



Universidade Federal de Ouro Preto
Escola de Minas
Engenharia de Controle e Automação



Laryssa Aparecida Gomes Fagundes

Mapeamento de Trajetória com Robô Seguidor de Linha Utilizando Sensores de Velocidade, Giroscópios e Acelerômetros

Ouro Preto
2026

Laryssa Aparecida Gomes Fagundes

Mapeamento de Trajetória com Robô Seguidor de Linha Utilizando Sensores de Velocidade, Giroscópios e Acelerômetros

Trabalho apresentado ao Colegiado do Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como parte dos requisitos para a obtenção do Grau de Engenheiro(a) de Controle e Automação.

Orientador: Prof. Dr. Alan Kardek Rêgo Segundo

Coorientador: Prof. Dr. Filipe Augusto Santos Rocha

Ouro Preto
2026



MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
ESCOLA DE MINAS
DEPARTAMENTO DE ENGENHARIA CONTROLE E
AUTOMACAO



FOLHA DE APROVAÇÃO

Laryssa Aparecida Gomes Fagundes

Mapeamento de Trajetória com Robô Seguidor de Linha Utilizando Sensores de Velocidade, Giroscópios e Acelerômetros

Monografia apresentada ao Curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Engenheiro de Controle e Automação

Aprovada em 06 de março de 2026

Membros da banca

Dr. Alan Kardek Rêgo Segundo - Orientador (Universidade Federal de Ouro Preto)
Dr. Filipe Augusto Santos Rocha - Coorientador (Universidade Federal de São João Del Rey)
Dra. Sílvia Grasiella Moreira Almeida - Convidada (Instituto Federal de Minas Gerais)
Dr. André Luis Barroso Almeida - Convidado (Instituto Federal de Minas Gerais)

Alan Kardek Rêgo Segundo, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 14/03/2026



Documento assinado eletronicamente por **Alan Kardek Rego Segundo, PROFESSOR DE MAGISTERIO SUPERIOR**, em 14/03/2026, às 11:32, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1070598** e o código CRC **9644576E**.

Referência: Caso responda este documento, indicar expressamente o Processo nº 23109.010523/2025-18

SEI nº 1070598

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: 3135591533 - www.ufop.br

Agradecimentos

Gostaria de começar agradecendo à minha mãe, que, sob o sol, caminhou por mim para que eu pudesse seguir pela sombra. Foi o seu cuidado, sua força silenciosa e seu amor inabalável que tornaram possível cada passo até aqui.

Aos meus irmãos, que de formas inimagináveis me impulsionaram a continuar e a acreditar. Cada gesto, cada palavra e cada presença fizeram diferença nessa jornada.

Ao meu tio Ronaldo e à minha tia Maria Inês, que ofereceram apoio constante e incentivo sincero. Sou profundamente grata por terem sido porto seguro quando precisei.

Ao meu namorado, Lucas, por estar presente em todas as etapas, sua parceria tornou o caminho mais leve e significativo.

Aos meus amigos, que compreenderam ausências, celebraram conquistas e, mesmo nos dias mais difíceis, foram riso, acolhimento e força.

À Escola de Minas da UFOP, pela formação, ensinamentos e oportunidades que contribuíram de maneira essencial para minha trajetória acadêmica e profissional.

Aos meus orientadores, que foram excepcionais na condução deste trabalho, compartilhando não apenas conhecimento, mas também incentivo e dedicação. Este trabalho carrega muito da sabedoria de vocês.

À Equipe FIRE do IFMG, que me apresentou a robótica e gentilmente cedeu a pista para testes.

Ao Laboratório IFABLab - Ouro Preto, especialmente ao Guilherme e à Mika, que contribuíram com a confecção das peças e com ideias tão criativas quanto ousadas, tornando este projeto possível na prática.

Este trabalho é resultado de muitas mãos. A todos vocês, minha gratidão.

A educação é o poder das mulheres.

— Malala Yousafzai.

Resumo

A estimativa precisa de localização representa um dos principais desafios na robótica móvel, especialmente em ambientes internos, onde sistemas de posicionamento global não estão disponíveis. Métodos de navegação relativa, como a odometria e os Sistemas de Navegação Inercial (INS), são amplamente utilizados, porém apresentam limitações decorrentes do acúmulo progressivo de erros ao longo do tempo. Nesse contexto, este trabalho tem como objetivo investigar e implementar métodos de reconstrução de trajetória em um robô seguidor de linha, utilizando dados de sensores de velocidade, giroscópios e acelerômetros. A estimativa da trajetória foi processada e analisada sob três abordagens distintas: odometria, INS e a combinação de ambas por meio de um Filtro de Kalman Estendido (EKF). Os resultados experimentais demonstraram que a odometria isolada apresenta deslocamento lateral e rotação do eixo principal em decorrência de erros angulares cumulativos. Já a INS pura diverge significativamente da trajetória real devido ao erro crescente associado à dupla integração de ruídos. Em contrapartida, a combinação das técnicas via EKF apresentou desempenho superior, reduzindo as deficiências individuais de cada método e proporcionando uma representação geométrica mais consistente com o percurso efetivamente realizado.

Palavras-chaves: Robótica Móvel. Odometria. Navegação Inercial. Filtro de Kalman Estendido. Mapeamento de Trajetória.

Abstract

Accurate localization estimation represents one of the main challenges in mobile robotics, especially in indoor environments where global positioning systems are not available. Relative navigation methods, such as odometry and Inertial Navigation Systems (INS), are widely used; however, they present limitations due to the progressive accumulation of errors over time. In this context, this work aims to investigate and implement trajectory reconstruction methods in a line-following robot using data from velocity sensors, gyroscopes, and accelerometers. The trajectory estimation was processed and analyzed under three distinct approaches: odometry, INS, and the combination of both through an Extended Kalman Filter (EKF). The experimental results demonstrated that standalone odometry exhibits lateral displacement and rotation of the main axis due to cumulative angular errors. Pure INS, on the other hand, significantly diverges from the actual trajectory because of the increasing error associated with the double integration of noise. In contrast, the combination of techniques via EKF showed superior performance, reducing the individual deficiencies of each method and providing a geometric representation more consistent with the path effectively traveled.

Key-words: Mobile Robotics. Odometry. Inertial Navigation. Extended Kalman Filter. Trajectory Mapping.

Lista de figuras

Figura 1 – Robô	15
Figura 2 – Sistemas de coordenadas	18
Figura 3 – Acelerômetro	19
Figura 4 – Giroscópio	20
Figura 5 – <i>Encoder</i>	21
Figura 6 – <i>Encoder</i> em quadratura	21
Figura 7 – <i>Encoder</i> resolução	22
Figura 8 – <i>Encoder</i> Magnético	22
Figura 9 – Navegação inercial transformação	25
Figura 10 – Projeto 3D novo robô	32
Figura 11 – Novo robô montado	33
Figura 12 – Fluxograma do código embarcado	34
Figura 13 – Pista	35
Figura 14 – Resultados da trajetória com odometria	39
Figura 15 – Resultados da trajetória com INS	41
Figura 16 – Resultados da trajetória com EKF	42

Lista de abreviaturas e siglas

CI	Circuito Integrado
DSHOT	Digital Shot
EKF	Extended Kalman Filter
ESC	Electroninc Speed Controllers
GNSS	Global Navigation Satellite System
GPS	Sistema de Posicionamento Global
HTTP	Hypertext Transfer Protocol
ICAO	Organização da Aviação Civil Internacional
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
IoT	Internet das Coisas
LiDAR	Light Detection and Ranging
MEMS	Micro Electro Mechanical Systems
MIT	Instituto de Tecnologia de Massachusetts
MQTT	Message Queuing Telemetry Transport
PID	Proporcional-Integral-Derivativo
RTOS	Real-Time Operating System
TLS	Transport Layer Security
ULA	Unidade Lógica e Aritmética

Lista de símbolos

θ	Letra grega theta
Δ	Letra grega Delta
ω	Letra grega omega

Sumário

1	INTRODUCAO	11
1.1	Justificativas e Relevância	12
1.2	Objetivos	13
1.3	Metodologia	13
1.3.1	Robô	14
1.4	Organização e Estrutura	15
2	REVISÃO DE LITERATURA	17
2.1	Percepção	17
2.2	Descrição espacial	18
2.3	Giroscópios e Acelerômetros	19
2.4	Encoder Óptico	20
2.5	Encoder Magnético	22
2.6	Métodos de navegação	23
2.6.1	Odometria	23
2.6.1.1	Cálculo de d e Δ teta	24
2.6.2	Navegação Inercial	24
2.7	Filtro de Kalman	26
2.8	Filtro de Kalman Estendido	27
2.8.1	Etapa de Predição	28
2.8.2	Etapa de Correção	28
2.9	Microcontroladores	29
2.10	FreeRTOS	29
2.11	Protocolo MQTT	30
3	DESENVOLVIMENTO	32
3.1	Modificações no robô	32
3.2	Código embarcado	33
3.2.1	Estrutura de Código e Organização em Classes	33
3.2.2	Fluxo de Execução na Função Principal	33
3.3	Processamento dos dados	35
3.3.1	Coleta de Dados e Comunicação	35
3.4	Estimativa da trajetória	36
4	RESULTADOS	38
4.1	Resultados odometria	38

4.2	Resultados navegação inercial	40
4.3	Resultados Filtro de Kalman Estendido	41
5	CONSIDERAÇÕES FINAIS	43
	Referências	44

1 Introdução

A capacidade de deslocamento e localização em um ambiente sempre representaram desafios fundamentais para a humanidade. Os primeiros métodos de navegação para superar esse desafio fundamentavam-se na observação de pontos de referência terrestres e no mapeamento estelar (GIBSON *et al.*, 2013). Com a expansão das expedições marítimas, surgiu a navegação estimada, técnica que permitia estimar a posição de uma embarcação utilizando os dados de velocidade estimada com informações direcionais fornecidas pela bússola (GOEL *et al.*, 2022).

Para superar as limitações desses métodos, pesquisadores e engenheiros desenvolveram, ao longo das últimas décadas, uma ampla gama de sensores e sistemas de posicionamento. Entre as principais técnicas destacam-se a odometria, o Sistema de Navegação Inercial (INS) e os sistemas baseados em satélites. No entanto, a viabilidade de cada técnica é ditada por suas capacidades operacionais e limitações específicas (MOHAMED *et al.*, 2019).

No início do século XX, o desenvolvimento da girobússola marcou um avanço significativo ao substituir a bússola magnética. Diferente de sua predecessora, a girobússola indicava o norte verdadeiro em vez do norte magnético, proporcionando maior precisão direcional para as embarcações. Contudo, os primeiros giroscópios ainda enfrentavam elevada taxa de deriva que restringia seu uso apenas à estabilização de plataformas e indicadores de curva e inclinação, sendo ainda insuficientes para sistemas complexos de navegação autônoma (GIBSON *et al.*, 2013).

A transição para a navegação inercial prática iniciou-se apenas após o término da Segunda Guerra Mundial, no final da década de 1940 (GIBSON *et al.*, 2013). Um dos marcos inaugurais dessa tecnologia ocorreu na aviação em 8 de fevereiro de 1953, quando um sistema desenvolvido pelo Laboratório de Instrumentação do Instituto de Tecnologia de Massachusetts (MIT) guiou um bombardeiro B-29 de Massachusetts até a Califórnia (GOEL *et al.*, 2022).

Posteriormente, o sucesso da navegação inercial foi demonstrado pelo submarino de propulsão atômica USS Nautilus. Em agosto de 1958, a embarcação operou por 95 horas sob a calota polar, atravessando o Polo Norte utilizando exclusivamente sistemas inerciais como meio de posicionamento. (GOEL *et al.*, 2022).

A partir da década de 1990, o surgimento de sensores inerciais baseados em tecnologia MEMS provocou uma evolução significativa e ampliou as possibilidades de aplicação. Atualmente, esses sensores podem ser encontrados em diversos dispositivos eletrônicos, como celulares, consoles de jogos e etc. (GOEL *et al.*, 2022).

