Universidade de Brasília

Faculdade de Tecnologia

Projeto de sistema embarcado para controle cinemático de robô móvel não-holonômico em competições VSSS

André Dornelas Sanches

PROJETO FINAL DE CURSO

ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Brasília

2025

Universidade de Brasília
Faculdade de Tecnologia

Projeto de sistema embarcado para controle cinemático de robô móvel não-holonômico em competições VSSS

André Dornelas Sanches

Projeto Final de Curso submetido como requisito parcial para obtenção do grau de Engenheiro de Controle e Automação.

Orientador: Prof. Dr. Geovany Araújo Borges
Coorientadora: Profa. Dra. Carla Maria Chagas E Cavalcante Koike

Brasília
2025

FICHA CATALOGRÁFICA

Dornelas Sanches, André.

Projeto de sistema embarcado para controle cinemático de robô móvel não-holonômico em competições VSSS / André Dornelas Sanches; orientador Geovany Araújo Borges; coorientadora Carla Maria Chagas E Cavalcante Koike. -- Brasília, 2025.

82 p.

Projeto Final de Curso (Engenharia de Controle e Automação) -- Universidade de Brasília, 2025.

1. Robótica móvel. 2. Competições de robótica. 3. Controle embarcado. 4. Very Small Size Soccer (VSSS). I. Borges, Geovany Araújo, orient. II. Koike, Carla Maria Chagas E Cavalcante, coorient. III. Título.

Universidade de Brasília
Faculdade de Tecnologia

Projeto de sistema embarcado para controle cinemático de robô móvel não-holonômico em competições VSSS

André Dornelas Sanches

Projeto Final de Curso submetido como requisito parcial para obtenção do grau de Engenheiro de Controle e Automação.

Trabalho aprovado. Brasília, 15 de Dezembro de 2025:

Prof. Dr. Geovany Araújo Borges
UnB/FT/ENE
Orientador

Profa. Dra. Carla Maria Chagas E Cavalcante Koike
UnB/CIC
Coorientadora

Prof. Dr. Hugo Tadashi Muniz Kubassa
UnB/FT/ENE
Examinador externo

Jones Yudi Mori Alves Da Silva
UnB/FT/ENM
Examinador externo

Este trabalho é dedicado à continuação da equipe UnBall, que tanto poderá colaborar para a formação de profissionais com conhecimentos multidisciplinares.

Agradecimentos

Agradeço primeiramente aos meus pais, Eniê e Jefferson, que me deram toda a base desde criança para que eu conseguisse alcançar meus objetivos, acadêmicos ou não. Agradeço também à minha tia avó Altiva, que sempre incentivou meus estudos e garantiu que eu tivesse tudo de que precisava para seguir em frente.

Sou muito grato aos meus orientadores, Geovany e Carla, por toparem me apoiar neste projeto pelo qual tenho tanta paixão. Sinto que encontrei o equilíbrio perfeito, com um lado que me faz mirar mais alto e outro que me ajuda a não viajar demais.

Também não posso deixar de agradecer a razão de existir deste projeto, a equipe UnBall. Quando me disseram que seria a melhor escolha da minha graduação eu duvidei, mas hoje tenho certeza de que foi, além de ter sido o maior incentivo para eu não desistir. Sou grato a todos que estiveram comigo nessa caminhada: Juliana, Gabriel, Luiz, Ayssa, Emme, Raul e, em especial, Maria Cláudia, que me deu a ideia de transformar este trabalho em um PFC e me incentivou a seguir com ele.

Agradeço ainda à Faculdade de Tecnologia e ao professor Edson pelo apoio constante ao trabalho das equipes de competição da UnB, e à Finatech, que tornou este projeto financeiramente possível.

E deixo meu agradecimento ao SINST e à SESUP da Câmara dos Deputados. Ao meu chefe Thiago, pela paciência e compreensão durante todo o processo, e ao Severino, pelo suporte técnico durante a construção do robô, sempre disposto a ajudar quando necessário.

Resumo

Este trabalho apresenta o desenvolvimento de um sistema embarcado para o controle cinemático de um robô móvel não holonômico de tração diferencial, projetado para competir em torneios de robótica na categoria Very Small Size Soccer (VSSS). O projeto abrange desde a definição da arquitetura eletrônica até a implementação de estratégias de controle digital no microcontrolador ESP32. A plataforma embarcada integra motores DC com encoders ópticos, uma unidade inercial e um driver de potência dedicado, todos organizados em uma placa de circuito impresso compacta, desenvolvida especificamente para as restrições físicas da competição. O sistema recebe comandos externos de velocidade linear e angular transmitidos sem fio e processa as medições dos sensores para executar o controle de movimento. O desenvolvimento contemplou a modelagem da dinâmica do robô e a identificação experimental dos atuadores, resultando em uma arquitetura de controle em cascata, na qual um controlador adaptativo por modelo de referência (MRAC) regula as velocidades das rodas enquanto o nível superior comanda as velocidades linear e angular do robô. Em seguida, foram avaliadas duas abordagens de controle nesse nível superior, baseadas em PID e em realimentação de estados. Os resultados experimentais demonstraram que os controladores implementados foram capazes de seguir referências de velocidade de forma estável e reprodutível, evidenciando a integração eficaz entre hardware, modelagem e algoritmos de controle embarcados.

Palavras-chave: Robótica móvel; Competições de robótica; Controle embarcado; Very Small Size Soccer (VSSS).

Abstract

This work presents the development of an embedded system for the kinematic control of a nonholonomic differential-drive mobile robot designed to compete in robotics tournaments in the Very Small Size Soccer (VSSS) category. The project encompasses the definition of the electronic architecture and the implementation of digital control strategies on an ESP32 microcontroller. The embedded platform integrates DC motors with optical encoders, an inertial measurement unit, and a dedicated motor driver, all arranged on a compact printed circuit board designed to meet the physical constraints of the competition. The system receives wireless commands of linear and angular velocities and processes sensor measurements to execute motion control. The development included the modeling of the robot's dynamics and the experimental identification of actuator parameters, leading to a cascade-control architecture in which a Model Reference Adaptive Controller (MRAC) regulates individual wheel velocities while a higher-level controller commands the robot's linear and angular motion. Two approaches were evaluated at this upper level, based on PID control and state-feedback control. Experimental results demonstrated that the implemented controllers were able to track velocity references in a stable and repeatable manner, confirming the effective integration between hardware design, system modeling, and embedded control algorithms.

Keywords: Mobile robotics; Robotics competitions; Embedded control; Very Small Size Soccer (VSSS).

Lista de figuras

Figura 1.1 O robô Spot, da Boston Dynamics 17
Figura 1.2 Robô móvel diferencial desenvolvido neste trabalho para a categoria VSSS 18
Figura 1.3 Robô móvel não-holonômico de tração diferencial da categoria *Robomagellan* 19
Figura 1.4 Robô móvel holonômico omnidirecional da categoria *Small Size League* 20
Figura 2.1 Modelo de um robô diferencial 23
Figura 2.2 Representação das forças e velocidades atuando em um robô diferencial quando o centro de massa C está deslocado do eixo das rodas por uma distância d 25
Figura 2.3 Diagrama em blocos do controlador de velocidade MRAC 27
Figura 2.4 Diagrama de blocos de um controlador PID 30
Figura 2.5 Diagrama de blocos de um sistema na forma de espaço de estados . . . 31
Figura 2.6 Diagrama do sistema em malha fechada para controle por realimentação de estados 32
Figura 3.1 Arquitetura geral do sistema embarcado. 37
Figura 3.2 Renderização 3D do projeto da PCB 40
Figura 3.3 PCB impressa 40
Figura 3.4 Resposta do motor ao degrau e modelo de primeira ordem ajustado. . . 41
Figura 3.5 Parâmetros adaptativos durante um experimento com referências variadas 47
Figura 3.6 Ação de controle durante experimento para o MRAC padrão e robusto . 47
Figura 3.7 Resposta do motor medida durante o experimento para o MRAC padrão e robusto 48
Figura 3.8 Diagrama de blocos do sistema de controle em cascata PID 48
Figura 3.9 Diagrama de blocos do sistema de controle em cascata com realimentação de estados 51
Figura 3.10 Resposta experimental da velocidade linear v a um degrau de entrada . 53
Figura 3.11 Ajuste do modelo exponencial aos dados experimentais de velocidade linear 53
Figura 3.12 Resposta experimental da velocidade angular ω a um degrau de entrada 54
Figura 3.13 Ajuste do modelo exponencial aos dados experimentais de velocidade angular 54
Figura 4.1 Referência de velocidade linear e velocidades medidas nas duas rodas durante o ensaio 1 59
Figura 4.2 Referência e velocidade medida da roda esquerda durante o ensaio 2 . . 60
Figura 4.3 Referência e velocidade medida da roda direita durante o ensaio 2 . . . 61

Figura 4.4 Referência senoidal e velocidades medidas nas duas rodas durante o ensaio 3 61
Figura 4.5 Referência e velocidade medida da roda direita durante o ensaio 4 . . . 62
Figura 4.6 Referência e velocidade medida da roda esquerda durante o ensaio 4 . . 63
Figura 4.7 Representação da fase 1: movimento linear 64
Figura 4.8 Representação da fase 2: trajetórias circulares 65
Figura 4.9 Representação da fase 3 e 4: trajetória quadrada 65
Figura 4.10 Referência e velocidade linear medida do robô sem controle de movimento 66
Figura 4.11 Referência e velocidade angular medida do robô sem controle de movimento 66
Figura 4.12 Referência e velocidade linear medida do robô com controle PID 67
Figura 4.13 Referência e velocidade angular medida do robô com controle PID . . . 68
Figura 4.14 Referência e velocidade linear medida do robô com controle por realimentação de estados 69
Figura 4.15 Referência e velocidade angular medida do robô com controle por realimentação de estados 69
Figura 4.16 *Frame* capturado do vídeo utilizado para aquisição dos dados 71
Figura 4.17 Comparação entre trajetória real e simulada usando o método sem controle 72
Figura 4.18 Comparação entre trajetória real e simulada usando o controlador PID . 73
Figura 4.19 Comparação entre trajetória real e simulada usando o controlador por espaço de estados 73

Lista de tabelas

Tabela 4.1 Comparação de precisão (MSE) e suavidade (σ) 70
Tabela 4.2 Erro quadrático médio (MSE) entre trajetória simulada e trajetória capturada pela câmera 72

Lista de códigos

Código B.1 Código utilizado na identificação experimental da planta 81

Lista de abreviaturas e siglas

IMU	Unidade de Medida Inercial (*Inertial Measurement Unit*)
LARC	Competição de Robótica da América Latina (Latin American Robotics Competition)
LTI	Linear e invariante no tempo (*Linear Time-Invariant*)
MCU	Microcontrolador (*Microcontroller Unit)*
MIMO	Múltiplas entradas e múltiplas saídas (*Multiple-Input Multiple-Output*)
MRAC	Controlador adaptativo por modelo de referência (*Model Reference Adaptive Controller*)
MSE	Erro quadrático médio (*Mean Squared Error*
PCB	Placa de circuito impresso (*Printed Circuit Board)*
PCI	Placa de circuito impresso
PWM	Modulação por largura de pulso (*Pulse-Width Modulation*)
VSSS	Very Small Size Soccer
ZOH	Segurador de ordem zero (Zero-Order Holder)

Símbolos gregos

γ	Ganho de adaptação dos parâmetros do controle MRAC
Ω	Velocidade de rotação das rodas no domínio de Laplace
ω	Velocidade rotacional ou angular do robô [rad/s]
ω^*	Velocidade angular desejada [rad/s]
ω^o	Velocidade angular de referência do robô [rad/s]
ω_d	Velocidade de rotação da roda direita [rad/s]
ω_e	Velocidade de rotação da roda esquerda [rad/s]
σ	Desvio padrão dos erros de rastreamento
σ_{total}	Soma dos desvios padrão dos erros
τ	Constante de tempo do modelo de primeira ordem
τ_m	Constante de tempo do modelo de referência
τ_p	Constante de tempo do modelo identificado da planta
θ^*	Parâmetro ideal para seguimento da referência no MRAC
θ_1	Parâmetro de adaptação do controlador MRAC
θ_2	Parâmetro de adaptação do controlador MRAC

Símbolos latinos

$\mathbf{A}$	Matriz de dinâmica do sistema

$\mathbf{B}$	Matriz de entrada
$\mathbf{C}$	Matriz de saída
$\mathbf{D}$	Matriz de transmissão direta
$\mathbf{K}$	Matriz de ganhos da realimentação de estados
$\mathbf{N}$	Matriz de ganhos de referência do controlador de realimentação de estados
$\mathbf{u}(\mathbf{t})$	Vetor de entradas de ação de controle em tempo contínuo
$\mathbf{x}(\mathbf{t})$	Vetor de estados em tempo contínuo
$\mathbf{y}(\mathbf{t})$	Vetor de saídas em tempo contínuo
ω_{ref}	Velocidade angular de referência a ser seguida pelo robô [rad/s]
a	Coeficiente do denominador da planta identificada do sistema de propulsão
a_1	Parâmetro dinâmico associado ao amortecimento/atrito na velocidade linear
a_2	Parâmetro dinâmico associado ao termo não linear ω^2
a_3	Ganho dinâmico do comando linear v^o
b	Coeficiente do numerador da planta identificada do sistema de propulsão
b_1	Parâmetro dinâmico associado ao amortecimento/atrito na velocidade angular
b_2	Parâmetro dinâmico associado ao acoplamento entre v e ω
b_3	Ganho dinâmico do comando angular ω^o
$E(s)$	Erro do controlador PID
e_ω	Erro da velocidade angular do robô [rad/s]
K	Ganho estático do modelo de primeira ordem
K_d	Ganho da ação derivativa no controle PID
K_i	Ganho da ação integrativa no controle PID
K_p	Ganho da ação proporcional no controle PID
n	Número total de amostras dos ensaios
$R(s)$	Referência do controlador PID
T	Período do controle
U	Ação de controle aplicada ao motor, no domínio de Laplace
u	Ação de controle aplicada ao motor
v^o	Velocidade linear de referência do robô [m/s]
V_d	Velocidade linear da roda direita [m/s]
V_e	Velocidade linear da roda esquerda [m/s]
v_{ref}	Velocidade linear de referência a ser seguida pelo robô [m/s]
$Y(s)$	Saída do controlador PID
d	Distância entre o centro de massa do robô e o eixo das suas rodas [m]
L	Distância entre as rodas do robô [m]
r	Raio da roda do robô [m]
v	Velocidade translacional ou linear do robô [m/s]

Subscritos

d	Referente à roda direita
e	Referente à roda esquerda
m	modelo de referência

Sumário

1 Introdução 17
1.1 Contextualização e motivação 17
1.2 Definição do problema 19
1.3 Objetivos 20
1.3.1 Objetivo geral 20
1.3.2 Objetivos específicos 20
1.4 Estrutura do trabalho 21

2 Fundamentação Teórica 22
2.1 Modelagem de robôs móveis 22
2.1.1 Cinemática de um robô com tração diferencial 22
2.1.2 Dinâmica de um robô com tração diferencial 24
2.2 Controle Adaptativo por Modelo de Referência 26
2.2.1 Estrutura geral do modelo e controlador 26
2.2.2 Métodos de robustez 29
2.3 Controle Proporcional Integral Derivativo 29
2.4 Controle no Espaço de Estados 30
2.4.1 Formulação geral do modelo no espaço de estados 30
2.4.2 Discretização do modelo 31
2.4.3 Controle por realimentação de estados 32
2.4.4 Controle com seguimento de referência 32
2.4.5 Controlabilidade e estabilidade 33
2.5 Conclusão do capítulo 33

3 Desenvolvimento 35
3.1 Planejamento e decisões de projeto 35
3.1.1 Requisitos gerais do sistema 35
3.1.2 Arquitetura geral do sistema 36
3.1.3 Critérios de escolha de componentes 37
3.2 Implementação do hardware embarcado 38
3.2.1 Processamento e atuação 38
3.2.2 Gestão de energia e interface 39
3.2.3 Projeto da placa de circuito impresso 39
3.3 Controle adaptativo por modelo de referência 39
3.3.1 Identificação experimental da planta do motor 40
3.3.2 Controlador adaptativo por modelo de referência 42

3.3.3 Parâmetros iniciais . . . 43
3.3.4 Métodos de robustez para o MRAC . . . 44
3.4 Controle PID . . . 48
3.5 Controle no espaço de estados . . . 50
3.5.1 Definição do modelo . . . 50
3.5.2 Identificação por resposta natural . . . 51
3.5.3 Determinação dos ganhos dos atuadores . . . 53
3.5.4 Projeto do controlador . . . 55
3.5.5 Controlabilidade . . . 55

4 Resultados . . . 58
4.1 Controle MRAC para as rodas . . . 58
4.1.1 Ensaio 1: Degrau em velocidade linear . . . 59
4.1.2 Ensaio 2: Degrau em velocidade angular . . . 60
4.1.3 Ensaio 3: Referência senoidal em velocidade linear . . . 61
4.1.4 Ensaio 4: Movimento circular com inversão de sentido . . . 62
4.2 Controladores de velocidade do robô . . . 63
4.2.1 Caso 1: Sem controle (malha aberta) . . . 65
4.2.2 Caso 2: Controlador PID na velocidade angular . . . 67
4.2.3 Caso 3: Controle no espaço de estados . . . 68
4.2.4 Comparação quantitativa . . . 70
4.3 Simulação de posição . . . 71

5 Conclusões . . . 75
5.1 Considerações gerais sobre o trabalho . . . 75
5.2 Avaliação dos objetivos alcançados . . . 75
5.3 Perspectivas e trabalhos futuros . . . 76

Referências . . . 77

Apêndices 79

Apêndice A Esquemático . . . 80

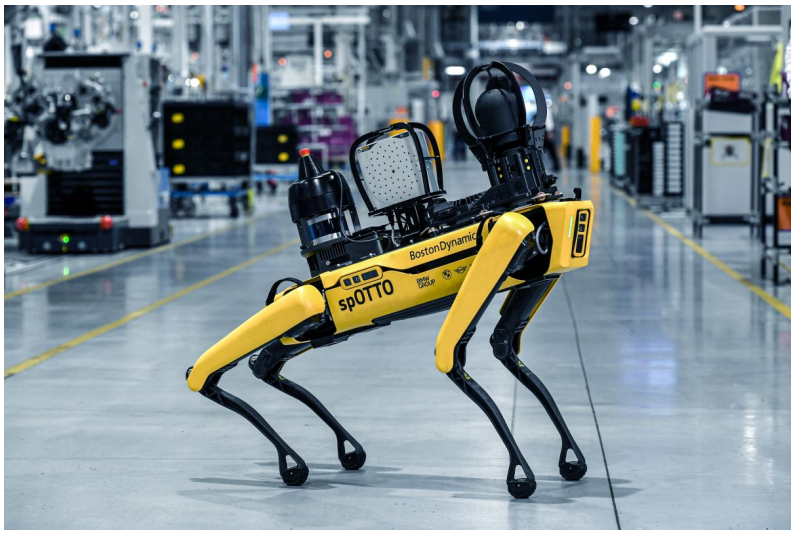
Apêndice B Códigos de programação . . . 81
B.1 Método de identificação de parâmetros de um sistema de primeira ordem . 81

1 Introdução

1.1 Contextualização e motivação

A robótica móvel autônoma tem desempenhado um papel central na transformação de setores industriais, logísticos e de serviços, impulsionando a demanda por sistemas de controle embarcado cada vez mais robustos, compactos e eficientes. Nesse contexto, o desenvolvimento de agentes autônomos capazes de operar em ambientes dinâmicos e não estruturados apresenta desafios significativos de engenharia, que vão desde a eletrônica de potência até algoritmos de controle em tempo real. Robôs comerciais de alto desempenho, como o Spot da Boston Dynamics, ilustrado na figura 1.1, constituem um exemplo icônico de robótica móvel, capaz de operar em ambientes complexos. Esses sistemas evidenciam o potencial dessas tecnologias e o grande interesse que despertam tanto na indústria quanto no público em geral.

Figura 1.1 – O robô Spot, da Boston Dynamics



Fonte: Dynamics (2024)

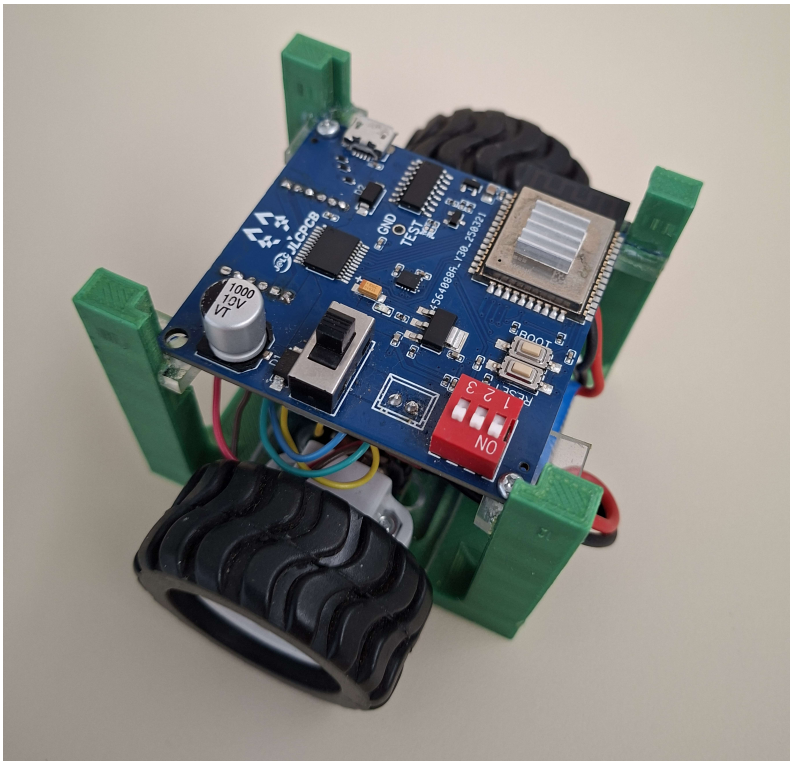
O desafio da robótica móvel tem impulsionado a criação de competições que visam o avanço na área, como a *Latin American Robotics Competition* (LARC), considerada a maior competição de robótica da América Latina. A LARC abriga diversas categorias, entre elas a *Very Small Size Soccer* (VSSS), que consiste em partidas de futebol entre robôs autônomos. Nessa categoria, duas equipes, cada uma composta por três robôs, competem em um jogo de futebol com regras semelhantes às do futebol tradicional.

Essas competições são fundamentais para incentivar o estudo de diversas áreas, como visão computacional, programação orientada a objetos, eletrônica, desenvolvimento de

software embarcado, controle e mecânica. Além disso, por ocorrerem em conjunto com simpósios acadêmicos, essas competições também desempenham um papel essencial na divulgação científica, inspirando jovens de todas as idades a seguir carreiras nas áreas de ciência, tecnologia, engenharia e matemática (Nardi *et al.*, 2016).

A grande complexidade da categoria VSSS reside nas restrições físicas impostas: os robôs devem caber em um cubo de 7,5 cm de aresta. Essa limitação de volume cria um desafio de integração severo, onde microcontroladores, drivers de potência, sensores inerciais e baterias devem coexistir sem interferências eletromagnéticas ou superaquecimento, mantendo a capacidade de processar malhas de controle de alta frequência. A figura 1.2 ilustra o robô desenvolvido neste trabalho, demonstrando a compactação exigida para atender às especificações da categoria, evidenciando a arquitetura em conformidade com as restrições de tamanho (7,5 cm^3).

