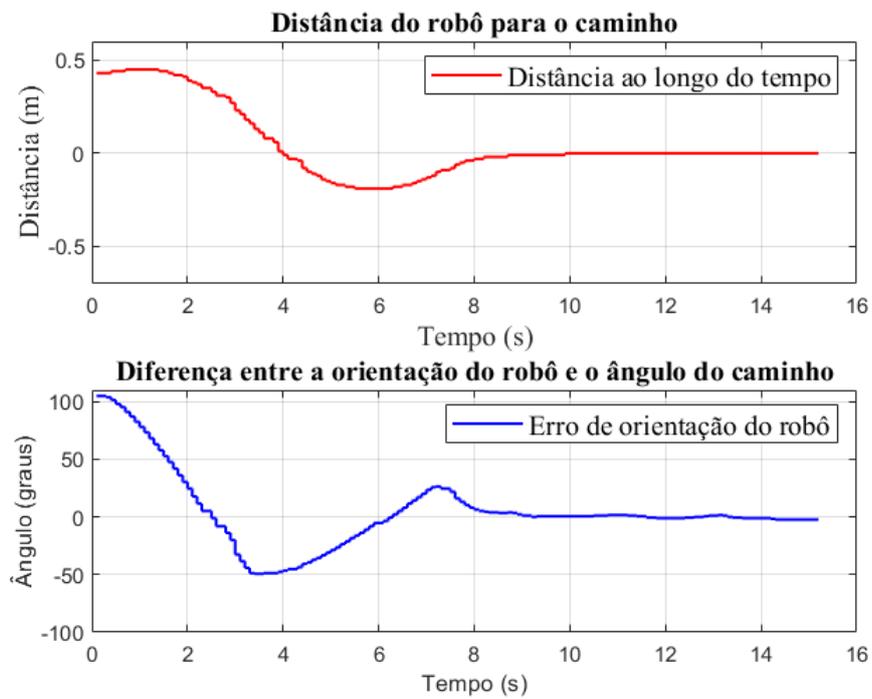
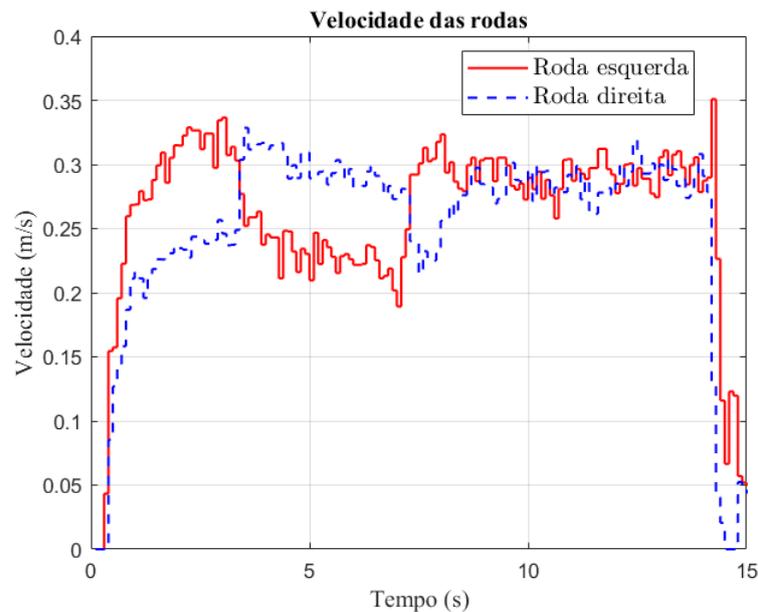


Figura 56 – Resultado do erro de orientação do robô para o teste prático I.



iniciais a roda das esquerda desenvolveu uma velocidade maior que a da direita, indicando que o robô fez uma curva para a direita, e em seguida, a velocidade da roda direita fica maior que a da esquerda indicando uma curva para esquerda. Por fim, as velocidades permanecem com valores bem próximos. Dessa forma o movimento descrito anteriormente com base nos valores de velocidade é justamente o ilustrado na [Figura 55](#).

Figura 57 – Resultado da velocidade das rodas durante o Teste I.