Outra técnica utilizada para localização é a odometria que diferente da navegação inercial, que se baseia na integração de forças de aceleração, fundamenta-se na medição do deslocamento físico por meio do cálculo da distância percorrida pelas rodas do veículo (J. BORENSTEIN; FENG, 1996). Na robótica móvel contemporânea, essa técnica consolidou-se como uma das formas mais simples e fundamentais de localização autônoma, utilizando codificadores incrementais para estimar a mudança de posição e orientação em relação a um ponto inicial conhecido (J. BORENSTEIN; FENG, 1996).

Uma aplicação prática dessa técnica ocorreu com os Mars Exploration Rovers, lançados em 2003, nos quais a odometria foi utilizada como um dos métodos para estimar o deslocamento e o posicionamento dos veículos em uma missão em marte (MOHAMED *et al.*, 2019).

Atualmente, a base da navegação aérea e terrestre fundamenta-se no Sistema Global de Navegação por Satélite (GNSS). Segundo a Organização da Aviação Civil Internacional (ICAO), o GNSS é definido como um sistema mundial de determinação de posição e tempo que inclui uma ou mais constelações de satélites, receptores embarcados e monitoramento da integridade do sistema, podendo contar com aumentos adicionais para atender ao desempenho exigido em operações específicas (INTERNATIONAL CIVIL AVIATION ORGANIZATION, 2007), sendo o GPS (Global Positioning System) o exemplo mais difundido de uma constelação GNSS (GIBSON *et al.*, 2013).

1.1 Justificativas e Relevância

A presença de robôs no cotidiano tem crescido significativamente nos últimos anos, ocupando espaços que abrangem desde a automação industrial até setores críticos como saúde e serviços (RIBEIRO, 2017). Esses avanços são possibilitados pelo desenvolvimento de tecnologias que permitem aos robôs realizarem tarefas cada vez mais complexas, aumentando sua utilidade e eficiência em múltiplos contextos.

No entanto, o principal desafio para a implementação e integração dos robôs móveis é a determinação de sua localização (CHEN; ZHANG, 2017). Esse processo é um dos mais críticos na implementação de robôs móveis, uma vez que a precisão na determinação de sua posição impacta diretamente na qualidade e segurança da execução de suas tarefas.

Embora existam diversas técnicas de navegação consolidadas, todas apresentam limitações intrínsecas que impedem uma localização totalmente precisa em todos os cenários. O Sistema de Posicionamento Global (GPS), por exemplo, apesar de oferecer precisão métrica em ambientes abertos, não é adequado para aplicações em ambientes internos, onde o sinal é severamente afetado por paredes e objetos (MOHAMED *et al.*, 2019). Além disso, o sinal de GPS enfrenta dificuldades críticas em ambientes subterrâneos, sob densa vegetação ou em locais sujeitos a múltiplos percursos de sinal e interferências

eletromagnéticas (GIBSON *et al.*, 2013).

Diante da inviabilidade de sistemas externos em ambientes internos, técnicas como a odometria e a navegação inercial tornam-se essenciais. Contudo, a odometria, embora seja uma técnica simples e amplamente utilizada, sofre com erros acumulados decorrentes do deslizamento das rodas e irregularidades do solo (GIBSON *et al.*, 2013). De forma análoga, a navegação inercial baseada em sensores também acumula erros provenientes do processo de dupla integração numérica das forças de aceleração e velocidade angular (GIBSON *et al.*, 2013).

Com base nisso, o presente trabalho visa realizar o mapeamento da trajetória de um robô seguidor de linha, utilizando sensores de velocidade, giroscópios e acelerômetros, para mapear a trajetória percorrida, combinando técnicas já conhecidas, como odometria e navegação inercial, a fim de implementar um sistema de navegação mais robusto.

1.2 Objetivos

Geral

O objetivo deste trabalho é investigar métodos de mapeamento de trajetória de um robô seguidor de linha utilizando dados dos sensores, com o intuito de reconstruir e representar graficamente o percurso realizado.

Específicos

1. Analisar e implementar adaptações no robô seguidor de linha existente, incluindo a substituição do encoder óptico por encoder magnético.
2. Desenvolver o código para leitura dos dados de odometria e sensores inerciais, bem como sua transmissão via MQTT em rede local.
3. Implementar *script* em *Python* para processar os dados recebidos e reconstruir o percurso em ambiente 2D.
4. Avaliar a reconstrução do trajeto obtido, comparando-o com o percurso real.
5. Discutir limitações e potenciais melhorias no processo de mapeamento de trajetória utilizando sensores inerciais e odometria.

1.3 Metodologia

A metodologia proposta neste trabalho consiste na utilização combinada de odometria e navegação inercial para estimar a trajetória percorrida pelo robô seguidor de

linha. A odometria será obtida por meio dos sensores de velocidade (*encoders*), enquanto a navegação inercial será realizada a partir dos dados fornecidos pela IMU (giroscópios e acelerômetros).

Os dados foram coletados durante a execução da trajetória, em modo online, e transmitidos para um servidor. Em seguida, são processados em um *script* desenvolvido em *Python*, que foi responsável por reconstruir e representar graficamente o percurso em um ambiente 2D.

Como critério de validação, foi utilizada a própria linha da pista. A trajetória reconstruída foi analisada visualmente em relação ao traçado real, avaliando sua capacidade de manter o formato original do percurso, bem como a presença de rotações indevidas e desvios perceptíveis ao longo da trajetória.

1.3.1 Robô

O robô utilizado neste trabalho foi originalmente desenvolvido por [Martins \(2024\)](#), conforme ilustrado na [Figura 1](#). O projeto foi desenvolvido em conformidade com as diretrizes da categoria de robôs seguidores de linha definidas pela Robocore.

Os principais componentes embarcados no robô são:

1. Placas de Circuito Impresso

O robô é composto por três placas de circuito impresso, denominadas *Head*, *Body* e *Side*:

- Placa *Head*:

Esta placa integra seis conjuntos de sensores QRE1113 dispostos em paralelo, cada conjunto composto por um emissor infravermelho e um fototransistor que é ativado pela reflexão parcial dos feixes infravermelhos. Sua função é realizar a leitura contínua do traçado que o robô segue.

- Placa *Side*:

Semelhante à placa *Head*, porém contendo apenas um conjunto de sensores. A placa *Side* é responsável por captar as marcações laterais da pista, identificando os pontos de início e fim do trajeto, bem como as transições de início e fim de curva.

- Placa *Body*:

Sendo a principal placa do projeto, a *Body* abriga diversos circuitos e componentes críticos, dentre os quais se destacam:

- a) Circuitos de proteção contra inversão de polaridade e acionamento da placa;
- b) Reguladores de tensão;

- c) Sensor de nível da bateria;
- d) Unidade de Medição Inercial (IMU);
- e) LEDs de debug;
- f) MOSFET para acionamento dos sensores frontais;
- g) Microcontrolador;
- h) Conectores para os controladores dos motores, *encoders* e para as placas de sensores de linha (laterais e de traçado).

2. Locomoção;

O sistema de locomoção do robô é composto por quatro engrenagens com uma redução de 1,538:1, além de dois motores *brushless* controlados por dois ESCs (*Electronic Speed Controllers*) via protocolo DSHOT e quatro rodas. Um aspecto fundamental do projeto foi o alinhamento dos motores com a IMU presente na placa *Body*, garantindo que o centro de rotação do robô coincida com o centro da IMU. Isso permite que a rotação do robô em seu próprio eixo corresponda diretamente à rotação no eixo Z do giroscópio.

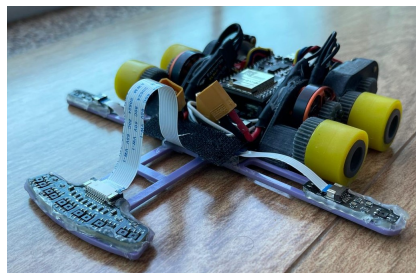


Figura 1 – Robô desenvolvido. Fonte: [Martins \(2024\)](#).

1.4 Organização e Estrutura

Este trabalho está organizado de forma a fornecer uma visão geral dos fundamentos teóricos da localização de robôs móveis, suas aplicações práticas e o desenvolvimento de um projeto para mapeamento da trajetória executada por um robô seguidor de linha.

No [Capítulo 1](#), é apresentado um breve histórico sobre os métodos de navegação, além da justificativa e da relevância do estudo do mapeamento de trajetórias em robôs móveis, evidenciando os desafios da localização em robótica móvel e a importância da integração de diferentes técnicas de navegação.

O [Capítulo 2](#) contempla a revisão de literatura, abordando tópicos como percepção e descrição espacial, funcionamento dos sensores utilizados, métodos de navegação estudados e os fundamentos dos microcontroladores e do sistema operacional FreeRTOS, que serviram de base para o desenvolvimento do projeto.

No [Capítulo 3](#), são detalhadas as modificações realizadas no robô seguidor de linha, bem como a implementação dos sensores e dos métodos de navegação adotados para viabilizar o mapeamento de trajetórias.

Por fim, o [Capítulo 4](#) apresenta os resultados obtidos, comparando a trajetória real com a trajetória reconstruída a partir dos dados dos sensores e discutindo-os, além de sugerir possíveis aprimoramentos para trabalhos futuros.

2 Revisão de literatura

2.1 Percepção

A capacidade de um robô de perceber o mundo ao seu redor e alterar seu comportamento com base nisso é o que torna um robô algo interessante de construir e um artefato útil quando concluído (EVERETT, 1995). Isto é feito por meio da utilização de sensores, que são a principal ferramenta para a percepção de espaço e localização no ambiente. Sem sensores, os robôs não seriam nada mais do que automação fixa, realizando a mesma tarefa repetidamente em um ambiente cuidadosamente controlado (EVERETT, 1995).

Para que o robô desempenhe suas funções com precisão geralmente ele é equipado com um conjunto de sensores, sendo a escolha de sensores adequados à aplicação é essencial para que ele possa executar suas tarefas.

Existe uma grande variedade de sensores usados em robôs móveis, por exemplo, para a detecção de linhas, é comum utilizar sensores infravermelhos ou câmeras, cada um com suas vantagens em termos de custo e desempenho. Na identificação de obstáculos, as opções variam desde sensores mais acessíveis, como os infravermelhos e ultrassônicos, até tecnologias mais avançadas, como o LiDAR, que oferecem maior precisão. Essa diversidade ressalta a importância de selecionar os sensores que melhor se adaptam às necessidades específicas do projeto (SIEGWART; NOURBAKHS, 2004).

Contudo, na prática, o uso desses sensores pode apresentar desafios. Um problema observado em robôs seguidores de linha, que utilizam sensores infravermelhos para detectar o traçado no chão, é que reflexos de luz na superfície podem ocasionar leituras equivocadas, confundindo o traçado com outros elementos.

Além disso, ruídos nas medições podem comprometer a interpretação do ambiente, enquanto limitações nos dados fornecidos pelos sensores podem resultar na ausência de informações essenciais para que o robô execute determinadas funções (SIEGWART; NOURBAKHS, 2004).

Para contornar essas questões, podem ser adotadas diversas estratégias, tais como aumentar o número de sensores, diversificar os tipos de sensores empregados e implementar técnicas de software, como filtros, para aprimorar a qualidade das medições.

2.2 Descrição espacial

Para descrever a posição e a orientação do robô no ambiente, é necessário estabelecer dois sistemas de coordenadas: um sistema global, que representa o ambiente, e um sistema local, que representa o próprio robô. Conforme ilustrado na [Figura 2](#), as coordenadas $\{X_I, Y_I\}$ determinam o referencial global, enquanto $\{X_R, Y_R\}$, definem o referencial local do robô. A posição do robô pode ser determinada escolhendo um ponto P em seu chassi, cujo deslocamento em relação ao sistema global é expresso pelo vetor $P = [x, y]$, onde x e y representam as coordenadas do ponto P no sistema global ([SIEGWART; NOURBAKHSH, 2004](#)).

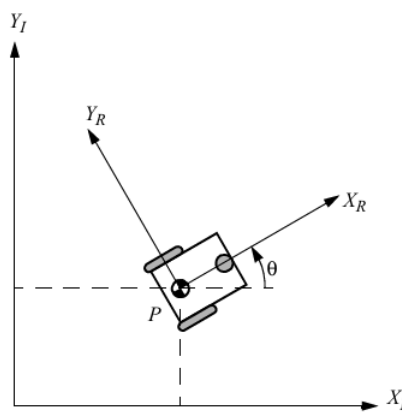


Figura 2 – Representação de sistemas de coordenadas para descrição espacial. Fonte: [Siegwart e Nourbakhsh \(2004\)](#).

