



Trabalho de Conclusão de Curso

Estimação de estado usando fusão de sensores para um robô holonômico

de Eduardo Henrique Farias Silva

orientado por

Prof. Doutor Ícaro Bezerra Queiroz de Araújo

Prof. Mestre Glauber Rodrigues Leite

Universidade Federal de Alagoas
Instituto de Computação
Maceió, Alagoas
10 de Agosto de 2023

UNIVERSIDADE FEDERAL DE ALAGOAS
Instituto de Computação

ESTIMAÇÃO DE ESTADO USANDO FUSÃO DE SENSORES PARA UM ROBÔ HOLONÔMICO

Trabalho de Conclusão de Curso submetido
ao Instituto de Computação da Universidade
Federal de Alagoas como requisito parcial
para a obtenção do grau de Engenheiro de
Computação.

Eduardo Henrique Farias Silva

Orientador: Prof. Doutor Ícaro Bezerra Queiroz de Araújo

Coorientador: Prof. Mestre Glauber Rodrigues Leite

Banca Avaliadora:

Andressa Martins Oliveira Profa. MSc.

Tiago Alves de Almeida Prof. Doutor

Maceió, Alagoas
10 de Agosto de 2023

Catálogo na fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico
Bibliotecária: Taciana Sousa dos Santos – CRB-4 – 2062

S586e Silva, Eduardo Henrique Farias.
Estimação de estado usando fusão de sensores para um robô holonômico
/ Eduardo Henrique Farias Silva. – 2023.
44 f. : il. color.

Orientador: Ícaro Bezerra Queiroz de Araújo.
Coorientador: Glauber Rodrigues Leite.
Monografia (Trabalho de Conclusão de Curso em Engenharia da
Computação) – Universidade Federal de Alagoas. Instituto de Computação.
Maceió, 2023.

Bibliografia: f. 43-44.

1. Robótica móvel. 2. Filtro de Kalman. 3. Sensor inercial. I. Título.

CDU: 007.52

UNIVERSIDADE FEDERAL DE ALAGOAS
Instituto de Computação

ESTIMAÇÃO DE ESTADO USANDO FUSÃO DE SENSORES PARA UM ROBÔ HOLONÔMICO

Trabalho de Conclusão de Curso submetido ao Instituto de Computação da Universidade Federal de Alagoas como requisito parcial para a obtenção do grau de Engenheiro de Computação.

Aprovado em 10 de Agosto de 2023:

Ícaro Bezerra Queiroz de Araújo,
Prof. Doutor, Orientador

Glauber Rodrigues Leite,
Prof. Mestre, Coorientador

Andressa Martins Oliveira,
Profa. MSc., Avaliador

Tiago Alves de Almeida,
Prof. Doutor

Agradecimentos

Agradeço, primeiramente, aos meus pais, por me ensinarem os princípios da vida e por terem sempre acreditado e me apoiado, sempre trabalhando para que eu pudesse estudar.

A minha irmã que desde sempre foi minha parceira, soube ouvir minhas frustrações e que também acabou por seguir o meu caminho, ingressando no curso de Engenharia de Computação.

Aos meus colegas do Instituto de Computação, que durante toda a graduação me incentivaram e permitiram que essa jornada fosse mais divertida.

Por fim, mas não menos importante, agradeço aos professores Glauber Rodrigues e Ícaro Araújo, que não somente são excelentes profissionais mas também ser humanos incríveis, se disponibilizaram a atuar como orientadores em uma das etapas mais importantes da minha vida.

01 de Julho de 2023, Maceió, Alagoas

Eduardo Henrique Farias Silva

Resumo

Com a chegada da Revolução Industrial 4.0, a robótica se tornou um mercado mais popular e desejado. Robôs articulados, veículos guiados automaticamente (*AGV*), humanóides, robôs móveis autônomos (*AMR*) entre outros; têm sido alvo de grandes estudos com objetivo de realizar trabalhos repetitivos, complicados ou que colocariam em risco a vida humana. Na robótica móvel, cujo objetivo é permitir ao robô sua locomoção em um ambiente de trabalho, desafios surgem no domínio da navegação. Este trabalho apresenta uma modelagem para o problema de rastreamento de posição para robôs móveis utilizando fusão de sensores, proprioceptivos e exteroceptivos, por meio do filtro de Kalman. A modelagem utiliza como sensores uma unidade de medida inercial (*IMU*) e uma câmera, ambos de baixo custo, para realizar os processos do filtro, e sua implementação é feita utilizando o *ROS*, visando a integração em qualquer dispositivo que seja compatível.

Palavras-chave: Robô móvel; Estimação de Estado; Filtro de Kalman.

Abstract

With the arrival of Fourth Industrial Revolution, robotics has become a more popular and desired market. Articulated robots, automatically guided vehicles (AGV), humanoids, autonomous mobile robots (AMR), and others; have been the subject of large studies with the aim of carrying out repetitive, complicated work or work that would endanger human life. In mobile robotics, whose objective is to allow the robot to move around in a work environment, challenges arise in the field of navigation. This work presents a modeling for the position tracking problem for mobile robots using sensor fusion, proprioceptive and exteroceptive sensors, through the Kalman filter. The modeling uses an inertial measurement unit (IMU) and a camera as sensors, both of low cost, to carry out the filter processes, and its implementation is done using ROS, aiming at integration in any device that is compatible.

Keywords: Mobile robot; State Estimation; Kalman filter.

Lista de Figuras

2.1	Áreas do conhecimento envolvidas na robótica móvel.	14
2.2	Tipos de rodas. (a) Roda fixa. (b) Roda giratória. (c) Roda sueca 90. (d) Roda sueca 45.	16
2.3	Robô omnidirecional Omnino	17
2.4	Modelo cinemático para robô holonômico com três rodas.	19
2.5	Pesquisas em problemas de localização.	21
2.6	Localização absoluta	24
2.7	Exemplo do Filtro de Kalman	25
3.1	Módulo GY-521.	29
3.2	Módulo Pi Camera.	30
3.3	Processos para detecção da tag ArUco. (a) Imagem original. (b) Resultado da segmentação. (c) Detecção de contorno. (d) Aproximação poligonal e remoção de contornos irrelevantes. (e) Exemplo de marcador após transformação de perspectiva. (f) Atribuição de bit.	31
3.4	Trajeto linear (em metros)	34
3.5	Trajeto em L (em metros)	35
3.6	Trajeto Lemniscata de Bernoulli (em metros)	35
4.1	Leituras da IMU e câmera durante o deslocamento em linha reta	36
4.2	Leituras da IMU e câmera durante o deslocamento em L	37
4.3	Leituras da IMU e câmera deslocamento seguindo a Lemniscata de Bernoulli	38
4.4	Acelerações no eixo x e y captadas da IMU MPU-6050 durante o percurso em L.	39
4.5	Comparação entre IMU, câmera e aplicação do filtro	40
4.6	Aplicação do filtro de Kalman utilizando somente a IMU	41

Lista de Abreviaturas

ADC	Analog-to-Digital Converter
AGV	Automated Guided Vehicle
AMR	Autonomous Mobile Robot
ASV	Autonomous Surface Vehicle
AUV	Autonomous Underwater Vehicle
DMP	Digital Motion Processor
DOF	Degree Of Freedom
FPS	Frame Per Second
IMU	Inertial Measurement Unit
IC	Instituto de Computação.
LIDAR	Light Detection and Ranging.
ROS	Robot Operating System.
SLAM	Simultaneous Localization and Mapping.
UAV	Unmanned Aerial Vehicle