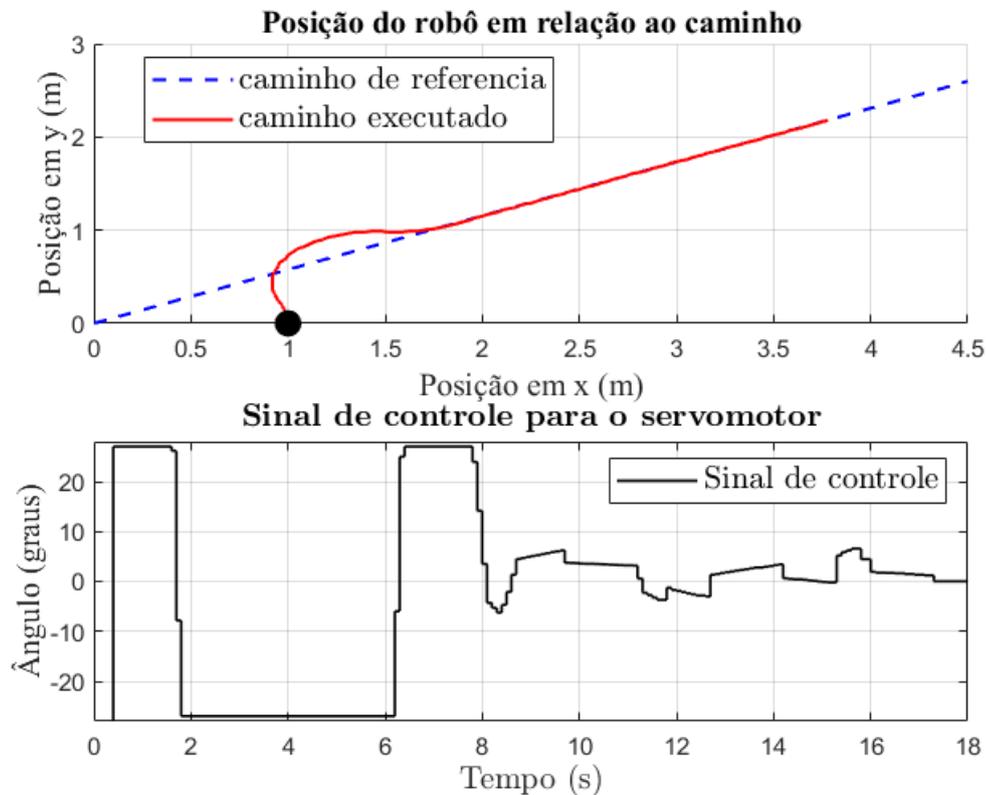


4.2.2 Teste Prático II

O Teste II foi conduzido, resultando na determinação da posição do robô em relação ao caminho desejado, que apresenta sobressinal, e sinal de controle gerado pelo controlador, conforme evidenciado pela [Figura 58](#). Em contraste com o sinal gerado pela simulação, este apresenta oscilações cujo motivo foi já explicado no resultado do Teste 1. A [Figura 59](#) destaca que a distância entre o robô e o caminho desejado aproximou-se de zero, no entanto, oscilou entre $\pm 0,010$. Quanto à orientação do robô em relação ao ângulo do caminho, manteve-se oscilando com pequenas amplitudes de $\pm 0,58^\circ$ em torno de zero.

A [Figura 60](#) mostra que, nos primeiros momentos, a roda esquerda desenvolveu uma velocidade superior à da direita, sugerindo que o robô realizou uma curva para a direita. Posteriormente, as velocidades permaneceram com valores bastante próximos, indicando a tentativa de seguir uma trajetória linear. Portanto, o movimento descrito anteriormente, com base nos valores de velocidade, é refletido na [Figura 58](#).

Figura 58 – Resultado da posição do robô e o sinal de controle para o teste prático II.



4.2.3 Teste Prático III

O Teste III foi realizado para determinar a posição do robô em relação ao caminho desejado e o sinal de controle produzido pelo controlador, como indicado pela [Figura 58](#). Em comparação com o sinal de controle gerado durante a simulação, observam-se oscilações tal qual ocorreu nos testes anteriores. A [Figura 62](#) mostra a distância entre o robô e o caminho desejado que ficou em torno de $\pm 0,025$ m. A orientação do robô em relação ao ângulo do caminho continuou oscilando com amplitudes reduzidas, mantendo-se dentro de $\pm 0,27^\circ$ em torno do zero. A [Figura 63](#) revela que, nos estágios iniciais, a roda esquerda apresentou uma velocidade superior à da roda direita, sugerindo uma curva para a direita realizada pelo robô. Em seguida, ocorreu o oposto, indicando uma curva à esquerda, seguida por mais uma curva à direita, até que as velocidades se aproximaram. Consequentemente, o padrão de movimento descrito com base nas velocidades é refletido na [Figura 61](#).