Para representar a rotação em torno do eixo z , utiliza-se uma matriz de rotação, que descreve como o referencial local está orientado em relação ao global. Essa matriz é dada por:

$$R_z = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

Entretanto, conforme o robô se move, é necessário representar sucessivamente suas rotações e translações. Para isso, utiliza-se a matriz homogênea, que combina uma matriz de rotação com o vetor de translação, incorporando também os fatores de perspectiva e escala. Essa abordagem simplifica o processo de aplicação de transformações lineares sucessivas e permite representar integralmente um dado sistema de coordenadas. A matriz homogênea é definida como:

$$T = \begin{bmatrix} R & P \\ 0 & 1 \end{bmatrix} \quad (2.2)$$

onde R é a matriz de rotação em relação ao sistema global; P é o vetor de translação e a última linha $[0 \ 1]$ é a perspectiva e fator de escala. Logo, para um sistema com translação em x e y , e rotação no eixo z , obtém-se como resultado a matriz:

$$T = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & x \\ \sin \theta & \cos \theta & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

2.3 Giroscópios e Acelerômetros

Acelerômetros são dispositivos capazes de medir a aceleração aplicada ao longo de um ou mais eixos (MARTINS, 2024). O princípio de funcionamento de um acelerômetro pode ser representado pelo modelo clássico de sistema massa-mola amortecido com força aplicada. Nesse modelo, uma massa de prova está presa a uma mola de rigidez definida e a um amortecedor, que controla oscilações indesejadas. Quando o sistema é submetido a uma aceleração linear, uma força proporcional ao produto da massa pela aceleração atua sobre a massa de prova, causando sua deflexão. Essa deflexão é detectada por um sensor capacitivo e convertida em um sinal elétrico equivalente (QAZIZADA; PIVARČIOVÁ, 2016). Conforme ilustrado na Figura 3, em repouso, as capacitâncias $C1$ e $C2$ são iguais. Sob a influência da aceleração, essas capacitâncias variam de acordo com a direção e intensidade das forças que causam a alteração no estado inercial do sistema. Essas variações de capacitância são então convertidas em um sinal elétrico, o qual é utilizado para determinar a aceleração do dispositivo (TORRES, 2015).

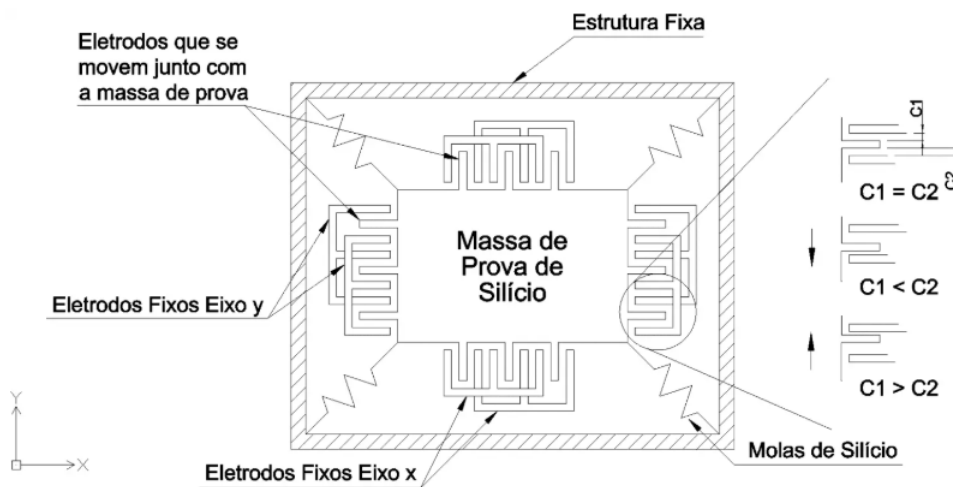


Figura 3 – Estrutura de um acelerômetro de dois eixos. Fonte: Torres (2015).

Um giroscópio é um dispositivo usado para medir ou manter a orientação, baseado nos princípios da conservação do momento angular (QAZIZADA; PIVARČIOVÁ, 2016). Ele se

utiliza de uma estrutura semelhante à do acelerômetro através da variação de capacitância, mas ao invés de medir o deslocamento, ele mede a velocidade angular através da força de Coriolis (STUHLER, 2022).

Conforme ilustrado na Figura 4, o giroscópio é um dispositivo com duas partes móveis, cada uma suportada por molas: uma que se move no sentido horizontal (eixo x) e outra que vibra na direção vertical (eixo y). Quando o dispositivo gira em torno do eixo z , uma força chamada Coriolis empurra essas partes para o lado (na direção x). Esse deslocamento muda a distância entre eletrodos fixos e móveis, alterando a capacitância – mesmo princípio de funcionamento do acelerômetro explicado anteriormente. Dessa maneira, o giroscópio consegue detectar e medir a rotação (TORRES, 2015).

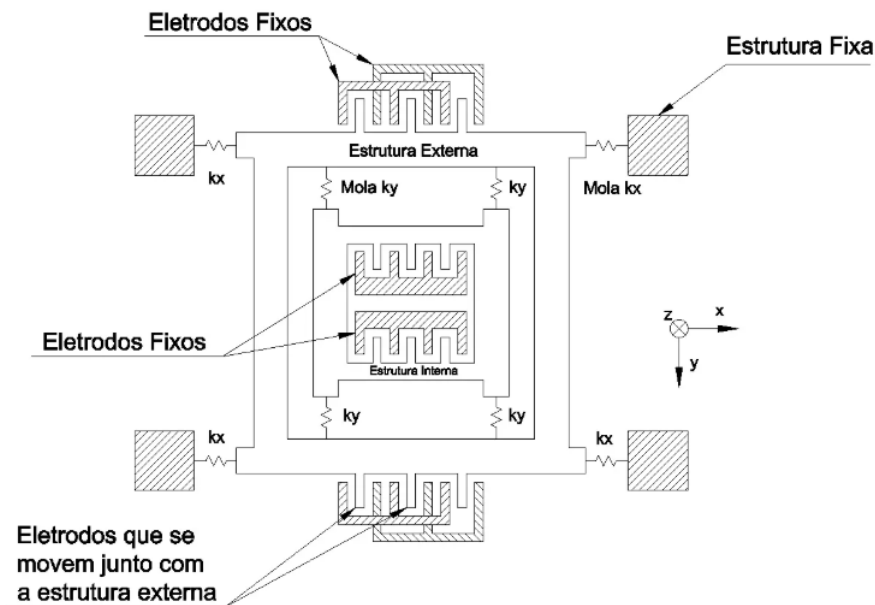


Figura 4 – Estrutura de um giroscópio de um eixo. Fonte: Torres (2015).

2.4 Encoder Óptico

O *encoder* óptico é um sensor composto por um emissor que gera um feixe de luz focalizado, o qual é direcionado a um fotodetector. Interposto entre esses dois elementos, há um disco acoplado ao eixo rotativo de interesse, que possui diversos orifícios. À medida que o disco gira, esses orifícios alternam a interrupção do feixe, produzindo um conjunto de pulsos a cada revolução do eixo (J. BORENSTEIN; FENG, 1996), conforme ilustrado na Figura 5.

Existem dois tipos básicos de codificadores ópticos: incremental e absoluto. O *encoder* incremental é um dispositivo que gera pulsos de sinal conforme o eixo rotaciona, ele mede a mudança na posição angular a partir de um ponto de referência e não mantém a referência da posição final em caso de falha de energia. Já o *encoder* absoluto fornece a

posição angular absoluta do eixo, ou seja, sua posição exata em relação a um ponto fixo. Sua principal vantagem é a capacidade de manter a informação de posição mesmo após uma queda de energia ([DYNAPAR ENCODERS, 2019](#)).

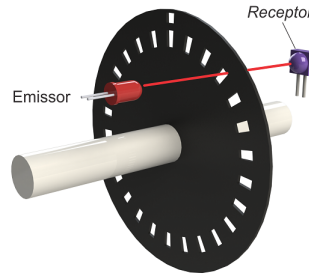


Figura 5 – Ilustração *encoder*. Fonte: [Almeida \(2016\)](#).

No caso dos *encoders* incrementais, há duas configurações: o tacômetro de canal único e o *encoder* em fase de quadratura. O tacômetro de canal único, por ser mais simples, possui apenas um canal, o que significa que ele gera apenas uma sequência de pulsos a cada rotação do eixo. Dessa forma, é possível medir a quantidade de movimento angular, mas não a direção, já que o sinal não traz nenhuma informação que diferencie o sentido do giro. Em contrapartida, os *encoders* em fase de quadratura dispõem de dois canais, com um deles deslocado em relação ao outro de forma que os pulsos gerados fiquem defasados em 90° , conforme ilustrado na [Figura 6](#). Essa técnica permite identificar qual canal está à frente do outro, possibilitando a determinação da direção da rotação ([J. BORENSTEIN; FENG, 1996](#)).

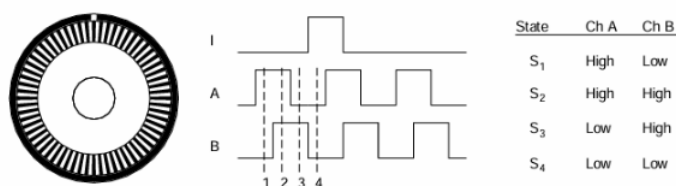


Figura 6 – *Encoder* Quadratura - Pulsos. Fonte: [J. Borenstein e Feng \(1996\)](#).

Ao aumentar o número de orifícios no disco, melhora-se a resolução do sensor, conforme ilustrado na [Figura 7](#); entretanto, esse aprimoramento costuma acarretar um aumento no custo do equipamento. Esse tipo de sensor é amplamente utilizado para quantificar distâncias, controlar velocidades, medir ângulos, contabilizar rotações e realizar posicionamentos ([ALMEIDA, 2016](#)).

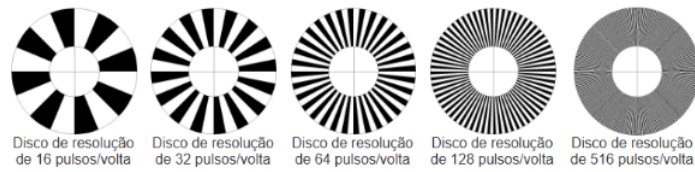


Figura 7 – *Encoder* Resolução - Pulsos. Fonte: [J. Borenstein e Feng \(1996\)](#).

2.5 *Encoder* Magnético

O *encoder* magnético é um sensor de posição e/ou velocidade angular que converte a rotação do eixo em variações de campo magnético, as quais são detectadas por um sensor magnético, tipicamente baseado em efeito Hall. Na implementação mais simples, um ímã permanente é fixado ao eixo girante (por exemplo, o eixo do motor) e o sensor é montado em uma placa estacionária próxima ao ímã, de modo a perceber as variações do campo magnético gerado ([AKM SEMICONDUCTOR, \[s.d.\]](#)), como representado na [Figura 8](#). À medida que o eixo gira, o vetor de campo magnético produzido pelo ímã também muda de orientação no entorno do sensor; o elemento Hall detecta essa variação na distribuição do campo magnético e a converte em um sinal elétrico. Dessa forma, para estimar a posição rotacional no plano XY , é necessário obter dois componentes ortogonais do campo, B_x e B_y , o que pode ser realizado por meio de elementos Hall dispostos de forma a medir essas componentes ([AKM SEMICONDUCTOR, \[s.d.\]](#)).

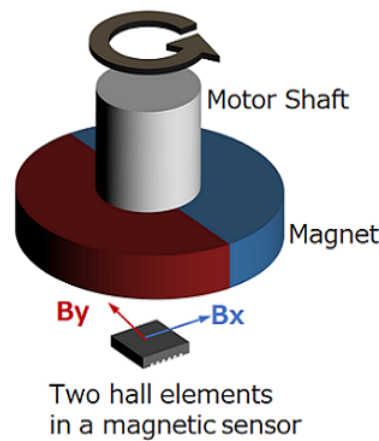


Figura 8 – *Encoder* Magnético. Fonte: [AKM Semiconductor \(\[s.d.\]\)](#).