Figura 1.2 – Robô móvel diferencial desenvolvido neste trabalho para a categoria VSSS



Fonte: Fotografia produzida pelo autor

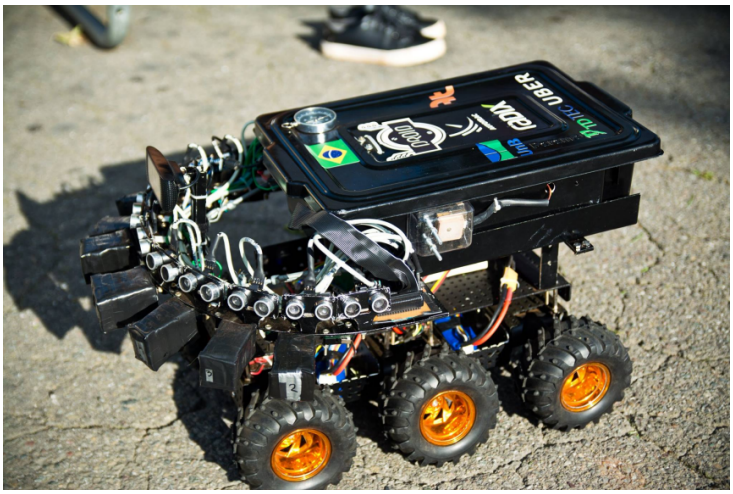
Neste cenário de contínua evolução tecnológica e educacional, a motivação deste trabalho vai além da construção de um robô. O projeto da plataforma embarcada proprietária, baseada no microcontrolador ESP32, surge como um meio para explorar e validar técnicas avançadas de controle, como o controlador adaptativo por modelo de referência e a realimentação de estados, em condições reais e restritas. Dessa forma, busca-se contribuir para o ecossistema da robótica acadêmica com uma arquitetura de hardware e firmware capaz de elevar o nível técnico das competições e servir de base para futuros pesquisadores.

1.2 Definição do problema

O problema central abordado neste trabalho reside na complexidade de controlar o comportamento dinâmico de um sistema robótico autônomo sob restrições físicas e temporais. A construção de um robô móvel para competições como a LARC não se limita à integração de componentes eletrônicos, ela exige o domínio sobre a interação entre o software de controle e a mecânica dos robôs em tempo real.

Um dos desafios primordiais, já explorado em trabalhos correlatos como o de Porto (2017), é lidar com a natureza não-holonômica do robô diferencial (ilustrado na figura 1.3). Diferentemente de sistemas holonômicos (figura 1.4), que possuem total liberdade de movimento instantâneo no plano, o robô diferencial impõe restrições cinemáticas que acoplam seus movimentos lineares e angulares. Essa característica exige que o sistema de controle seja capaz de planejar e executar trajetórias suaves e precisas, compensando a incapacidade do robô de se deslocar lateralmente sem realizar manobras de reorientação.

Figura 1.3 – Robô móvel não-holonômico de tração diferencial da categoria *Robomagellan*

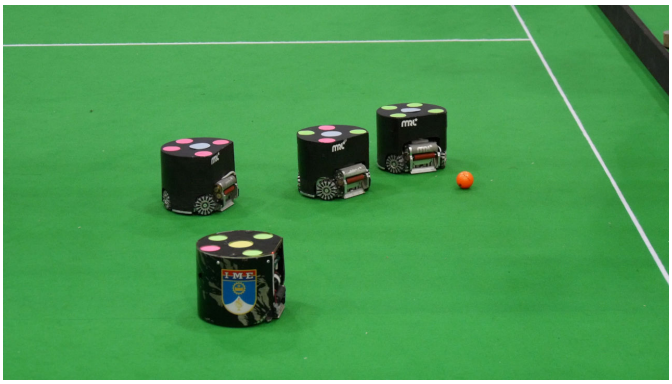


Nota: Robô desenvolvido pela equipe DROID da UnB

Fonte: Adaptada de Porto (2017)

Além das restrições cinemáticas, o projeto enfrenta a dificuldade de garantir a estabilidade e a robustez do controle diante das incertezas e não-linearidades inerentes aos componentes de tamanho reduzido. A variabilidade nos parâmetros dos motores, o atrito com a superfície onde ocorre o jogo e a latência na comunicação e no processamento dos sensores introduzem perturbações que podem degradar significativamente o desempenho do robô se não forem adequadamente tratadas. O problema, portanto, consiste em desenvolver uma arquitetura embarcada que integre o hardware necessário em um espaço exíguo e que também forneça a inteligência de controle necessária para transformar comandos em ações físicas precisas.

Figura 1.4 – Robô móvel holonômico omnidirecional da categoria *Small Size League*



Fonte: (Gigazine, 2017)

1.3 Objetivos

1.3.1 Objetivo geral

O objetivo deste trabalho é construir um robô autônomo, focando na sua parte eletrônica e na programação embarcada para garantir seu correto funcionamento. Para isso, serão abordadas etapas fundamentais, como o desenvolvimento do esquemático eletrônico, o projeto da placa de circuito impresso e a programação do microcontrolador que controlará o robô, incluindo o projeto do controlador de movimento embarcado.

O robô deve ser capaz de receber comandos de velocidade de um sistema externo, processar essas informações e executar movimentos precisos por meio do acionamento dos seus motores, de acordo com as velocidades lineares e angulares de referência. A conclusão bem-sucedida do projeto será avaliada pela capacidade do robô de processar os sinais recebidos e realizar o movimento de forma eficiente e autônoma.

1.3.2 Objetivos específicos

1. Projetar e manufaturar uma placa de circuito impresso (PCI/PCB) compacta integrando microcontrolador, drivers de motor e sensores inerciais;
2. Realizar a modelagem dinâmica e a identificação de parâmetros dos motores DC e do sistema robótico;
3. Projetar e validar um controlador adaptativo por modelo de referência para as rodas;
4. Implementar e comparar o desempenho de estratégias de controle digital para o controle cinemático do robô: PID e realimentação de estados; e
5. Validar o sistema através de ensaios experimentais através de rastreamento de trajetórias completas para avaliar sua robustez a perturbações.

1.4 Estrutura do trabalho

Este trabalho está organizado em cinco capítulos, estruturados da seguinte forma: O capítulo 2 apresenta a fundamentação teórica para os conceitos que vão ser apresentados em seguida, abordando a modelagem cinemática e dinâmica do robô diferencial, além das teorias de controle PID, realimentação de estados e controle adaptativo por modelo de referência. O capítulo 3 detalha o desenvolvimento do hardware e a implementação do firmware, incluindo a identificação da planta e o projeto dos controladores. O capítulo 4 expõe os resultados experimentais obtidos do robô físico, comparando o desempenho das diferentes leis de controle em bancada e em pista. São apresentados gráficos comparativos de desempenho para os diferentes controladores em cenários de teste variados. Por fim, o capítulo 5 apresenta as conclusões retiradas a partir dos resultados, discutindo as limitações do sistema desenvolvido e propondo sugestões para trabalhos futuros.

2 Fundamentação Teórica

Introdução do capítulo

Este capítulo apresenta os principais conceitos teóricos necessários para o desenvolvimento do sistema embarcado de controle de movimento do robô móvel proposto neste trabalho. Inicialmente, são discutidos os modelos de robôs com tração diferencial, abordando a cinemática e a dinâmica do robô. Nessa etapa, são estabelecidas as relações entre as velocidades das rodas e as velocidades linear e angular do robô, bem como um modelo dinâmico simplificado adequado ao projeto de controladores.

Em seguida, é introduzido o controle adaptativo por modelo de referência (*Model Reference Adaptive Control* – MRAC), destacando sua estrutura geral, as leis de adaptação dos parâmetros e os critérios de estabilidade com base na teoria de Lyapunov, além de métodos para aumento de robustez em aplicações práticas. Na sequência, é revisado o controlador Proporcional Integral Derivativo (PID), utilizado como referência clássica para comparação de desempenho.

Por fim, são apresentados os fundamentos do controle no espaço de estados, incluindo a formulação de sistemas lineares invariante no tempo, o processo de discretização para implementação em microcontroladores e a síntese de controladores por realimentação de estados com seguimento de referência

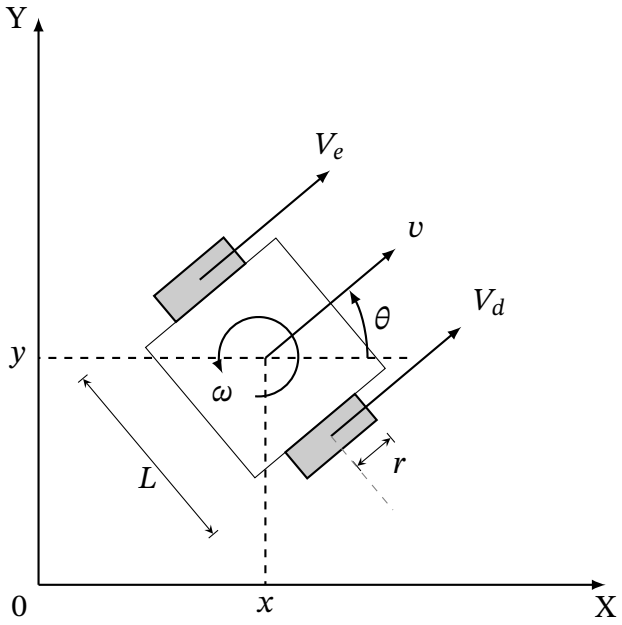
2.1 Modelagem de robôs móveis

2.1.1 Cinemática de um robô com tração diferencial

A cinemática é o ramo da mecânica que descreve o movimento dos corpos sem considerar as forças que o produzem. Seu foco está nas relações geométricas que vinculam posições, velocidades e orientações ao longo do tempo. No contexto de robôs móveis, a cinemática estabelece como as velocidades impostas aos atuadores se traduzem no movimento global do robô no ambiente. Para robôs com tração diferencial, esse estudo é essencial, pois permite relacionar as velocidades angulares das rodas com as velocidades linear e angular do robô, constituindo a base para o planejamento de trajetórias e para o projeto de controladores de movimento.

De acordo com o modelo cinemático de robôs diferenciais não holonômicos (Kim *et al.*, 2004), e assumindo a condição de não escorregamento, as velocidades lineares das rodas esquerda e direita são dadas por:

Figura 2.1 – Modelo de um robô diferencial



Fonte: Elaborada pelo autor

$$V_e = r\,\omega_e, \text{ e} \tag{2.1}$$

$$V_d = r\,\omega_d\,, \tag{2.2}$$

onde r é o raio das rodas e ω_d e ω_e as respectivas velocidades angulares para a roda direita e esquerda.

Para calcular velocidades de referência a serem definidas por trajetórias, deseja-se encontrar as velocidades linear e angular do robô, v e ω, como visto na figura 2.1. Tem-se então a seguinte relação:

$$\begin{aligned} v &= \frac{V_d + V_e}{2} = r\left(\frac{\omega_e + \omega_d}{2}\right), \text{ e} \\ \omega &= \frac{V_d - V_e}{L} = r\left(\frac{\omega_d - \omega_e}{L}\right), \end{aligned} \tag{2.3}$$

ou na forma matricial:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{r}{2} & \frac{r}{2} \\ -\frac{r}{L} & \frac{r}{L} \end{bmatrix} \begin{bmatrix} \omega_e \\ \omega_d \end{bmatrix}, \tag{2.4}$$

onde L é a distância entre as duas rodas.

Portanto, para transformar novamente as velocidades lineares e angulares do robô para velocidades angulares de cada uma das rodas, podem ser usadas as seguintes relações:

$$\begin{aligned} \omega_d &= \frac{v + \left(\frac{L}{2} \cdot w\right)}{r}, \text{ e} \\ \omega_e &= \frac{v - \left(\frac{L}{2} \cdot w\right)}{r}, \end{aligned} \tag{2.5}$$

ou na forma matricial,

$$\begin{bmatrix} \omega_d \\ \omega_e \end{bmatrix} = \begin{bmatrix} \frac{1}{r} & \frac{L}{2r} \\ \frac{1}{r} & -\frac{L}{2r} \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}. \tag{2.6}$$

Esta cinemática será usada para calcular as velocidades angulares de cada uma das rodas dentro do software embarcado após a lei de controle.

2.1.2 Dinâmica de um robô com tração diferencial

Enquanto a cinemática, abordada na seção anterior, descreve a geometria do movimento, a dinâmica descreve a causa desse movimento, relacionando os comandos de torque ou tensão dos motores com as acelerações e velocidades resultantes do robô. A modelagem dinâmica é fundamental para o projeto de controladores onde serão feitas análises de identificação de modelos e para a criação de simulações realistas.

Autores como Alves, Lages e Henriques (2018) e Pedone (2023) demonstram derivações detalhadas do modelo dinâmico de um robô de tração diferencial. Essas derivações, frequentemente baseadas em formulações como a de Newton-Euler, resultam em equações que capturam as complexidades do sistema.

Por exemplo, um modelo *gray-box* (caixa-cinza) de um robô de tração diferencial acionado por motores de corrente contínua pode ser representado como:

$$\dot{v} = a_1 v + a_2 \omega^2 + a_3 v^o, \text{ e}$$

$$\dot{\omega} = b_1 \omega + b_2 v + b_3 \omega^o,$$

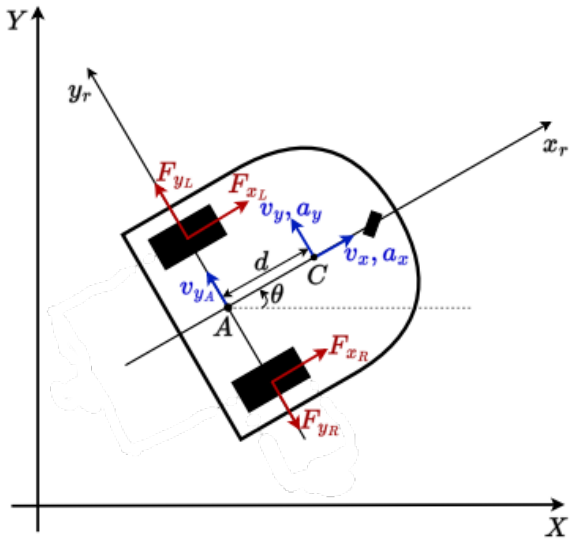
onde v^o e ω^o representam, respectivamente, as velocidades linear e angular de referência impostas ao robô pelo nível superior de controle. De acordo com Pedone (2023) esse modelo é uma simplificação que assume-se que os controladores de velocidade dos motores são controladores PD. Esses valores correspondem aos *set-points* desejados para o movimento do robô, definidos por um planejador de trajetória ou por um controlador cinemático, e são utilizados como entradas para o modelo dinâmico. No sistema real, v^o e ω^o são fornecidos diretamente pelo módulo de controle.

Os coeficientes a_1, a_2 e a_3, bem como b_1, b_2 e b_3, representam parâmetros dinâmicos que agregam diversas características físicas do robô. Conforme indicado em Pedone (2023), essas constantes resultam de combinações entre propriedades como massa total, posição do

centro de massa, momento de inércia, atritos mecânicos, raio das rodas, distância entre as rodas e parâmetros elétricos e mecânicos dos motores. Embora suas expressões analíticas completas sejam extensas, o papel desses coeficientes é determinar como as velocidades linear e angular respondem aos comandos v^o e ω^o, incorporando efeitos como inércia, atrito e acoplamentos dinâmicos.

Note-se a presença de termos não lineares ($a_2\omega^2$) e acoplados ($b_2\omega v$). Esses termos surgem fisicamente quando o centro de massa do robô está localizado a uma distância d do eixo das rodas ($d \neq 0$), conforme ilustrado na figura 2.2. Embora esse modelo seja mais preciso, ele também aumenta a complexidade da identificação dos parâmetros, pois exige a estimação de termos acoplados (a_2 e b_2).

Figura 2.2 – Representação das forças e velocidades atuando em um robô diferencial quando o centro de massa C está deslocado do eixo das rodas por uma distância d



Fonte: Adaptado de Pedone (2023)

No entanto, conforme apontado em Pedone (2023), é possível realizar uma simplificação significativa se assumirmos que o centro de massa do robô está posicionado diretamente sobre o eixo das rodas ($d \approx 0$). Sob esta condição, os termos não-lineares e de acoplamento tornam-se desprezíveis ($a_2 \approx 0$ e $b_2 \approx 0$). Esta suposição simplifica drasticamente as equações dinâmicas para o modelo de velocidade linear:

$$\dot{v} = a_1 v + a_2 v^o, \text{ e} \tag{2.7}$$

$$\dot{\omega} = b_1 \omega + b_2 \omega^o. \tag{2.8}$$

O resultado é um modelo dinâmico linear e, principalmente, desacoplado. Nessa formulação, a dinâmica da velocidade linear v passa a depender exclusivamente do comando de velocidade linear v^o, enquanto a dinâmica da velocidade angular ω é influenciada apenas

pelo comando angular ω^o. Essa separação permite tratar as duas dinâmicas como sistemas independentes, o que simplifica substancialmente o processo de identificação dos parâmetros, dividindo-o em dois problemas menores: um para o subsistema linear e outro para o subsistema angular.

2.2 Controle Adaptativo por Modelo de Referência

O controle de movimento de robôs móveis pode ser estruturado em um sistema de cascata. No nível superior, um controlador de trajetória cinemático determina as velocidades linear e angular necessárias para que o robô siga um caminho pré-definido, que pode ser composto por segmentos de reta e arcos. Para que esta estratégia funcione, é fundamental que um controlador de velocidade, de nível inferior, garanta que os atuadores do robô, os motores, respondam de forma rápida e precisa a essas velocidades de referência.

Quando os parâmetros dinâmicos do sistema da planta são desconhecidos ou variam com o tempo, seja devido a mudanças no atrito, variações na tensão da bateria ou alterações no terreno, controladores com ganhos fixos podem não ser suficientes. Nesses casos, o Controle Adaptativo por Modelo de Referência (MRAC), do inglês *Model Reference Adaptive Controller*, apresenta-se como uma solução robusta. Esta técnica ajusta os parâmetros do controlador em tempo real para forçar o sistema controlado a se comportar como um modelo de referência ideal, mesmo diante de incertezas na planta.

2.2.1 Estrutura geral do modelo e controlador

O princípio fundamental do MRAC é definir um modelo de referência, $H_m(s)$, que encapsula o comportamento dinâmico desejado para o sistema. A estrutura geral desta abordagem, conforme ilustrada na figura 2.3, utiliza uma lei de controle para atualizar os parâmetros e forçar a saída do sistema real, $\omega(t)$, a rastrear a saída do modelo de referência, $\omega_m(t)$.

Considerando para este projeto um sistema de primeira ordem dado pela seguinte função de transferência no domínio de Laplace:

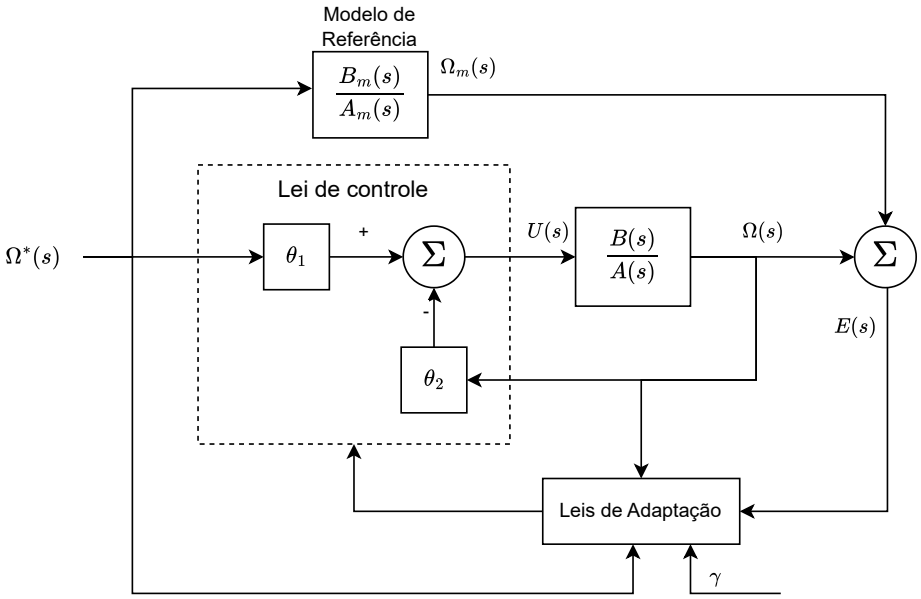
$$G(s) = \frac{\Omega(s)}{U(s)}, \tag{2.9}$$

sendo $\Omega(s)$ a velocidade angular das rodas e $U(s)$ a ação de controle, o modelo pode ser descrito pela seguinte equação diferencial:

$$\dot{\omega}(t) = -a\omega(t) + bu(t), \tag{2.10}$$

com a e b representando, respectivamente, os coeficientes do denominador e do numerador da função de transferência do sistema de propulsão.

Figura 2.3 – Diagrama em blocos do controlador de velocidade MRAC



Fonte: Borges (1998)

Deseja-se que com a lei de controle:

$$U(s) = \theta_1 \Omega^*(s) - \theta_2 \Omega, \tag{2.11}$$

o sistema em malha fechada seja:

$$H_m(s) = \frac{b_m}{s + a_m} = \frac{\Omega(s)}{\Omega^*(s)}, \tag{2.12}$$

com $H_m(s)$ sendo o modelo de referência ao qual a planta deverá seguir.

Portanto, os parâmetros do controlador em 2.11 que levam o sistema em malha fechada a apresentar o comportamento do modelo de referência são:

$$\theta_1 = \frac{b_m}{b}, \text{ e} \tag{2.13}$$

$$\theta_2 = \frac{a_m - a}{b}. \tag{2.14}$$

Como a estimativa inicial dos parâmetros a e b do modelo da planta podem variar ao longo do funcionamento do sistema devido à perturbações e etc, o controlador fará com que o sistema se comporte como o modelo $H_m(s)$ alterando os parâmetros θ_1 e θ_2. O modelo de referência é excitado com a mesmo entrada $\omega^*(t)$ do sistema em malha fechada e produz uma saída $\omega_m(t)$, que, assim como 2.10, pode ser descrita pela equação:

$$\dot{\omega_m}(t) = -a_m \omega_m(t) + b_m \omega^*(t). \tag{2.15}$$

O erro de rastreamento $e(t)$ é a diferença entre a saída real e a saída do modelo:

$$e(t) = \omega(t) - \omega_m(t). \tag{2.16}$$

A garantia de estabilidade do sistema adaptativo é a principal característica do MRAC. A estabilidade, ou seja, a garantia de que o erro de rastreamento $e(t)$ convirja para zero com erro inicial nulo, é demonstrada em (Borges, 1998) através do Segundo Método de Lyapunov. Introduz-se então a equação quadrática:

$$V(e, \theta_1, \theta_2) = \frac{1}{2}\left(e^2 + \frac{1}{b\gamma}(b\theta_1 - b_m)^2 + \frac{1}{b\gamma}(b\theta_2 + a - a_m)^2\right) \tag{2.17}$$

Esta função é positiva definida e só é nula quando o erro $e(t)$ é zero e os parâmetros θ_1 e θ_2 atingem seus valores ideais, definidos nas equações 2.13 e 2.14, respectivamente. Para garantir a estabilidade assintótica, a derivada temporal deve ser negativa definida ($\frac{dV}{dt}$ deve ser estritamente menor que 0). Logo,

$$\frac{dV}{dt} = e(t)\frac{de(t)}{dt} + \frac{1}{\gamma}(b\theta_2 + a - a_m))\frac{d\theta_2}{dt} + \frac{1}{\gamma}(b\theta_1 - b_m)\frac{d\theta_1}{dt}, \tag{2.18}$$

onde γ é o ganho de adaptação dos parâmetros do controlador.