Figura 59 – Resultado do erro de orientação do robô para o teste prático II.

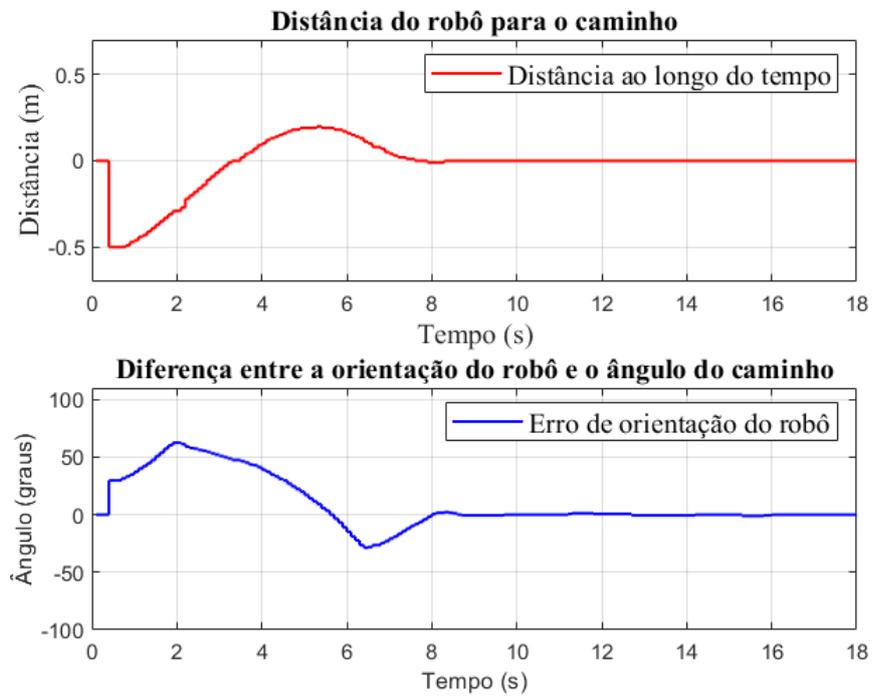


Figura 60 – Resultado da velocidade das rodas durante o Teste II.

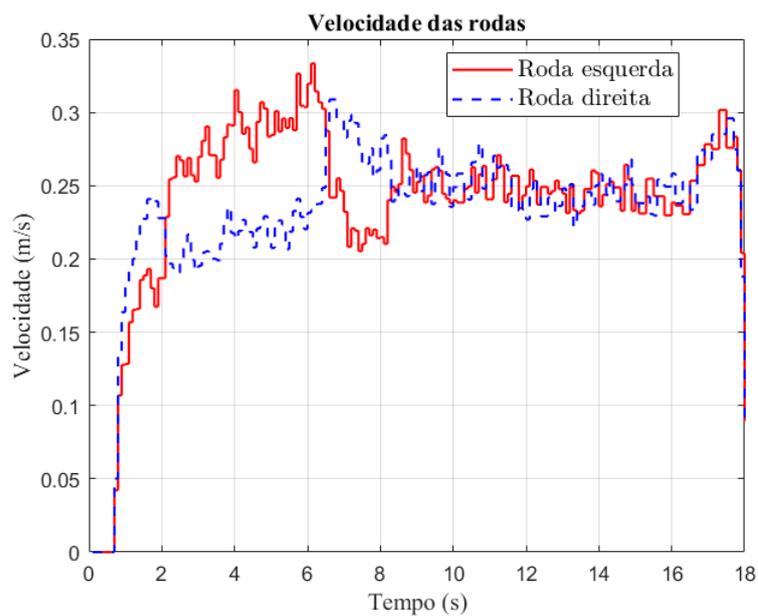


Figura 61 – Resultado da posição do robô e o sinal de controle para o teste prático III.