Além disso, por serem sensores baseados em campo magnético, *encoders* magnéticos apresentam boa imunidade a poeira, óleo e vibração quando comparados a *encoders* ópticos, e permitem arquiteturas compactas com baixa latência de leitura, a depender do CI e do processamento adotado ([TEXAS INSTRUMENTS, 2024](#)). Por outro lado, a precisão angular depende fortemente do alinhamento mecânico, incluindo a centralização relativa entre ímã

e sensor e a distância entre eles, bem como da presença de campos parasitas; esses fatores podem introduzir erros sistemáticos e exigem cuidados de montagem e calibração ([AKM SEMICONDUCTOR](#), [s.d.]).

2.6 Métodos de navegação

A localização é um dos problemas fundamentais na navegação de robôs móveis. Em ambientes internos, esse desafio consiste em estimar a pose do robô, ou seja, sua posição e orientação ([GANGANATH; LEUNG, 2012](#)). Para atender a essa necessidade, foram desenvolvidas diversas técnicas de localização, que podem ser classificadas em dois principais métodos: localização relativa e localização absoluta ([J. BORENSTEIN; FENG, 1996](#)). No método de localização relativa, a posição e a orientação do robô são estimadas com base em seu estado inicial, enquanto no método de localização absoluta, esses parâmetros são determinados em relação a um objeto fixo no ambiente ([BEZERRA, 2004](#)). Neste capítulo, são apresentados dois métodos amplamente utilizados para localização relativa: a Odometria e a Navegação Inercial.

2.6.1 Odometria

A odometria é uma técnica de localização relativa que pressupõe que a posição inicial do robô seja conhecida. Ela estima a posição e a orientação dos robôs por meio da medição das rotações das rodas, que podem ser convertidas com precisão em deslocamento linear relativo ao solo ([J. BORENSTEIN; FENG, 1996](#)). Sendo amplamente utilizada, a odometria é um dos métodos mais comuns para a estimativa de posição relativa e pode ser implementada de forma independente, sem a necessidade de informações externas, o que a torna simples, econômica e de fácil execução em tempo ([CHEN; ZHANG, 2017](#)). Contudo, essa técnica apresenta limitações devido a diversos erros sistemáticos e não sistemáticos, tais como o desalinhamento das rodas, diferenças nos diâmetros das rodas, deslocamento sobre superfícies irregulares e escorregamento das rodas, entre outros ([J. BORENSTEIN; FENG, 1996](#)).

Para definir o modelo de odometria, é necessário levar em consideração algumas características físicas do robô. No caso deste trabalho, é adotada uma simplificação do modelo cinemático. Embora o robô possua quatro rodas, as rodas de cada lado estão suficientemente próximas para que o comportamento do sistema possa ser aproximado ao de um robô diferencial de duas rodas. Assim, o modelo de odometria utilizado será o modelo diferencial simplificado, em que o robô é representado como um ponto localizado no centro geométrico entre as rodas.

A partir dessas considerações, o modelo de movimento absoluto do robô é definido pelas seguintes equações:

$$x_t = x_{t-1} + d \cos \left(\theta_{t-1} + \frac{\Delta\theta_t}{2} \right) \quad (2.4)$$

$$y_t = y_{t-1} + d \sin \left(\theta_{t-1} + \frac{\Delta\theta_t}{2} \right) \quad (2.5)$$

$$\theta_t = \theta_{t-1} + \Delta\theta_t \quad (2.6)$$

onde x_t e y_t representam a posição do robô no instante t , θ_t é a sua orientação, d corresponde à distância percorrida entre os instantes $t - 1$ e t , e $\Delta\theta_t$ é a variação angular no mesmo intervalo de tempo (AHMED, 2023).

2.6.1.1 Cálculo de d e $\Delta\theta$

Para o cálculo de d e $\Delta\theta$, é determinada a distância percorrida por cada roda, esquerda (d_l) e direita (d_r), durante o intervalo entre medições. Uma vez obtidas essas distâncias, calculam-se:

$$d = \frac{d_r + d_l}{2}$$

$$\Delta\theta = \frac{d_r - d_l}{dw}$$

em que dw é calculado a partir do centro geométrico do sistema de rodas, obtido pela interseção das diagonais formadas entre as rodas dianteiras e traseiras. Dessa forma, o parâmetro dw é definido como a distância entre cada roda e esse ponto central (AHMED, 2023).

2.6.2 Navegação Inercial

A navegação inercial é um método de localização relativa que estima a posição e a orientação do robô por meio das medições realizadas por acelerômetros e giroscópios. Os giroscópios fornecem informações sobre a taxa angular, enquanto os acelerômetros medem as variações na velocidade (QAZIZADA; PIVARČIOVÁ, 2016). Ao integrar a taxa angular ao longo do tempo, obtém-se a orientação em ângulos; já a dupla integração da aceleração gera os dados referentes à distância percorrida. Essa abordagem parte do pressuposto de que a posição inicial do robô é conhecida, uma vez que os cálculos são efetuados a partir desse ponto de referência (BARSHAN; DURRANT-WHYTE, 1993).

Para definir o modelo de navegação inercial para o robô em duas dimensões é necessário realizar a medição das acelerações em dois eixos ortogonais no plano de

movimento e da rotação em torno do eixo perpendicular a esse plano. Para isso, são utilizados dois acelerômetros, responsáveis por medir as acelerações nos eixos do corpo do robô, e um giroscópio, que mede a velocidade angular associada à mudança de direção.

Como os acelerômetros medem as acelerações no referencial do robô, é necessário transformá-las para um referencial de navegação fixo. Essa transformação depende da orientação instantânea do robô, representada pelo ângulo de azimuth A , que descreve a rotação do corpo em relação ao referencial de navegação. O valor de A é obtido pela integração da velocidade angular medida pelo giroscópio ao longo do tempo (NOURELDIN; KARAMAT; GEORGY, 2013). A transformação entre os sistemas de coordenadas é apresentada na Figura 9.

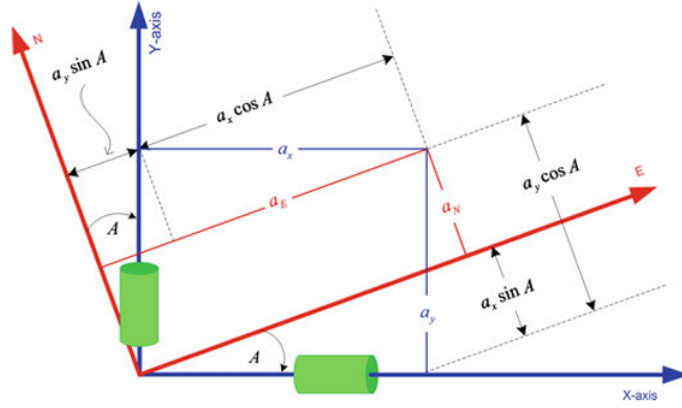


Figura 9 – Transformação do sistema de coordenadas do veículo (X-Y) para o sistema de coordenadas de navegação (E-N). Fonte: (NOURELDIN; KARAMAT; GEORGY, 2013).

Sejam a_x e a_y as acelerações medidas pelos acelerômetros no referencial do corpo do robô, e a_E e a_N as acelerações expressas no referencial de navegação, orientado nos eixos leste (E) e norte (N). A relação entre essas grandezas pode ser descrita pelas seguintes equações de transformação:

$$a_E = a_x \cos(A) + a_y \sin(A) \quad (2.7)$$

$$a_N = a_y \cos(A) - a_x \sin(A) \quad (2.8)$$

Uma vez obtidas as acelerações no referencial de navegação, estas são integradas ao longo do tempo para o cálculo das velocidades nos eixos leste e norte:

$$v_E(t) = \int a_E(t) dt \quad (2.9)$$

$$v_N(t) = \int a_N(t) dt \quad (2.10)$$

A posição do robô é então obtida por meio de uma segunda integração das velocidades:

$$x_E(t) = \int v_E(t) dt \quad (2.11)$$

$$x_N(t) = \int v_N(t) dt \quad (2.12)$$

A orientação $\theta(t)$ do robô é calculada pela integração da velocidade angular $\omega(t)$ fornecida pelo giroscópio:

$$A(t) = \int \omega(t) dt + A_0 \quad (2.13)$$

em que A_0 representa a orientação inicial do robô ([NOURELDIN; KARAMAT; GEORGY, 2013](#)).

Entretanto, um desafio dessa técnica é que, embora as medições de taxa sejam confiáveis por longos períodos, elas precisam ser integradas para fornecer dados de orientação, posição e velocidade. Como consequência, a navegação inercial apresenta erros de posição que se acumulam com o tempo e com a distância percorrida. Para superar esse problema, é necessário redefinir periodicamente os sensores inerciais por meio de outros mecanismos de sensoramento absoluto, eliminando assim o erro acumulado ([BARSHAN; DURRANT-WHYTE, 1993](#)).

2.7 Filtro de Kalman

O Filtro de Kalman é um algoritmo projetado para estimar e prever o estado de um sistema dinâmico na presença de incertezas, como ruídos de medição ([BECKER, 2024](#)). Desenvolvido pelo engenheiro húngaro Rudolf Kalman, uma de suas principais características é o funcionamento de forma recursiva, o que dispensa o armazenamento do histórico completo de dados, pois o algoritmo precisa apenas da estimativa do passo anterior e a medição atual para calcular o novo estado ([MATLAB, 2017](#)).

Este processo é dividido em duas etapas fundamentais. Na Predição, o modelo matemático do sistema é utilizado para estimar o estado atual com base no estado anterior, sendo essa projeção realizada antes que a medição sensorial do momento atual seja processada. Já na etapa de Atualização, o algoritmo incorpora a medição real para ajustar e refinar a estimativa previamente calculada ([MATLAB, 2017](#)).

O grande diferencial do Filtro de Kalman reside na sua capacidade de equilibrar incertezas ([MATHWORKS, 2025](#)). Na prática, o algoritmo avalia a confiabilidade das informações: caso o ruído de medição seja muito elevado, o filtro tende a priorizar o valor estimado pelo modelo matemático, por outro lado, se a incerteza do modelo for maior que a do sensor, o filtro deposita maior confiança na medição realizada ([MATLAB, 2017](#)).

2.8 Filtro de Kalman Estendido

O Filtro de Kalman Estendido (*Extended Kalman Filter* - EKF) é uma extensão do Filtro de Kalman clássico desenvolvida para sistemas não lineares. Enquanto o filtro tradicional é aplicável apenas a sistemas descritos por modelos lineares, o EKF permite a estimação de estados em sistemas cuja dinâmica e/ou modelo de medição são não lineares ([MASOOLEH et al., 2022](#)).

No contexto deste trabalho, o modelo cinemático do robô diferencial e as equações que relacionam sensores como *encoders* e IMU ao vetor de estados apresentam natureza não linear, tornando necessária a utilização do EKF.

O sistema dinâmico não linear pode ser descrito por ([MASOOLEH et al., 2022](#)):

$$\mathbf{x}_k = f(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1} \quad (2.14)$$

$$\mathbf{y}_k = h(\mathbf{x}_k) + \mathbf{v}_k \quad (2.15)$$

- $\mathbf{x}_k$ é o vetor de estados do sistema no instante discreto k , contendo as variáveis que descrevem completamente a dinâmica do robô;
- $\mathbf{y}_k$ é o vetor de medições obtido pelos sensores no instante k ;
- $\mathbf{u}_{k-1}$ é o vetor de entrada de controle aplicado ao sistema no instante $k - 1$;
- $f(\cdot)$ representa a função não linear de transição de estados, responsável por descrever a evolução temporal do sistema a partir do estado anterior e das entradas de controle;
- $h(\cdot)$ representa a função não linear de medição, que relaciona o vetor de estados às grandezas efetivamente medidas pelos sensores;
- $\mathbf{w}_{k-1}$ é o ruído do processo, geralmente modelado como ruído branco gaussiano de média zero e covariância $\mathbf{Q}$;
- $\mathbf{v}_k$ é o ruído de medição, também modelado como ruído branco gaussiano de média zero e covariância $\mathbf{R}$.

Diferentemente do filtro linear, o EKF realiza uma linearização local do sistema a cada iteração por meio da expansão em série de Taylor de primeira ordem. Essa aproximação permite aplicar a estrutura recursiva do Filtro de Kalman tradicional ao modelo linearizado.