Com

$$\frac{de(t)}{dt} = -a_m e(t) - (b\theta_2 + a - a_m)\omega(t) + (b\theta_1 - b_m)\omega^*(t), \tag{2.19}$$

obtém-se

$$\begin{aligned}\frac{dV}{dt} = -a_m e^2(t) &+ \frac{1}{\gamma}\left(b\theta_2 + a - a_m\right)\left(\frac{d\theta_2}{dt} - \gamma\omega(t)e(t)\right) \\ &+ \frac{1}{\gamma}\left(b\theta_1 - b_m\right)\left(\frac{d\theta_1}{dt} + \gamma\omega^*(t)e(t)\right).\end{aligned} \tag{2.20}$$

Com as regras de adaptação dos parâmetros do controlador dadas por:

$$\frac{d\theta_1}{dt} = -\gamma\omega^*(t)e(t), \text{ e} \tag{2.21}$$

$$\frac{d\theta_2}{dt} = \gamma\omega(t)e(t), \tag{2.22}$$

tem-se que:

$$\frac{dV}{dt} = -a_m e^2(t) < 0. \tag{2.23}$$

Como $a_m > 0$ pois o modelo de referência é estável e $e^2(t) \geq 0$, a condição $\frac{dV}{dt} < 0$ é satisfeita. Em (Åström; Wittenmark, 2008) mostra-se que isso garante que o erro $e(t)$ convirja para zero quando $t \to \infty$.

É importante notar que a convergência do erro para zero não implica, por si só, que os parâmetros θ_1 e θ_2 convirjam para seus valores ideais corretos. A convergência dos parâmetros só é garantida se o sinal de referência $\omega^*(t)$ for persistentemente excitado, ou seja, suficientemente rico em frequências para excitar todas as dinâmicas do sistema.

2.2.2 Métodos de robustez

As leis de adaptação clássicas são sensíveis a problemas práticos, como ruídos de medição, dinâmicas não modeladas e perturbações externas. Na presença de ruído, o erro $e(t)$ pode nunca ser exatamente zero, fazendo com que os parâmetros $\theta(t)$ sejam integrados indefinidamente, num fenômeno conhecido como deriva paramétrica (*parameter drift*).

Para aumentar a robustez do MRAC em implementações reais, diversas modificações na lei de adaptação foram propostas na literatura (Nguyen, 2018). As mais comuns incluem:

- Zona Morta (*Dead-Zone*): A adaptação é desligada se o erro $e(t)$ estiver abaixo de um limiar pré-definido, a zona de ruído ou zona morta;
- Projeção (*Projection*): Os parâmetros são impedidos, ou "projetados de volta", ao sair de uma faixa de valores conhecidos e seguros; e
- *e-Modification*: Adiciona um termo de vazamento na lei de adaptação, que puxa os parâmetros de volta para um valor nominal caso o erro aumente por muito tempo.

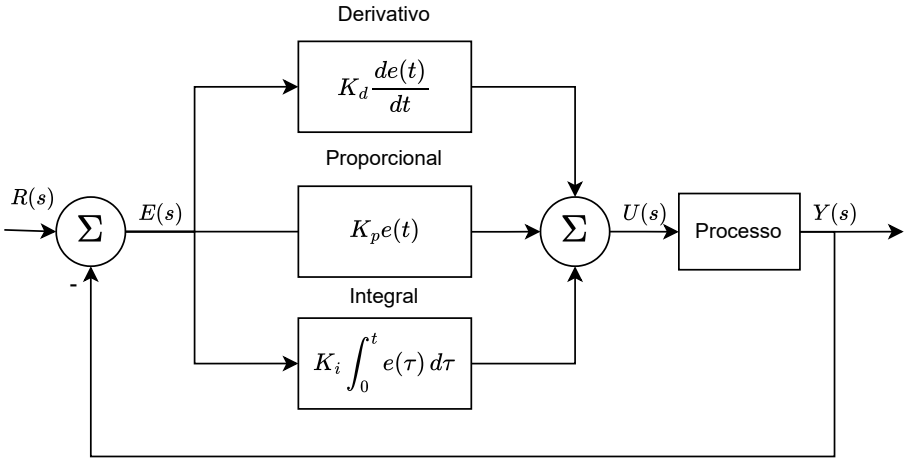
2.3 Controle Proporcional Integral Derivativo

O controlador Proporcional-Integral-Derivativo (PID) é uma estratégia de controle em malha fechada que atua calculando o desvio entre o valor desejado e a variável de processo. Conforme apresentado na figura 2.4, o erro $E(s)$ é obtido pela diferença entre o sinal de referência $R(s)$ e a saída medida do processo $Y(s)$.

O sinal de controle final $U(s)$, que atua sobre o processo, é resultante da soma ponderada de três ações distintas:

- Ação proporcional (K_p): Gera um sinal proporcional à magnitude do erro atual $e(t)$, sendo responsável pela velocidade de resposta inicial do sistema;
- Ação integral (K_i): Baseia-se no histórico acumulado do erro, tendo como principal função eliminar o erro em regime permanente; e
- Ação derivativa (K_d): Atua sobre a taxa de variação do erro ($\frac{de(t)}{dt}$), antecipando o comportamento futuro do sistema e contribuindo para o amortecimento de oscilações.

Figura 2.4 – Diagrama de blocos de um controlador PID



Fonte: Elaborada pelo autor

Matematicamente, a lei de controle no domínio do tempo contínuo, representada pelo diagrama, é expressa pela equação:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}, \tag{2.24}$$

onde K_p, K_i e K_d são os ganhos ajustáveis que determinam a intensidade de cada ação de controle.

2.4 Controle no Espaço de Estados

2.4.1 Formulação geral do modelo no espaço de estados

O controle no espaço de estados é uma abordagem de análise e projeto de sistemas que permite representar sua dinâmica de forma matricial e compacta. Diferentemente dos métodos clássicos baseados em funções de transferência, essa representação possibilita o tratamento de sistemas com múltiplas entradas e saídas, além de permitir a utilização direta das variáveis internas do sistema, conhecidas como estados (Ogata, 2010).

Um sistema linear invariante no tempo (LTI), na forma contínua, pode ser visualizado na figura 2.5 e descrito por:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t), \end{aligned} \tag{2.25}$$

onde:

$\mathbf{x}(t)$ vetor de estados, que representa as variáveis internas necessárias para descrever o comportamento dinâmico do sistema;

$\mathbf{u}(t)$ vetor de entradas, correspondente aos sinais de controle aplicados ao sistema;

$\mathbf{y}(t)$ vetor de saídas, composto pelas variáveis medidas ou observadas;

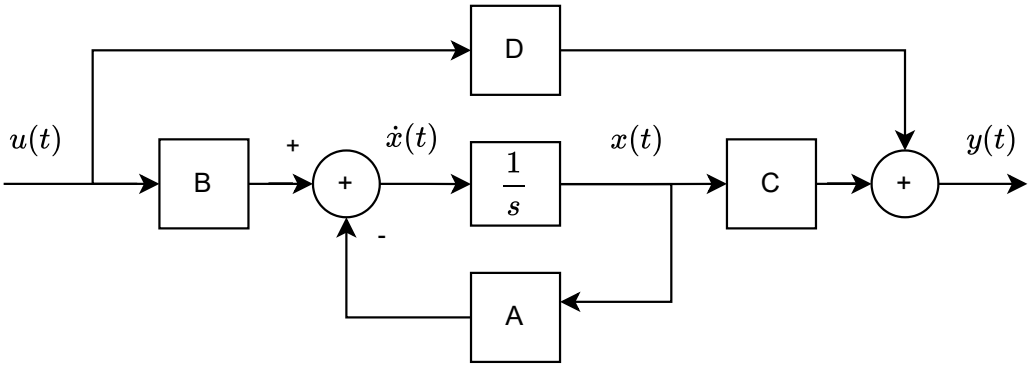
A matriz de dinâmica do sistema, ou matriz de estados, responsável por definir a evolução dos estados ao longo do tempo;

B matriz de entrada, que determina como as entradas influenciam os estados;

C matriz de saída, que relaciona os estados às variáveis observáveis; e

D matriz de transmissão direta, ou alimentação direta, representando a influência imediata da entrada sobre a saída.

Figura 2.5 – Diagrama de blocos de um sistema na forma de espaço de estados



Fonte: Elaborada pelo autor

2.4.2 Discretização do modelo

Como o controle será implementado em um microcontrolador digital, o sistema contínuo descrito em 2.25 precisa ser convertido para a forma discreta, representando sua evolução em instantes k, separados por um período de amostragem T.

Assumindo um segurador de ordem zero (ZOH), que mantém o sinal de entrada $\mathbf{u}(t)$ constante durante o intervalo $kT \leq t < (k+1)T$, a discretização do sistema contínuo resulta nas matrizes **G** e **H** (Ogata, 1995):

$$\mathbf{G} = e^{\mathbf{A}T}, \tag{2.26}$$

$$\mathbf{H} = \left(\int_0^T e^{\mathbf{A}\tau} \, d\tau \right) \mathbf{B}. \tag{2.27}$$

As matrizes de saída e transmissão direta geralmente permanecem as mesmas:

$$\mathbf{C_d} = \mathbf{C}, \qquad \mathbf{D_d} = \mathbf{D}. \tag{2.28}$$

Com isso, o modelo linear invariante no tempo discreto equivalente é dado por:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{G}\mathbf{x}(k) + \mathbf{H}\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{C_d}\mathbf{x}(k) + \mathbf{D_d}\mathbf{u}(k). \end{aligned} \tag{2.29}$$

2.4.3 Controle por realimentação de estados

O controle por realimentação de estados utiliza as variáveis de estado $\mathbf{x}(k)$ para calcular o sinal de controle $\mathbf{u}(k)$. O objetivo é modificar a dinâmica do sistema de malha aberta para garantir a estabilidade e o desempenho transitório desejados.

Isso pode ser alcançado através da alocação de polos, que é a técnica de posicionar os polos de malha fechada em locais desejados no plano Z para garantir o desempenho e a estabilidade do sistema. Para um problema de regulação, onde o objetivo é levar os estados do sistema a zero ou mantê-los em um ponto de equilíbrio nulo, a lei de controle é definida como:

$$\mathbf{u}(k) = -\mathbf{K}\mathbf{x}(k), \tag{2.30}$$

onde $\mathbf{K}$ é a matriz de ganhos de realimentação de estados.

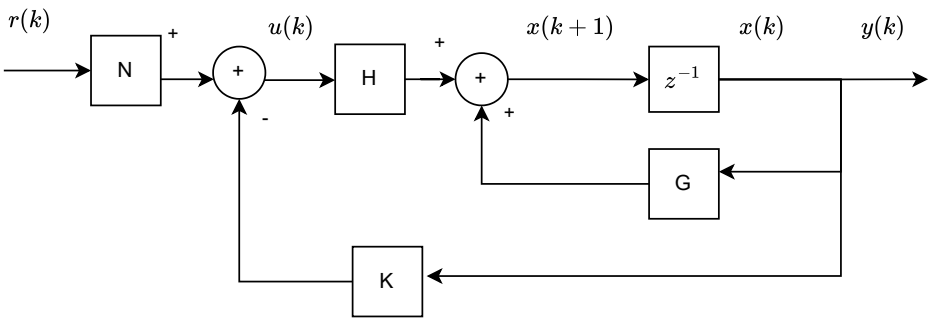
2.4.4 Controle com seguimento de referência

No cenário de seguimento de uma referência $\mathbf{r}(k)$, a lei de controle 2.30 é modificada para incluir um termo de referência, visando que a saída do sistema $\mathbf{y}(k)$ siga $\mathbf{r}(k)$:

$$\mathbf{u}(k) = -\mathbf{K}\mathbf{x}(k) + \mathbf{N}\mathbf{r}(k). \tag{2.31}$$

O ganho de referência $\mathbf{N}$ é introduzido para eliminar o erro em regime permanente, garantindo $\mathbf{y}(k) \to \mathbf{r}(k)$ quando $k \to \infty$. O diagrama de controle do sistema em malha fechada pode ser visto na figura 2.6.

Figura 2.6 – Diagrama do sistema em malha fechada para controle por realimentação de estados



Fonte: Elaborada pelo autor

Matematicamente, o ganho $\mathbf{N}$ é obtido por:

$$\mathbf{N} = \left[\mathbf{C_d}(\mathbf{I} - \mathbf{G} + \mathbf{HK})^{-1}\mathbf{H}\right]^{-1}. \tag{2.32}$$

Este termo garante que, para um sistema estável e sem perturbações, a saída siga corretamente o valor de referência em regime permanente, compensando o efeito de atenuação da realimentação e assegurando erro nulo em estado estacionário. Caso haja uma perturbação, é garantido apenas que o erro será constante (Ogata, 2010).

2.4.5 Controlabilidade e estabilidade

Antes de projetar um controlador por realimentação de estados, é fundamental verificar se o sistema é controlável.

Um sistema linear invariante no tempo discreto $(\mathbf{G}, \mathbf{H})$ é dito completamente controlável por estados se for possível transferir qualquer estado inicial $\mathbf{x}[0]$ para qualquer estado final $\mathbf{x}[k]$ em um número finito de passos (Ogata, 1995).

A condição necessária e suficiente para tal é que a matriz de controlabilidade, representada por:

$$C = \begin{bmatrix}\mathbf{H} & \mathbf{GH} & \mathbf{G}^2\mathbf{H} & \cdots & \mathbf{G}^{n-1}\mathbf{H}\end{bmatrix}, \tag{2.33}$$

possua posto completo, ou seja, $\text{rank}(C) = n$.

Se esta condição for satisfeita, é possível determinar uma matriz $\mathbf{K}$ que posicionará os n polos do sistema em malha fechada em quaisquer localizações desejadas no plano z.

A estabilidade do sistema em malha fechada $(\mathbf{G} - \mathbf{HK})$ depende da localização de seus autovalores. O sistema é assintoticamente estável se e somente se todos os autovalores estiverem estritamente dentro do círculo unitário no plano z, conforme a Equação 2.34.

$$|\lambda_i(\mathbf{G} - \mathbf{HK})| < 1 \quad \forall i = 1, 2, \ldots, n \tag{2.34}$$

A realimentação de estados permite, portanto, garantir a estabilidade do sistema ao mover os polos instáveis para a região interna do círculo unitário.

2.5 Conclusão do capítulo

Este capítulo apresentou os conceitos teóricos necessários ao desenvolvimento do sistema de controle de velocidade para um robô móvel de tração diferencial. Foram descritos a modelagem cinemática e dinâmica do robô, incluindo o modelo simplificado linear e desacoplado, mais adequado ao projeto e à implementação dos controladores.

Em seguida, foram introduzidos o controle adaptativo por modelo de referência (MRAC), com suas leis de adaptação baseadas em Lyapunov e métodos de robustez, e o controlador PID, utilizado como referência clássica para comparação de desempenho. Por fim, foram revisados os fundamentos de controle no espaço de estados, discretização e realimentação de estados com seguimento de referência, destacando-se as condições de controlabilidade e estabilidade.

No próximo capítulo, esses conceitos serão aplicados ao projeto e à implementação dos controladores e do sistema embarcado, permitindo avaliar experimentalmente o desempenho das estratégias de controle estudadas.

3 Desenvolvimento

Introdução

Este capítulo apresenta o processo de desenvolvimento do robô, detalhando as etapas que conduziram desde o planejamento inicial até a implementação dos diferentes subsistemas de hardware e controle. Primeiramente, são definidos os requisitos funcionais e as restrições impostas pela categoria VSSS, que orientaram as decisões de projeto mecânicas, eletrônicas e de arquitetura do sistema embarcado. Em seguida, descrevem-se a implementação da placa de circuito impresso, os módulos de processamento, sensoriamento e atuação, bem como o gerenciamento de energia.

As seções seguintes tratam do desenvolvimento dos sistemas de controle: o controlador adaptativo por modelo de referência (MRAC), o controlador PID para estabilidade rotacional e o controlador baseado em espaço de estados para rastreamento de velocidades linear e angular. Os métodos de identificação experimental, modelagem e projeto de cada controlador são detalhados de forma a evidenciar os critérios empregados e as limitações consideradas na implementação embarcada.

3.1 Planejamento e decisões de projeto

O planejamento do robô partiu da definição dos objetivos funcionais e das restrições impostas pela categoria VSSS uma sequência de decisões voltadas para atender às demandas funcionais e às restrições impostas pela categoria VSSS. Esta seção apresenta a visão geral das decisões de projeto que guiaram o desenvolvimento do sistema, abrangendo desde os requisitos iniciais até a definição da arquitetura geral de hardware e firmware.

3.1.1 Requisitos gerais do sistema

O ponto de partida foi a definição dos requisitos funcionais e construtivos e das limitações. O robô deve ser capaz de atuar de forma autônoma, executando comandos de movimentação e controle de trajetória de maneira confiável e precisa. Para isso, é necessário que o sistema embarcado seja capaz de realizar leitura sensorial em tempo real, controle de velocidade dos motores e comunicação sem fio com o computador central que processa as informações de visão e define as ações estratégicas do robô.

A principal restrição física do projeto está associada ao tamanho máximo do robô, que deve caber em um cubo com arestas de 75 mm conforme estabelecido pelas regras da competição (VSSS Technical Committee, 2023). Essa limitação impactou diretamente a

escolha de componentes e o layout da placa, exigindo a priorização de módulos compactos, a organização cuidadosa dos conectores em uma única região e o uso de rotas curtas para minimizar interferências. O espaço reduzido impôs desafios quanto ao posicionamento dos componentes, gerenciamento de cabos e dissipação térmica dos dispositivos eletrônicos. Além disso, o sistema deve apresentar baixo consumo de energia, de modo a maximizar a autonomia da bateria durante as partidas, e ser robusto o suficiente para suportar vibrações e impactos provenientes das colisões em campo.

Outros requisitos relevantes incluem a necessidade de comunicação sem fio estável, alimentação independente via bateria, facilidade de gravação e depuração do firmware, e medições confiáveis de velocidade e orientação para o controle de movimento. O custo e a disponibilidade comercial dos componentes também foram considerados, buscando-se um equilíbrio entre desempenho técnico e viabilidade econômica.

3.1.2 Arquitetura geral do sistema

A partir dos requisitos e dos componentes definidos, foi estabelecida a arquitetura geral do sistema embarcado, que integra os módulos de hardware e software responsáveis pela execução das funções de comunicação, sensoriamento, controle e atuação do robô. Essa arquitetura visa garantir o processamento em tempo real das informações de movimento e o acionamento preciso dos motores, mantendo a confiabilidade e modularidade do sistema.

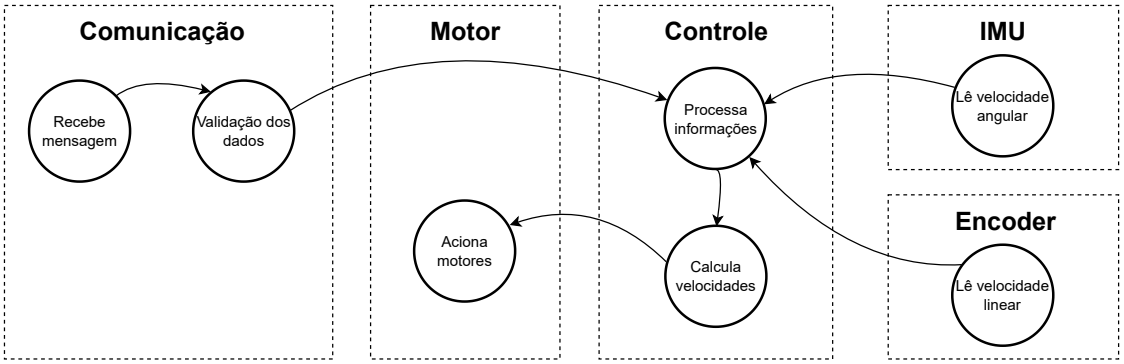
O sistema é organizado em quatro subsistemas principais:

- **Comunicação**: recebe as referências de velocidade enviadas pelo computador central, valida a integridade das mensagens, atualiza os valores de comando e identifica perdas de conexão para acionar mecanismos de segurança.
- **Atuação**: converte os sinais de controle em tensão adequada para os motores, aplica o sentido e a intensidade do comando recebido e garante que a ação física corresponda ao valor calculado pelo controlador.
- **Controle**: processa as medições dos sensores, atualiza periodicamente os algoritmos de controle e calcula os sinais de comando necessários para que as velocidades linear e angular sigam as referências recebidas.
- **Sensoriamento**: mede continuamente a velocidade das rodas e a velocidade angular do robô, filtra e disponibiliza essas informações ao controlador, permitindo a realimentação do sistema.

Todos os subsistemas são integrados pelo microcontrolador **ESP32-WROOM-32E**, que atua como unidade central de processamento, coordenando as tarefas e garantindo o fluxo contínuo de dados entre os módulos. A figura 3.1 apresenta a arquitetura geral do sistema embarcado, destacando o fluxo de informações entre os blocos funcionais.

O diagrama da figura 3.1 mostra o ciclo de operação do sistema: o módulo de comuni-

Figura 3.1 – Arquitetura geral do sistema embarcado.



Fonte: Elaborada pelo autor

cação fornece as referências de velocidade ao módulo de controle, que processa as leituras dos sensores e calcula as ações de controle. Em seguida, os sinais são enviados ao módulo de atuação, que aciona os motores. As medições dos encoders e da IMU são realimentadas ao controle, fechando o ciclo de *feedback* do sistema.

Essa organização modular facilita o desenvolvimento e a depuração de cada parte do sistema de forma independente, além de permitir futuras expansões, como a integração de novos sensores, estratégias de controle ou protocolos de comunicação.

3.1.3 Critérios de escolha de componentes

Com base nos requisitos definidos, os principais componentes do sistema foram selecionados considerando fatores como desempenho, integração, eficiência energética e compatibilidade com o ambiente de desenvolvimento.

O microcontrolador escolhido foi o **ESP32-WROOM-32E**, devido à sua elevada capacidade de processamento, conectividade Wi-Fi integrada e suporte a múltiplas interfaces de comunicação: UART, SPI, I^2C (Espressif Systems, 2024). A opção pelo módulo completo, em vez do chip isolado, deve-se ao fato de o **ESP32-WROOM-32E** já incorporar antena, cristal oscilador, circuito de RF e demais componentes de suporte, o que simplifica o projeto da placa de circuito impresso e aumenta a confiabilidade do sistema. Essas características o tornam especialmente adequado para aplicações embarcadas, reduzindo a necessidade de circuitos externos e facilitando a integração com os demais módulos eletrônicos do robô.

Para o acionamento dos motores, foi selecionado o **TB6612FNG**, um driver de ponte H dupla capaz de fornecer corrente contínua de até 1,2 A por canal com alta eficiência. Esse componente apresenta menores perdas por aquecimento em comparação a alternativas como o **L298N**, além de ser amplamente utilizado em projetos de robótica móvel de pequeno porte.

A unidade de medição inercial (IMU) escolhida foi o **LSM6DS3TR-C**, que combina

acelerômetro e giroscópio em um único encapsulamento e oferece comunicação via barramento I^2C. Esse sensor permite estimar a velocidade angular do robô, auxiliando no controle de trajetória. Além disso, foram utilizados encoders ópticos **enc01a** da Pololu acoplados aos motores de corrente contínua, responsáveis pela medição da velocidade angular das rodas, possibilitando a implementação de controladores de malha fechada.