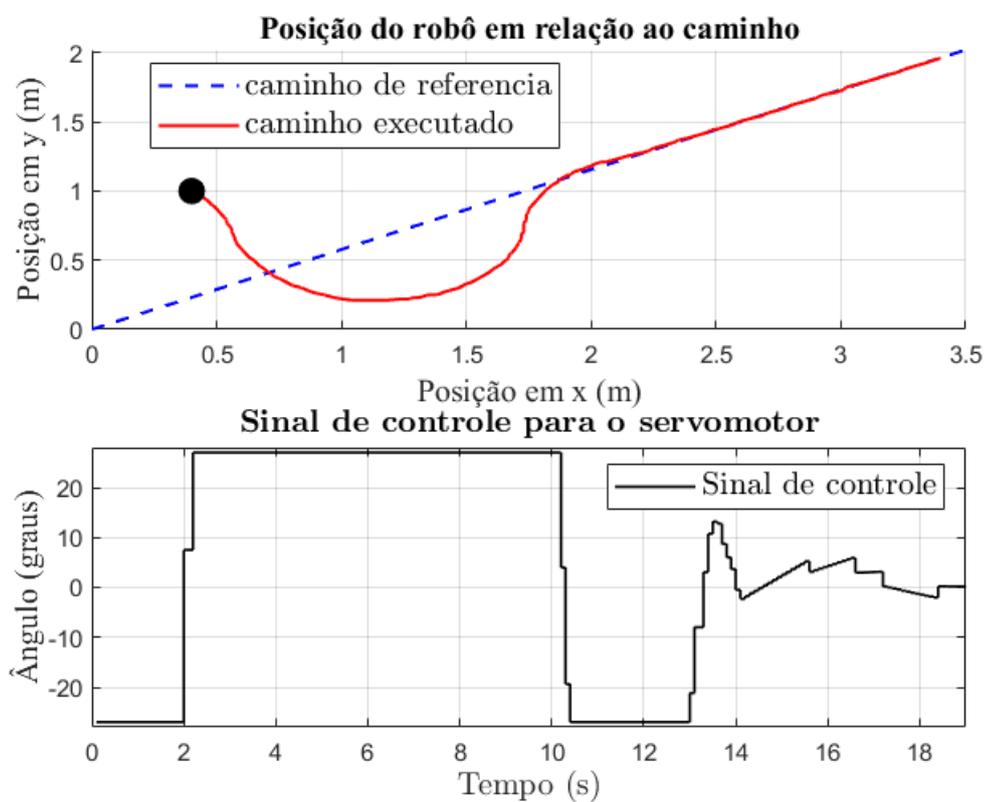


Figura 62 – Resultado do erro de orientação do robô para o teste prático III.