2.8.1 Etapa de Predição

A predição do estado é realizada diretamente a partir da função não linear:

$$\hat{\mathbf{x}}_k^- = f(\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k) \quad (2.16)$$

Para atualizar a matriz de covariância do erro, utiliza-se a matriz Jacobiana da função de transição de estados:

$$\mathbf{F}_k = \left. \frac{\partial f}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_{k-1}, \mathbf{u}_k} \quad (2.17)$$

A covariância *a priori* é então calculada por:

$$\mathbf{P}_k^- = \mathbf{F}_k \mathbf{P}_{k-1} \mathbf{F}_k^T + \mathbf{Q} \quad (2.18)$$

A utilização da Jacobiana $\mathbf{F}_k$ permite aproximar localmente a propagação da incerteza em torno do ponto de operação atual.

2.8.2 Etapa de Correção

Na etapa de atualização, calcula-se inicialmente a matriz Jacobiana do modelo de medição:

$$\mathbf{H}_k = \left. \frac{\partial h}{\partial \mathbf{x}} \right|_{\hat{\mathbf{x}}_k^-} \quad (2.19)$$

O ganho de Kalman é então obtido por:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (2.20)$$

O estado estimado *a posteriori* é então atualizado:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k (\mathbf{y}_k - h(\hat{\mathbf{x}}_k^-)) \quad (2.21)$$

Por fim, a matriz de covariância é corrigida:

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \mathbf{P}_k^- \quad (2.22)$$

2.9 Microcontroladores

Os microcontroladores são circuitos integrados que reúnem, em um único chip, os principais componentes necessários ao funcionamento de um sistema computacional. Esses dispositivos incorporam unidade de processamento, memória e periféricos de entrada e saída, dependendo apenas de uma fonte de alimentação externa para operar, o que os caracteriza como computadores completos em um único encapsulamento ([KERSCHBAUMER et al., 2013](#)).

Esses dispositivos reúnem processador (Unidade Lógica e Aritmética – ULA), memória para armazenamento de dados e programas, interfaces de entrada e saída, temporizadores, conversores analógico-digitais (ADC), conversores digital-analógicos (DAC) e módulos de comunicação serial, entre outros recursos ([CARVALHO CASTRO PENIDO; TRINDADE, 2013](#)). Essa arquitetura integrada permite a execução de tarefas específicas de controle e monitoramento com elevada eficiência e baixo consumo de energia.

Devido à sua versatilidade, dimensões reduzidas e baixo custo, os microcontroladores são amplamente empregados em sistemas embarcados, estando presentes em dispositivos eletrônicos de uso cotidiano, como televisores, automóveis e aeronaves, além de aplicações industriais, médicas e de automação ([KERSCHBAUMER, 2025](#)). No contexto da robótica móvel, esses dispositivos desempenham papel fundamental na aquisição de dados de sensores, no processamento de algoritmos de controle e na comunicação com outros sistemas.

2.10 FreeRTOS

Os sistemas operacionais de tempo real (RTOS) surgiram nas décadas de 1960 e 1970 para atender às demandas de aplicações críticas e sensíveis ao tempo, especialmente nos setores militar, aeroespacial e industrial ([SUSNJARA; SMALLEY, 2025](#)). Diferentemente dos sistemas operacionais de uso geral, que priorizam o desempenho médio do sistema, os RTOS são projetados para garantir previsibilidade temporal, assegurando que tarefas sejam executadas dentro de prazos previamente definidos.

O FreeRTOS é um sistema operacional de tempo real amplamente utilizado em sistemas embarcados, nos quais a resposta a eventos deve ser rápida e determinística ([DENARDIN; BARRIQUELLO, 2019](#)). Em comparação com sistemas operacionais convencionais, os RTOS são menores e mais leves, sendo ideais para dispositivos com restrições de memória, capacidade de processamento e consumo de energia. A partir da década de 2000, com o avanço dos sistemas embarcados e a consolidação da Internet das Coisas (IoT), houve um aumento significativo na demanda por soluções eficientes e de baixo consumo de recursos, contexto no qual o FreeRTOS ganhou destaque ([SUSNJARA; SMALLEY, 2025](#)).

Entre as principais características de um RTOS destacam-se o determinismo na execução das tarefas, garantindo que cada processo seja concluído dentro de um intervalo de tempo previsível, o escalonamento baseado em prioridades, que assegura a execução preferencial de tarefas críticas, e a baixa latência tanto na resposta a interrupções quanto na troca de contexto entre tarefas, reduzindo atrasos e aumentando a eficiência do sistema (SUSNJARA; SMALLEY, 2025).

No contexto deste trabalho, a utilização do FreeRTOS é justificada pela complexidade e pela quantidade de tarefas executadas pelo robô seguidor de linha. Para realizar o seguimento da trajetória, o sistema precisa executar a coleta e conversão dos sinais de seis sensores analógicos, o processamento para o cálculo da ação de controle e a atuação nos motores, garantindo a permanência do robô sobre a linha.

Além disso, para realizar o mapeamento da pista, objetivo principal deste trabalho, o robô deve coletar e processar dados provenientes de outros sensores, como *encoders*, giroscópio e acelerômetro. Diante da complexidade de gerenciar simultaneamente esses processos, o uso de um *loop* convencional se mostra inviável, reforçando a necessidade de um sistema operacional de tempo real como o FreeRTOS.

2.11 Protocolo MQTT

O MQTT surgiu em 1999, desenvolvido por Andy Stanford-Clark e Arlen Nipper, com o objetivo de viabilizar a comunicação eficiente com oleodutos via satélite, minimizando o consumo de bateria e o uso de largura de banda. Desde sua concepção, o protocolo foi projetado para ser leve, simples de implementar e eficiente em termos de utilização de rede. Inicialmente aplicado em sistemas embarcados proprietários, o MQTT evoluiu para se tornar amplamente utilizado em aplicações abertas de Internet das Coisas (IoT), consolidando-se como um dos principais protocolos de mensageria nesse contexto (HIVEMQ, 2024).

Do ponto de vista técnico, o MQTT é um protocolo de mensagens leve que utiliza um formato binário para comunicação entre clientes e servidores. Diferentemente de protocolos baseados em texto, como o HTTP, o uso de mensagens binárias reduz o tamanho dos pacotes transmitidos e diminui o processamento necessário para a interpretação dos dados. Essa característica torna o protocolo particularmente adequado para redes com largura de banda limitada e dispositivos com recursos computacionais restritos, como sensores e sistemas embarcados típicos de aplicações IoT (HIVEMQ, 2024).

A arquitetura do MQTT é composta por dois elementos principais: o *broker* e os clientes. O *broker* atua como intermediário responsável por receber, gerenciar e encaminhar mensagens aos destinatários apropriados, enquanto os clientes são as entidades que publicam mensagens ou se inscrevem para recebê-las. Esse modelo promove o desacoplamento entre

emissores e receptores, aumentando a flexibilidade e a escalabilidade do sistema ([HIVEMQ, 2024](#)).

A comunicação no MQTT é baseada em tópicos e assinaturas. Os tópicos são organizados de forma hierárquica, estruturados como *strings* separadas por barras, de maneira semelhante a caminhos de diretórios. Quando um cliente publica uma mensagem, ela é associada a um tópico específico, e outros clientes podem se inscrever nesse tópico para receber automaticamente as mensagens correspondentes ([QUINCOZES; EMILIO; KAZIENKO, 2019](#)).

O MQTT define três níveis de Qualidade de Serviço (QoS): QoS 0 - entrega no máximo uma vez; QoS 1 - entrega pelo menos uma vez e QoS 2 - entrega exatamente uma vez ([HIVEMQ, 2024](#)). Esses níveis permitem ajustar o equilíbrio entre confiabilidade e consumo de recursos, de acordo com a criticidade da aplicação.

Entre as principais vantagens do MQTT destacam-se sua leveza e eficiência, a possibilidade de comunicação bidirecional, a facilidade de escalabilidade decorrente do desacoplamento entre clientes, a entrega confiável de mensagens por meio dos diferentes níveis de Qualidade de Serviço e os recursos de segurança, como a possibilidade de criptografia utilizando TLS ([HIVEMQ, 2024](#)).

O MQTT tem sido amplamente utilizado em diferentes aplicações de IoT, como em casas inteligentes, na agricultura e em sistemas de transporte, demonstrando sua versatilidade e adequação a cenários com restrições de rede e recursos computacionais ([HIVEMQ, 2024](#)).

3 Desenvolvimento

3.1 Modificações no robô

Para a implementação da navegação, foi necessário realizar algumas modificações no projeto originalmente desenvolvido por [Martins \(2024\)](#). A principal alteração consistiu na substituição do *encoder* óptico pelo encoder magnético.

Na configuração anterior, cada lado do robô utilizava um conjunto de quatro engrenagens: duas acopladas às rodas, uma ao motor e outra ao sensor óptico. Entretanto, como o processo de fabricação das peças é caseiro, após a montagem do robô observou-se uma movimentação lateral indesejada no eixo do *encoder*, o que comprometia a leitura do sensor óptico.

Dessa forma, a estrutura do robô foi redesenhada para acomodar o novo tipo de *encoder*, no qual o ímã está acoplado diretamente ao eixo do motor. Nesse sistema, foi implementado um sensor de efeito Hall em cada lado do robô, responsável por detectar a variação do campo magnético gerada pela rotação do ímã e, consequentemente, medir a velocidade angular das rodas. A [Figura 10](#) apresenta o novo projeto 3D do robô após as modificações estruturais.

Dessa forma, a estrutura do robô foi redesenhada para acomodar o novo tipo de *encoder*, no qual foi implementado um encoder magnético em cada lado do robô, responsável por detectar a variação do campo magnético gerada pela rotação do ímã, que está acoplado diretamente ao eixo do motor e, consequentemente, medir a velocidade angular das rodas. A [Figura 10](#) apresenta o novo projeto 3D do robô após as modificações estruturais.

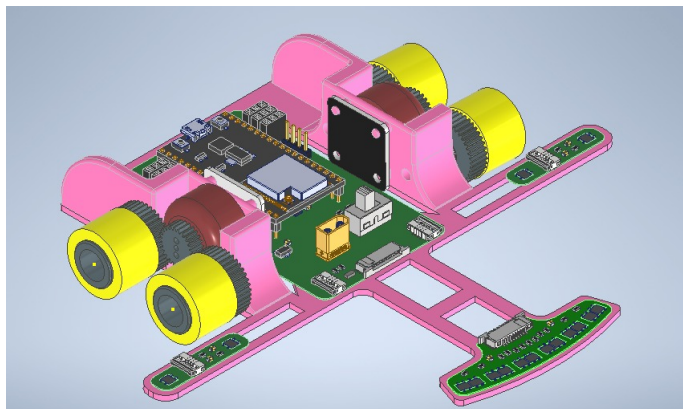


Figura 10 – Novo projeto 3D. Fonte: Autor.

Essa solução reduziu a complexidade mecânica do sistema, diminuindo o conjunto

de engrenagens de quatro para três de cada lado, além de tornar a montagem mais simples e a leitura do sensor mais estável e confiável. O resultado final da montagem é apresentado na [Figura 11](#).



Figura 11 – Foto do robô montado. Fonte: Autor.

3.2 Código embarcado

Nesta seção, é descrita a estrutura do software embarcado implementado no robô e as decisões de projeto adotadas para atender aos seus objetivos principais: seguir a trajetória, coletar os dados dos sensores e realizar o envio dessas informações.

3.2.1 Estrutura de Código e Organização em Classes

O código foi estruturado com uma arquitetura modular, organizada em classes, na qual cada classe é responsável por configurar um componente específico do robô. Essa abordagem melhora a legibilidade e a manutenção do código, além de favorecer a reutilização de componentes.

Além disso, a aplicação foi implementada utilizando o sistema operacional de tempo real FreeRTOS. Desse modo, o código pôde ser dividido em tarefas menores e independentes, cada uma com prioridade definida conforme sua criticidade. Essa estratégia também contribuiu para a organização do código e garante que atividades essenciais tenham prioridade sobre rotinas secundárias, diferentemente do modelo sequencial tradicional baseado em um único *loop* principal, comum em aplicações de microcontroladores.