A alimentação do sistema é feita por uma bateria Li-Po de duas células. A tensão nominal de 7,4V é compatível com a faixa de operação dos motores e do driver, garantindo boa margem de desempenho sem comprometer a segurança elétrica do circuito. O dimensionamento da bateria, de 450 mAh, levou em consideração não apenas o consumo médio dos motores e do microcontrolador, mas também as restrições físicas do robô, já que o componente deve se ajustar ao espaço disponível junto aos demais módulos eletrônicos. Com essa capacidade e considerando uma corrente média de operação entre 1 A e 2 A, a autonomia estimada situa-se entre aproximadamente 10 e 20 minutos, valor suficiente para sustentar uma partida completa de 10 minutos.

Em síntese, os principais requisitos estabelecidos para o projeto são:

- Comunicação sem fio confiável com o computador central;
- Medição precisa das velocidades das rodas e da velocidade angular do robô;
- Baixo consumo de energia e autonomia suficiente para uma partida completa;
- Robustez mecânica e elétrica para suportar colisões e vibrações;
- Modularidade para facilitar manutenção e substituição de componentes; e
- Conformidade com a restrição dimensional de 75 mm da categoria VSSS.

3.2 Implementação do hardware embarcado

A implementação física do robô concentra todos os subsistemas em uma única placa de circuito impresso (PCI), desenhada para maximizar o aproveitamento do espaço interno do chassi. A integração dos componentes segue o diagrama detalhado no esquemático elétrico apresentado no apêndice A, que apresenta as interconexões entre processamento, potência e sensores.

3.2.1 Processamento e atuação

O **ESP32**, núcleo do sistema, é responsável pela gestão dos atuadores através de sinais PWM e pinos de direção. Para o acionamento dos motores DC, optou-se pelo driver **TB6612FNG** em oposição às tradicionais pontes H baseadas em tecnologia bipolar, como o **L298N**. O **TB6612FNG** utiliza MOSFETs em sua estrutura interna, o que reduz a queda de tensão e a dissipação térmica, garantindo que a maior parte da energia da bateria seja convertida em trabalho mecânico, suportando correntes contínuas de até 1,2 A por canal.

O sistema de sensoriamento, essencial para o controle de malha fechada que será detalhado nos capítulos seguintes, é composto por duas frentes. Para o cálculo da velocidade das rodas, o circuito recebe os sinais em quadratura dos encoders ópticos **enc01a**, da Pololu através dos conectores dedicados, com a opção de inclusão de capacitores no meio do caminho entre eles e a porta no MCU, mas que não se mostrou necessário. Complementarmente, a velocidade angular e aceleração do chassi é monitorada pela IMU **LSM6DS3TR-C**, comunicando-se via barramento I^2C. A fusão desses sensores permite compensar o deslizamento das rodas e garantir a precisão da medida das velocidades do robô.

3.2.2 Gestão de energia e interface

A autonomia é garantida por uma bateria Li-Po de duas células (7,4 V), dimensionada para suportar partidas de 10 minutos com margem de segurança. A gestão de energia inclui proteção contra inversão de polaridade e um estágio de regulação de tensão de 3,3 V via CI **AMS1117**, fundamental para a estabilidade do microcontrolador e dos sensores. O circuito prevê ainda capacitores de desacoplamento próximos aos pinos de alimentação de todos os CIs para filtrar ruídos de alta frequência gerados pelos motores.

Para facilitar o desenvolvimento e a depuração em campo, a placa integra um conversor USB-Serial **CH340C**. Isso permite que o firmware seja atualizado e que dados de telemetria sejam visualizados via porta USB sem a necessidade de programadores externos, uma característica necessária para ajustes rápidos durante as competições.

3.2.3 Projeto da placa de circuito impresso

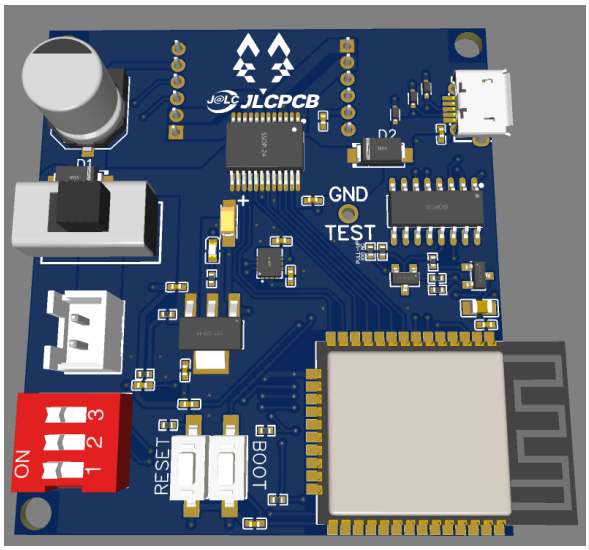
O layout da PCB foi desenvolvido no ambiente *EasyEDA*, utilizando uma estratégia de duas camadas. A face superior acomoda a maioria dos componentes e as trilhas de sinal, enquanto a face inferior é dedicada majoritariamente a um plano de terra (*GND*) sólido. Essa técnica reduz a impedância de retorno e minimiza a interferência eletromagnética (EMI) dos motores sobre o microcontrolador (Johnson; Graham, 1993).

As figuras 3.2 e 3.3 mostram, respectivamente, o projeto da placa em sua renderização 3D e sua versão real fabricada. Nota-se a concentração dos conectores de potência e chaves na borda esquerda para facilitar o acesso manual, enquanto a parte de comunicação foi posicionada na extremidade oposta para minimizar a obstrução da antena.

3.3 Controle adaptativo por modelo de referência

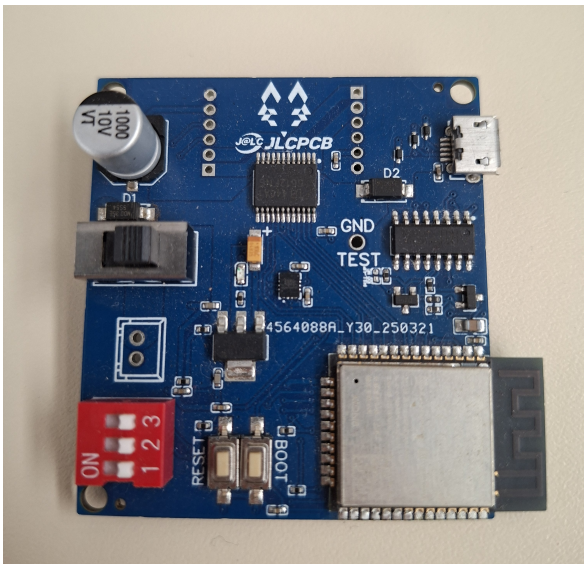
Esta seção descreve o projeto e a implementação do controlador adaptativo por modelo de referência (MRAC) aplicado ao sistema de propulsão do robô. Inicialmente, apresenta-se o procedimento de identificação experimental da planta do motor, utilizado para obter o

Figura 3.2 – Renderização 3D do projeto da PCB



Fonte: Elaborada pelo autor

Figura 3.3 – PCB impressa



Fonte: Fotografia produzida pelo autor

modelo dinâmico necessário ao projeto do controlador. Em seguida, é detalhada a formulação do MRAC em sua versão contínua e discreta, incluindo o modelo de referência adotado, as leis de adaptação e a lei de controle. Na sequência, discute-se a definição dos parâmetros iniciais do controlador e, por fim, são apresentadas as técnicas de robustez incorporadas ao esquema adaptativo, bem como sua influência no comportamento dos parâmetros e na ação de controle.

3.3.1 Identificação experimental da planta do motor

Para o projeto do controle adaptativo das rodas por meio de um modelo de referência, é necessária a identificação do sistema dinâmico. Essa etapa consiste em determinar uma função de transferência que represente o comportamento do conjunto motor–roda, permitindo estimar sua saída a partir de uma dada entrada.

A identificação foi realizada a partir de um experimento com entrada do tipo degrau, de modo a extrair os parâmetros do modelo de primeira ordem da forma da forma:

$$\Omega(s) = \frac{K}{\tau s + 1} U(s). \tag{3.1}$$

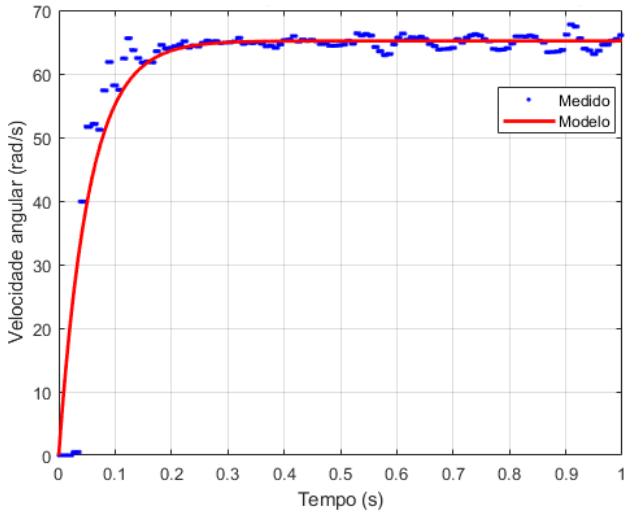
A constante de tempo τ foi obtida pelo ponto em que a resposta ao degrau atinge 63,21% de seu valor em regime permanente. Já o ganho estático K foi determinado como a razão entre o valor em regime permanente da velocidade e a amplitude do degrau aplicado, conforme a equação:

$$K = \frac{\omega(\infty) - \omega(0)}{A}. \tag{3.2}$$

Para reduzir ruídos e obter uma resposta mais estável, aplicou-se inicialmente uma baixa tensão ao motor. No entanto, devido à baixa resolução do encoder, observou-se uma limitação na precisão das medições em baixas velocidades. Assim, adotou-se um valor de PWM mais elevado, gerando aproximadamente 6V e resultando em um maior número de pontos de amostragem durante o regime transitório e, consequentemente, em uma estimativa mais precisa dos parâmetros do modelo.

A figura 3.4 apresenta a curva experimental da velocidade angular medida pelo encoder durante o ensaio de três segundos.

Figura 3.4 – Resposta do motor ao degrau e modelo de primeira ordem ajustado.



Fonte: Elaborada pelo autor

Com base na equação 3.2 e no script MATLAB apresentado no apêndice B.1, obteve-se inicialmente $\tau = 0{,}0475$ s e $K = 0{,}3043$. Esses valores foram então utilizados como condições iniciais para o ajuste fino por meio do método dos mínimos quadrados no **MATLAB**, resultando nos parâmetros finais $\tau = 0{,}0534$ s e $K = 0{,}3047$, empregados nas etapas seguintes de modelagem e controle. Com isso, o modelo contínuo estimado de primeira ordem para os motores é:

$$G_p(s) = \frac{\Omega_p(s)}{U_p(s)} = \frac{0.3047}{0.0534s + 1}. \tag{3.3}$$

O modelo de primeira ordem montado na equação 3.3 foi adotado como representação padrão para todos os motores do robô. Essa decisão deve-se ao fato de que a plataforma utilizada é altamente modular, permitindo a substituição frequente dos motores e rodas durante manutenções e ajustes mecânicos inclusive durante a partida. Como consequência, pequenas variações individuais nos parâmetros dinâmicos tendem a ocorrer a cada troca de componente. Assim, optou-se por empregar um modelo contínuo único e representativo,

de modo a simplificar a implementação do sistema de controle e garantir reprodutibilidade entre diferentes configurações de hardware.

3.3.2 Controlador adaptativo por modelo de referência

Para o projeto do controlador adotou-se o seguinte modelo de referência:

$$H_m(s) = \frac{-p_0}{s - p_0}. \tag{3.4}$$

A relação entre o parâmetro p_0 e a constante de tempo do modelo de referência τ_m é dada por

$$\tau_m = -\frac{1}{p_0}. \tag{3.5}$$

Comparando τ_m com a constante de tempo da planta identificada, $\tau_p = 0{,}0534$ s, deseja-se que o modelo de referência seja ligeiramente mais rápido que o sistema real, mas ainda compatível com sua dinâmica. Essa escolha segue a recomendação de utilizar um modelo desejado fisicamente realizável, capaz de impor um comportamento mais responsivo sem exigir do atuador uma velocidade de resposta além de suas limitações. Adicionalmente, essa definição leva em conta a natureza genérica do modelo da planta adotado em virtude da modularidade do robô, assegurando que a dinâmica de referência imposta seja alcançável mesmo diante das variações paramétricas inerentes às trocas de componentes. Portanto, escolhe-se $p_0 = -100$, obtendo $\tau_m = 0{,}01$ s e o modelo na equação 3.4 torna-se:

$$H_m(s) = \frac{100}{s + 100}. \tag{3.6}$$

A discretização do modelo de referência contínuo foi realizada através do método de segurador de ordem zero (ZOH) com tempo de amostragem $T = 0{,}01$ s. Este procedimento resultou na função de transferência discreta apresentada na equação 3.7. A partir desta, derivou-se a equação de diferenças recursiva 3.8, necessária para a implementação computacional no microcontrolador:

$$G_{disc}(z) = \frac{\Omega_m(z)}{\Omega^*(z)} = \frac{0{,}6321}{z - 0{,}3679}, \text{ e} \tag{3.7}$$

$$\omega_m(k) = 0{,}3679 \cdot \omega_m(k-1) + 0{,}6321 \cdot \omega^*(k-1). \tag{3.8}$$

A implementação digital das regras de adaptação baseia-se nas equações diferenciais 2.21 e 2.22. Estas foram discretizadas por meio da aproximação de Euler de primeira ordem, resultando nas equações de diferenças a seguir:

$$\theta_1(k) = \theta_1(k-1) - T\,\gamma\,\omega^*(k)\,(\omega(k) - \omega_m(k))\,, \text{ e} \tag{3.9}$$

$$\theta_2(k) = \theta_2(k-1) + T\,\gamma\,\omega(k)\,(\omega(k) - \omega_m(k))\,, \tag{3.10}$$

devendo-se substituir as variáveis $\omega^*(k)$, $\omega(k)$ e $\omega_m(k)$ pela velocidade de referência, pela velocidade medida das rodas de tração e pela velocidade calculada pelo modelo de referência, respectivamente, de acordo com o sistema de propulsão do motor esquerdo ou direito.

O ganho de adaptação foi escolhido como $\gamma = 0{,}005$ através de análise experimental com o robô se movimentando por vários minutos. Considerando o período de atualização de controle $T = 0{,}01$s, isso faz com que a relação de adaptação se dê por:

$$\gamma T = 0{,}00005.$$

Por fim, a lei de controle apresentada na equação 2.11 é disposta da forma discreta:

$$u(k) = \theta_1(k) \cdot \omega^* - \theta_2(k) \cdot \omega. \tag{3.11}$$

3.3.3 Parâmetros iniciais

A definição dos valores iniciais dos parâmetros adaptativos $\theta_1(0)$ e $\theta_2(0)$ é fundamental para o desempenho do controlador. Esses parâmetros influenciam diretamente a velocidade de resposta do sistema ao ser inicializado: valores muito baixos podem tornar a resposta lenta e ineficiente, enquanto valores excessivamente altos podem gerar oscilações ou movimentos descoordenados do robô no início da partida.

Neste trabalho, optou-se pela abordagem de casamento de modelos (*Model Matching*) para determinar os valores iniciais. Esta é considerada uma boa prática, pois utiliza o conhecimento da planta para calcular os ganhos ideais θ^* que seriam necessários se o modelo do sistema fosse perfeitamente conhecido. Essa técnica, descrita por Åström e Wittenmark (2008), calcula os parâmetros ideais que tornam a função de transferência de malha fechada idêntica ao modelo de referência desejado.

O modelo de referência $G_m(s)$, que define o comportamento desejado para o sistema em malha fechada, foi escolhido como:

$$G_m(s) = \frac{K_m}{\tau_m s + 1} = \frac{1}{0{,}01s + 1}.$$

Foi adotado um ganho de referência $K_m = 1$ para garantir o seguimento da referência em regime permanente. Com isso, a função de transferência de malha fechada $G_{cl}(s)$ resultante é:

$$G_{cl}(s) = \frac{\Omega(s)}{\Omega^*(s)} = \frac{G_p(s)\theta_1}{1 + G_p(s)\theta_2} = \frac{K_m\theta_1}{\tau_m s + (1 + K_m\theta_2)}.$$

Para que o casamento de modelos seja perfeito ($G_{cl}(s) \equiv G_m(s)$), igualamos as duas equações:

$$\frac{K_m\theta_1}{\tau_m s + (1 + K_m\theta_2)} = \frac{K_m}{\tau_m s + 1}.$$

Igualando os coeficientes dos denominadores e numeradores, obtemos as fórmulas para os parâmetros ideais. Essas fórmulas são a representação em função de transferência do que é apresentado em (Åström; Wittenmark, 2008), que define as condições para o seguimento perfeito do modelo ("*perfect model-following*").

$$\theta_1(0) = \theta_1^* = \frac{K_m \cdot \tau_m}{K_m \cdot \tau_m}, \text{ e}$$
$$\theta_2(0) = \theta_2^* = \frac{(\tau_m/\tau_m) - 1}{K_m}.$$

Substituindo os valores identificados da planta e os valores do modelo de referência tem-se:

$$\theta_1(0) = \frac{1 \cdot 0{,}0534}{0{,}3047 \cdot 0{,}01} \approx 17{,}53$$
$$\theta_2(0) = \frac{(0{,}0534/0{,}01) - 1}{0{,}3047} \approx 14{,}24$$

Embora o método de casamento de modelos sugerisse um ganho inicial de realimentação $\theta_2^* \approx 14{,}24$, optou-se por uma estratégia de inicialização conservadora, definindo $\theta_2(0) = 0$. Essa decisão justifica-se pela necessidade de garantir uma melhor estabilidade no instante de partida, verificada experimentalmente. Ganhos elevados de realimentação inicial tendem a excitar dinâmicas não modeladas e amplificar ruídos de medição, podendo levar à instabilidade antes que o mecanismo adaptativo atue eficazmente. Ao iniciar com $\theta_2 = 0$, o controlador opera inicialmente de forma predominante guiado pelo ganho de *feedforward* θ_1, permitindo que o mecanismo de adaptação ajuste o ganho de realimentação progressivamente e de forma segura conforme a dinâmica do erro se desenvolve.

3.3.4 Métodos de robustez para o MRAC

Um dos principais desafios do controle adaptativo por modelo de referência (MRAC) é o fenômeno conhecido como *parameter drift*. Esse efeito ocorre quando os parâmetros adaptativos crescem indefinidamente ao longo do tempo, mesmo que a resposta do sistema permaneça aparentemente estável. Em geral, isso acontece porque a análise de estabilidade de Lyapunov garante a convergência do erro de rastreamento, mas não assegura limites superiores para os parâmetros adaptativos, especialmente na presença de ruídos, perturbações ou incertezas não modeladas (Nguyen, 2018).

Para lidar com problemas como o *parameter drift*, oscilações ou instabilidades provocadas por incertezas não modeladas e ruídos, diversas modificações robustas foram propostas ao esquema de controle adaptativo por modelo de referência. Neste trabalho são abordadas

três dessas técnicas, baseadas no princípio de limitação dos parâmetros adaptativos ou na adição de termos de amortecimento: o método de **dead-zone**, o método de **projeção** e o método **e-Modification**.

3.3.4.1 Método de dead-zone

O método de dead-zone interrompe o processo de adaptação quando a norma do erro de rastreamento $\|e\|$ é inferior a um limiar e_0. Isso evita a adaptação desnecessária causada por ruídos ou erros residuais pequenos, que podem levar ao crescimento indefinido dos parâmetros adaptativos (conhecido como *parameter drift*). As leis de adaptação para os parâmetros θ_1 e θ_2 com o método de dead-zone são dadas por:

$$\begin{aligned} \frac{d\theta_1}{dt} &= \begin{cases} -\gamma\, \omega^*(t)\, e(t), & \text{se } |e(t)| > e_0; \\ 0, & \text{caso contrário; e} \end{cases} \\ \frac{d\theta_2}{dt} &= \begin{cases} \gamma\, \omega(t)\, e(t), & \text{se } |e(t)| > e_0; \\ 0, & \text{caso contrário.} \end{cases} \end{aligned} \tag{3.12}$$

Sob hipóteses apropriadas, pode-se mostrar, via análise de estabilidade de Lyapunov, que os parâmetros adaptativos permanecem limitados (Nguyen, 2018). O valor de $e_0 = 0{,}01$ foi escolhido experimentalmente baseando-se em que, se muito alto, a adaptação é prematuramente desligada; se muito baixo, o sistema continua vulnerável ao ruído.

3.3.4.2 Método de projeção

O método de projeção impõe que os parâmetros adaptativos θ_i permaneçam dentro de um conjunto convexo S previamente definido, com base em conhecimentos prévios sobre os limites da incerteza do sistema.

O conjunto convexo S utilizado no método de projeção é definido como:

$$S = \left\{ \theta_i \in \mathbb{R}^n : (\theta_i - \theta_i^*)^2 \leq r_i^2, i = 1{,}2 \right\}. \tag{3.13}$$

Essa definição impõe um limite individual r_i para cada parâmetro adaptativo θ_i, permitindo maior flexibilidade na restrição da adaptação com base em conhecimentos prévios sobre as incertezas do sistema.

Quando os parâmetros atingem a fronteira de S e tendem a sair dessa região, o vetor de adaptação é projetado em uma direção tangente à fronteira, impedindo o crescimento indefinido dos parâmetros sem interromper a adaptação.

As leis contínuas com projeção para os parâmetros θ_1 e θ_2 são escritas como:

$$\frac{d\theta_1}{dt} = \begin{cases} -\gamma\, \omega^*(t)\, e(t), & \text{se } (\theta_1 - \theta_1^*)^2 < r_1^2 \text{ ou direção de retorno;} \\ 0, & \text{caso contrário.} \end{cases} \tag{3.14}$$

$$\frac{d\theta_2}{dt} = \begin{cases} \gamma\, \omega(t)\, e(t), & \text{se } (\theta_2 - \theta_2^*)^2 < r_2^2 \text{ ou direção de retorno;} \\ 0, & \text{caso contrário.} \end{cases} \tag{3.15}$$

Neste caso, a projeção é ativada sempre que θ_i tenta sair da esfera, garantindo que $\|\theta_i - \theta_i^*\| \leq r_i$.

3.3.4.3 Método *e*-Modification

O método *e*-Modification adiciona um termo de amortecimento proporcional ao erro de rastreamento à lei de adaptação, com o objetivo de limitar o crescimento dos parâmetros adaptativos mesmo na presença de ruídos ou dinâmica não modelada. Diferente dos métodos anteriores, que interrompem ou restringem diretamente a adaptação, essa abordagem suaviza a atualização dos parâmetros.

As equações 2.21 e 2.22 então são modificadas para:

$$\frac{d\theta_1}{dt} = -\gamma\, \omega^*(t)\, e(t) + \sigma_e\, |e(t)|\, \theta_1(t), \text{ e} \tag{3.16}$$

$$\frac{d\theta_2}{dt} = \gamma\, \omega(t)\, e(t) - \sigma_e\, |e(t)|\, \theta_2(t), \tag{3.17}$$

onde $\sigma_e > 0$ representa o ganho de amortecimento.

O termo $-\sigma_e |e(t)| \theta_i(t)$ reduz a taxa de adaptação de θ_i proporcionalmente à magnitude do erro e ao próprio valor do parâmetro, atuando como um mecanismo de amortecimento. Isso contribui para aumentar a robustez do sistema e evitar deriva paramétrica. Como o ganho de amortecimento é aplicado aos dois parâmetros, θ_1 e θ_2, o sistema ainda gera uma ação de controle proporcional à que geraria caso não fosse usado o método *e*-Modification, então a única mudança é na estabilização dos parâmetros.