Sumário

1	Introdução	10
1.1	Motivação	11
1.2	Objetivos	11
1.2.1	Objetivo geral	11
1.2.2	Objetivos específicos	12
1.3	Organização do texto	12
2	Definição	13
2.1	Robótica móvel	13
2.1.1	Robôs aéreos	14
2.1.2	Robôs aquáticos	15
2.1.3	Robôs terrestres	15
2.2	Omnino	17
2.2.1	Plataforma móvel	17
2.2.2	Modelo cinemático	18
2.3	Localização	20
2.3.1	Problemas de localização	21
2.3.2	Técnicas de localização	22
2.4	Filtro de Kalman	24
3	Arquitetura proposta	28
3.1	Descrição	28
3.2	Metodologia	28
3.2.1	Unidade de medida inercial	28
3.2.2	Câmera	29
3.2.3	Modelagem	31
3.3	Cenários	34
3.3.1	Cenário 1	34
3.3.2	Cenário 2	35
3.3.3	Cenário 3	35

4	Resultados	36
4.1	Cenário 1 - Percurso linear	36
4.2	Cenário 2 - Percurso em L	37
4.3	Cenário 3 - Percurso Lemniscata de Bernoulli	37
4.4	Discussões	38
5	Considerações finais	42
	Bibliografia	43

Capítulo 1

Introdução

Desde a introdução dos robôs industriais, o avanço em pesquisas, a popularização e barateamento da tecnologia e, principalmente, as novas necessidades e mercados tornaram mais comum a presença de robôs no cotidiano, seja para realização de tarefas perigosas e exaustivas, explorações, conforto, lazer entre outros [Garcia et al. (2007)].

Com a modernização, tem sido exigido novos serviços que implicam na maneira em que os robôs são vistos pela indústria e o ponto de vista social e pessoal. Robôs manipuladores como os industriais e médicos, robôs móveis e robôs biologicamente inspirados, apesar de serem modelados por estruturas diferentes, buscam ser um sistema adaptativo e inteligente com objetivo de melhor prestar o seu serviço, independentemente da área de atuação [Garcia et al. (2007)].

Os robôs podem ser classificados com base em sua locomoção. Os robôs de primeira geração são fixos e não podem se mover pela fábrica. Em contraste, robôs móveis podem se mover pelo mundo usando várias formas de mobilidade. Outra classificação, é de acordo com a função que o robô executa. Os robôs de manufatura operam em fábricas e são os sucessores tecnológicos dos robôs de primeira geração. Os robôs de serviço trabalham com limpeza, cuidados pessoais, reabilitação médica ou busca e transporte. Robôs de campo trabalham ao ar livre em tarefas como monitoramento ambiental, agricultura, mineração e construção. Os humanóides têm a forma física de um ser humano, operam como robôs móveis e de serviço [Corke (2011)].

Apesar de todo o sucesso dos manipuladores, esses robôs comerciais sofrem de uma desvantagem fundamental: falta de mobilidade. Um manipulador fixo tem uma amplitude de movimento limitada que depende de onde ele é fixado. Essa desvantagem se reflete no aumento de interesse nas áreas relacionadas com manipulações em ambientes não estruturados e em robôs móveis [Roland Siegwart and Scaramuzza (2011)].

Conforme descrito por Siciliano et al. (2009), o termo robô móvel descreve um sistema robótico capaz de realizar tarefas em diferentes lugares e consistindo de uma plataforma movida por elementos locomotivos. Essa mobilidade fornece aos robôs maior capacidade operacional e abre novas áreas de investigação. Algumas dessas áreas são comuns a todos

os robôs móveis, como exemplo o problema de navegação.

1.1 Motivação

De acordo com Roland Siegwart and Scaramuzza (2011), um dos principais pilares da robótica móvel é a navegação. Para uma navegação bem sucedida, o robô passa por diferentes etapas: percepção, localização, cognição e controle de movimento. Dessas etapas, a localização ainda é a área mais estudada e conseqüentemente a que mais possui avanços.

Os problemas de localização baseiam na informação da posição inicial de um robô em um momento arbitrário e se classificam em três tipos principais: rastreamento de posição, localização/posicionamento global e problema do robô sequestrado [Panigrahi and Bisoy (2022)].

A solução mais simples para os problemas de localização seria a instalação de um GPS no robô, já que a todo momento teríamos a informação precisa de posicionamento e orientação em relação ao sistema observado. Entretanto, essa tecnologia nem sempre está disponível, como por exemplo carros autônomos que perdem conexão com o GPS quando em ambientes fechados, túneis, entre edifícios altos e próximos (causando oclusão) entre outros. Ainda, para melhorar a precisão da localização de uma posição, mais satélites devem ser adquiridos e rastreados pelo receptor. Entretanto, o aumento de canais utilizados para rastrear os satélites, no receptor, resulta em um maior consumo de energia, este que é limitado pelas baterias de um dispositivo móvel, e, conseqüentemente, aquecimento do sistema. Logo, a implantação do GPS em um robô móvel é mais custosa, visto que é preciso um sistema mais robusto para alcançar precisões superiores [Rahmatullah et al. (2017)].

Para contornar esse problema, outros sensores e técnicas de localização são empregados no intuito de rastrear a posição do robô. Técnicas de localização relativa fazem uso de sensores (proprioceptivos) como encoders e IMU para encontrar a posição em um deslocamento curto, para evitar erros. Enquanto que a localização absoluta utiliza sensores (exteroceptivos) para localizar o robô a partir de fontes externas [Panigrahi and Bisoy (2022)].

1.2 Objetivos

1.2.1 Objetivo geral

A finalidade desse trabalho é analisar o comportamento e precisão de um sensor inercial de baixo custo ao estimar a posição de um robô omnidirecional durante seu percurso em diferentes trajetórias e o impacto que é causado à estimação quando há o auxílio de uma câmera por meio de fusão de sensores utilizando o Filtro de Kalman.

1.2.2 Objetivos específicos

- Integrar uma IMU de baixo custo ao robô Omnino;
- Rastrear as posições do robô usando a informação da IMU;
- Rastrear as posições do robô usando a informação da câmera;
- Aplicar o Filtro de Kalman, com a IMU na etapa de previsão e a câmera na etapa de correção;
- Análise dos rastreamentos e validação dos efeitos do filtro.

1.3 Organização do texto

O estudo está organizado em capítulos descrevendo cada etapa do desenvolvimento do trabalho, onde são fornecidas informações teóricas e a respeito da implementação da solução. No Capítulo 2, encontra-se a teoria e a definição do problema. O Capítulo 3 apresenta a proposta do trabalho bem como a metodologia abordada e as situações em que o robô atuará. No Capítulo 4 são discutidos os resultados obtidos com os cenários de simulação utilizados, de acordo com os objetivos propostos. Por fim, o Capítulo 5 conclui o trabalho, apontando dificuldades encontradas e trabalhos futuros.