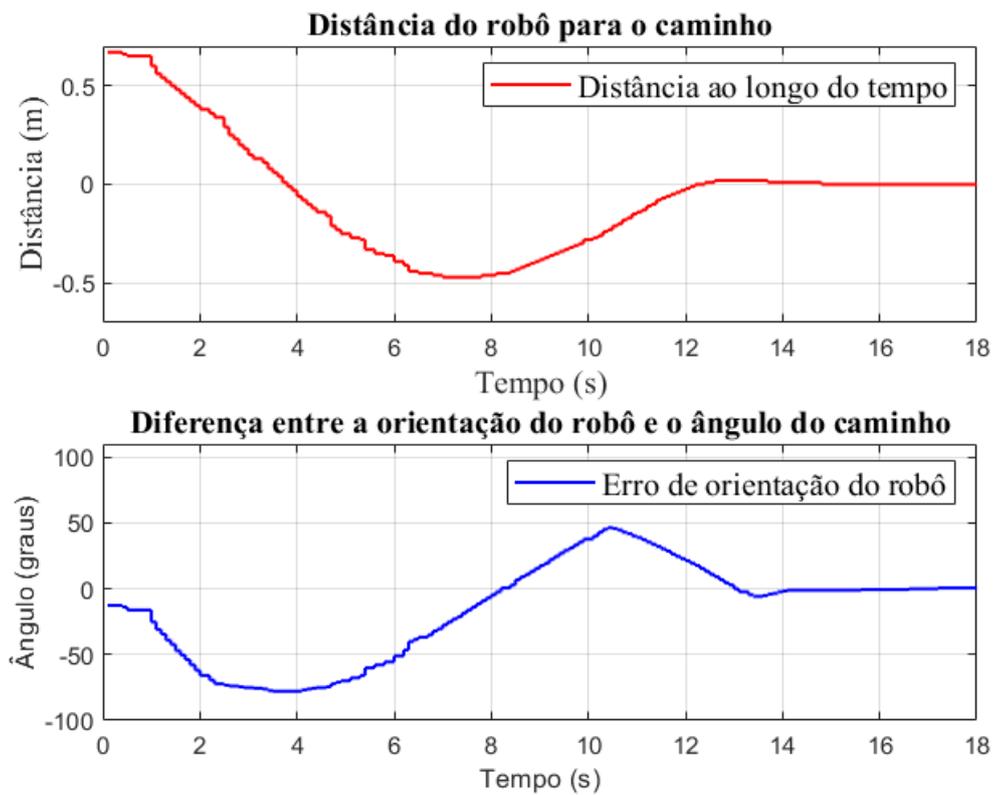
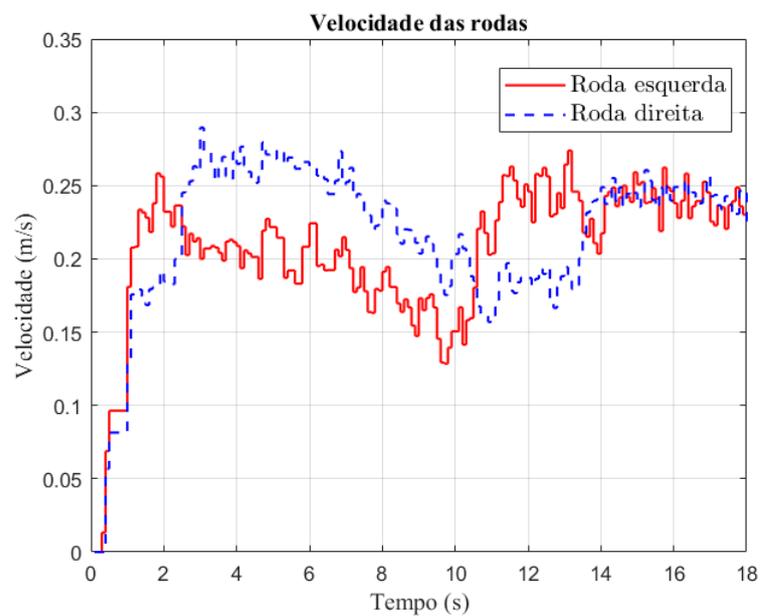


Figura 63 – Resultado da velocidade das rodas durante o Teste III.



5 Conclusões

Neste trabalho, demonstrou-se a implementação de um controlador lateral para rastreamento de caminho com um robô móvel do tipo *Car-Like*.

Com base nos dados coletados tanto nos testes simulados quanto nos práticos, observa-se que, no Teste I, a distância do robô em relação ao caminho desejado é 0,0455 m menor em comparação com o valor obtido na simulação. Enquanto o erro de orientação estabilizou em $\pm 1,14^\circ$. No que diz respeito ao Teste II, a distância obtida também foi menor que a da simulação em 0,0463 m e o erro de orientação estabilizou em $\pm 0,58^\circ$. No Teste III o resultado obtido foi uma distância de 0,0355 m menor que a da simulação e um erro de orientação estabilizado em $\pm 0,27^\circ$.

O motivo para os resultados dos testes práticos serem diferentes da simulação é devido a utilização de um ângulo de deslizamento, α_1 , constante. Porém, este é um parâmetro variante no tempo que está relacionado ao ângulo de esterçamento das rodas do robô, ou seja, quando este está em uma curva. Como α_1 foi levado em consideração para a criação do polítopo utilizado no projeto do controlador, o desempenho deste foi satisfatório na aplicação prática. Outra diferença observada foram os sinais de controle dos testes práticos e os das simulações. O motivo é a presença da zona morta do servomotor que foi uma não-linearidade não considerada no projeto do controlador, então o sinal de controle apresentou uma oscilação.