A equação geral de atualização dos parâmetros adaptativos com todas as modificações de robustez consideradas neste trabalho pode então ser escrita como:

$$\theta_i[k+1] = \begin{cases} \theta_i[k] + \Delta\theta_i[k], & \text{se } |e[k]| > e_0 \text{ e } \Big[(\theta_i[k] - \theta_i^*)^2 < r_i^2 \\ & \quad \text{ou } (\theta_i[k] - \theta_i^*) \cdot \omega_i[k] \cdot e[k] \leq 0 \Big]; \\ \theta_i[k], & \text{caso contrário,} \end{cases} \tag{3.18}$$

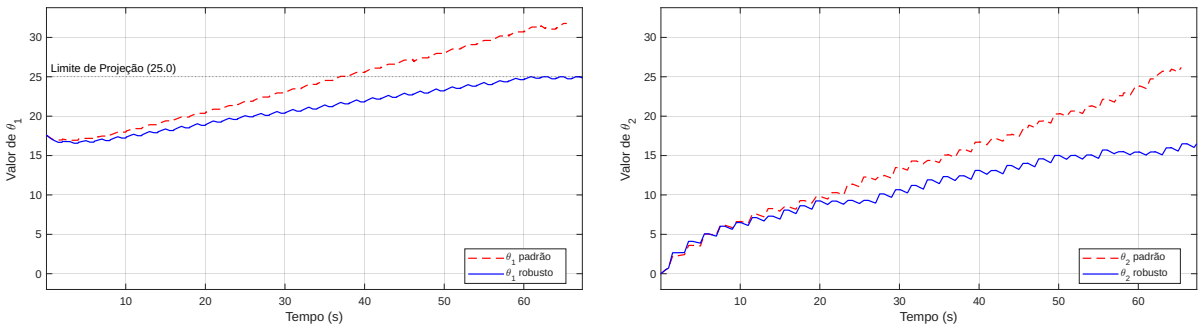
onde $i = 1, 2$ e o termo $\Delta\theta_i[k]$ é dado por:

$$\Delta\theta_i[k] = \begin{cases} -T \cdot \gamma \cdot (\omega^*[k] \cdot e[k] + \sigma_e\, |e[k]|\, \theta_1[k]), & i = 1 \\ T \cdot \gamma \cdot (\omega[k] \cdot e[k] - \sigma_e \cdot |e[k]| \cdot \theta_2[k]), & i = 2 \end{cases} \tag{3.19}$$

Após a implementação dos métodos de robustez pode-se verificar experimentalmente uma melhora no sistema devido a melhor estabilização dos parâmetros θ_1 e θ_2. Na figura 3.5a, o método da projeção não permitiu o aumento de θ_1, fazendo com que o θ_2 se estabilizasse, como pode ser visto na figura 3.5b. Também há uma redução significativa no crescimento de ambos parâmetros, dados os métodos *e-modification* e *deadzone*.

Figura 3.5 – Parâmetros adaptativos durante um experimento com referências variadas

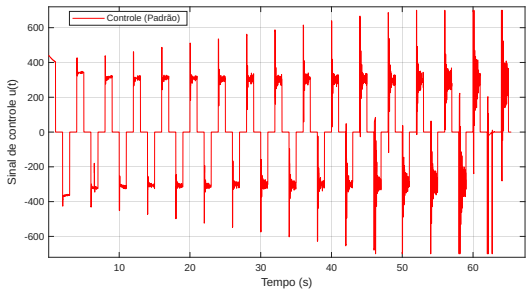
(a) Evolução de θ_1 (b) Evolução de θ_2



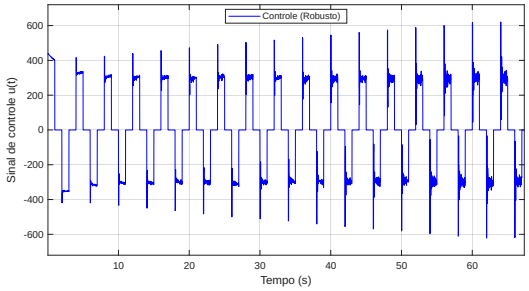
Fonte: Elaborada pelo autor

A ação de controle foi suavizada, como pode ser visto na figura 3.6, evitando picos muito grandes na situação onde há uma mudança brusca do sinal de entrada, o que é muito comum durante a partida.

Figura 3.6 – Ação de controle durante experimento para o MRAC padrão e robusto



Fonte: Elaborada pelo autor

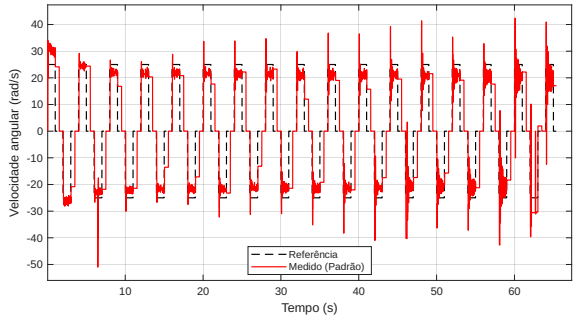


Fonte: Elaborada pelo autor

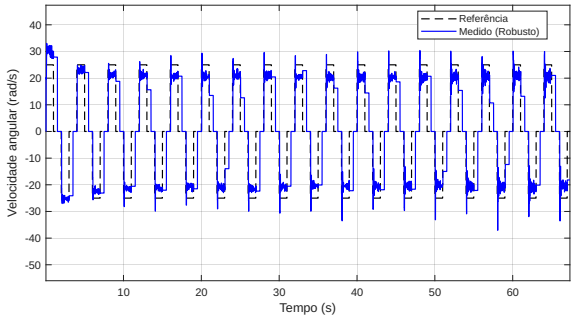
Consequentemente, a ação de controle resultou num rastreamento mais preciso do

sinal e com menos *overshoot*, como mostra a figura 3.7.

Figura 3.7 – Resposta do motor medida durante o experimento para o MRAC padrão e robusto



Fonte: Elaborada pelo autor

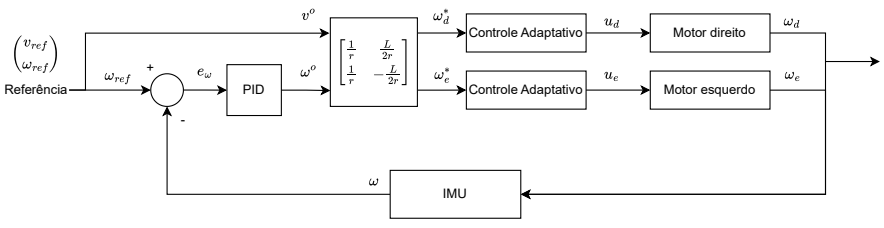


Fonte: Elaborada pelo autor

3.4 Controle PID

Dado que o controle adaptativo atua apenas individualmente em cada uma das rodas, deseja-se projetar um controlador que leve em consideração o sistema como um todo, pois os comandos de velocidade de uma roda influencia diretamente na outra. O controlador escolhido para se iniciar o projeto é o controle proporcional integrativo derivativo, por sua simplicidade de implementação, que envolve apenas somar as três ações de controle que são facilmente calculadas a partir do erro medido.

Figura 3.8 – Diagrama de blocos do sistema de controle em cascata PID



Fonte: Elaborada pelo autor

Para esse trabalho foi projetado um controlador PID que corrige a velocidade rotacional

do robô utilizando a IMU presente na PCI e calculando o erro a partir da própria referência recebida, assim como mostrado no diagrama de blocos da figura 3.8.

Para melhorar a leitura da velocidade rotacional da IMU, foi implementado um filtro complementar assim como descrito por Varaschim *et al.* (2022) a fim de se obter uma estimativa mais estável da velocidade angular utilizando a velocidade calculada pelos encoders, já que o girômetro fornece uma leitura precisa a curto prazo mas que deriva com o tempo e a leitura dos encoders fornece uma leitura ruidosa mas confiável a longo prazo.

O controlador PID implementado utiliza uma abordagem de tempo discreto com ganhos empíricos, onde a lei de controle é executada com período fixo $T = 20\ ms$. A implementação não utiliza a discretização tradicional, que incluiria explicitamente o período de amostragem nos termos integral e derivativo, por duas razões práticas também discutidas na literatura de controle digital (Åström; Wittenmark, 2013):

1. O período de amostragem é fixo no microcontrolador **ESP32**. Em sistemas embarcados com temporização determinística, é possível tratar os ganhos como constantes ajustadas diretamente para o período escolhido, simplificando o código sem comprometer o desempenho (Franklin; Powell; Emami-Naeini, 2015).
2. Os ganhos foram ajustados automaticamente utilizando o algoritmo *Twiddle*, uma estratégia amplamente empregada para sintonia experimental de controladores em robôs móveis por sua simplicidade e eficiência computacional (Thrun; Burgard; Fox, 2005). Dessa forma, os parâmetros K_p, K_i e K_d já incorporam o efeito do período de amostragem, tornando desnecessária a discretização analítica tradicional.

Para prevenir o crescimento indiscriminado do termo acumulador de erro, foi implementado um sistema *anti-windup*, que o reseta caso ele exceda um certo valor $\omega_{\max}$. Este valor foi calculado a partir da estimativa da velocidade angular máxima do robô, que ocorre quando as rodas giram em sentidos opostos com velocidade máxima.

$$\omega_{\max} = \frac{2 \cdot v_{\max}}{L} = \frac{2 \cdot 1.429}{0.0825} \approx 34.6\ \text{rad/s},$$

onde $v_{\max}$ é calculado com base na velocidade máxima do motor e no raio da roda:

$$v_{\max} = 2\pi R \cdot \frac{650\ \text{RPM}}{60} = 2\pi \cdot 0{,}0215 \cdot 10{,}8333 \approx 1{,}429\ \text{m/s}. \tag{3.20}$$

Define-se o erro e_ω como a diferença entre a velocidade de referência ω_{ref} e a velocidade medida ω:

$$e_\omega(t) = \omega_{ref}(t) - \omega(t), \tag{3.21}$$

e a partir da equação 2.24 é montada a lei de controle:

$$u(t) = K_p e_\omega(t) + K_i \int_0^t e_\omega(\tau) d\tau + K_d \frac{de_\omega(t)}{dt}, \tag{3.22}$$

com ω^o sendo a velocidade angular de referência imposta ao robô pelo controlador da camada superior, nesse caso, o controlador PID. Os ganhos obtidos experimentalmente foram $K_p = 1{,}02$, $K_i = 0{,}1$ e $K_p = 0{,}001$

Então, usando a conversão das velocidades linear v e angular ω do robô para as velocidades de rotação das rodas ω_d e ω_e mostradas em 2.5, o comando de controle a ser enviado para a camada do controle adaptativo:

$$\begin{aligned} \omega_R^* &= \frac{v^* + \left(\frac{L}{2} \cdot w^o\right)}{r}, \text{ e} \\ \omega_L^* &= \frac{v^* - \left(\frac{L}{2} \cdot w^o\right)}{r}. \end{aligned} \tag{3.23}$$

3.5 Controle no espaço de estados

Essa seção descreve o desenvolvimento do sistema de controle de velocidade para o robô móvel diferencial, fundamentado na modelagem dinâmica no espaço de estados. A abordagem adotada envolve a definição do modelo matemático da planta, a identificação experimental dos seus parâmetros dinâmicos através de ensaios de resposta ao degrau e o projeto do controlador utilizando alocação de polos.

3.5.1 Definição do modelo

O sistema do robô móvel diferencial possui natureza MIMO (*Multiple-Input Multiple-Output*), com duas entradas e duas saídas. Dessa forma, a modelagem no espaço de estados apresenta-se como uma ferramenta robusta para o projeto do controlador, permitindo tratar o acoplamento dinâmico entre as variáveis.

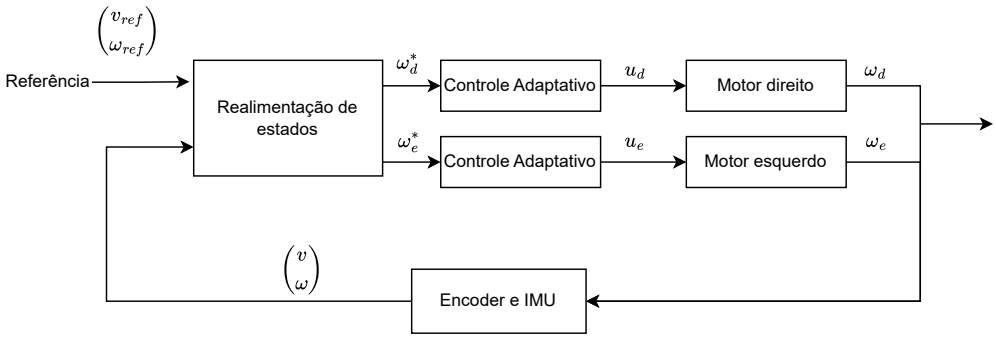
Optou-se pelo projeto de um controlador por realimentação de estados com rastreamento de referência. Embora a inclusão de ação integral pudesse garantir erro nulo em regime permanente frente a distúrbios, escolheu-se a arquitetura com ganho de *feedforward* N pela simplicidade de implementação no microcontrolador.

Para a representação em espaço de estados, utilizou-se o modelo reduzido proposto em (Pedone, 2023) como descrito em 2.8. As equações dinâmicas são apresentadas em 2.7 e 2.8. Em forma matricial, tem-se:

$$\begin{bmatrix} \dot{v}(t) \\ \dot{\omega}(t) \end{bmatrix} = \begin{bmatrix} a_1 & 0 \\ 0 & b_1 \end{bmatrix} \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} + \begin{bmatrix} a_2 & 0 \\ 0 & b_2 \end{bmatrix} \begin{bmatrix} v^o(t) \\ \omega^o(t) \end{bmatrix}. \tag{3.24}$$

O controle em cascata representado no diagrama da figura 3.9 fará com que o controlador no espaço de estados envie as referências de velocidade angular para cada uma das rodas. Logo, para facilitar a visualização do sistema, escreve-se em função das velocidades das rodas. Portanto, substituindo a equação 2.4 em 3.24, obtém-se:

Figura 3.9 – Diagrama de blocos do sistema de controle em cascata com realimentação de estados



Fonte: Elaborada pelo autor

$$\begin{bmatrix} \dot{v}(t) \\ \dot{w}(t) \end{bmatrix} = \begin{bmatrix} a_1 & 0 \\ 0 & b_1 \end{bmatrix} \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} + \begin{bmatrix} a_2\frac{r}{2} & a_2\frac{r}{2} \\ b_2\frac{r}{L} & -b_2\frac{r}{L} \end{bmatrix} \begin{bmatrix} \omega_d \\ \omega_e \end{bmatrix}. \tag{3.25}$$

Por conveniência notacionaol e para ilustrar melhor a dinâmica do modelo, substitui-se a_1 e b_1 por α_v e α_ω, representando o coeficiente de dinâmica, ou seja, a taxa de decaimento natural devido ao atrito e arrasto. E a_2 e b_2 por k_v e k_w, que representam os ganhos dos atuadores.

A medição do estado é feita diretamente, através dos sensores, portanto a equação de saída $\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t)$ pode ser escrita como:

$$\begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix}. \tag{3.26}$$

3.5.2 Identificação por resposta natural

A identificação dos parâmetros do sistema foi realizada experimentalmente através de ensaios de resposta ao degrau em malha aberta, uma metodologia validada por Archela (2019) para a caracterização de sistemas de acionamento elétrico. O procedimento consistiu em aplicar degraus de referência conhecidos aos motores através do controle MRAC e registar a evolução temporal das velocidades linear e angular resultantes.

A figura 3.10 ilustra a resposta experimental da velocidade linear do robô a um degrau de velocidade conhecido. Observa-se o comportamento aproximado de um sistema de primeira ordem, caracterizado por um decrescimento exponencial até um valor de regime estacionário.

A dinâmica do robô em malha fechada pode ser aproximada (Pedone, 2023) pela equação diferencial linear de primeira ordem:

$$\dot{v} = -\alpha v + k \cdot v_{ref}, \tag{3.27}$$

onde α é o coeficiente de dinâmica (taxa de decaimento natural da velocidade devido ao atrito e perdas do sistema) e k é o ganho estático do atuador, relacionando o comando v_{ref} com a velocidade de regime permanente do robô.

Durante a fase de aceleração, a evolução temporal da velocidade depende simultaneamente de ambas as incógnitas. A identificação simultânea destes parâmetros a partir da curva de subida pode resultar em múltiplos pares de soluções que satisfazem o ajuste numérico, reduzindo a confiabilidade individual de cada parâmetro.

Em contrapartida, durante a fase de decaimento, quando $v_{ref} = 0$, o termo de entrada e o ganho k são eliminados da Equação 3.27, simplificando a dinâmica para:

$$\dot{v} = -\alpha v. \tag{3.28}$$

Este procedimento isola o parâmetro α como a única variável desconhecida, permitindo sua determinação direta através do ajuste de uma curva exponencial aos dados experimentais. Uma vez obtido o valor de α, este é reinserido na equação de estado estacionário ($\dot{v} = 0$) da fase forçada para determinar o ganho k, garantindo um modelo final fisicamente consistente.

Para a caracterização dinâmica, os dados experimentais recolhidos foram ajustados a um modelo exponencial de primeira ordem decrescente da forma:

$$y(t) = a \cdot e^{-b(t-c)}, \tag{3.29}$$

onde $y(t)$ representa a velocidade (linear ou angular), e a, b e c são os coeficientes determinados pelo ajuste de curva realizado no MATLAB usando o *curve fitting*. Esta abordagem gráfica e analítica permite extrair as constantes de tempo e ganhos estáticos do sistema, alinhando-se com métodos clássicos de identificação como o de Ziegler-Nichols e Hägglund, discutidos em (Archela, 2019).

A figura 3.11 apresenta a sobreposição dos dados experimentais com a curva do modelo ajustado, demonstrando a adequação do modelo de primeira ordem para representar a dinâmica do sistema.

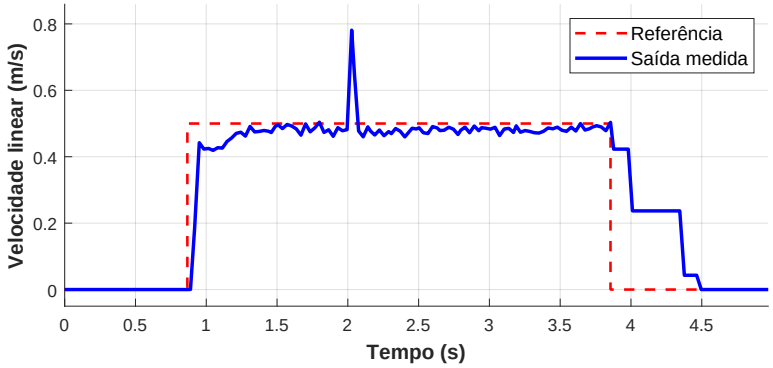
A partir deste ajuste, foram obtidos os parâmetros físicos que descrevem a dinâmica do robô: as constantes de tempo, representadas por α_v e α_ω, e os ganhos estáticos do sistema, representados por k_v e k_ω. Os valores identificados para o protótipo foram:

- Coeficiente de dinâmica linear: $\alpha_v = 2.6166$, e
- Ganho estático linear: $k_v = 2.7563$.

Devido à baixa qualidade do ajuste R^2 na medição do decaimento da velocidade angular, ilustrada na figura 3.13, os parâmetros dinâmicos rotacionais (α_ω, k_ω) foram assumidos iguais aos parâmetros lineares (α_v, k_v), uma aproximação válida dada a simetria do sistema de acionamento e a utilização de motores supostamente idênticos.

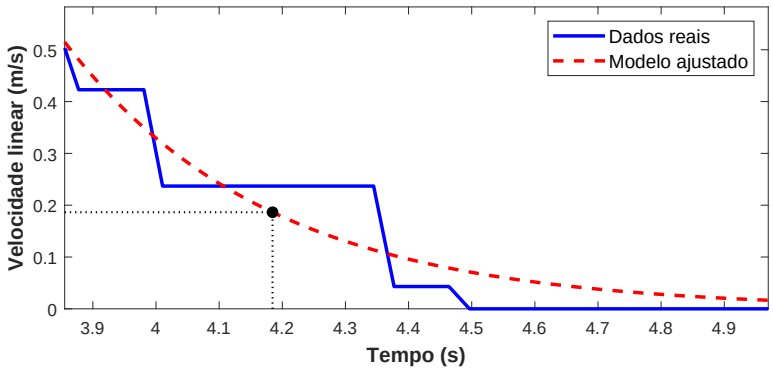
Assim, definiu-se:

Figura 3.10 – Resposta experimental da velocidade linear v a um degrau de entrada



Fonte: Elaborada pelo autor

Figura 3.11 – Ajuste do modelo exponencial aos dados experimentais de velocidade linear



Fonte: Elaborada pelo autor

- Coeficiente de dinâmica angular: $\alpha_\omega = 2.6166$, e
- Ganho estático angular: $k_\omega = 2.7563$.

3.5.3 Determinação dos ganhos dos atuadores

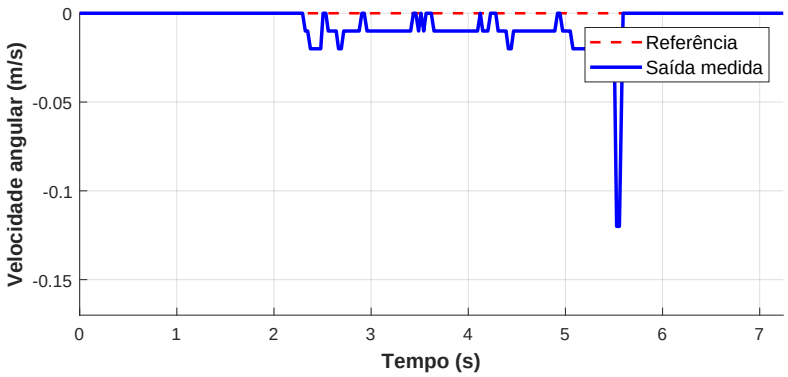
Os parâmetros k_v e k_w foram determinados experimentalmente através da análise da resposta em regime permanente ($t \to \infty$) para o sistema permanentemente excitado. Quando a velocidade estabiliza ($\dot{v} = 0$ e $\dot{w} = 0$) para uma entrada degrau constante, as equações diferenciais se tornam algébricas:

$$0 = -\alpha_v v_{ss} + k_v v_{ref} \implies k_v = \alpha_v \frac{v_{ss}}{v_{ref}}, \text{ e} \tag{3.30}$$

$$0 = -\alpha_\omega \omega_{ss} + k_\omega \omega_{ref} \implies k_\omega = \alpha_\omega \frac{\omega_{ss}}{\omega_{ref}}, \tag{3.31}$$

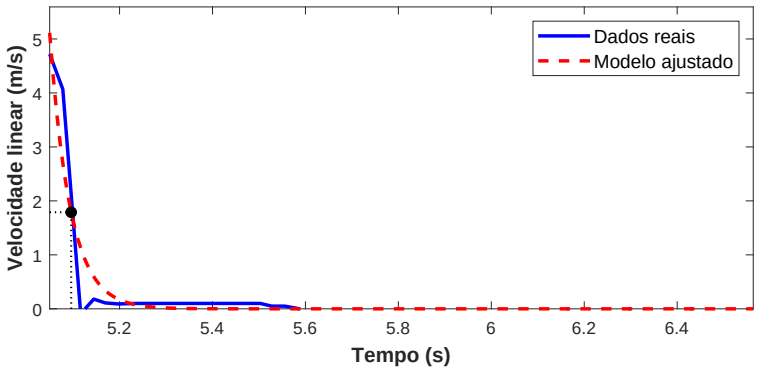
onde v_{ss} e ω_{ss} são as velocidades medidas em estado estacionário e v_{ref} e ω_{ref} são as amplitudes dos degraus aplicados.