Capítulo 2

Definição

2.1 Robótica móvel

Atualmente, a robótica móvel tem sido um dos campos de pesquisa científica com mais rápida expansão. Devido às suas habilidades de intervenção autônomas, os robôs móveis podem substituir os humanos em muitos campos [Siciliano et al. (2009)]. As aplicações incluem exploração planetária, patrulhamento, operações de resgate de emergência, aplicações petroquímicas, construção, entretenimento, serviços pessoais, intervenção em ambientes extremos, transporte, cuidados médicos entre outros [Francisco Rubio and Llopis-Albert (2019)], como visto na figura 2.1. Ainda assim, apesar de robôs móveis possuírem amplo conjunto de aplicações e diferentes mercados, um fato em comum entre eles é a necessidade do seu projeto envolver a integração de diversas áreas de conhecimento, tornando a robótica móvel uma área extremamente interdisciplinar [Roland Siegwart and Scaramuzza (2011)].

O termo robótica móvel descreve um sistema robótico capaz de realizar tarefas em diferentes locais [Garcia et al. (2007)] e sua principal característica é a presença de uma base móvel que permite ao robô movimentar-se livremente no ambiente. Segundo Siciliano et al. (2009), do ponto de vista mecânico, um robô autônomo móvel consiste de um ou mais corpos rígidos acoplado a um sistema de locomoção. A escolha do sistema locomotivo é importante pois não depende somente do ambiente (seja ele aéreo, terrestre ou aquático) em que o robô irá operar, mas também de alguns critérios técnicos, como estabilidade, eficiência, controlabilidade, condições do ambiente e assim por diante [Francisco Rubio and Llopis-Albert (2019)].

Um robô é autônomo quando possui a capacidade de determinar as ações a serem tomadas para realizar uma tarefa, utilizando um sistema de percepção que o auxilia e uma unidade de cognição ou um sistema de controle para coordenar todos os subsistemas que compõem o robô [Francisco Rubio and Llopis-Albert (2019)]. De acordo com Roland Siegwart and Scaramuzza (2011), os fundamentos da robótica móvel consistem nas áreas de

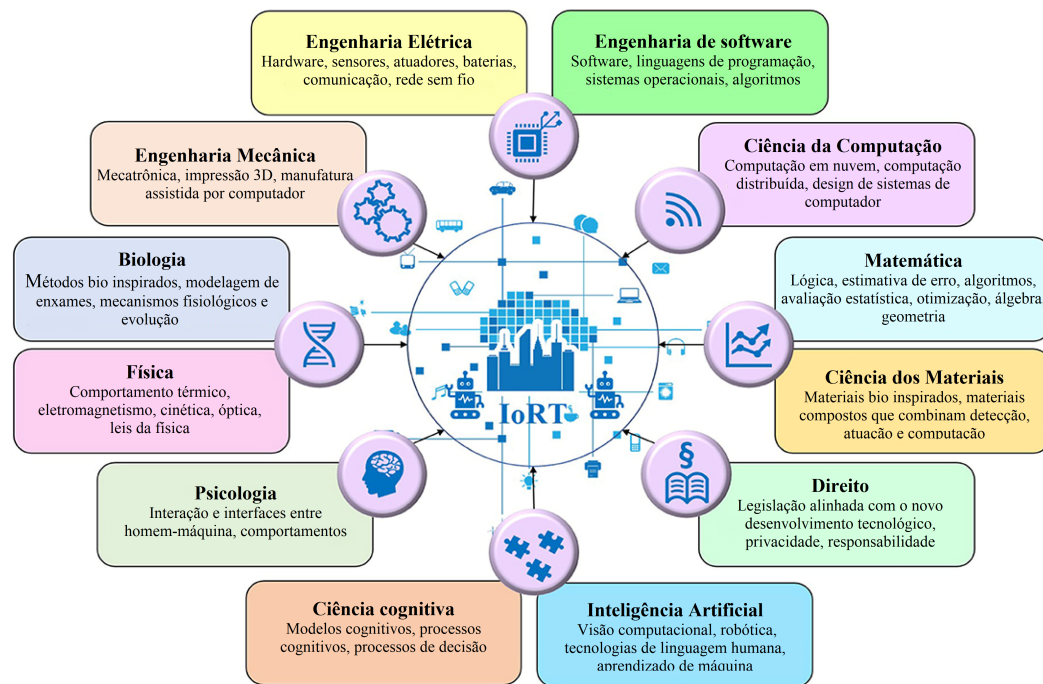


Figura 2.1:
Áreas do conhecimento envolvidas na robótica móvel.
Fonte: Adaptado de Raj and Kos (2022)

locomoção, percepção, cognição e navegação.

- Os problemas de locomoção são resolvidos pela compreensão do mecanismo e cinemática, dinâmica e teoria de controle.
- A percepção envolve as áreas de análise de sinal e campos especializados, como visão computacional e tecnologias de sensores.
- A cognição é responsável por analisar os dados de entrada dos sensores e tomar as ações correspondentes para atingir os objetivos do robô móvel. É responsável pelo esquema do sistema de controle.
- A navegação requer conhecimento de algoritmos de planejamento, teoria da informação e inteligência artificial.

Esses robôs móveis são muito semelhantes em termos do que fazem e como fazem [Corke (2011)], no entanto, se diferem na base móvel, cujo seu planejamento está sujeito ao ambiente e condições que estará atuando. A partir do seu sistema de locomoção, o robô pode ser classificado, dentre as principais categorias, como: aéreo; terrestre; ou aquático.

2.1.1 Robôs aéreos

Robôs voadores ou veículos aéreos não tripulados (UAV) tem se tornado mais populares e diversificados em tamanho e forma, segundo Corke (2011). Popularmente conhecido

como *drone*, desempenham tarefas pré-programadas com ou sem interação humana, com os modelos mais avançados podendo decolar e pousar independentemente das ações de um operador. Inicialmente direcionado as atividades militares, hoje encontra-se na agricultura, distribuição e logística, imagens aéreas, vigilância, além do aumento do uso pessoal [Francisco Rubio and Llopis-Albert (2019)].

2.1.2 Robôs aquáticos

Mais de 70% da terra é coberta por água e uma das mais antigas metas do homem tem sido a exploração dos oceanos e áreas submersas que lhe são inacessíveis [Wang et al. (2017)]. Durante os últimos anos, o uso de *autonomous surface vehicles* (ASVs) e *autonomous underwater vehicles* (AUV) tem aumentado rapidamente, desde que tais robôs podem operar em áreas mais profundas e perigosas, no qual mergulhadores teriam dificuldades em alcançar. Suas aplicações são encontradas, mas não limitadas, em pesca, monitoramento da poluição subaquática, limpeza e tratamento de resíduos no oceano [Garcia et al. (2007)].

2.1.3 Robôs terrestres

Robôs movidos por pernas

Pernas são outro tipo comum de locomoção, permitindo a criação de robôs que podem andar. Apesar de normalmente serem mais custosas quando comparadas com rodas, pernas acabam por possuir algumas vantagens. De acordo com Francisco Rubio and Llopis-Albert (2019), a maior vantagem das pernas sobre as rodas é a sua transversalidade, além de que é capaz de se deslocar em ambientes irregulares, possuir melhor mobilidade e estabilidade. Os robôs móveis com pernas são feitos de múltiplos corpos rígidos, conectados por juntas prismáticas ou juntas de revolução [Siciliano et al. (2009)]. Existe uma grande variedade de estruturas mecânicas para a classe de robô com pernas, essas que frequentemente são inspiradas pelo estudo de organismos vivos partindo de humanóides bípedes até insetos hexapódes.