3.2.2 Fluxo de Execução na Função Principal

O fluxo geral de execução do sistema é apresentado no fluxograma da [Figura 12](#). A inicialização começa com a configuração dos periféricos do microcontrolador e dos

protocolos de comunicação utilizados. Nessa etapa também são estabelecidas a conexão com a rede Wi-Fi e a comunicação com o *broker* MQTT.

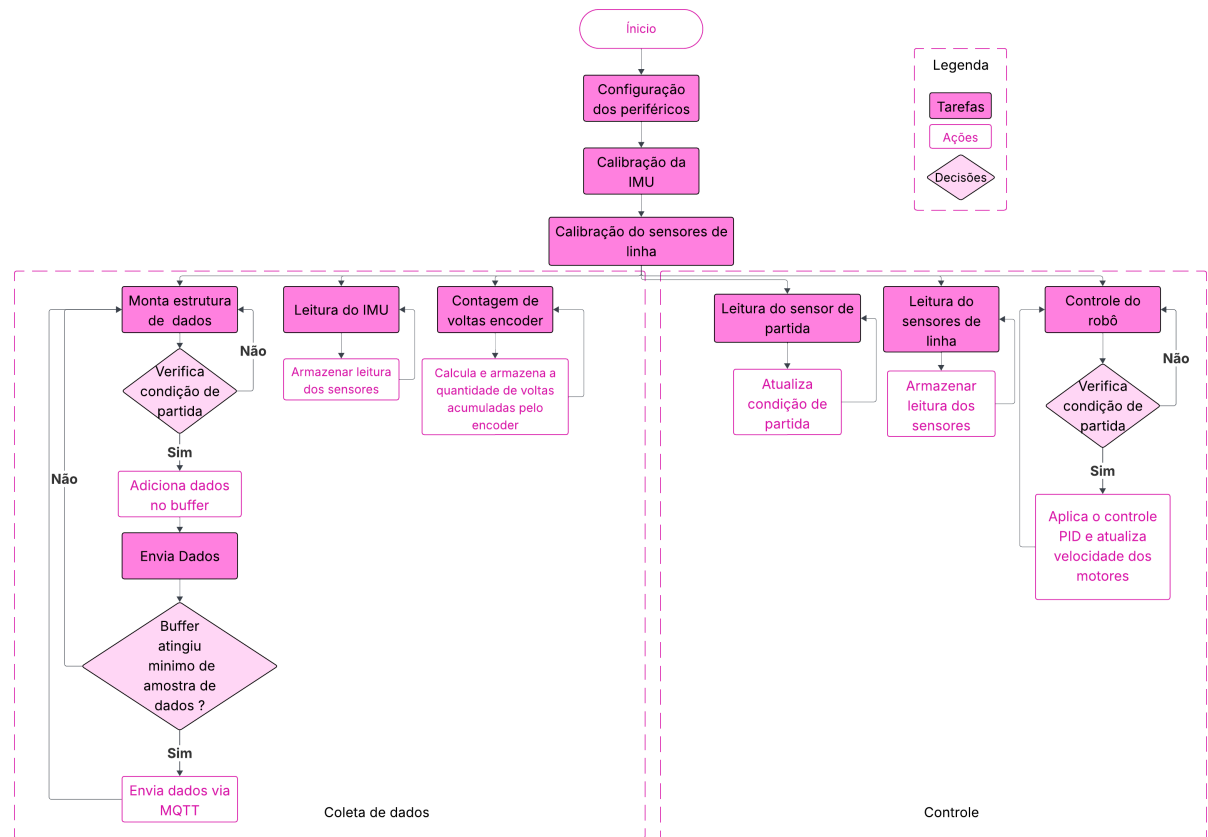


Figura 12 – Fluxograma do código. Fonte: Autor.

A primeira tarefa executada é a tarefa de calibração da Unidade de Medição Inercial (IMU). Com o robô mantido em repouso sobre a pista, são coletadas múltiplas amostras do acelerômetro e do giroscópio. Essas leituras são acumuladas, posteriormente é realizado o cálculo da média, e o resultado é usado como correção do offset nas medições subsequentes.

Concluída a calibração da IMU, realiza-se a calibração dos sensores de linha. Para isso, o robô gira em torno do próprio eixo, garantindo que todos os sensores passem sobre a superfície da linha e do fundo da pista, neste momento são registrados os valores máximos e mínimos obtidos por cada sensor, os quais são posteriormente empregados na normalização das leituras. Após a etapa de inicialização e calibração, são criadas e iniciadas as tarefas responsáveis pelo controle do robô e pela aquisição e envio de dados.

As tarefas de controle implementam a função principal do robô de seguir a linha. Inicialmente, é realizada a leitura do sensor infravermelho responsável por definir a condição de partida, habilitando ou não o robô de se locomover e enviar dados. Paralelamente, ocorre a leitura dos sensores de linha, cujos valores são armazenados em variáveis para uso

pelo algoritmo de controle. Por fim, a tarefa que implementa o controle do robô, verifica a condição de partida e, quando verdadeira, utiliza as leituras dos sensores para calcular o erro de posição, aplicar o controlador PID e atualizar as velocidades dos motores.

Já nas tarefas de coleta e envio de dados, primeiramente são realizadas as leituras do acelerômetro e do giroscópio, com o armazenamento dos valores obtidos. De forma paralela, em outra tarefa é efetuada a leitura dos *encoders* das rodas, realizando-se o cálculo do deslocamento angular acumulado.

A cada período de amostragem, as medições inerciais e de odometria são organizadas em uma estrutura de dados e inseridas em um *buffer*. Na tarefa de envio de dados, inicialmente é verificada a condição de partida; quando verdadeira e quando o *buffer* de dados atinge a quantidade mínima de amostras, os dados coletados da IMU e dos *encoders* são enviados via protocolo MQTT.

3.3 Processamento dos dados

3.3.1 Coleta de Dados e Comunicação

A coleta dos dados ocorre por meio de um *script* em *Python* utilizando o protocolo MQTT. O *script* se inscreve no tópico MQTT para receber as mensagens publicadas e armazená-las, os dados coletados são salvos em um arquivo para processamento posterior.

As coletas de dados foram realizadas em uma pista de testes, conforme apresentado na [Figura 13](#). O robô percorreu o trajeto diversas vezes, em velocidade constante, sendo registrados os dados correspondentes a cada execução com uma taxa de amostragem fixa, permitindo posteriormente a análise e comparação dos resultados.

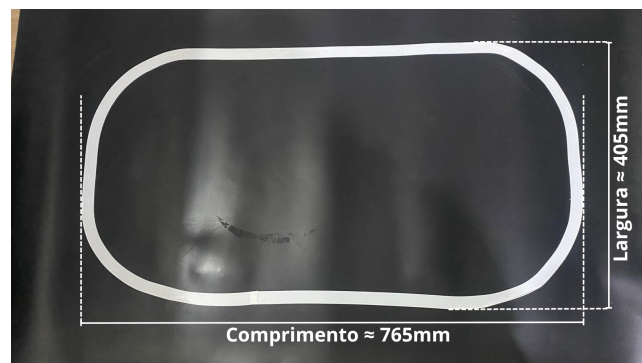


Figura 13 – Foto da pista utilizada. Fonte: Autor.

3.4 Estimativa da trajetória

A determinação da trajetória do robô foi realizada por meio da combinação de técnicas de navegação inercial, odometria e combinação da técnicas baseada no Filtro de Kalman Estendido (EKF). Inicialmente, cada método fornece uma estimativa independente da trajetória.

No método de Navegação Inercial, os dados brutos provenientes do sensor MPU6050 são inicialmente convertidos para unidades do Sistema Internacional (SI). A aceleração é convertida para m/s^2 e a velocidade angular para rad/s , utilizando-se as constantes de sensibilidade configuradas no sensor. A conversão da aceleração é dada por:

$$acc_{ms^2} = \left(\frac{\text{valor bruto}}{SENSITIVITY_{acc}} \right) \cdot 9.80665 \quad (3.1)$$

De forma análoga, a velocidade angular é obtida por:

$$\omega_{rad/s} = \left(\frac{\text{valor bruto}}{SENSITIVITY_{gyro}} \right) \quad (3.2)$$

A orientação θ é estimada pela integração da velocidade angular no eixo Z . Com base nesse ângulo, as acelerações medidas no referencial local (a_x, a_y) são rotacionadas para o referencial global. Em seguida, aplica-se dupla integração numérica, utilizando o método trapezoidal, para a obtenção das coordenadas cartesianas (X, Y).

Paralelamente, a estimativa por odometria é realizada a partir das medições dos *encoders*. A cada intervalo de amostragem, calcula-se a diferença entre os ângulos acumulados no instante atual e no instante anterior, obtendo-se o deslocamento angular de cada motor. Considerando que os *encoders* estão acoplados ao eixo do motor, aplica-se a relação de transmissão mecânica para determinar o deslocamento angular efetivo das rodas. A distância percorrida por cada roda é então calculada pela multiplicação do deslocamento angular pelo raio da roda.

A partir dos deslocamentos individuais das rodas, determinam-se o deslocamento linear e a variação angular do robô no referencial local. Esses valores são projetados no referencial global, permitindo a atualização da pose.

Por fim, as estimativas provenientes da navegação inercial e da odometria são integradas por meio do Filtro de Kalman Estendido (EKF). Inicialmente, é realizada a sincronização temporal das medições por meio de interpolação linear, de modo a alinhar os instantes de tempo das estimativas provenientes da navegação inercial e da odometria.

Na etapa de predição, a futura posição e orientação do robô são estimadas por meio de um modelo cinemático não linear, utilizando o deslocamento linear e a orientação fornecidos pela odometria.

Na etapa de atualização, as medições de posição e orientação provenientes do INS e da odometria são reunidas em um único vetor de observação. Em seguida, calcula-se o Ganho de Kalman, que define a contribuição de cada técnica de acordo com suas incertezas, permitindo corrigir a estimativa e reduzir o erro acumulado ao longo do tempo.

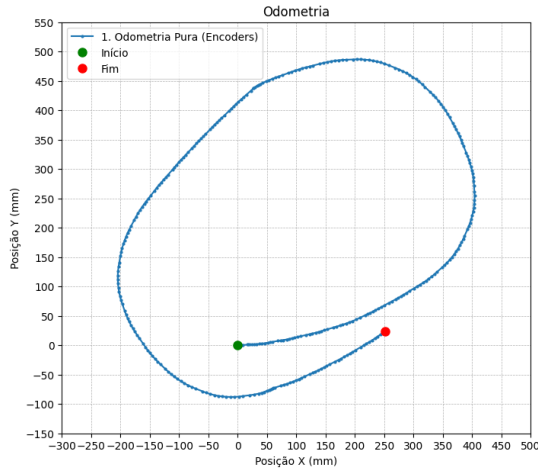
4 Resultados

Nesta seção, apresentam-se os resultados obtidos a partir da implementação dos três métodos de estimativa de trajetória: Odometria, Navegação Inercial (INS) e a combinação das técnicas via Filtro de Kalman Estendido (EKF).

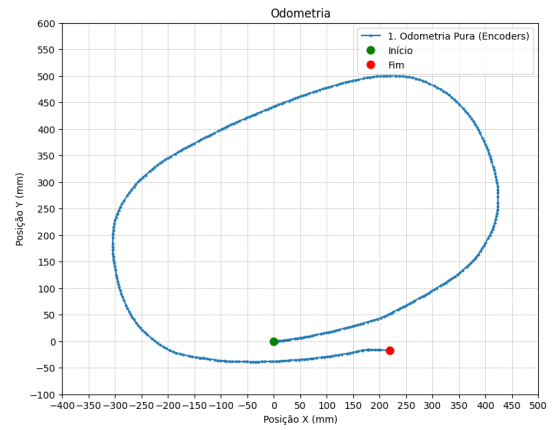
4.1 Resultados odometria

A trajetória estimada pelos *encoders* dos motores, por meio da odometria, demonstra capacidade de preservar, de forma aproximada, o formato geométrico do percurso conforme é possível observar na [Figura 14](#). Entretanto, observa-se um deslocamento lateral progressivo acompanhado de rotação do eixo principal da trajetória estimada. Como consequência, na representação computacional, o robô dificilmente retorna ao ponto inicial, ainda que fisicamente tenha passado novamente por essa posição.