Quanto à estrutura mecânica criada, revelou-se apropriada, especialmente considerando que foi construída integralmente a partir do zero. O diferencial mecânico desempenhou sua função conforme evidenciado no [Capítulo 4](#). No que concerne à estrutura eletrônica desenvolvida, a placa de circuito impresso operou de maneira satisfatória, não exercendo efeitos negativos no funcionamento do sistema.

Com base nos resultados apresentados, pode-se afirmar que a plataforma de tipo *Car-like* desenvolvida atendeu às expectativas em relação ao uso eficaz do diferencial mecânico e da placa de circuito impresso projetada. Apesar da presença da zona morta do servomotor, o rastreamento do caminho desejado durante os testes foi alcançado.

5.1 Trabalhos Futuros

A partir do aprimoramento do sistema, surgem oportunidades para o desenvolvimento de pesquisas baseadas nas descobertas deste estudo. Recomenda-se a exploração de métodos alternativos de localização ou a implementação de um Filtro de *Kalman* Estendido em conjunto com a odometria já existente. Isso se justifica pelo fato de que a odometria

diferencial sofre suscetibilidade a erros, os quais podem se acumular ao longo do tempo. Além disso, a resolução do problema inerente ao sistema de direção pode ser alcançada por meio da utilização de um outro atuador.

No que diz respeito ao desafio de controle, é possível explorar métodos adicionais de controle robusto para restringir ainda mais as condições de operação do sistema. Adicionalmente, propõe-se a elaboração de um controlador longitudinal, visando ajustar a velocidade do robô de maneira adequada para cada ponto do caminho. Essa abordagem busca reduzir o sobressinal, especialmente quando o RMR tenta alinhar-se ao caminho.

Por fim, é possível a implementação de sensores para detectar objetos no caminho, já que neste trabalho considerou-se que o ambiente estava livre de obstáculos, mas essa não é a realidade em muitas aplicações práticas.

Referências

- AGUIRRE, L. A. *Introdução à identificação de sistemas: Técnicas lineares e não-lineares aplicadas a sistemas reais*. [S.l.]: Editora UFMG, 2004. Citado 2 vezes nas páginas 62 e 63.
- AGUIRRE, L. A. *Enciclopédia de automática, v. 1: Controle e automação*. [S.l.]: Editora Blucher, 2007. Citado 4 vezes nas páginas 28, 29, 30 e 31.
- AGUIRRE, L. A. *Enciclopédia de automática, v. 3: Controle e automação*. [S.l.]: Editora Blucher, 2007. Citado 2 vezes nas páginas 11 e 15.
- AGUIRRE, L. A. *Controle de Sistemas Amostrados*. [S.l.]: E-Book Kindle, 2023. Citado 4 vezes nas páginas 24, 25, 26 e 27.
- ALCIATORE, M. B.; HISTAND, D. G. *Introduction to Mechatronics and Measurement Systems*. [S.l.: s.n.], 2012. Citado 3 vezes nas páginas 36, 41 e 45.
- ALIEXPRESS. *Chassi de carro inteligente com motor duplo*. 2024. Disponível em: <<https://encurtador.com.br/noxN1>>. Citado na página 53.
- AMAZON. *Haste de Tração de direção*. 2024. Disponível em: <<https://www.amazon.com.br/Dire%C3%A7%C3%A3o-Realista-Traseira-Superior-Tirante/dp/B0C968TSCJ>>. Citado na página 51.
- AROGETI, S. A.; BERMAN, N. Path following of autonomous vehicles in the presence of sliding effects. *IEEE Transactions on Vehicular Technology*, v. 61, n. 4, p. 1481–1492, 2012. Citado 5 vezes nas páginas 19, 20, 23, 58 e 59.
- BESSAS, A.; BENALIA, A.; BOUDJEMA, F. Robust output trajectory tracking of car-like robot mobile. *Journal of Electrical Systems*, v. 12, n. 3, 2016. Citado na página 11.
- BISHOP, R. H. *Mechatronics: an introduction*. [S.l.]: CRC Press, 2017. Citado na página 45.
- BOYD, S. et al. *Linear matrix inequalities in system and control theory*. [S.l.]: SIAM, 1994. Citado 2 vezes nas páginas 28 e 57.
- BRANDAO, A. S. *Controle descentralizado com desvio de obstáculos para uma formação líder-seguidor de robôs móveis*. Dissertação (Mestrado) — Universidade Federal do Espírito Santo, Vitória, 2008. Citado na página 16.
- CAMPBELL, T. et al. Could 3D printing change the world. *Technologies, Potential, and Implications of Additive Manufacturing, Atlantic Council, Washington, DC, JSTOR*, v. 3, 2011. Citado 3 vezes nas páginas 32, 33 e 34.
- CASTRO, F. P. d. *Servo motores: visão geral, controle e aplicação*. Universidade Tecnológica Federal do Paraná, 2016. Citado 2 vezes nas páginas 44 e 45.