Robôs movidos por rodas

Rodas são um dos mais importantes sistemas de locomoção para robôs. Seu uso é mais simples do que as pernas ou as *treads* e são mais fáceis de projetar, construir e programar quando o robô se move em um terreno plano, além do seu menor custo quando comparado os outros sistemas de locomoção terrestre. Outra vantagem do uso de rodas ocorre pelo contato direto que o corpo do robô terá com a superfície, evitando maiores problemas em termos de equilíbrio [Francisco Rubio and Llopis-Albert (2019)].

Embora muito utilizada, o uso de rodas implica em problemas de locomoção e odometria quando necessário navegar sobre obstáculos, terreno rochoso ou áreas com baixo atrito [Francisco Rubio and Llopis-Albert (2019)].

Existem quatro tipos básicos de rodas, segundo Roland Siegwart and Scaramuzza (2011):

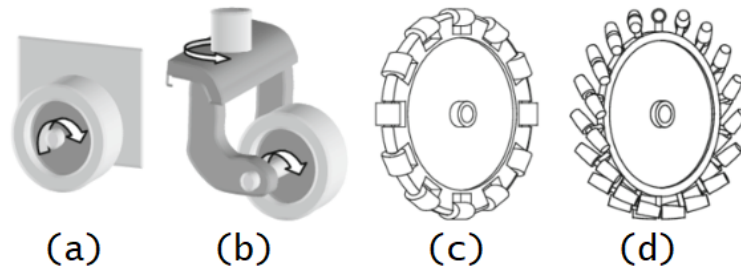


Figura 2.2:
Tipos de rodas. (a) Roda fixa. (b) Roda giratória.
(c) Roda sueca 90. (d) Roda sueca 45.
Fonte: Nava Rodriguez et al. (2011)

- I. Roda fixa padrão: Rodas convencionais com um único grau de liberdade (DOF) e rotação ao redor do ponto de contato [Francisco Rubio and Llopis-Albert (2019)]. Com a roda fixa ao chassi, a sua orientação é sempre constante [Siciliano et al. (2009)].
- II. Roda giratória (*Castor wheel*): Possui dois graus de liberdade e seu centro de rotação está deslocado por meio de uma junta rotacional, permitindo um giro de 360° [Francisco Rubio and Llopis-Albert (2019)]. Esse tipo de roda fornece um ponto de suporte para o equilíbrio estático sem afetar a mobilidade da base [Siciliano et al. (2009)]. Frequentemente usada em cadeiras e carrinhos de compras.
- III. Roda sueca: Contém três graus de liberdade [Francisco Rubio and Llopis-Albert (2019)]. A roda sueca funciona como uma roda normal, mas possui baixa resistência em outra direção, podendo ser perpendicular à direção convencional ou com um ângulo intermediário de acordo com a disposição que os rolos auxiliares são distribuídos na roda [Roland Siegwart and Scaramuzza (2011)].
- IV. Roda esférica: A roda esférica é uma roda verdadeiramente omnidirecional, frequentemente projetada para girar em qualquer direção. Sua implementação é complexa [Francisco Rubio and Llopis-Albert (2019)] e é realizada por meio de rolos, localizados na superfície superior da esfera, responsáveis pela transmissão da força rotacional [Roland Siegwart and Scaramuzza (2011)].

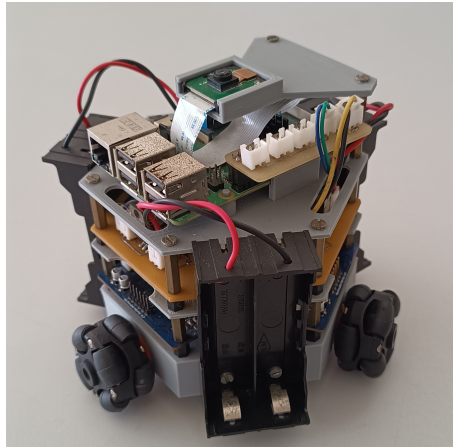


Figura 2.3:
Robô omnidirecional Omnino
Fonte: Ramalho (2020)

2.2 Omnino

O Omnino, figura 2.3, é um robô móvel omnidirecional autônomo voltado para atividades de pesquisa em enxame de agentes inteligentes e educacionais, projetado para ser de baixo custo e com acesso aberto ao software e hardware [Ramalho (2020)].

2.2.1 Plataforma móvel

A escolha dos tipos das rodas para um robô móvel está fortemente conectada com a escolha do arranjo ou da geometria das rodas [Roland Siegwart and Scaramuzza (2011)]. A partir dessa escolha, três características fundamentais são observadas: estabilidade; manobrabilidade; e controlabilidade.

Estabilidade

Segundo Roland Siegwart and Scaramuzza (2011), duas rodas são o número mínimo para a estabilidade estática. Contudo, em circunstâncias normais, tal solução requer diâmetros de roda impraticáveis.

A estabilidade pode ser melhorada adicionando mais rodas, embora uma vez que o números de pontos de contato seja maior que três, será necessário um sistema de suspensão flexível para lidar com terrenos irregulares [Roland Siegwart and Scaramuzza (2011)].

Manobrabilidade

Alguns robôs são omnidirecionais, ou seja, podem se mover em qualquer direção a qualquer momento independentemente de sua orientação. Para alcançar esse nível de manobrabilidade se faz imprescindível o uso de rodas que possam se mover em mais de uma

direção, neste caso os robôs frequentemente são equipados com rodas suecas ou esféricas [Roland Siegwart and Scaramuzza (2011)].

Controlabilidade

Ainda de acordo com Roland Siegwart and Scaramuzza (2011), existe uma correlação inversa entre manobrabilidade e controlabilidade. Projetos omnidirecionais requerem processamento significativo para converter as velocidades rotacionais e translacionais desejadas em comandos de roda individuais. Ademais, os graus de liberdade proveniente das rodas holonômicas causam acúmulo de derrapagens, tendem a reduzir a precisão do cálculo morto e aumentam a complexidade do projeto.

Sistema de locomoção do Omnino

A plataforma móvel do robô é baseada no modelo proposto por Roland Siegwart and Scaramuzza (2011), conforme representado na Figura 2.4, e utiliza três rodas suecas, do tipo que os rolos externos formam um eixo de rotação de 90° em relação ao plano da roda, distribuídas uniformemente em ângulos de 120° . Em cada roda são ligados motores de corrente contínua com redução e encoders acoplados, para que seja possível medir e controlar a velocidade empregada em cada uma de suas rodas. Na Figura 2.4, é mostrado o modelo base do chassi desenvolvido para o robô omnidirecional Omnino.

2.2.2 Modelo cinemático

A cinemática é o estudo mais básico de como os sistemas mecânicos se comportam [Roland Siegwart and Scaramuzza (2011)]. Na robótica móvel, é preciso entender o comportamento mecânico do robô tanto para melhor projetá-lo visando a realização de suas tarefas quanto criar um software de controle para uma instância de *hardware* de um robô móvel.