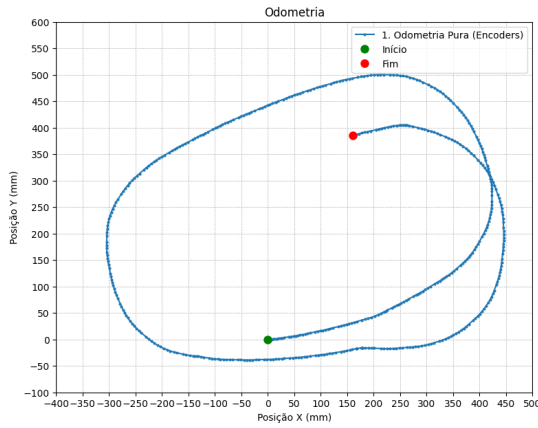
Esse comportamento está diretamente relacionado ao caráter cumulativo do erro na estimação. Como a orientação é calculada a partir da diferença de deslocamento entre as rodas, um pequeno erro angular nos instantes iniciais propaga-se para todas as estimativas subsequentes de posição. O resultado prático é um mapa que aparece rotacionado em relação ao referencial global, evidenciando a elevada sensibilidade da odometria a erros cumulativos de orientação. Esse efeito pode ser observado na [Figura 14](#) (d) na qual, mesmo com o robô percorrendo repetidamente a mesma trajetória, o mapeamento reconstruído apresenta rotação progressiva em relação a orientação original.



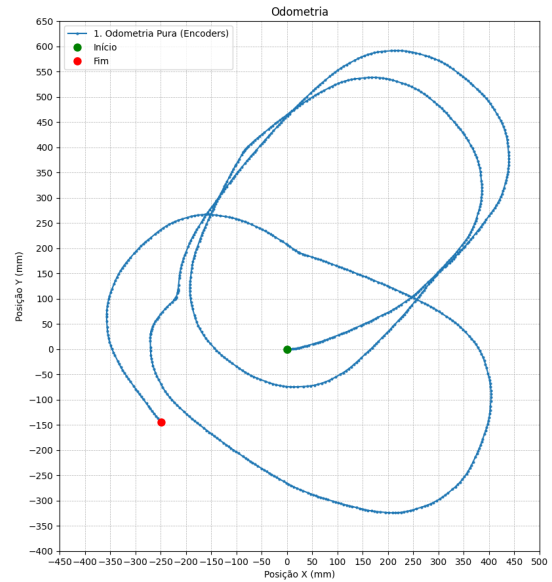
(a) Teste 1



(b) Teste 2



(c) Teste 3



(d) Teste 4

Figura 14 – Reconstrução da trajetória utilizando odometria. Fonte: Autor.

Essa divergência é inerente ao método de odometria, conforme discutido por [J. Borenstein e Feng \(1996\)](#). Segundo os autores, existem diversas razões para imprecisões na conversão das leituras dos *encoders* em deslocamento linear, às quais podem ser classificadas em erros sistemáticos e erros não sistemáticos.

Os erros sistemáticos estão associados a imperfeições estruturais do robô, como variações nos raios das rodas, diferença entre a distância medida entre as rodas e a distância real e desalinhamentos entre rodas. Conforme destacado pelos autores, esses erros são particularmente críticos porque se acumulam continuamente ao longo do deslocamento, provocando desvios progressivos na orientação e na posição estimada.

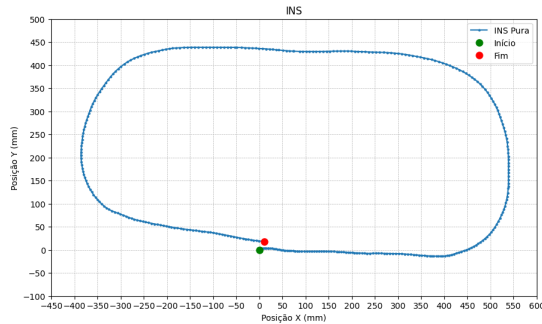
Já os erros não sistemáticos estão relacionados a fatores externos e imprevisíveis, como deslocamento sobre pisos irregulares ou escorregamento das rodas. Embora não

apresentem um padrão acumulativo constante como os erros sistemáticos, podem ocorrer de forma inesperada e provocar variações significativas na estimativa de posição (J. BORENSTEIN; FENG, 1996).

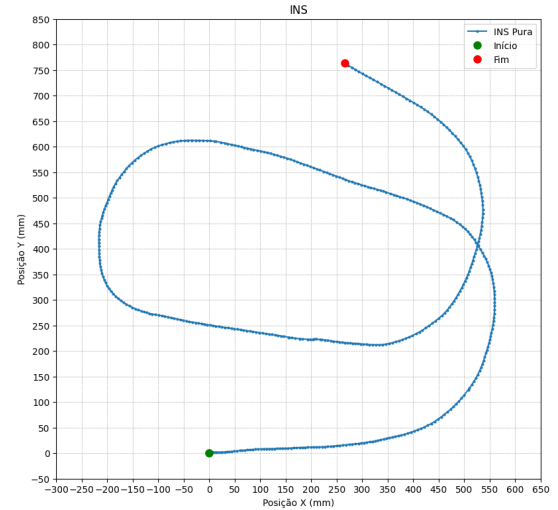
4.2 Resultados navegação inercial

A estimativa de trajetória realizada pelo Sistema de Navegação Inercial (INS) apresentou maior divergência em relação ao tamanho real da pista. Embora o sensor forneça dados em alta frequência, as medições de taxa angular e de aceleração linear precisam ser integradas uma vez para a obtenção da orientação e duas vezes para a estimativa da posição, respectivamente. Esse processo de integração sucessiva torna o método altamente sensível a pequenas imprecisões nas medições. Assim, mesmo erros muito pequenos nas taxas medidas podem resultar em crescimento progressivo e não limitado do erro nas variáveis integradas (J. BORENSTEIN; FENG, 1996).

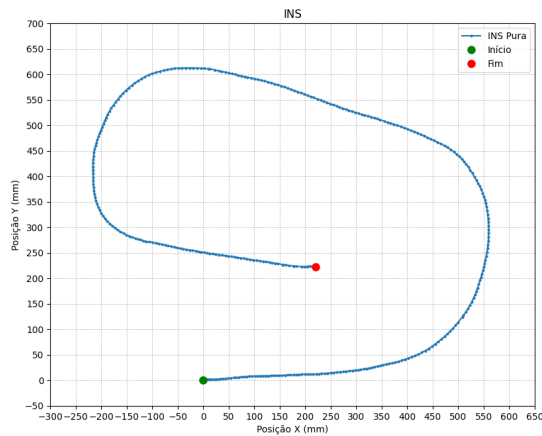
Como observado na Figura 15, a trajetória estimada preserva parcialmente a geometria do percurso, especialmente no Teste 1, no qual o formato geral da pista foi bem representado. No entanto, observa-se que a trajetória reconstruída apresentou alterações nas dimensões do trajeto estimado. Esse comportamento é consistente com o erro acumulado característico da navegação inercial, no qual pequenas imprecisões nas medições de aceleração e velocidade angular se propagam ao longo do processo de integração, resultando em desvios progressivos na estimativa de posição.



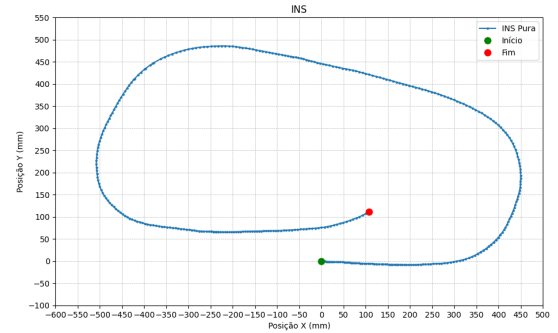
(a) Teste 1



(b) Teste 2



(c) Teste 3



(d) Teste 4

Figura 15 – Reconstrução da trajetória da pista utilizando INS. Fonte: Autor.

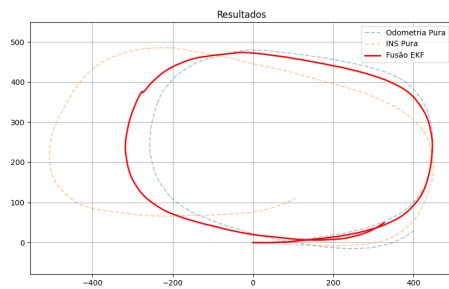
4.3 Resultados Filtro de Kalman Estendido

A trajetória estimada utilizando o Filtro de Kalman Estendido (EKF) apresentou desempenho superior em relação às estimativas obtidas isoladamente pela odometria e pela navegação inercial. A combinação das medições permitiu reduzir parte dos efeitos individuais observados anteriormente, como o erro acumulativo de orientação da odometria e o crescimento acelerado do erro da INS ao longo do tempo.

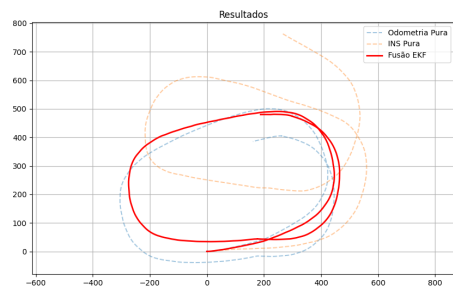
Conforme discutido [Barshan e Durrant-Whyte \(1993\)](#), a utilização de sensores inerciais em conjunto com outros mecanismos de sensoriamento é uma estratégia eficaz para mitigar problemas de deriva e instabilidade inerentes aos métodos individuais. Nesse contexto, o EKF pondera as incertezas associadas a cada fonte de medição e produz uma estimativa mais consistente do estado do robô.

Como observa-se na [Figura 15](#), a trajetória estimada pelo EKF mantém maior

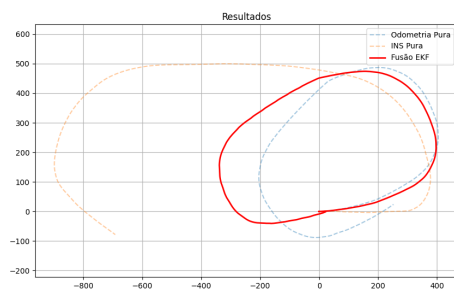
coerência geométrica com o percurso real quando comparada às demais abordagens. Ainda assim, permanecem discrepâncias visíveis entre o trajeto físico e o mapeamento estimado. Essa diferença pode ser observada por meio da comparação visual entre a forma e as dimensões da trajetória reconstruída e do percurso real, indicando que, embora o EKF tenha melhorado o desempenho do sistema, ainda existem desvios na estimativa da posição.



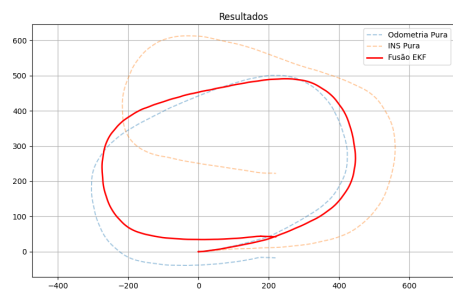
(a) Teste 1



(b) Teste 2



(c) Teste 3



(d) Teste 4

Figura 16 – Reconstrução da trajetória da pista por meio do Filtro de Kalman Estendido (EKF). Fonte: Autor.

5 Considerações finais

O presente trabalho mostrou a importância da estimativa de trajetória em sistemas de robótica móvel. A análise comparativa entre os métodos demonstrou que abordagens baseadas em apenas um tipo de sensor apresentam limitações inerentes. A Navegação Inercial mostrou-se fortemente suscetível ao erro acumulado decorrente da integração sucessiva das medições, enquanto a odometria falhou em manter a orientação global devido ao acúmulo contínuo de erros nos *encoders*.

A aplicação do Filtro de Kalman Estendido (EKF) demonstrou que a combinação de diferentes métodos de navegação é capaz de reduzir parte dessas limitações, produzindo estimativas mais coerentes com o percurso real quando comparadas aos métodos individuais. Ainda assim, os resultados obtidos indicam que o sistema permanece sujeito ao crescimento gradual da incerteza, característica inerente a métodos de navegação estimada. Nesse contexto, a literatura aponta que a incorporação de referências absolutas, como sistemas baseados em marcos ([BEZERRA, 2004](#)), pode contribuir para a limitação do erro acumulado.