Figura 3.12 – Resposta experimental da velocidade angular ω a um degrau de entrada



Fonte: Elaborada pelo autor

Figura 3.13 – Ajuste do modelo exponencial aos dados experimentais de velocidade angular



Fonte: Elaborada pelo autor

Utilizando os dados experimentais obtidos: Para a velocidade linear: Com $\alpha_v = 2.6166$ s^{-1}, $v_{ref} = 0.5$ m/s e uma velocidade estacionária média medida de $v_{ss} \approx 0.5267$ m/s, obteve-se:

$$k_v = 2.6166 \cdot \frac{0.5267}{0.5} \approx 2.7563\ s^{-1}. \tag{3.32}$$

Para a velocidade angular também assumiu-se por simetria do sistema de acionamento. Assim, adotou-se $k_\omega \approx k_v$:

$$k_\omega = 2.7563\ s^{-1}. \tag{3.33}$$

Substituindo os valores obtidos nos experimentos e os parâmetros r e L do robô na equação 3.24, obtemos a representação no espaço de estados do robô:

$$\begin{bmatrix} \dot{v}(t) \\ \dot{\omega}(t) \end{bmatrix} = \begin{bmatrix} -2.6166 & 0 \\ 0 & -2.6166 \end{bmatrix} \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} + \begin{bmatrix} 0.0296 & 0.0296 \\ -0.7183 & 0.7183 \end{bmatrix} \begin{bmatrix} \omega_d(t) \\ \omega_e(t) \end{bmatrix}. \tag{3.34}$$

Discretizando-a por meio de um segurador de ordem zero com período $T = 20\ ms$, se torna:

$$\begin{bmatrix} v(k+1) \\ \omega(k+1) \end{bmatrix} = \begin{bmatrix} 0.9490 & 0 \\ 0 & 0.9490 \end{bmatrix} \begin{bmatrix} v(k) \\ \omega(k) \end{bmatrix} + \begin{bmatrix} 0.0006 & 0.0006 \\ -0.0140 & 0.0140 \end{bmatrix} \begin{bmatrix} \omega_d(k) \\ \omega_e(k) \end{bmatrix}. \tag{3.35}$$

3.5.4 Projeto do controlador

Para fechar a malha e controlar o sistema, projeta-se uma realimentação de estados em tempo discreto com descrito pela equação 2.31. A escolha dos polos foi feita experimentalmente levando em consideração a estabilidade do sistema mais do que a rapidez da sua resposta, pois devido às muitas simplificações feitas ao longo do projeto, o sistema perderia sua estabilidade muito facilmente durante o funcionamento. Os polos escolhidos foram $p_1 = 0{,}94$ e $p_2 = 0{,}95$. Após a escolha dos polos de malha fechada e do cálculo de K, é possível calcular o valor de N, assim como mostra a equação 2.32. O diagrama em malha fechada pode ser ilustrado pela figura 2.6.

3.5.5 Controlabilidade

Para implementar o controlador é necessário analisar sua controlabilidade, verificando o posto da matriz de controlabilidade C. Como o sistema é de ordem $n = 2$, ou seja, possui dois estados, a definição da matriz de controlabilidade de acordo com a equação 2.33 é:

$$C = \begin{bmatrix} H & GH \end{bmatrix}. \tag{3.36}$$

Dadas as matrizes G e H do sistema calculadas em 3.35:

$$G = \begin{bmatrix} 0.9490 & 0 \\ 0 & 0.9490 \end{bmatrix} \text{ e } H = \begin{bmatrix} 0.0006 & 0.0006 \\ -0.0140 & 0.0140 \end{bmatrix}, \tag{3.37}$$

primeiramente calcula-se o termo GH:

$$GH = \begin{bmatrix} 0.9490 & 0 \\ 0 & 0.9490 \end{bmatrix} \begin{bmatrix} 0.0006 & 0.0006 \\ -0.0140 & 0.0140 \end{bmatrix} = \begin{bmatrix} 0.0005694 & 0.0005694 \\ -0.013286 & 0.013286 \end{bmatrix}.$$

Concatenando as colunas:

$$C = \begin{bmatrix} 0.0006 & 0.0006 & 0.0005694 & 0.0005694 \\ -0.0140 & 0.0140 & -0.013286 & 0.013286 \end{bmatrix}. \tag{3.38}$$

Como o sistema possui $n = 2$ estados, é necessário que $posto(C) = 2$. A análise mostra que as duas primeiras colunas são linearmente independentes, enquanto as demais são múltiplos destas e não acrescentam novas direções ao espaço de estados. Portanto,

$$\text{posto}(C) = 2.$$

Conclui-se que o sistema é completamente controlável.

Além disso, com os valores dos polos do sistema de malha fechada escolhidos anteriormente, é possível calcular K e N:

$$K = \begin{bmatrix} 42{,}4460 & -1{,}7509 \\ 42{,}446 & 1.7509 \end{bmatrix}, \text{ e } N = \begin{bmatrix} 86{,}6003 & -3{,}5723 \\ 86{,}6003 & 3{,}5723 \end{bmatrix}, \tag{3.39}$$

e a estabilidade do sistema de acordo com a equação 2.34:

$$HK = \begin{bmatrix} 0{,}0006 & 0{,}0006 \\ -0{,}0140 & 0{,}0140 \end{bmatrix} \begin{bmatrix} 42{,}4460 & -1{,}7509 \\ 42{,}4460 & 1{,}7509 \end{bmatrix} \approx \begin{bmatrix} 0{,}0509352 & \approx 0 \\ \approx 0 & 0{,}0490252 \end{bmatrix}.$$

Então,

$$A_{cl} = G - HK \approx \begin{bmatrix} 0{,}9490 & 0 \\ 0 & 0{,}9490 \end{bmatrix} - \begin{bmatrix} 0{,}0509352 & 0 \\ 0 & 0{,}0490252 \end{bmatrix} = \begin{bmatrix} 0{,}8980648 & 0 \\ 0 & 0{,}8999748 \end{bmatrix}.$$

Os autovalores são simplesmente os elementos da diagonal:

$$\lambda_1 \approx 0{,}8981, \quad \lambda_2 \approx 0{,}9000.$$

Como se trata de sistema discreto, a condição de estabilidade assintótica é

$$|\lambda_i| < 1 \quad \text{para todo } i.$$

Logo, o sistema é estável pois:

$$|\lambda_1| \approx 0{,}8981 < 1, \quad |\lambda_2| \approx 0{,}9000 < 1.$$

Conclusão do capítulo

Neste capítulo foram apresentados os fundamentos e etapas práticas que estruturaram o desenvolvimento do robô. Partiu-se da definição dos requisitos funcionais e das restrições impostas pela categoria VSSS, que guiaram a escolha de componentes, a organização da arquitetura embarcada e o planejamento do layout da placa. A implementação eletrônica integrou processamento, atuação e sensoriamento em uma única plataforma compacta e adequada às demandas de competição.

No âmbito do controle, foram desenvolvidas três abordagens complementares: o MRAC para controle individual das rodas, o controlador PID para estabilização da velocidade angular do robô, e o controlador no espaço de estados para rastreamento das velocidades linear e angular. Cada método foi fundamentado em modelagem e identificação experimental da planta, garantindo coerência entre o modelo adotado e o comportamento real do sistema. Técnicas de robustez também foram incorporadas ao MRAC, assegurando estabilidade paramétrica durante a operação.

De forma integrada, o desenvolvimento apresentado estabelece a base hardware-software necessária para os experimentos, análises de desempenho e validações mostrados no capítulo seguinte, no qual são avaliados os resultados práticos do robô operando sob as estratégias de controle projetadas.

4 Resultados

Introdução

Este capítulo apresenta os resultados experimentais obtidos com o robô desenvolvido, visando verificar se o sistema atende aos objetivos de controle de movimento estabelecidos no Capítulo 1. São avaliados, inicialmente, o desempenho do controlador adaptativo por modelo de referência (MRAC) aplicado às rodas, em diferentes perfis de referência, e, em seguida, as estratégias de controle de velocidade do robô (malha aberta, PID e controle no espaço de estados), comparadas por meio de métricas quantitativas. Por fim, são apresentados ensaios de seguimento de trajetória com auxílio de visão computacional, que permitem avaliar o efeito dos controladores no deslocamento espacial do robô.

4.1 Controle MRAC para as rodas

Esta seção apresenta os ensaios realizados com o robô físico para avaliar o desempenho adaptativo do controlador MRAC implementado. Os testes foram conduzidos em ambiente controlado, com o robô operando sobre superfície plana e recebendo referências de velocidade linear e angular previamente definidas. Os comandos foram enviados a partir de um módulo *ESP32* transmissor, enquanto o robô retornava, a cada 10 ms, um pacote de dados contendo as medições relevantes para análise, incluindo velocidades das rodas, velocidades linear e angular e parâmetros adaptativos estimados.

O objetivo é verificar a capacidade do controlador de seguir diferentes perfis de referência, incluindo degraus e sinais variáveis no tempo, além de observar a atuação adaptativa em situações que exigem aceleração, desaceleração e coordenação entre as rodas. Cada ensaio inclui gráficos contendo as referências impostas e as velocidades medidas, permitindo analisar o comportamento dinâmico do sistema sob diferentes condições. De forma geral, a avaliação cuidadosa do MRAC é fundamental, pois os controladores de movimento do robô dependem diretamente de sua capacidade de ajustar-se às variações dinâmicas e manter o rastreamento estável das velocidades linear e angular, contendo as referências impostas e as velocidades medidas, permitindo analisar o comportamento dinâmico do sistema sob diferentes condições.

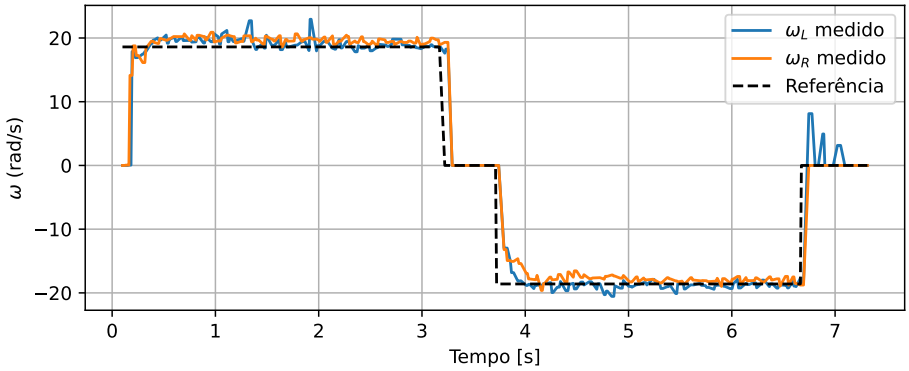
Para cada cenário de teste, foram realizadas três repetições independentes do ensaio, mantendo-se as mesmas condições de operação, referências impostas e ambiente de teste. O objetivo dessas repetições foi verificar a reprodutibilidade do comportamento do robô e identificar possíveis variações decorrentes de ruído de medição, pequenas diferenças de

atrito no solo ou variações espontâneas no desempenho dos motores. Entre as três execuções, observou-se que todas apresentaram a mesma tendência dinâmica e o mesmo padrão qualitativo de resposta, diferindo apenas em pequenas flutuações naturais dos sensores. Assim, para fins de apresentação gráfica neste capítulo, foi selecionada a repetição mais representativa, definida como aquela cujo sinal medido apresentava menor quantidade de ruído e melhor legibilidade visual, sem alterar o comportamento geral observado nas demais execuções. Essa escolha não afeta a interpretação dos resultados, uma vez que todas as repetições exibiram o mesmo desempenho do controlador.

4.1.1 Ensaio 1: Degrau em velocidade linear

O primeiro ensaio teve como objetivo avaliar o desempenho do controlador MRAC diante de uma referência de velocidade linear aplicada igualmente às duas rodas. Nesse teste, foram utilizados valores de referência próximos de 0,40 m/s para avanço e −0,40 m/s para recuo, o que corresponde aproximadamente a 19 rad/s nas rodas, considerando o raio de 0,0215 m. O robô foi posicionado no solo e recebeu um comando composto por quatro fases: avanço, parada curta, recuo e parada final. Essa configuração permite observar o comportamento adaptativo do controlador tanto para acelerações positivas quanto negativas, bem como analisar a simetria entre as duas rodas.

Figura 4.1 – Referência de velocidade linear e velocidades medidas nas duas rodas durante o ensaio 1



Fonte: Elaborada pelo autor

A figura 4.1 evidencia que a transição para velocidade zero ocorre de forma estável. Isso ocorre porque o valor aplicado ao motor é forçado a zero sempre que a referência de velocidade é nula, interrompendo a corrente e resultando em frenagem passiva, que se mostrou mais estável que a frenagem ativa utilizada originalmente pelo controlador. Desse modo, o driver recebe um comando nulo e interrompe o fornecimento de corrente aos motores, produzindo uma frenagem puramente passiva, que se mostrou mais estável que a frenagem ativa utilizada anteriormente, a qual aplicava torque contrário e gerava oscilações indesejadas.

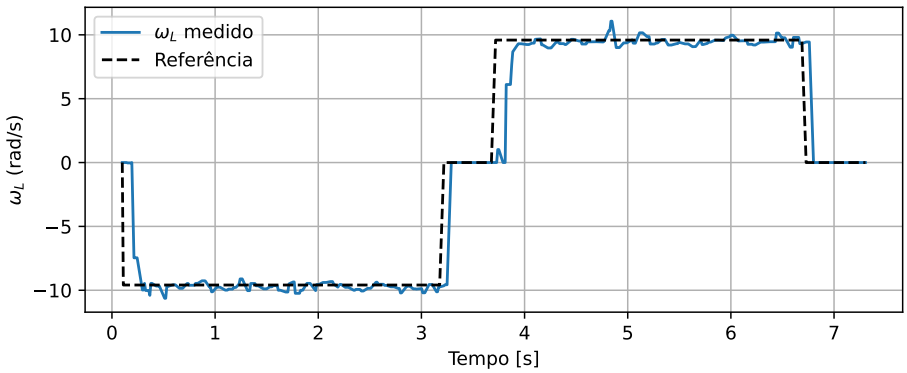
Observa-se também um pequeno pico oscilatório na velocidade da roda esquerda ao final do ensaio. Esse efeito ocorre porque, ao cortar a corrente dos motores, o robô ainda apresenta um leve movimento residual antes de parar completamente, o que é captado pelo encoder como uma pequena variação. Como a roda esquerda oferece um atrito um pouco menor, o robô tende a girar ligeiramente para a direita nesse instante. Essa oscilação é de baixa amplitude e não compromete a estabilidade do sistema.

4.1.2 Ensaio 2: Degrau em velocidade angular

O segundo ensaio avaliou a resposta do controlador MRAC para um comando de velocidade angular $\omega = 5$ rad/s, um teste especialmente relevante por representar uma condição típica de manobras rápidas no campo de competição para um comando de velocidade angular, no qual o robô realiza giros em torno do próprio eixo. A referência é composta por quatro fases: giro no sentido horário, parada curta, giro no sentido anti-horário e parada final. Como a geração de velocidade angular exige comandos opostos nas duas rodas, as referências aplicadas não são idênticas.

As figuras 4.2 e 4.3 apresentam, respectivamente, o comportamento das rodas esquerda e direita durante o ensaio de velocidade angular. Observa-se que ambas acompanham adequadamente a referência, tanto no giro horário quanto no giro anti-horário, mantendo transições suaves e sem oscilações significativas.

Figura 4.2 – Referência e velocidade medida da roda esquerda durante o ensaio 2

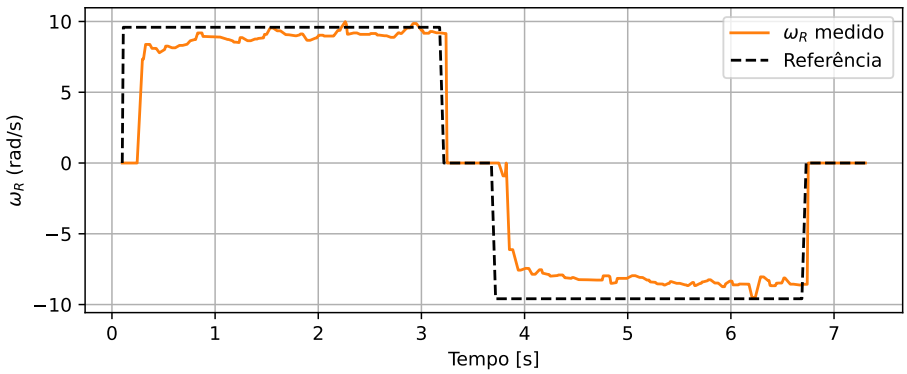


Fonte: Elaborada pelo autor

A roda direita apresenta uma resposta visualmente satisfatória. Já a roda esquerda exibe pequenas oscilações próximas ao instante de parada, o que é comum em baixas velocidades devido ao movimento residual e ao fato de o encoder registrar poucos pulsos por segundo nessa condição. Essa diferença também pode ser atribuída ao atrito ligeiramente menor na roda esquerda, fazendo com que ela demore um pouco mais a estabilizar.

Além disso, este ensaio torna mais perceptível uma limitação do método de medição de velocidade utilizado com o encoder. O cálculo baseado no tempo entre pulsos do encoder

Figura 4.3 – Referência e velocidade medida da roda direita durante o ensaio 2



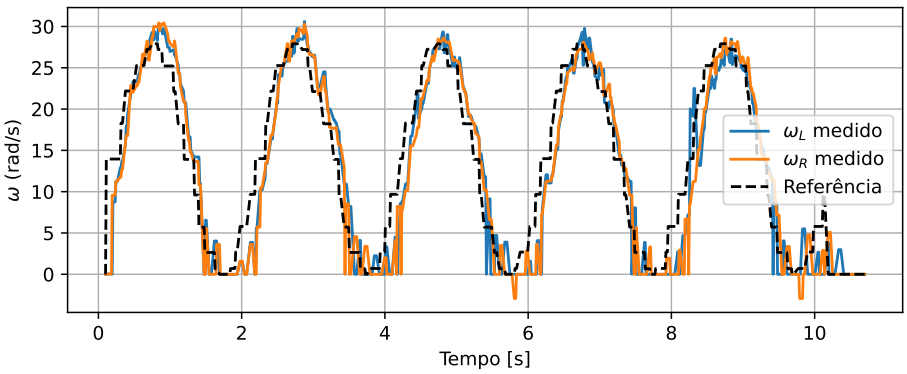
Fonte: Elaborada pelo autor

depende de um mecanismo de *watchdog* que zera a velocidade quando não há pulsos após certo intervalo. Isso faz com que a leitura demore um pouco para retornar a zero após a queda real da velocidade, gerando um atraso visível nos gráficos. Embora esse comportamento não comprometa a estabilidade do controle, ele explica a diferença entre o instante em que o robô para e o momento em que a velocidade medida efetivamente chega a zero.

4.1.3 Ensaio 3: Referência senoidal em velocidade linear

O terceiro ensaio aplica uma referência senoidal de velocidade linear com frequência de 0,5 Hz, enviada igualmente às duas rodas, de velocidade linear, enviada igualmente às duas rodas, com o objetivo de avaliar o desempenho do controlador MRAC em um regime de variação contínua de velocidade. Esse tipo de estímulo permite verificar se o controlador mantém estabilidade e precisão mesmo quando a referência não é constante, exigindo adaptações sucessivas ao longo do tempo.

Figura 4.4 – Referência senoidal e velocidades medidas nas duas rodas durante o ensaio 3



Fonte: Elaborada pelo autor

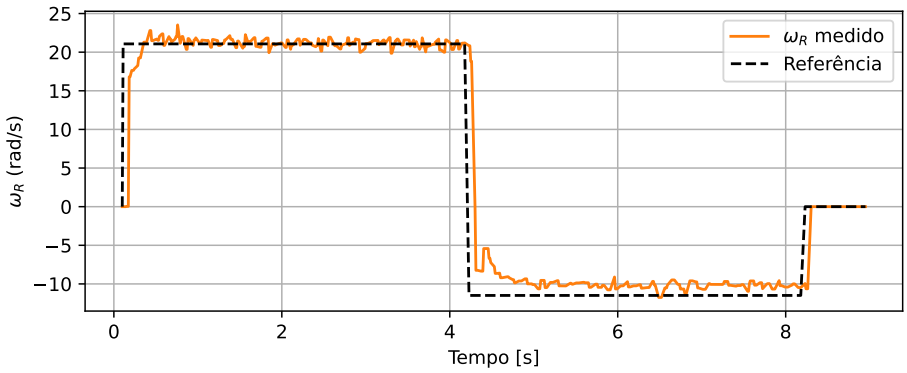
A figura 4.4 apresenta o comportamento do robô diante dessa referência senoidal.

Observa-se que o MRAC acompanha adequadamente a forma da referência, mantendo o movimento durante todo o ensaio. As rodas esquerda e direita apresentam respostas semelhantes, praticamente sobrepostas ao longo de todo o ciclo, o que indica boa simetria na correção da mecânica e correta atuação do controlador nos dois lados do robô. Pequenas diferenças próximas aos picos e vales da senoide são compatíveis com a dinâmica do motor e com o tempo necessário para atingir os extremos da variação de velocidade.

4.1.4 Ensaio 4: Movimento circular com inversão de sentido

O quarto ensaio avaliou o comportamento do controlador MRAC diante de uma trajetória circular com $v \neq 0$, escolhida por representar um movimento comum em ações evasivas ou de posicionamento do robô. A trajetória é composta por duas fases: um movimento circular para a frente e, em seguida, um movimento circular para trás, ambos com a mesma velocidade angular. O objetivo foi verificar se o controlador mantém estabilidade e precisão mesmo quando a velocidade linear muda de sinal enquanto a velocidade angular permanece constante, condição que exige coordenação simultânea das duas rodas.

Figura 4.5 – Referência e velocidade medida da roda direita durante o ensaio 4



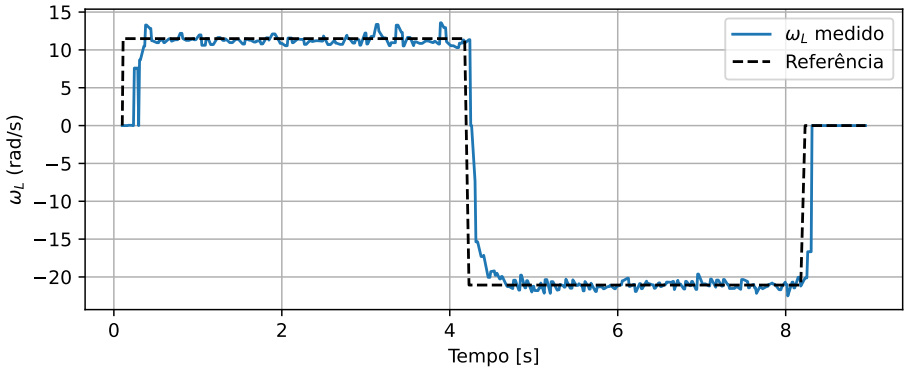
Fonte: Elaborada pelo autor

As figuras 4.5 e 4.6 mostram que o controlador segue adequadamente a referência nas duas fases do movimento. A transição entre avanço e recuo ocorre de maneira suave, sem oscilações significativas, demonstrando que o MRAC mantém estabilidade mesmo sob inversão do sinal da velocidade linear enquanto a velocidade angular permanece constante.