O processo de compreensão dos movimentos de um robô inicia com a descrição da contribuição que cada roda fornece para o movimento. Se por um lado cada roda permite todo o robô se mover, pelo outro será imposto uma restrição na locomoção do robô; como exemplo, o impedimento do robô derrapar lateralmente [Roland Siegwart and Scaramuzza (2011)].

Segundo Lynch and Park (2017), é necessário que um robô móvel omnidirecional possua pelo menos três rodas para atingir uma velocidade arbitrária, $\dot{q} = (\dot{\phi}, \dot{x}, \dot{y})$, desde que cada roda possua um único motor.

Ainda conforme o autor, para um robô holonômico com $m \geq 3$ rodas, a matriz $H(\phi) \in \mathbb{R}^{m \times 3}$ mapeando a velocidade do chassi desejada $\dot{q} \in \mathbb{R}^3$ para o vetor das velocidades das rodas $u \in \mathbb{R}^m$ é construída adicionando m linhas $h_i(\phi)$:

$$u = H(\phi)\dot{q} = \begin{bmatrix} h_1(\phi) \\ h_2(\phi) \\ \vdots \\ h_m(\phi) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{x} \\ \dot{y} \end{bmatrix} \quad (2.1)$$

Sendo $h_i(\phi)$, o modelo de cada roda.

A velocidade angular de cada roda é encontrada por meio da transformação:

$$u_i = h_i(\phi)\dot{q} = \left(\begin{bmatrix} \frac{1}{r_i} & \frac{\tan \gamma_i}{r_i} \end{bmatrix} \begin{bmatrix} \cos \beta_i & \sin \beta_i \\ -\sin \beta_i & \cos \beta_i \end{bmatrix} \begin{bmatrix} -y_i & 1 & 0 \\ x_i & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \right) \begin{bmatrix} \dot{\phi} \\ \dot{x} \\ \dot{y} \end{bmatrix} \quad (2.2)$$

Em que, lendo da direita para a esquerda, tem-se: o vetor de velocidade do robô; transformação que expressa a velocidade em *twist* (vetor que combina velocidade linear e angular); a transformação em velocidade linear; a transformação que expressa a velocidade linear no sistema de coordenadas das rodas; e por fim é calculado a velocidade angular de condução da roda.

Simplificando:

$$h_i(\phi) = \frac{1}{r_i \cos \gamma_i} \begin{bmatrix} x_i \sin(\beta_i + \phi) - y_i \cos(\beta_i + \phi) \\ \cos(\beta_i + \gamma_i + \phi) \\ \sin(\beta_i + \gamma_i + \phi) \end{bmatrix}^T \quad (2.3)$$

Sendo, β o ângulo da roda com relação ao eixo inicial do chassi, γ o ângulo do rolamento auxiliar com o eixo de rotação da roda e ϕ a orientação do chassi, x e y as coordenadas das rodas em relação ao centro do chassi do robô.

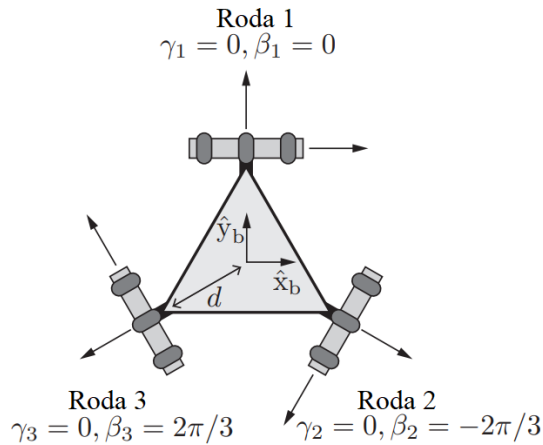


Figura 2.4:
Modelo cinemático para robô holonômico com três rodas.
Fonte: Lynch and Park (2017)

Aplicando a configuração da figura 2.4 na equação 2.3, é obtido o modelo cinemático para um robô móvel com três rodas omnidirecionais, no caso em que os rolos auxiliares formam um ângulo de 0 grau com o eixo da roda ($\gamma = 0$) e que a orientação do chassis do robô não interfere no mapeamento ($\phi = 0$):

- Para $\beta_1 = 0$, $\gamma_3 = 0$ e $\phi = 0$

$$h_1(\phi) = \frac{1}{r_1} \begin{bmatrix} -d \\ 1 \\ 0 \end{bmatrix}^T \quad (2.4)$$

- Para $\beta_2 = \frac{-2\pi}{3}$, $\gamma_3 = 0$ e $\phi = 0$

$$h_2(\phi) = \frac{1}{r_1} \begin{bmatrix} -d \\ \frac{-1}{2} \\ -\frac{\sqrt{3}}{2} \end{bmatrix}^T \quad (2.5)$$

- Para $\beta_3 = \frac{2\pi}{3}$, $\gamma_3 = 0$ e $\phi = 0$

$$h_3(\phi) = \frac{1}{r_1} \begin{bmatrix} -d \\ \frac{-1}{2} \\ \frac{\sqrt{3}}{2} \end{bmatrix}^T \quad (2.6)$$

Como o centro da base robô está a uma distância fixa das rodas e não interfere no mapeamento da cinemática, a expressão $x_i \sin(\beta_i + \phi) - y_i \cos(\beta_i + \phi)$ pode ser substituída pela própria distância d .

Portanto, a cinemática do Omnino é descrita por:

$$u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = H(0)\dot{q} = \frac{1}{r} \begin{bmatrix} -d & 1 & 0 \\ -d & -\frac{1}{2} & -\frac{\sqrt{3}}{2} \\ -d & -\frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{x} \\ \dot{y} \end{bmatrix} \quad (2.7)$$

Com r sendo o raio da roda, d o raio do centro da roda ao centro do robô, \dot{x} a velocidade no eixo-x, \dot{y} a velocidade no eixo-y e $\dot{\phi}$ a rotação no eixo-z.

2.3 Localização

Hoje os robôs não se limitam a só compartilhar espaço com os humanos, como atuam em ambientes onde os humanos não podem ir. Portanto, a capacidade de manter um senso de posição e navegar sem intervenção humana é indispensável [Roland Siegwart and Scaramuzza (2011)].

A localização é outro problema fundamental encontrado em robôs móveis que também requer atenção. A parte desafiadora da localização é estimar a posição e a orientação do

robô, cujas informações podem ser adquiridas de sensores e outros sistemas [Alatise and Hancke (2020)].

Segundo Roland Siegwart and Scaramuzza (2011), se um sistema de GPS pudesse ser instalado em um robô, o problema de localização seria resolvido, pois sempre saberia onde estava. Mas, no momento, esse sistema não está disponível ou não é preciso o suficiente para trabalhar. Ainda assim, a localização implica não apenas conhecer a posição absoluta do robô na Terra, mas também sua posição relativa em relação a um destino [Francisco Rubio and Llopis-Albert (2019)].

2.3.1 Problemas de localização

Com base na informação da posição inicial do robô, é possível classificar os problemas de localização em rastreamento de posição, posicionamento global e problema do robô sequestrado [Panigrahi and Bisoy (2022)]. Ainda de acordo com o estudo realizado pelo autor, dentro das categorias descritas anteriormente, o problema de rastreamento de posição é o mais abordado (Fig. 2.5).