- COSTA, J. P. S. Análise do dispositivo mecânico diferencial. Universidade Federal da Paraíba, 2019. Citado na página 52.
- DIAS, J. E. A. *Modelagem longitudinal e controle de velocidade de um carro autônomo*. Dissertação (Mestrado) — Universidade Federal de Minas Gerais, 2013. Citado 2 vezes nas páginas 20 e 21.
- DIAS, M. B. Proposta de desenvolvimento de uma máquina de corte a laser para facilitar a prototipagem. 2015. Citado na página 35.
- DIXON, J. C. *Suspension geometry and computation*. [S.l.]: John Wiley & Sons, 2009. Citado na página 17.
- DULLERUD, G. E.; PAGANINI, F. *A course in robust control theory: a convex approach*. [S.l.]: Springer Science & Business Media, 2013. v. 36. Citado na página 57.
- EMAM, M.; FAKHARIAN, A. Solving path following problem for car-like robot in the presence of sliding effect via LMI formulation. *Journal of Computer & Robotics*, Qazvin Islamic Azad University, v. 10, n. 2, p. 11–22, 2017. Citado na página 22.
- EMAM, M.; FAKHARIAN, A. Solving path following problem for car-like robot in the presence of sliding effect via lmi formulation. *Journal of Computer & Robotics*, Qazvin Islamic Azad University, v. 10, n. 2, p. 11–22, 2017. Citado 2 vezes nas páginas 59 e 60.
- ESPINOZA, M.; SCHAEFFER, L. Uso do CAD/CAE/CAM na produção de matrizes para os processos novos de conformação mecânica. *Revista del Instituto de investigación de la Facultad de minas, metalurgia y ciencias geográficas*, v. 7, n. 14, p. 84–91, 2004. Citado na página 31.
- EVERETT, H. *Sensors for mobile robots*. [S.l.]: CRC Press, 1995. Citado 2 vezes nas páginas 14 e 37.
- FAISAL, I. A.; PURBOYO, T. W.; ANSORI, A. S. R. A review of accelerometer sensor and gyroscope sensor in imu sensors on motion capture. *J. Eng. Appl. Sci*, v. 15, n. 3, p. 826–829, 2019. Citado 3 vezes nas páginas 39, 40 e 42.
- FILIPPESCHI, A. et al. Survey of motion tracking methods based on inertial sensors: A focus on upper limb human motion. *Sensors*, v. 17, n. 6, p. 1257, 2017. Citado na página 39.
- FLOYD, T. *Sistemas digitais: fundamentos e aplicações*. [S.l.]: Bookman Editora, 2009. Citado na página 24.
- GILLESPIE, T. *Fundamentals of vehicle dynamics*. [S.l.]: SAE international, 2021. Citado na página 17.
- GOLDENEYE. *Berkeley Autonomous Race Car*. 2024. Disponível em: <<https://goldeneye.berkeley.edu/barc.html>>. Citado na página 17.
- GRIDLING, G.; WEISS, B. Introduction to microcontrollers. *Vienna University of Technology Institute of Computer Engineering Embedded Computing Systems Group*, 2007. Citado na página 46.