Observa-se também que as rodas apresentam comportamentos complementares, coerentes com a cinemática do movimento circular: a roda direita opera com valores mais elevados por estar no lado externo da curva, enquanto a roda esquerda permanece com velocidades menores. As respostas são consistentes durante todo o período de teste e mantêm boa correspondência com a referência aplicada.

Pequenas diferenças entre as velocidades medidas e as referências, especialmente nos instantes de parada, são compatíveis com o método de estimativa baseado no intervalo entre

Figura 4.6 – Referência e velocidade medida da roda esquerda durante o ensaio 4



Fonte: Elaborada pelo autor

pulsos do encoder, que introduz um pequeno atraso para atingir o valor nulo. Esses efeitos não comprometem a estabilidade do sistema nem a avaliação do controlador.

Análise geral dos ensaios em relação aos objetivos do controlador

Os quatro ensaios apresentados permitem avaliar o desempenho geral do controlador MRAC frente aos objetivos definidos. De maneira consistente, o controlador demonstrou boa capacidade de rastreamento, acompanhando referências constantes e variáveis tanto em movimentos retilíneos quanto em giros puros e trajetórias combinadas. Em todos os casos, o sistema manteve estabilidade, sem oscilações significativas e com erros compatíveis com a dinâmica dos motores.

O MRAC também se mostrou robusto às variações mecânicas entre as rodas e ao ruído inerente ao método de medição por encoder, especialmente em baixas velocidades. Mesmo com atrasos ou pequenas oscilações no sinal medido, o controlador não amplificou erros e manteve o comportamento suave, atendendo ao objetivo de operar de forma segura frente a incertezas.

Além disso, o controlador lidou bem com inversões de sentido e mudanças rápidas de referência, ajustando-se às novas condições sem perda de estabilidade. Isso confirma sua capacidade adaptativa, essencial para o funcionamento do robô em situações típicas de partida e manobras ágeis.

4.2 Controladores de velocidade do robô

Esta seção apresenta e discute o desempenho de três abordagens distintas para o rastreamento das velocidades linear e angular do robô: operação sem controle, controlador PID e controle no espaço de estados. Todas as estratégias foram avaliadas sob a mesma

rotina de movimento, projetada para excitar diferentes dinâmicas do robô ao longo de uma execução de aproximadamente 16 segundos.

Para a avaliação quantitativa das estratégias de controle, foram definidas duas métricas de desempenho calculadas a partir dos dados discretos coletados. Inicialmente, define-se o erro instantâneo de rastreamento para as velocidades linear (e_v) e angular (e_ω) na k-ésima amostra como:

$$e_v[k] = v_{ref}[k] - v_{med}[k] \quad \text{e} \quad e_\omega[k] = \omega_{ref}[k] - \omega_{med}[k] \tag{4.1}$$

A precisão global do sistema foi avaliada através do Erro Quadrático Médio (MSE - *Mean Squared Error*), que penaliza grandes desvios e fornece uma medida da energia do erro. O MSE total foi calculado pela soma das contribuições linear e angular:

$$MSE_{total} = \frac{1}{n}\sum_{k=1}^{n}(e_v[k])^2 + \frac{1}{n}\sum_{k=1}^{n}(e_\omega[k])^2, \tag{4.2}$$

onde n é o número total de amostras.

Para quantificar a suavidade e a estabilidade da resposta, utilizou-se o desvio padrão dos erros de rastreamento (σ). Um valor elevado de σ indica que o erro varia bruscamente em torno de sua média, sintoma típico de instabilidade ou amplificação de ruído. A métrica de suavidade combinada (σ_{total}) é dada por:

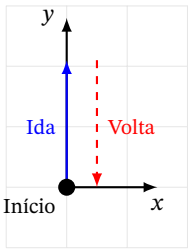
$$\sigma_{total} = \sigma(e_v) + \sigma(e_\omega), \tag{4.3}$$

onde σ representa o desvio padrão amostral do vetor de erros ao longo de todo o experimento.

A trajetória de teste é composta por quatro fases distintas:

Resposta linear (Fase 1): O robô é comandado a deslocar-se longitudinalmente para frente com velocidade linear $v = 0.3$ m/s por 0.7 s, seguido de uma pausa para estabilização, e um retorno à posição original com velocidade inversa ($v = -0.3$ m/s). A estimativa da trajetória pode ser visualizada na figura 4.7.

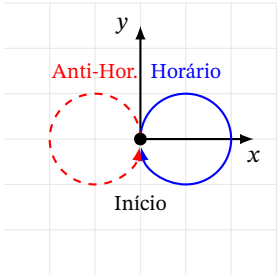
Figura 4.7 – Representação da fase 1: movimento linear



Fonte: Elaborada pelo autor

Trajetória curvilínea (Fase 2): São executados movimentos circulares completos. Primeiramente, uma rotação no sentido horário ($v = 0.25\,\text{m/s}$, $\omega = -2.5\,\text{rad/s}$) e, após uma pausa, uma rotação no sentido anti-horário com os mesmos módulos de velocidade. O resultado é um desenho de um "8", como ilustrado na figura 4.8.

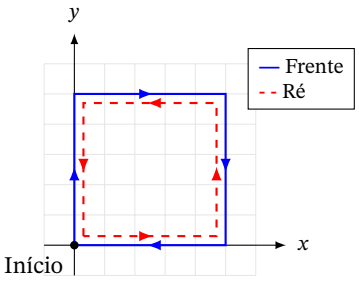
Figura 4.8 – Representação da fase 2: trajetórias circulares



Fonte: Elaborada pelo autor

Trajetória quadrada (Fases 3 e 4): O robô descreve uma trajetória em formato de quadrado, alternando segmentos de reta ($v = 0.3\,\text{m/s}$) com rotações em torno do próprio eixo (giros de 90° com $\omega = \pm 5.0\,\text{rad/s}$). Esta etapa é executada primeiramente com deslocamento frontal e, na sequência, de ré, testando a simetria do controle. A figura 4.9 representa uma estimativa da trajetória.

Figura 4.9 – Representação da fase 3 e 4: trajetória quadrada



Fonte: Elaborada pelo autor

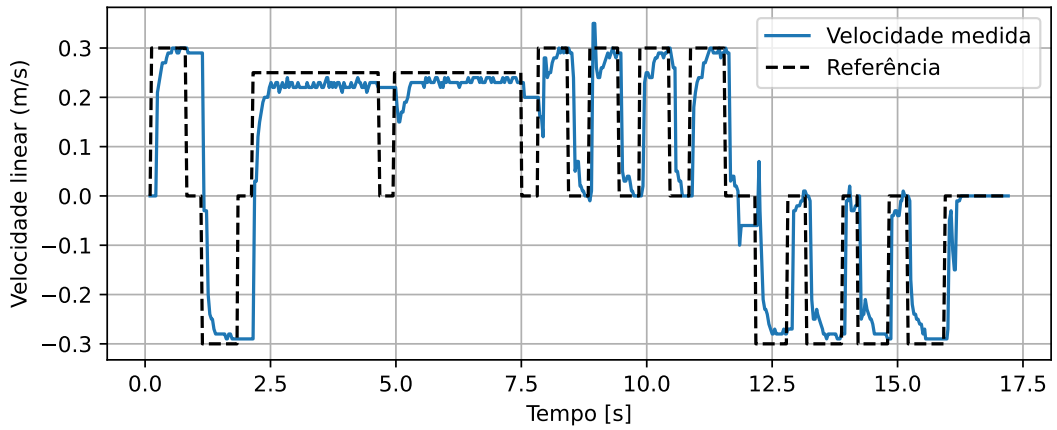
A aplicação dessa rotina padronizada permite uma comparação direta entre estabilidade, precisão e suavidade da resposta de cada estratégia. A análise apresentada a seguir estabelece as bases para a escolha do método de controle mais adequado às necessidades do sistema.

4.2.1 Caso 1: Sem controle (malha aberta)

No primeiro caso, o robô executou a rotina de movimento em malha aberta, utilizando apenas a transformação cinemática inversa (Equação 2.5) para converter as velocidades de

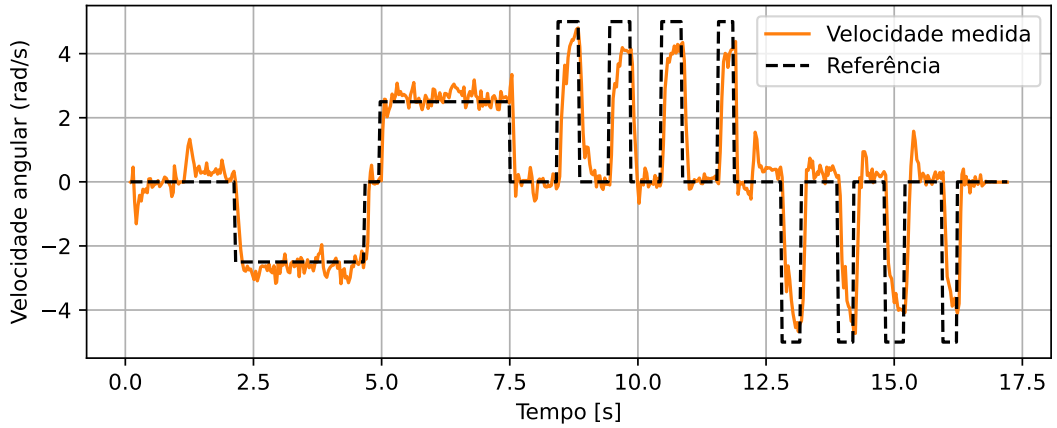
referência em comandos para os motores. As figuras 4.10 e 4.11 apresentam o desempenho obtido para as velocidades linear e angular.

Figura 4.10 – Referência e velocidade linear medida do robô sem controle de movimento



Fonte: Elaborada pelo autor

Figura 4.11 – Referência e velocidade angular medida do robô sem controle de movimento



Fonte: Elaborada pelo autor

A análise da velocidade linear (figura 4.10) demonstra que o modelo cinemático utilizado é representativo do sistema, visto que a velocidade medida acompanha a referência em regime permanente. Observa-se, contudo, a influência da dinâmica natural dos motores: nas transições abruptas de velocidade, há um atraso físico na resposta devido à inércia do sistema, que não é compensada ativamente por nenhum controlador.

O comportamento da velocidade angular (figura 4.11) corrobora essa característica. A ausência de realimentação impede a correção de erros instantâneos, resultando em um perfil de movimento que depende exclusivamente da calibração prévia dos atuadores. Nota-se que, apesar dos atrasos dinâmicos nas inversões de sentido, o sinal medido apresenta baixa

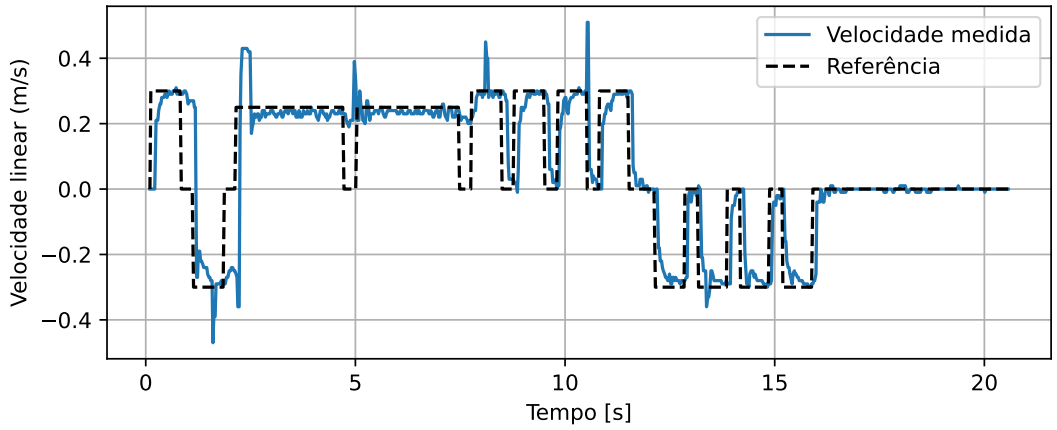
variabilidade em torno da trajetória.

Esse comportamento explica o desempenho quantitativo superior observado nesse caso, refletido pelo menor MSE e pelo menor desvio padrão. Como o sistema opera em malha aberta, não ocorre realimentação do ruído proveniente dos encoders para o sinal de atuação, o que reduz a variabilidade da resposta. Ainda assim, apesar da suavidade resultante, a ausência de correção por realimentação torna o robô suscetível a perturbações externas e ao acúmulo de erros ao longo do tempo.

4.2.2 Caso 2: Controlador PID na velocidade angular

No segundo cenário, o controle foi aplicado exclusivamente à dinâmica rotacional do robô. A velocidade angular ω foi estimada através de uma IMU, cujos dados brutos do giroscópio e acelerômetro foram fundidos por um filtro complementar para mitigar ruídos de alta frequência e a deriva (*drift*). O erro calculado ($e_\omega = \omega_{ref} - \omega_{IMU}$) alimenta um controlador PID, cuja saída ajusta a tensão dos motores de forma diferencial, enquanto a velocidade linear permanece em malha aberta.

Figura 4.12 – Referência e velocidade linear medida do robô com controle PID

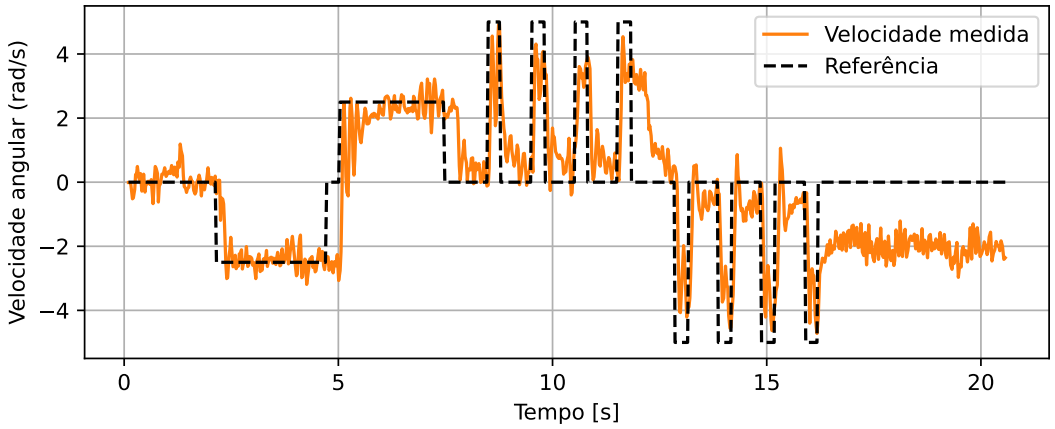


Fonte: Elaborada pelo autor

A análise da velocidade linear (figura 4.12) revela o impacto do acoplamento dinâmico entre as variáveis. Embora o PID atue apenas na orientação, as correções aplicadas às rodas para girar o robô afetam a velocidade linear resultante. Nota-se que a oscilação do laço de controle angular se propaga para a velocidade linear, introduzindo perturbações em uma variável que não está sob controle direto.

O comportamento da velocidade angular (figura 4.13) evidencia limitações tanto no hardware quanto no ajuste do controlador. Observa-se um erro de regime permanente significativo nos instantes de repouso, onde o sensor indica aproximadamente $-2\,\text{rad/s}$. Esse fenômeno é atribuído à combinação de um viés (*bias*) do giroscópio com uma sintonia

Figura 4.13 – Referência e velocidade angular medida do robô com controle PID



Fonte: Elaborada pelo autor

inadequada do termo integral (K_i). O ganho integral utilizado mostrou-se insuficiente para acumular a ação de controle necessária para vencer a zona morta dos motores e zerar o erro estacionário proveniente da leitura deslocada do sensor.

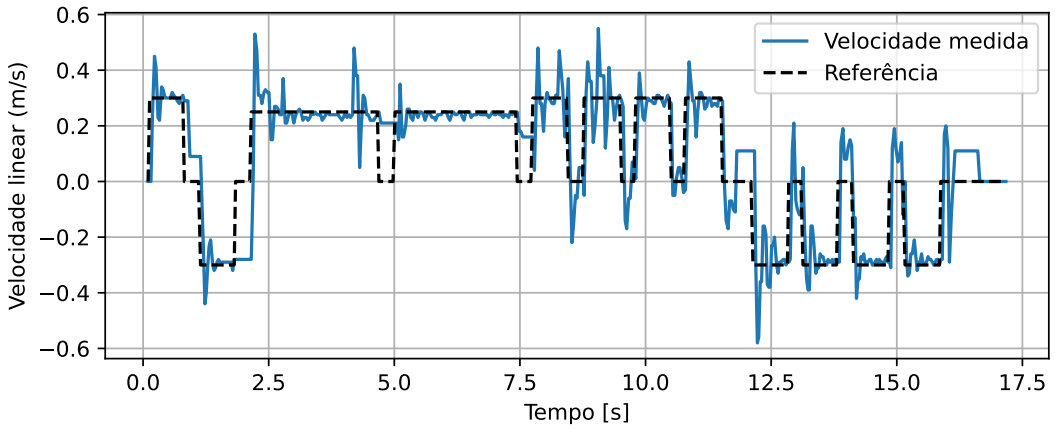
Adicionalmente, as fortes oscilações durante o movimento sugerem que os ganhos proporcional (K_p) e derivativo (K_d) podem estar excessivamente agressivos para o nível de ruído da IMU. O controlador reage às vibrações estruturais captadas pelo sensor, gerando comandos bruscos que realimentam a instabilidade mecânica. Esse ciclo resulta em um comportamento com excesso de atividade de controle, indicando que uma sintonia mais conservadora dos ganhos, priorizando a filtragem em detrimento da velocidade de resposta, poderia ter mitigado o ruído observado e reduzido o MSE.

4.2.3 Caso 3: Controle no espaço de estados

No terceiro caso, o robô executou a mesma rotina de movimento utilizando um controlador baseado em espaço de estados, que emprega a realimentação dos estados do sistema, v e ω, para calcular as velocidades a serem enviadas para o controle adaptativo. As figuras 4.14 e 4.15 apresentam o desempenho obtido.

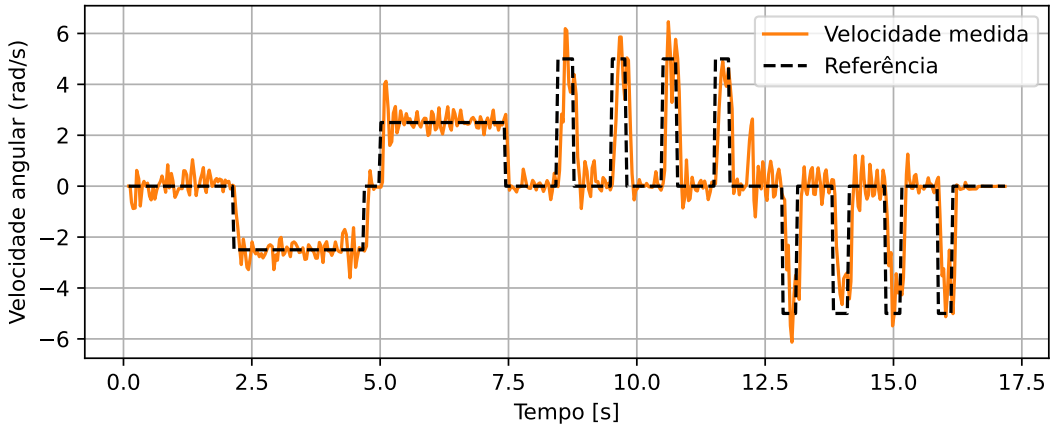
A análise da velocidade linear (figura 4.14) demonstra a capacidade do controlador de lidar com o sistema multivariável, tratando o acoplamento dinâmico de forma nativa. Nota-se que a resposta é agressiva pela existência dos picos transientes, os *overshoots*, expressivos nas mudanças de referência. Por exemplo, na transição em $t \approx 2{,}5s$, a velocidade linear atinge um pico de 0,53 m/s para uma referência de 0,25m/s, caracterizando um *overshoot* de aproximadamente 112%. Esse comportamento indica que a matriz de ganhos K impõe uma dinâmica de correção muito rápida, gerando esforços de controle elevados que superam momentaneamente a inércia do robô.

Figura 4.14 – Referência e velocidade linear medida do robô com controle por realimentação de estados



Fonte: Elaborada pelo autor

Figura 4.15 – Referência e velocidade angular medida do robô com controle por realimentação de estados



Fonte: Elaborada pelo autor

O gráfico de velocidade angular (figura 4.15) apresenta uma melhoria em relação ao caso PID: a eliminação do erro de regime permanente no final do ensaio. Após $t = 16s$, a velocidade medida converge corretamente para zero, sem o erro estacionário observado anteriormente. Isso sugere que a estrutura de realimentação de estados foi mais robusta às não-linearidades do sistema nessa faixa de operação.

O erro quadrático médio (MSE) se justifica por esse comportamento oscilatório de alta frequência do sinal. Embora o controlador garanta um rastreamento fiel da trajetória sem desvios constantes, ele o faz com alta variância e correções bruscas. O controlador no espaço de estados priorizou a rapidez de resposta, resultando em um sistema preciso, mas energeticamente custoso devido às variações abruptas nos atuadores.

4.2.4 Comparação quantitativa

A Tabela 4.1 sumariza o desempenho das três estratégias. Para eliminar a subjetividade da análise visual, foram consideradas duas métricas complementares: o Erro Quadrático Médio (MSE), que quantifica a precisão global de rastreamento, e a soma dos Desvios Padrão dos erros (σ_{total}), que serve como indicador direto da suavidade e estabilidade do comportamento do robô.

Tabela 4.1 – Comparação de precisão (MSE) e suavidade (σ)

Estratégia	**MSE Total**	**Suavidade (σ_{total})**
Malha Aberta (Sem controle)	1.640	1.396
Espaço de Estados	1.677	1.405
PID	2.591	1.701

Fonte: Elaborada pelo autor

A análise quantitativa revela um *trade-off* entre a capacidade de correção e a amplificação de ruído:

1. **Referência de Suavidade (Malha Aberta):** O caso sem controle estabelece a linha de base para a suavidade do sistema, com $\sigma_{total} \approx 1.396$. Como não há realimentação, o sistema atua como um filtro passa-baixas natural, ignorando o ruído dos sensores. O baixo MSE (1.640) indica que, em um ambiente controlado e por curtos períodos, o modelo cinemático é fidedigno, mas a estratégia é incapaz de rejeitar distúrbios externos.
2. **Degradação pelo PID:** O controlador PID apresentou o pior desempenho em ambas as métricas. O aumento substancial na métrica de oscilação para $\sigma_{total} \approx 1.701$ (um incremento de aproximadamente 22% em relação à base) confirma que a ação do controlador, particularmente o termo derivativo, amplificou o ruído de quantização dos encoders. Isso resultou em um esforço de controle oscilatório que prejudicou a precisão global, elevando o MSE para 2.591.
3. **Eficiência do Espaço de Estados:** Esta estratégia apresentou o resultado mais equilibrado. O controlador foi capaz de manter a suavidade do movimento praticamente inalterada em relação à malha aberta ($\sigma_{total} \approx 1.405$, uma diferença inferior a 1%), provando que as correções aplicadas foram suaves e proporcionais. O MSE de 1.677, muito próximo ao caso ideal sem controle, indica que o sistema ganhou robustez sem pagar o preço da instabilidade ou do *chattering* observado no PID.

Em conclusão, os dados validam o controle no espaço de estados como a estratégia mais adequada para a aplicação no robô VSSS. Diferente do PID, que se mostrou agressivo diante do ruído sensorial, a realimentação de estados conseguiu aliar a precisão necessária para competições à suavidade de atuação essencial para a preservação dos atuadores e estabilidade

da odometria.