Figura 2.5:
Pesquisas em problemas de localização.
Fonte: Panigrahi and Bisoy (2022)

No rastreamento de posição, a posição inicial do robô é conhecida e o objetivo é rastrear o robô a cada instante ao longo de sua navegação no ambiente. Durante a navegação, o algoritmo de localização utiliza a posição anterior do robô para atualizar sua localização atual. Isso é possível monitorando continuamente a rota do robô. O rastreamento de posição utiliza odometria e dados dos sensores, entretanto, em caso de grande incerteza, o robô pode não ser localizado [Panigrahi and Bisoy (2022)].

A realocação implica no rastreamento de robôs móveis sem qualquer informação sobre sua *pose* inicial. Isso dá origem a outra classe de problemas de localização: a localização global. Nesta categoria, o robô não tem nenhum conhecimento sobre sua posição inicial e

deve ser capaz de se localizar globalmente dentro de um ambiente conhecido. Em algumas situações, o robô é localizado em um local arbitrário durante o rastreamento de pose ou pode ter sido movido (sequestrado) para um lugar não identificado. Esta questão surge no problema do robô sequestrado no qual o robô sabe que está sendo sequestrado ao não reconhecer o ambiente em que se encontra [Panigrahi and Bisoy (2022)].

2.3.2 Técnicas de localização

Assim, para enfrentar a questão da localização, uma boa técnica deve ser proposta para lidar com erros, fatores de rebaixamento, medições e estimativas inadequadas. As técnicas são divididas em duas categorias: localização relativa e localização absoluta [Alatise and Hancke (2020)].

Técnicas de localização relativa

Este método estima a posição e orientação do robô móvel integrando informações produzidas e combinadas por diversos sensores, geralmente encoder ou sensores inerciais. A integração começa a partir da posição inicial e atualiza continuamente no tempo. O posicionamento relativo sozinho pode ser usado apenas por um curto período de tempo [Alatise and Hancke (2020)].

- I. Odometria: A odometria é o método de navegação mais utilizado para posicionamento de robôs móveis; fornece boa precisão a curto prazo, é barato e permite taxas de amostragem muito altas. No entanto, a ideia fundamental da odometria é a integração do movimento incremental ao longo do tempo, o que leva inevitavelmente ao acúmulo ilimitado de erros, que aumentam proporcionalmente com a distância percorrida [Borenstein et al. (1997)].
- II. Sensores inerciais: A navegação inercial usa giroscópios e acelerômetros para medir a taxa de rotação e aceleração, respectivamente. As medições são integradas para determinar a posição e por serem autossuficientes [Borenstein et al. (1997)], não precisam de referências externas. Todavia, os dados do sensor inercial variam com o tempo e qualquer pequeno erro constante aumenta sem limites devido à necessidade da integração. Sendo assim, são inadequados para um posicionamento preciso durante um longo período de tempo [Borenstein et al. (1997)].

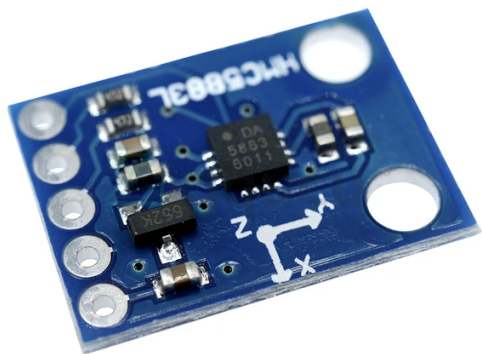
Técnicas de localização absoluta

A abordagem absoluta permite que o robô móvel procure sua localização diretamente do ambiente do sistema móvel. Seus métodos dependem de *beacons* de navegação, pontos de referência, correspondência de mapas ou sinais baseados em satélite. A posição do robô é determinada externamente e sua precisão geralmente independe do tempo e da

localização, portanto, não há agregação de erro com tempo ou distância percorrida. A limitação é que não é possível acompanhar o robô por curtas distâncias [Panigrahi and Bisoy (2022)].

- I. **Bússolas magnéticas:** A direção do veículo é o mais significativo dos parâmetros de navegação (x, y, θ) em termos de sua influência nos erros acumulados de estimativa. Por esta razão, sensores que fornecem uma medida de rumo absoluto, como a bússola magnética, são importantes para a navegação de plataformas autônomas. Uma desvantagem é a dificuldade do seu uso para aplicações internas devido a interferências por estruturas metálicas e sua suscetibilidade à vibração [Roland Siegwart and Scaramuzza (2011)]. Um exemplo é o módulo GY-273, voltado para projetos de baixo custo imagem (a) da figura 2.6.
- II. **Beacons:** Sistemas de navegação de farol (*Beacons*) ativo são os auxílios de navegação mais comuns em navios e aviões, bem como em sistemas de robôs móveis comerciais. Beacons podem ser detectados de forma confiável e fornecer informações de posicionamento precisas com processamento mínimo. Como resultado, esta abordagem permite altas taxas de amostragem e produz alta confiabilidade, mas também incorre em alto custo de instalação e manutenção [Borenstein et al. (1997)]. Os faróis ultrassônicos estacionários, imagem (b) da figura 2.6, são um exemplo de *beacons* voltados para ambientes controlados.
- III. **Sistemas de posicionamento global:** O Sistema de Posicionamento Global (GPS) é uma tecnologia para navegação ao ar livre. O sistema compreende 30 satélites que transmitem sinais de onda de rádio codificados. Um receptor GPS funciona medindo o tempo de viagem dos sinais de rádio de quatro ou mais satélites. Com quatro pontos conhecidos no espaço e quatro atrasos de tempo medidos, é possível calcular a posição (x, y, z) do receptor [Corke (2011)].
- IV. **Ponto de referência:** Pontos de referência são características distintas que um robô pode reconhecer a partir de seus sensores. As referências podem ser formas geométricas e podem incluir informações adicionais. Em geral, os pontos de referência têm uma posição fixa e conhecida, e são escolhidos para serem fáceis de identificar. Antes que um robô possa usar pontos de referência para navegação, as características dos pontos de referência devem ser conhecidas e armazenadas na memória do robô [Borenstein et al. (1997)]. Um tipo de ponto de referência muito utilizado são as *tags* de realidade aumentada, como mostra a imagem (c) da figura 2.6.
- V. **Correspondência de mapas:** Posicionamento baseado em mapa, também conhecido como "mapa correspondente", é uma técnica na qual o robô usa seus sensores para criar um mapa de seu ambiente local e o compara com um mapa armazenado na

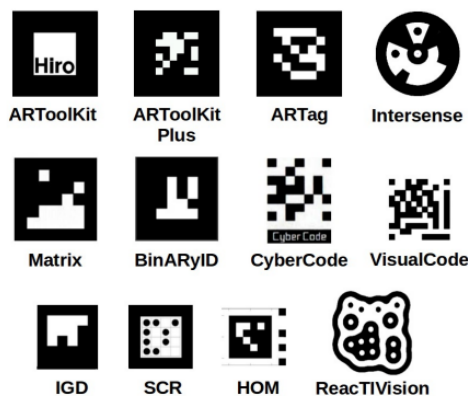
memória ou um mapa gerado por outro sensor. A correspondência normalmente ocorre em três etapas: primeiro, pontos de interesse preliminares são comparados até que uma condição seja alcançada; em seguida é realizada a correspondência de pontos distantes das interseções encontradas no primeiro passo, mas que ainda estejam dentro do segmento analisado; por fim são comparados pontos próximos a outros segmentos [Hashemi and Karimi (2014)]. Exemplo mostrado na imagem (d) da figura 2.6, em que um mapa é gerado usando um sensor LIDAR.