No contexto específico do robô desenvolvido, vale destacar que a modelagem adotada na odometria considerou a aproximação de um robô diferencial. Entretanto, o chassi utilizado pertence à classe *skid-steer*, cuja dinâmica difere significativamente desse modelo idealizado. Conforme discutido por ([RABIEE; BISWAS, 2019](#)), veículos *skid-steer* apresentam deslizamento e derrapagem inerentes à sua configuração construtiva, o que introduz incertezas adicionais não contempladas pelo modelo cinemático diferencial clássico. Dessa forma, uma melhoria relevante consiste na reformulação da modelagem cinemática, de modo a torná-la mais representativa ao comportamento real do robô.

Dessa forma, conclui-se que o trabalho atingiu seu objetivo ao investigar e implementar métodos de mapeamento de trajetória para um robô seguidor de linha a partir de dados de odometria e sensores inerciais, permitindo a reconstrução gráfica do percurso realizado. Os resultados experimentais evidenciaram as limitações dos métodos individuais de navegação e os ganhos obtidos com a combinação dessas informações por meio do Filtro de Kalman Estendido. Ainda assim, os resultados indicam que melhorias adicionais na modelagem do robô e nas estratégias de correção de erros podem melhorar o desempenho do sistema.

Referências

AHMED. *Wheel Odometry Model for Differential Drive Robotics*. Mar. 2023. <https://medium.com/@nahmed3536/wheel-odometry-model-for-differential-drive-robotics-91b85a012299>. Citado 2 vezes na página 24.

AKM SEMICONDUCTOR. *Principle and advantages of magnetic encoder — Tutorials*. [s.d.] Disponível em: <https://www.akm.com/us/en/products/rotation-angle-sensor/tutorial/magnetic-encoder/>. Acesso em: 15 jan. 2026. Citado 3 vezes nas páginas 22, 23.

ALMEIDA, Fernanda. *O que é Encoder? Para que serve, como escolher e como interfacear?* Artigo publicado no blog da HI Tecnologia abordando o funcionamento, aplicações e critérios de escolha para encoders. HI Tecnologia. Nov. 2016. Disponível em: <https://materiais.hitecnologia.com.br/blog/o-que-%C3%A9-encoder-para-que-serve-como-escolher-como-interfacear/>. Citado 1 vez na página 21.

BARSHAN, B.; DURRANT-WHYTE, H.F. An Inertial Navigation System for a Mobile Robot. *IFAC Proceedings Volumes*, v. 26, n. 1, p. 54–59, 1993. 1st IFAC International Workshop on Intelligent Autonomous Vehicles, Hampshire, UK, 18-21 April. ISSN 1474-6670. DOI: [https://doi.org/10.1016/S1474-6670\(17\)49275-5](https://doi.org/10.1016/S1474-6670(17)49275-5). Disponível em: <https://www.sciencedirect.com/science/article/pii/S1474667017492755>. Citado 3 vezes nas páginas 24, 26, 41.

BECKER, Alex. *Kalman Filter from the Ground Up*. Accessed: 2026-01-25. 2024. Disponível em: <https://kalmanfilter.net/>. Citado 1 vez na página 26.

BEZERRA, Clauber Gomes. *Localização de um robô móvel usando odometria e marcos naturais*. 2004. f. 122. Mestrado em Automação e Sistemas; Engenharia de Computação; Telecomunicações – Universidade Federal do Rio Grande do Norte, Natal. Citado 2 vezes nas páginas 23, 43.

CARVALHO CASTRO PENIDO, Édilus de; TRINDADE, Ronaldo Silva. *Microcontroladores*. Ouro Preto, Santa Maria: Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais; Universidade Federal de Santa Maria, Colégio Técnico Industrial de Santa Maria; Rede e-Tec Brasil, 2013. p. 80. ISBN 978-85-86473-12-8. Citado 1 vez na página 29.

CHEN, Weihua; ZHANG, Tie. An indoor mobile robot navigation technique using odometry and electronic compass. *International Journal of Advanced Robotic Systems*, v. 14, n. 3, p. 1729881417711643, 2017. DOI: [10.1177/1729881417711643](https://doi.org/10.1177/1729881417711643). eprint: <https://doi.org/10.1177/1729881417711643>. Disponível em: <https://doi.org/10.1177/1729881417711643>. Citado 2 vezes nas páginas 12, 23.

DENARDIN, G.W.; BARRIQUELLO, C.H. *Sistemas operacionais de tempo real e sua aplicação em sistemas embarcados*. Editora Blucher, 2019. ISBN 9788521213970. Disponível em: <https://books.google.com.br/books?id=FrqxDwAAQBAJ>. Citado 1 vez na página 29.

DYNAPAR ENCODERS. *Tipos de encoders: o que são, como funcionam e qual escolher*. Mar. 2019. Disponível em: <https://dynaparencoders.com.br/diferenca-encoder-incremental-absoluto/>. Citado 1 vez na página 21.

EVERETT, H. R. *Sensors for Mobile Robots: Theory and Application*. A K Peters, Ltd., 1995. ISBN 1-56881-048-2. Citado 2 vezes na página 17.

GANGANATH, N.; LEUNG, H. Mobile robot localization using odometry and Kinect sensor. *In: IEEE International Conference on Emerging Signal Processing Applications, ESPA 2012*. IEEE, 2012. p. 91–94. Citado 1 vez na página 23.

GIBSON, Chris *et al.* Demonstrating Practical Inertial Navigation: The Beginnings and Beyond. *In: AMERICAN INSTITUTE OF AERONAUTICS e ASTRONAUTICS (AIAA). PROCEEDINGS of the AIAA Guidance, Navigation, and Control Conference*. Boston, MA: AIAA, ago. 2013. The Charles Stark Draper Laboratory, Inc. DOI: [10.2514/6.2013-5122](https://doi.org/10.2514/6.2013-5122). Disponível em: <https://arc.aiaa.org/doi/10.2514/6.2013-5122>. Citado 7 vezes nas páginas 11–13.

GOEL, Ankit *et al.* *An Introduction to Inertial Navigation From the Perspective of State Estimation*. 2022. Acesso em: 19 jan. 2026. Disponível em: <https://dsbaero.engin.umich.edu/wp-content/uploads/sites/441/2022/01/InertialNavigationTutorialCSM.pdf>. Citado 4 vezes na página 11.

HIVEMQ. *MQTT Essentials: The Ultimate Guide to the MQTT Protocol for IoT Messaging*. 2024. <https://www.hivemq.com/mqtt/>. Accessed: 20 Feb. 2026. Citado 6 vezes nas páginas 30, 31.

INTERNATIONAL CIVIL AVIATION ORGANIZATION. *Annex 10 to the Convention of International Civil Aviation: Volume I: Radio Navigation Aids*. Montreal, PQ, Canada, jul. 2007. Amendment 82. Citado 1 vez na página 12.

J. BORENSTEIN, H. R. Everett; FENG, L. *Where am I? Sensors and Methods for Mobile Robot Positioning*. Edição: J. Borenstein. Oak Ridge National Lab (ORNL) D&D Program e United States Department of Energy's Robotics Technology Development Program, abr. 1996. Citado 10 vezes nas páginas 12, 20–23, 39, 40.

KERSCHBAUMER, Ricardo *et al.* *Microcontroladores*. Santa Catarina, Brasil, 2013. Citado 1 vez na página 29.

KERSCHBAUMER, Ricardo. *Microcontroladores*. INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA CATARINENSE, 2025. Citado 1 vez na página 29.

MARTINS, Lucas José Nunes. *Construção de um robô seguidor de linha com enfoque em desempenho e coleta de dados inerciais*. Dez. 2024. Trabalho de Diploma – Instituto de Engenharia de Sistemas e Tecnologia da Informação (IESTI), Itajubá. Citado 3 vezes nas páginas 14, 15, 19, 32.

MASOOLEH, Leila Samandari *et al.* Distributed state estimation in large-scale processes decomposed into observable subsystems using community detection. *Computers & Chemical Engineering*, v. 156, p. 107544, 2022. ISSN 0098-1354. DOI: <https://doi.org/10.1016/j.compchemeng.2021.107544>. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0098135421003227>. Citado 2 vezes na página 27.

MATHWORKS. *What Is a Kalman Filter?* Accessed: 2026-01-25. 2025. Disponível em: <https://www.mathworks.com/discovery/kalman-filter.html>. Citado 1 vez na página 27.

MATLAB. *Understanding Kalman Filters [YouTube Playlist]*. 2017. YouTube. Playlist de vídeos sobre filtro de Kalman. Disponível em: <https://www.youtube.com/playlist?list=PLn8PRpmsu08pzi6EMiYnR-076Mh-q3tWr>. Citado 3 vezes nas páginas 26, 27.

MOHAMED, Sherif A. S. *et al.* A Survey on Odometry for Autonomous Navigation Systems. *IEEE Access*, v. 7, p. 97466–97486, 2019. DOI: [10.1109/ACCESS.2019.2929133](https://doi.org/10.1109/ACCESS.2019.2929133). Citado 3 vezes nas páginas 11, 12.

NOURELDIN, Aboelmagd; KARAMAT, Tashfeen B.; GEORGY, Jacques. *Fundamentals of Inertial Navigation, Satellite-based Positioning and their Integration*. 1. ed. Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2013. p. xviii, 314. ISBN 978-3-642-30466-8. DOI: [10.1007/978-3-642-30466-8](https://doi.org/10.1007/978-3-642-30466-8). Citado 2 vezes nas páginas 25, 26.

QAZIZADA, Mohammad Emal; PIVARČIOVÁ, Elena. Mobile Robot Controlling Possibilities of Inertial Navigation System. *Procedia Engineering*, v. 149, p. 431–436, 2016. DOI: [10.1016/j.proeng.2016.06.729](https://doi.org/10.1016/j.proeng.2016.06.729). Citado 3 vezes nas páginas 19, 24.

QUINCOZES, Silvio; EMILIO, Tubino; KAZIENKO, Juliano. MQTT Protocol: Fundamentals, Tools and Future Directions. *IEEE Latin America Transactions*, v. 17, n. 09, p. 1439–1448, 2019. DOI: [10.1109/TLA.2019.8931137](https://doi.org/10.1109/TLA.2019.8931137). Citado 1 vez na página 31.

RABIEE, Sadegh; BISWAS, Joydeep. A Friction-Based Kinematic Model for Skid-Steer Wheeled Mobile Robots. *In: 2019 International Conference on Robotics and Automation (ICRA)*. 2019. p. 8563–8569. DOI: [10.1109/ICRA.2019.8794216](https://doi.org/10.1109/ICRA.2019.8794216). Citado 1 vez na página 43.

RIBEIRO, Daniel Veloso. *Robô Seguidor de Linha para Mapeamento de Ambientes Internos*. 2017. Monografia (Graduação) – Engenharia Elétrica – Universidade Federal do Maranhão, São Luís. Citado 1 vez na página 12.

SIEGWART, Roland; NOURBAKHSH, Illah R. Introduction to Autonomous Mobile Robots. *In: A Bradford Book*. 1st. London, England: The MIT Press, 2004. Citado 3 vezes nas páginas 17, 18.

STUHLER, Jackson. Utilização de MEMS (Micro-Electro Mechanical Systems – sistemas microeletromecânicos) para aplicações de circuitos de baixo custo de movimentação e sensoriamento. *Revista Metodologias e Aprendizado*, 2022. Citado 1 vez na página 20.

SUSNJARA, Stephanie; SMALLEY, Ian. *O que é um Sistema Operacional em Tempo Real (RTOS)?* 2025. <https://www.ibm.com/br-pt/think/topics/real-time-operating-system>. Citado 3 vezes nas páginas 29, 30.

TEXAS INSTRUMENTS. *High Resolution Low Latency Compact Absolute Angle ...* 2024. Disponível em: <https://www.ti.com/lit/pdf/tiduf97>. Acesso em: 15 jan. 2026. Citado 1 vez na página 22.

TORRES, Henrique. *Sensores Inerciais – Parte 2*. Acesso em: 15 mar. 2025. 2015. Disponível em: <https://embarcados.com.br/sensores-inerciais-parte-2/>. Citado 2 vezes nas páginas 19, 20.