- GUERRA, P. N. *Modelagem linear e identificação do modelo dinâmico de um robô móvel com acionamento diferencial*. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Norte, 2005. Citado na página 11.
- HALLGRIMSSON, B. *Prototyping and modelmaking for product design*. [S.l.]: Hachette UK, 2012. Citado na página 32.
- LUCA, A. D.; ORIOLO, G.; SAMSON, C. Feedback control of a nonholonomic car-like robot. *Robot motion planning and control*, p. 171–253, 2005. Citado na página 21.
- MARTINS, R. J. et al. Cinemática e localização em robótica terrestre usando múltiplos encoders. In: *Simpósio Brasileiro de Automação Inteligente (SBAI)*. [S.l.: s.n.], 2011. Citado 5 vezes nas páginas 18, 20, 52, 60 e 61.
- ORCIUOLI, A. O impacto das tecnologias de fabricação digital nos processos de design. *AU—Arquitetura e Urbanismo: digital: Ed*, v. 183, 2009. Citado na página 31.
- PRUSA, J. *3D Printers*. 2024. Disponível em: <<https://www.prusa3d.com/category/3d-printers/>>. Citado na página 34.
- RASHAD, R. Dynamic trajectory tracking of a car-like robot. In: *Trabajo de grado*. [S.l.]: Institute of Computer Science, Bayerische Julius Maximilians Universität . . . , 2012. Citado 6 vezes nas páginas 11, 14, 19, 21, 22 e 58.
- RODRIGUES, A. P. C. Desenvolvimento de uma interface para acionamento de atuadores e leitura de encoders para um exoesqueleto de membro inferior com a plataforma SoC-FPGA Zybo. 2017. Citado 2 vezes nas páginas 38 e 39.
- ROS. *ROS Noetic Ninjemys*. 2024. Disponível em: <<https://wiki.ros.org/noetic>>. Citado na página 64.
- SANTOS, J. W.; JUNIOR, R. C. d. L. *Sistema de automatização residencial de baixo custo controlado pelo microcontrolador esp32 e monitorado via smartphone*. Dissertação (B.S. thesis) — Universidade Tecnológica Federal do Paraná, 2019. Citado na página 46.
- SASAKI, A. S. Estratégia de desvio de obstáculo para navegação autônoma de um robô móvel do tipo car-like. 2012. Citado na página 16.
- SHAHRUBUDIN, N.; LEE, T. C.; RAMLAN, R. An overview on 3D printing technology: Technological, materials, and applications. *Procedia Manufacturing*, Elsevier, v. 35, p. 1286–1296, 1 2019. ISSN 2351-9789. Citado na página 32.
- SILVA, S. R. d. C. *Sistema de monitoramento e controle de energia utilizando o microcontrolador ESP32*. Dissertação (Trabalho de Conclusão de Curso) — Universidade Tecnológica Federal do Paraná, 2021. Citado na página 47.
- SILVA, T. M. da et al. Estudo das propriedades de atenuação de materiais para desenvolvimento de um simulador de tireoide-pescoço. *Brazilian Journal of Radiation Sciences*, v. 9, n. 1, 2021. Citado na página 35.
- SINGHAL, K.; KUMAR, V.; RANA, K. Robust trajectory tracking control of non-holonomic wheeled mobile robots using an adaptive fractional order parallel fuzzy pid controller. *Journal of the Franklin Institute*, Elsevier, v. 359, n. 9, p. 4160–4215, 2022. Citado na página 11.

TAVARES, G. V. *Uma Contribuição para o Desenvolvimento de Próteses Mioelétricas Ativas de Membros Superiores*. Dissertação (Mestrado) — Universidade Estadual de Campinas, 2022. Citado 3 vezes nas páginas 41, 43 e 44.

THOMSEN, A. *Motor DC com Driver Ponte H L298N*. 2013. Disponível em: <<https://www.makerhero.com/blog/motor-dc-arduino-ponte-h-l298n/>>. Citado na página 43.

WANG, R. et al. Robust \mathcal{H}_∞ path following control for autonomous ground vehicles with delay and data dropout. *IEEE Transactions on Intelligent Transportation Systems*, v. 17, n. 7, p. 2042–2050, 2016. Citado na página 23.

YADAV, P. S. et al. A theoretical review of mobile robot locomotion based on mecanum wheels. Transdisciplinary Research and Education Center for Green Technologies, 2022. Citado na página 15.

ZANARDI, M. *Análise das geometrias do sistema de direção de um veículo de movimentação de cargas aplicado a construção civil*. Dissertação (Trabalho de Conclusão de Curso) — Universidade de Caixas do Sul, 1013. Citado na página 18.