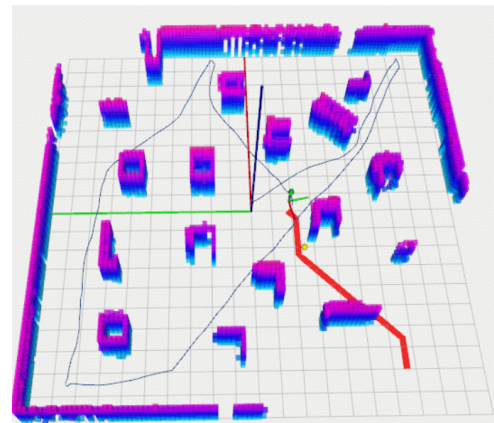
(a) Módulo de bússola magnética GY-273



(b) Marvelmind Beacon Mini-RX



(c) Pontos de referência do tipo realidade aumentada.



(d) SLAM

Figura 2.6: Localização absoluta

2.4 Filtro de Kalman

A localização baseada em filtro de Kalman resolve o problema de rastreamento de posição de maneira eficiente e precisa. É um estimador de estados ótimo - ou seja, infere parâmetros de interesse de observações indiretas, imprecisas e incertas, e por isso, é muito utilizado em técnicas de fusão de sensores [Roland Siegwart and Scaramuzza (2011)] para rastrear robôs a partir de um local inicialmente conhecido. No entanto, em certas situações, a incerteza do robô é muito grande (uma colisão com obstáculo, a exemplo) e,

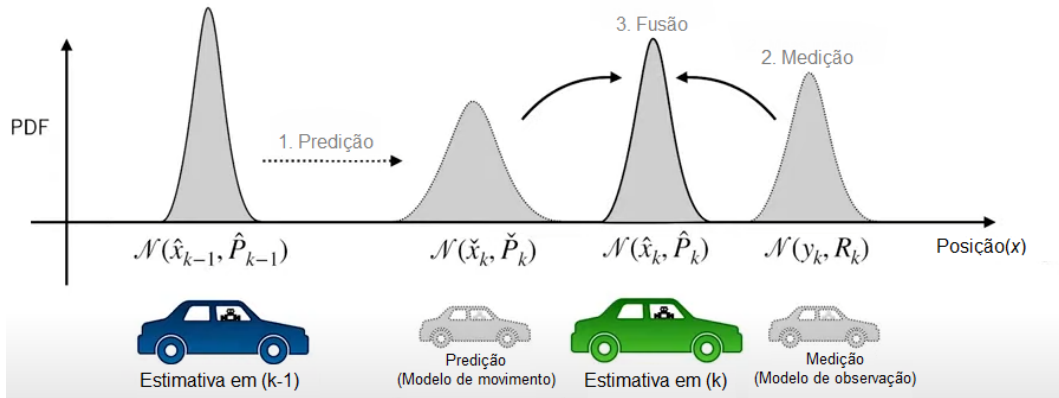


Figura 2.7:

Exemplo do Filtro de Kalman

Fonte: Adaptado de Waslander and Kelly (2019)

portanto, a localização não consegue capturar as possíveis posições do robô [Panigrahi and Bisoy (2022)].

O filtro de Kalman é uma das ferramentas de estimativa mais úteis disponíveis atualmente. Ele fornece um método recursivo de estimar o estado de um sistema dinâmico na presença de ruído. Uma característica chave é que ele simultaneamente mantém estimativas do vetor de estado (\hat{x}) e da matriz de covariância de erro de estimativa (P), ou seja, a saída de um filtro de Kalman é uma função de densidade de probabilidade gaussiana (PDF) com média \hat{x} e covariância P . No contexto da localização, a saída do filtro é uma distribuição de prováveis posições do robô ao invés de uma única estimativa de posição [Choset et al. (2005)].

O filtro assume que as variáveis do estado (\hat{x}) são aleatórias e possuem distribuição Gaussiana. Ou seja, cada variável possui média μ , o centro da distribuição aleatória e estado mais provável, e variância σ^2 , representando a incerteza. Algumas variáveis podem possuir relações entre si, e essa correlação é registrada na matrix de covariância. Para cada elemento da matriz existe um grau de correlação entre a i -ésima variável e a j -ésima variável.

A ideia por trás do filtro é ilustrada na figura 2.7. Iniciando de uma estimativa $k - 1$, a nova posição do veículo será estimada usando a combinação de uma predição e por uma medição. A predição é realizada por meio de equações de movimento, muitas vezes derivadas de sensores inerciais ou odometria das rodas. A medição, utilizada para corrigir a predição, é obtida do modelo de observação, extraído de sensores mais precisos como o GPS ou o LIDAR. Desta maneira, o filtro de Kalman é visto como uma técnica de fusão de informações provenientes de diferentes sensores para produzir uma estimativa ótima, considerando a incerteza de seu movimento e sua observação [Waslander and Kelly (2019)].

A representação da Gaussiana (μ, σ^2) é dada pela melhor estimativa do vetor de estado

(\hat{x}) e sua matriz de covariância, incerteza do estado, P em um instante k :

$$\hat{x}(k) = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad P(k) = \begin{bmatrix} \Sigma_{x_1 x_1} & \dots & \Sigma_{x_1 x_n} \\ \vdots & \ddots & \vdots \\ \Sigma_{x_n x_1} & \dots & \Sigma_{x_n x_n} \end{bmatrix} \quad (2.8)$$

As etapas do filtro são descritas pelas seguintes equações [Choset et al. (2005)]:

Predição

$$\begin{aligned} \hat{x}(k+1|k) &= F(k)\hat{x}(k|k) + G(k)u(k) \\ P(k+1|k) &= F(k)P(k|k)F(k)^T + V(k) \end{aligned} \quad (2.9)$$

Na previsão, nossa estimativa de estado é atualizada de acordo com a dinâmica do sistema. Além disso, a covariância de erro, P , aumenta com a matriz de covariância de ruído do processo, V . Valores maiores de V farão com que a covariância de erro cresça mais rapidamente.

- F é a matriz que armazena a dinâmica do sistema;
- u é o vetor de entrada;
- G é a matriz de entrada. Transforma o vetor de entrada para o mesmo espaço do sistema;
- V é a matriz de covariância de ruído de processo.