4.3 Simulação de posição

Além dos testes baseados na coleta direta das velocidades linear e angular via comunicação sem fio, realizou-se também uma análise da trajetória espacial do robô a partir de uma câmera posicionada verticalmente acima do campo de jogo. O objetivo foi quantificar o erro de seguimento de trajetória durante a execução de curvas, comparando a trajetória ideal simulada com a trajetória efetivamente medida.

Para esse experimento, foram enviados comandos constantes de velocidade linear e angular, definidos como $v = 0{,}22$ m/s e $\omega = 1{,}5$ rad/s, alternando-se apenas o sinal de ω de modo a produzir duas curvas circulares consecutivas, uma no sentido horário e outra no sentido anti-horário, formando uma trajetória aproximada de um “8”.

Os pontos da trajetória foram obtidos por meio de um algoritmo de visão computacional que rastreia o LED presente na placa do robô durante toda a execução do movimento. A figura 4.16 apresenta um *frame* utilizado na aquisição.



Figura 4.16 – *Frame* capturado do vídeo utilizado para aquisição dos dados

A partir dos comandos $v(t)$ e $\omega(t)$, gerou-se uma trajetória ideal utilizando o modelo cinemático do robô diferencial. A trajetória simulada foi então comparada com os dados capturados pela câmera, e o erro quadrático médio (MSE) foi calculado para quantificar a precisão de seguimento para cada método avaliado: malha aberta, controle PID e controle por realimentação de estados. Os resultados estão apresentados na Tabela 4.2.

Tabela 4.2 – Erro quadrático médio (MSE) entre trajetória simulada e trajetória capturada pela câmera

Controlador	**MSE**
Malha aberta (Sem controle)	0,20127
Espaço de estados	0,03780
PID	0,13855

A figura 4.17 apresenta o resultado do robô operando sem compensação. O erro quadrático médio obtido foi de 0,20127 m, indicando um afastamento significativo da trajetória ideal ao longo do movimento.

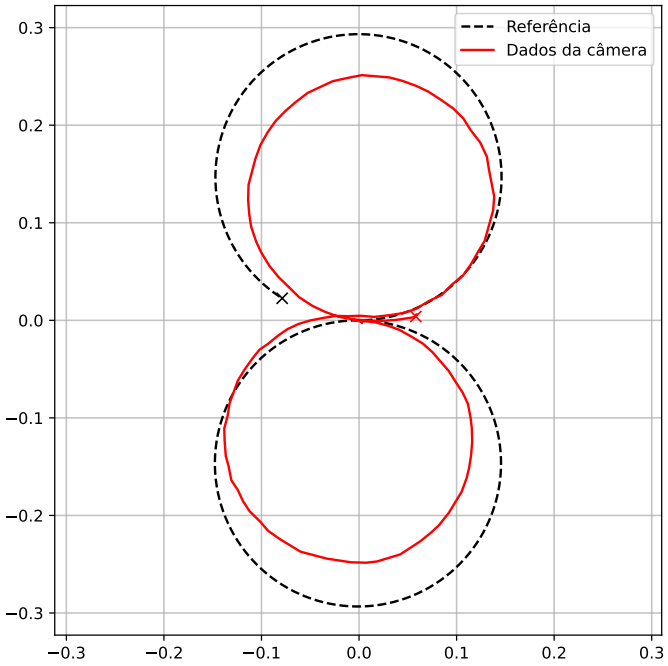


Figura 4.17 – Comparação entre trajetória real e simulada usando o método sem controle

Para o controlador PID, mostrado na figura 4.18, o erro reduziu-se para 0,13855 m, representando uma melhora em relação ao método em malha aberta, embora ainda exista deriva acumulada entre a trajetória real e a ideal.

O controlador por espaço de estados apresentou o menor erro entre os métodos avaliados. Como mostrado na Tabela 4.2, o erro quadrático médio foi de apenas 0,03780 m, correspondendo a uma redução de 81,2% em relação à malha aberta e de 72,7% em relação ao PID. Esse valor corresponde a um erro médio inferior a 4 cm ao longo de toda a trajetória capturada.

A análise quantitativa demonstra que o controlador por espaço de estados apresentou o melhor desempenho no seguimento da trajetória planejada, seguido pelo controlador PID, enquanto o método sem compensação apresentou o maior erro absoluto. Esses resultados complementam as análises baseadas em velocidade, reforçando que a realimentação de

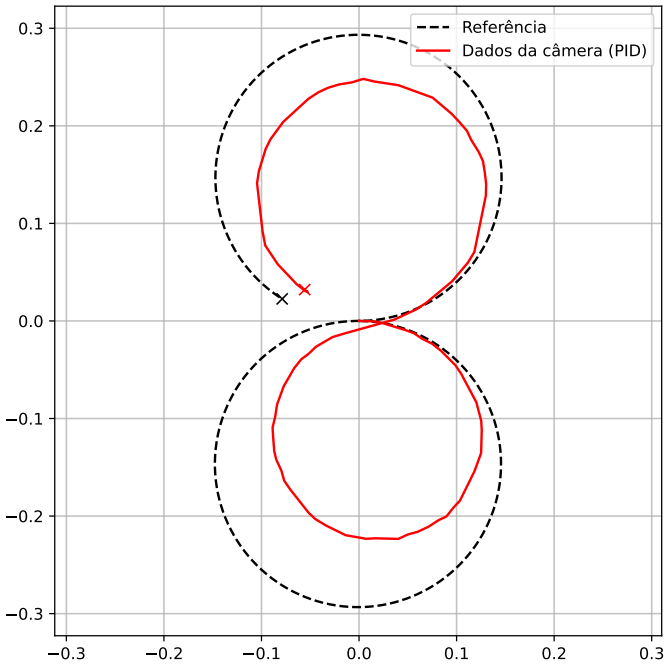


Figura 4.18 – Comparação entre trajetória real e simulada usando o controlador PID

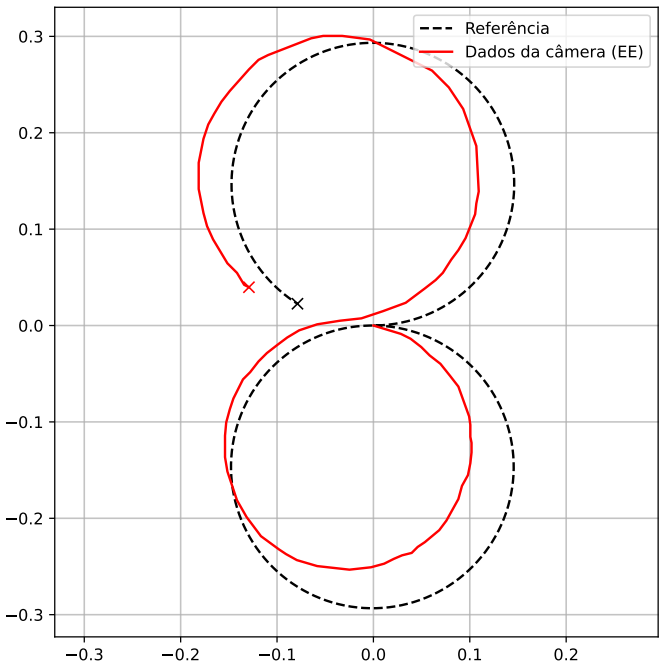


Figura 4.19 – Comparação entre trajetória real e simulada usando o controlador por espaço de estados

estados é a estratégia que proporciona o rastreamento mais preciso da trajetória espacial do robô.

Conclusão do capítulo

Ao longo deste capítulo, verificou-se que o controlador MRAC implementado foi capaz de rastrear diferentes referências de velocidade nas rodas, mantendo estabilidade mesmo em mudanças bruscas de sinal e em trajetórias mais complexas. Na camada de controle de velocidade do robô, os resultados mostraram que a malha aberta oferece movimento suave, porém sem robustez; o controlador PID é mais sensível ao ruído e à calibração da IMU; e o controle no espaço de estados apresentou o melhor compromisso entre precisão e suavidade, destacando-se tanto nas métricas de erro quanto nos testes de trajetória com visão computacional. Assim, os ensaios confirmam que a arquitetura de controle proposta é adequada para o robô desenvolvido, atendendo aos objetivos de garantir rastreamento consistente de velocidades linear e angular em operação real.

5 Conclusões

Introdução do capítulo

Este capítulo apresenta as principais conclusões do trabalho, avaliando o cumprimento dos objetivos propostos e sintetizando os resultados obtidos ao longo do desenvolvimento do robô móvel da categoria VSSS. São discutidos o desempenho do sistema embarcado, a eficácia das estratégias de controle implementadas e a coerência entre a modelagem adotada e o comportamento experimental observado. Ao final, são indicadas perspectivas de continuidade que podem aprimorar o projeto em versões futuras.

5.1 Considerações gerais sobre o trabalho

O desenvolvimento do robô integrou de forma coerente todas as etapas necessárias para a construção de um sistema embarcado funcional na categoria VSSS, abrangendo eletrônica, modelagem, identificação, programação e controle. A placa de circuito impresso projetada, os módulos de sensoriamento e atuação e o firmware desenvolvido mostraram-se adequados às restrições de tamanho, consumo e desempenho estabelecidas pela competição.

A integração entre hardware e software permitiu a execução confiável dos ensaios experimentais, com o robô recebendo comandos de velocidade, processando-os em tempo real e registrando dados consistentes para análise. Os testes com diferentes métodos de controle evidenciaram a capacidade do sistema de seguir referências lineares e angulares mesmo em condições dinâmicas, como inversões de sentido ou trajetórias variáveis no tempo.

5.2 Avaliação dos objetivos alcançados

Os objetivos estabelecidos no seção 1.3 foram atingidos de forma satisfatória. Inicialmente, foi projetada e construída uma placa de circuito impresso compacta, integrando microcontrolador, drivers de motor, sensores e interfaces auxiliares. Esse desenvolvimento atendeu ao objetivo de consolidar um sistema embarcado completo para o robô.

A modelagem cinemática e dinâmica, aliada à identificação experimental dos parâmetros do sistema, permitiu a construção de modelos adequados à síntese dos controladores. Em seguida, foram implementadas e testadas três abordagens de controle: operação em malha aberta, controlador PID e controle baseado em espaço de estados. Todas foram avaliadas sob a mesma rotina de movimento, garantindo uma comparação justa entre elas.

Os resultados apresentados no capítulo 4 demonstram que o comportamento experimental do robô é coerente com o modelo adotado e que os controladores implementados alcançam rastreamento consistente das referências impostas. O controlador por realimentação de estados destacou-se como o que apresentou o melhor compromisso entre precisão e suavidade, enquanto o controlador PID mostrou maior sensibilidade ao ruído dos sensores. Já a operação sem controle, embora suave, apresentou erros mais perceptíveis em transições rápidas. Dessa forma, os objetivos de análise, implementação e comparação das estratégias de controle foram cumpridos.

5.3 Perspectivas e trabalhos futuros

O trabalho abre possibilidades claras para aprimoramentos. Um caminho promissor é a substituição do método atual de estimativa de velocidade por técnicas de fusão sensorial envolvendo dados de *encoder* e IMU, de modo a reduzir atrasos e melhorar a robustez das medições. Esse avanço pode ser complementado pelo uso de algoritmos de detecção de bordas ascendentes e descendentes nos encoders, explorando máxima resolução.

Outra perspectiva relevante é a reestruturação do firmware para o framework *ESP-IDF*, utilizando o sistema operacional *FreeRTOS*. A adoção de tarefas com prioridades distintas pode mitigar o *jitter* da malha de controle e garantir maior determinismo na execução do sistema.

No âmbito mecânico, melhorias na rigidez do chassi e no acoplamento dos motores podem reduzir vibrações estruturais, aumentando a qualidade das medições e a estabilidade dinâmica do robô. Também se abrem oportunidades para explorar novos controladores, técnicas de otimização de trajetórias e integração com sistemas de visão para navegação plenamente autônoma.

Referências

ALVES, T. G.; LAGES, W. F.; HENRIQUES, R. V. Parametric identification and controller design for a differential-drive mobile robot**this work has been funded by the capes procad research project on assistive technologies. **IFAC-PapersOnLine**, v. 51, n. 15, p. 437–442, 2018. ISSN 2405-8963. 18th IFAC Symposium on System Identification SYSID 2018. Disponível em: https://www.sciencedirect.com/science/article/pii/S2405896318318512. Citado na p. 24.

ARCHELA, A. **Técnicas de modelagem, estimação de parâmetros e controle embarcado aplicados em veículos elétricos com trações traseira independentes**. Dissertação (Mestrado) — Universidade Estadual de Londrina, Programa de Pós-graduação em Engenharia Elétrica, 2019. Tese de Mestrado. Disponível em: https://pos.uel.br/meel/wp-content/uploads/2024/04/2019-Alex_Archela.pdf. Citado nas pp. 51 e 52.

ÅSTRÖM, K.; WITTENMARK, B. **Adaptive Control**. Dover Publications, 2008. (Dover Books on Electrical Engineering). ISBN 9780486462783. Disponível em: https://books.google.com.br/books?id=L0m_CR-IK24C. Citado nas pp. 29, 43 e 44.

ÅSTRÖM, K.; WITTENMARK, B. **Computer-Controlled Systems: Theory and Design, Third Edition**. Dover Publications, 2013. (Dover Books on Electrical Engineering). ISBN 9780486284040. Disponível em: https://books.google.com.br/books?id=TynEAgAAQBAJ. Citado na p. 49.

BORGES, G. A. **Um Sistema Optico de Reconhecimento de Trajetórias para Veículos Automáticos**. Dissertação (Dissertação de Mestrado) — Universidade Federal da Paraíba, Campina Grande, Paraíba, Brasil, 1998. Disponível em: https://www.ene.unb.br/gaborges/arquivos/msc.geovany.borges.1998.pdf. Citado nas pp. 27 e 28.

DYNAMICS, B. **Spot robot used for inspection in BMW facilities**. 2024. https://humanoidroboticstechnology.com/wp-content/uploads/2024/05/Spot-BMW.jpg. Acesso em: 1 dez. 2025. Citado na p. 17.

Espressif Systems. **ESP32-WROOM-32E Datasheet**. [*S.l.*], 2024. Acesso em: 20 Set. 2024. Disponível em: https://documentation.espressif.com/esp32-wroom-32e_esp32-wroom-32ue_datasheet_en.pdf. Citado na p. 37.

FRANKLIN, G.; POWELL, J. D.; EMAMI-NAEINI, A. **Feedback Control of Dynamic Systems**. [*S.l.*]: Pearson, 2015. Citado na p. 49.

GIGAZINE. **RoboCup 2017 Nagoya World Tournament 'Soccer Small Size League'**. 2017. Disponível em: https://gigazine.net/gsc_news/en/

20170727-robocup-soccer-small-size-league-rc2017. Acesso em: 20 nov. 2025. Citado na p. 20.

JOHNSON, H.; GRAHAM, M. **High-speed Digital Design: A Handbook of Black Magic**. Prentice Hall, 1993. (Prentice Hall Modern Semiconductor Design). ISBN 9780133957242. Disponível em: https://books.google.com.br/books?id=H5SsQgAACAAJ. Citado na p. 39.

KIM, J.; KIM, D.; KIM, Y.; SEOW, K. **Soccer Robotics**. Springer Berlin Heidelberg, 2004. (Springer Tracts in Advanced Robotics). ISBN 9783540218593. Disponível em: https://books.google.com.br/books?id=C6WeRYvu6dAC. Citado na p. 22.

NARDI, D.; ROBERTS, J.; VELOSO, M.; FLETCHER, L. Robotics competitions and challenges. *In*: ____. **Springer Handbook of Robotics**. [*S.l.*]: Springer International Publishing, 2016. ISBN 978-3-319-32552-1. Citado na p. 18.

NGUYEN, N. **Model Reference Adaptive Control - a Primer**. [*S.l.*]: Springer International Publishing, 2018. ISBN 978-3-319-56392-3. Citado nas pp. 29, 44 e 45.

OGATA, K. **Discrete-time Control Systems**. Prentice Hall, 1995. ISBN 9780130342812. Disponível em: https://books.google.com.br/books?id=owQqAQAAMAAJ. Citado nas pp. 31 e 33.

OGATA, K. **Modern Control Engineering**. 5th. ed. Upper Saddle River, NJ: Prentice Hall, 2010. Citado nas pp. 30 e 33.

PEDONE, R. **System identification for mobile robots: a comparative study of gray-box and black-box approaches**. Dissertação (Mestrado) — Politecnico di Milano, 2023. Citado nas pp. 24, 25, 50 e 51.

PORTO, L. H. S. **Controle de Movimento de um Robô Móvel Não-Holonômico com Tração Diferencial**. Dissertação (Trabalho de Graduação) — Universidade de Brasília, Brasília, 2017. Citado na p. 19.

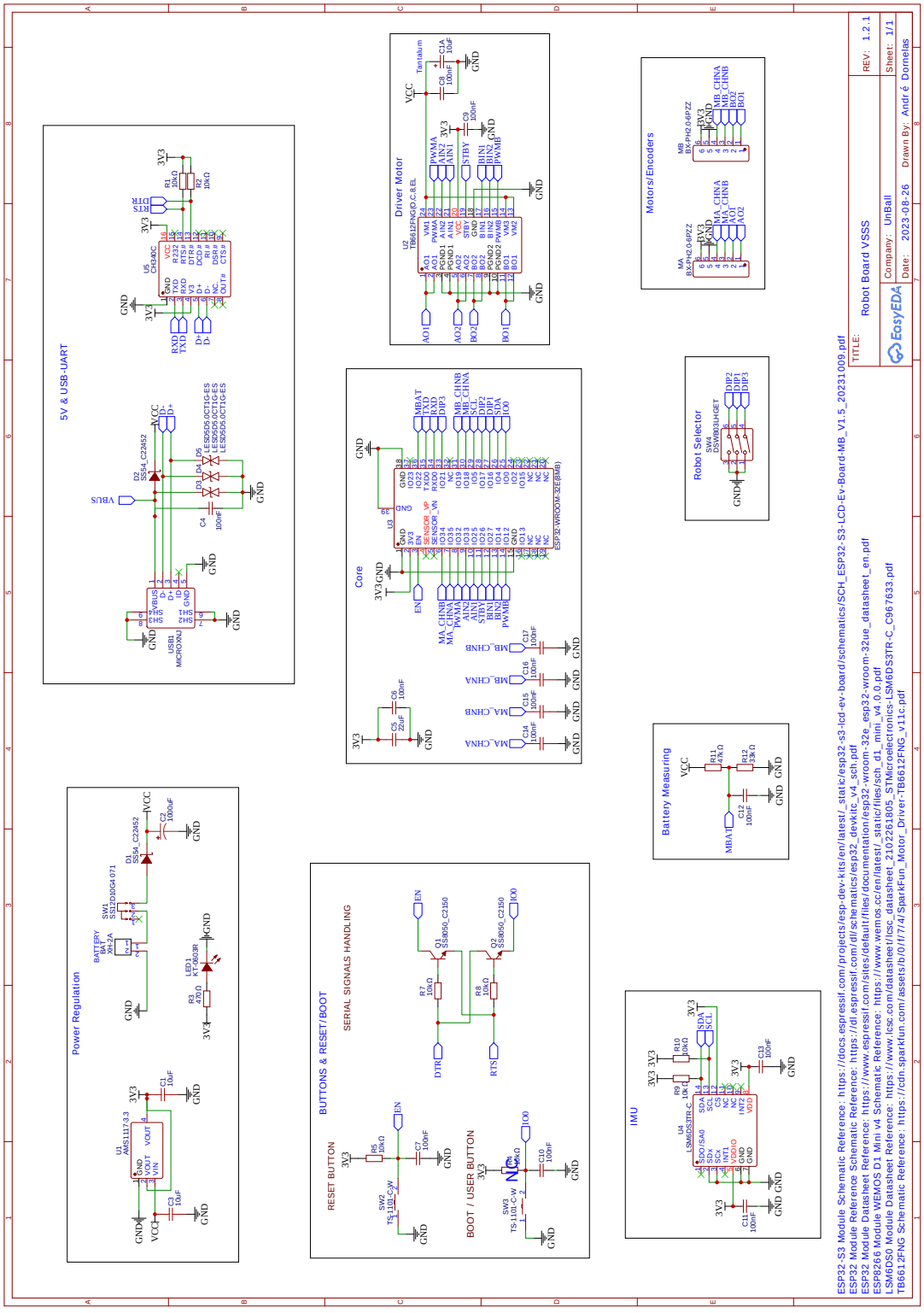
THRUN, S.; BURGARD, W.; FOX, D. **Probabilistic Robotics**. [*S.l.*]: MIT Press, 2005. Citado na p. 49.

VARASCHIM, F. H.; SOARES, J. W. M.; POSSOLI, F. A. de A.; PIVOVAR, L. E.; BREGANON, R.; ALVES, U. N. L. T. Utilização de um filtro complementar junto ao sensor mpu 6050. **Revista Mundi Engenharia, Tecnologia e Gestão**, v. 7, n. 7, p. 459–1–459–12, 2022. Disponível em: https://periodicos.ifpr.edu.br/index.php/MundiETG/article/view/2341/992. Citado na p. 49.

VSSS Technical Committee. **Regras Oficiais da Categoria Very Small Size Soccer (VSSS) – Série A**. [*S.l.*], 2023. Versão oficial disponibilizada em PDF. Disponível em: https://lars-larc2024.ucsp.edu.pe/rules/regrasVSS23.pdf. Citado na p. 35.

Apêndices

Apêndice A – Esquemático



Apêndice B – Códigos de programação

B.1 Método de identificação de parâmetros de um sistema de primeira ordem

Código B.1 – Código utilizado na identificação experimental da planta

```
% Identificação de um sistema de primeira ordem a partir de um arquivo .csv
% contendo tempo, comando PWM e velocidade medida (rad/s).
% O método utiliza o ponto de 63,21% e um refinamento via lsqcurvefit.

%% === 1. Carregar dados do arquivo ===
data = readtable('logs/motor_log.csv');
t = data.time;      % tempo [s]
u = data.pwm;       % entrada PWM
w = data.w;         % velocidade medida [rad/s]

%% === 2. Recortar dados a partir do instante do degrau ===
idx_step = find(u > 0, 1, 'first');
t0 = t(idx_step);

t = t(idx_step:end) - t0;
w = w(idx_step:end);
u = u(idx_step:end);

%% === 3. Valor do degrau aplicado (PWM = 212) ===
u_step = 212;

%% === 4. Estimativa inicial de K e tau (método dos 63,21%) ===
w_inf = mean(w(end-20:end));        % valor em regime
target = 0.6321 * w_inf;            % 63,21% do regime
idx_tau = find(w >= target, 1);

tau_init = t(idx_tau);              % constante de tempo inicial
K_init = w_inf / u_step;            % ganho inicial (rad/s por unidade PWM)

%% === 5. Normalizar saída para fitting com degrau unitário ===
w_norm = w / u_step;

%% === 6. Ajuste fino com lsqcurvefit ===
% Modelo de primeira ordem: w(t) = K*(1 - exp(-t/tau))
w_model = @(p, t) p(1) * (1 - exp(-t / p(2)));  % p(1)=K, p(2)=tau

p0 = [K_init, tau_init];    % estimativas iniciais
opts = optimoptions('lsqcurvefit', 'Display', 'off');

p_fit = lsqcurvefit(w_model, p0, t, w_norm, [], [], opts);

```

```
K_fit  = p_fit(1);
tau_fit = p_fit(2);

%% === 7. Modelo final identificador ===
s = tf('s');
G_pwm = K_fit / (tau_fit * s + 1)
```


UnB