Medição

$$\begin{aligned} \hat{x}(k+1|k+1) &= \hat{x}(k+1|k) + Rv \\ P(k+1|k+1) &= P(k+1|k) - RH(k+1)P(k+1|k) \end{aligned} \quad (2.10)$$

Em que

$$\begin{aligned} v &= y(k+1) - H(k+1)x(k+1|k) \\ S &= H(k+1)P(k+1|k)H(k+1)^T + W(k+1) \\ R &= P(k+1|k)H(k+1)^T S^{-1} \end{aligned} \quad (2.11)$$

Na etapa de correção, a estimativa de estado é atualizada para incluir novas informações de medição. Essa nova informação é ponderada em relação à estimativa de estado \hat{x} usando o ganho de Kalman R . Valores grandes de R pesam mais as medições recebidas, enquanto valores menores dão mais peso a previsão. Como o ganho de Kalman (R) está relacionado a incerteza do estado \hat{x} , valores maiores de P aumentarão o ganho (R) e isto implica que as observações são mais precisas, conseqüentemente as estimativas da previsão não são estáveis. Por fim, a covariância de erro P diminui para aumentar a confiança na estimativa do estado.

- y é o vetor de saída;
- H é a matriz de saída. Transforma o vetor de saída para o mesmo espaço do sistema;
- W é a matriz de covariância de ruído de medição;
- v é a inovação;
- R é o ganho de Kalman.

Capítulo 3

Arquitetura proposta

3.1 Descrição

A ideia da proposta surge do problema que os carros autônômicos sofrem ao utilizar o GPS em ambientes fechados, túneis ou entre edifícios altos, causando interferência no sinal. O experimento consiste em realizar o rastreamento de posição como visto na seção 2.3.1, em que se deseja estimar a posição do robô durante alguns trajetos pré-definidos utilizando o filtro de Kalman como estimador, que permite a fusão de sensores. Para a etapa de previsão, o filtro é alimentado com as leituras providas por uma IMU. Na etapa de atualização, a correção da previsão é feita por meio das leituras da câmera utilizando a técnica de localização por ponto de referência.

3.2 Metodologia

3.2.1 Unidade de medida inercial

Para a localização relativa foi utilizado o módulo GY-521 que possui o sensor inercial *MPU-6050* integrada a ele.

O MPU-6050 é um dispositivo integrado de rastreamento de movimento de 6-eixos que combina um giroscópio de 3-eixos, acelerômetro de 3-eixos e um processador de movimento digital (DMP). Com seu barramento de sensor *I²C* dedicado, ele aceita diretamente entradas de uma bússola externa com 3-eixos para fornecer uma saída completa com 9 DOF, utilizando a sua implementação *MotionFusion*. Ainda possui uma porta *I²C* auxiliar que permite a interação com outros sensores não inerciais, como sensores de pressão [InvenSense (2012)].

O sensor inercial possui três conversores analógico-digital (ADCs) de 16 bits para digitalizar as saídas do giroscópio e três ADCs de 16 bits para digitalizar as saídas do acelerômetro. Para rastreamento preciso de movimentos rápidos e lentos, as peças apresentam um giroscópio programável pelo usuário em escala total de

$\pm 250^\circ/\text{seg}$, $\pm 500^\circ/\text{seg}$, $\pm 1000^\circ/\text{seg}$ e $\pm 2000^\circ/\text{seg}$ (graus por segundo) e um acelerômetro programável pelo usuário com alcance de $\pm 2g$, $\pm 4g$, $\pm 8g$ e $\pm 16g$ (força-g) [InvenSense (2012)].

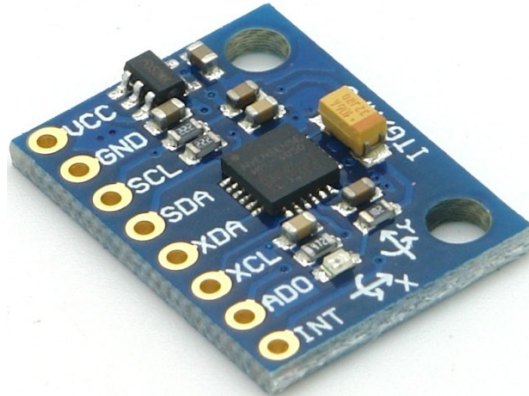


Figura 3.1:
Módulo GY-521.
Fonte: InvenSense (2012)

3.2.2 Câmera

Para a localização absoluta do robô foi utilizado o módulo de câmera *Raspberry Pi Camera Rev 1.3*, que possui compatibilidade com os modelos A e B da placa *Raspberry Pi*. Oferece alta sensibilidade, baixa interferência e captura de imagem com baixo ruído, e se conecta à placa Raspberry Pi por meio do conector CSI projetado especificamente para interface com câmeras [Pi (2013)].

A câmera conta com um sensor de imagem Omnivision 5647 CMOS em um módulo de foco fixo com filtro infravermelho integrado, com resolução de 5 megapixels e captura de imagens a 60 frames por segundo (FPS) no modo 720p ou 30 FPS em 1080p. Ainda oferece as funções automáticas de controle de exposição, balanço da cor branca, calibração dos níveis de preto entre outras [Pi (2013)].

Tag ArUco

Como ponto de referência foi utilizado marcador fiducial. Eles oferecem um padrão distinguível com características visuais, são simples, rápidos e precisos em sua detecção, e em alguns casos apresentam uma codificação como proteção contra falhas de detecções incorretas, [Raj and Kos (2021), Garrido-Jurado et al. (2014)].

Para este trabalho, foi utilizado o marcador ArUco, uma *tag* baseada em quadrados com códigos binários. A *tag* possui a vantagem de facilitar a obtenção da *pose* por meio do seus quatro pontos proeminentes, enquanto que o centro fornece a sua identificação,



Figura 3.2:
Módulo Pi Camera.
Fonte: Pi (2013)

diferente de um marcador circular, por exemplo, que necessita de várias *tags* para estimar a *pose* [Garrido-Jurado et al. (2014)].

A detecção dos marcadores funciona por etapas [Garrido-Jurado et al. (2014)]:

1. Segmentação de imagem: Primeiramente, os contornos mais proeminentes na imagem em escala de cinza são extraídos.
2. Extração de contorno e filtragem: Uma extração de contorno é realizada na imagem limiarizada, seguida de uma aproximação poligonal. Como os marcadores são incluídos em contornos retangulares, aqueles que não são aproximados a polígonos de 4 vértices são descartados. Por fim, é simplificado os contornos próximos deixando apenas os externos.
3. Extração do código do marcador: Consiste em analisar a região interna dos contornos para extrair seu código interno. Primeiro, a projeção em perspectiva é removida pelo cálculo da matriz de homografia. Em seguida, a imagem binarizada é dividida em uma grade e cada elemento recebe o valor 0 ou 1, dependendo dos valores da maioria dos pixels nela. Se todos os bits da borda forem zero, a grade interna será analisada.
4. Identificação do marcador e correção de erros: Neste ponto, é necessário determinar quais dos candidatos a marcador obtidos realmente pertencem ao dicionário de *tags*. Uma vez extraído o código de um candidato a marcador, são obtidos quatro identificadores diferentes (um para cada rotação possível). Se algum deles for encontrado no dicionário, será um candidato a marcador válido.

5. Refinamento de canto e estimativa de pose: Uma vez detectado um marcador, é possível estimar sua pose em relação à câmera minimizando iterativamente o erro de reprojeção dos cantos.

Ainda segundo o autor Garrido-Jurado et al. (2014), os algoritmos utilizados para o processos descritos anteriormente ficam a critério do desenvolvedor. Neste trabalho foram utilizadas as funcionalidades disponíveis na biblioteca *OpenCV* [OpenCV (2012)].

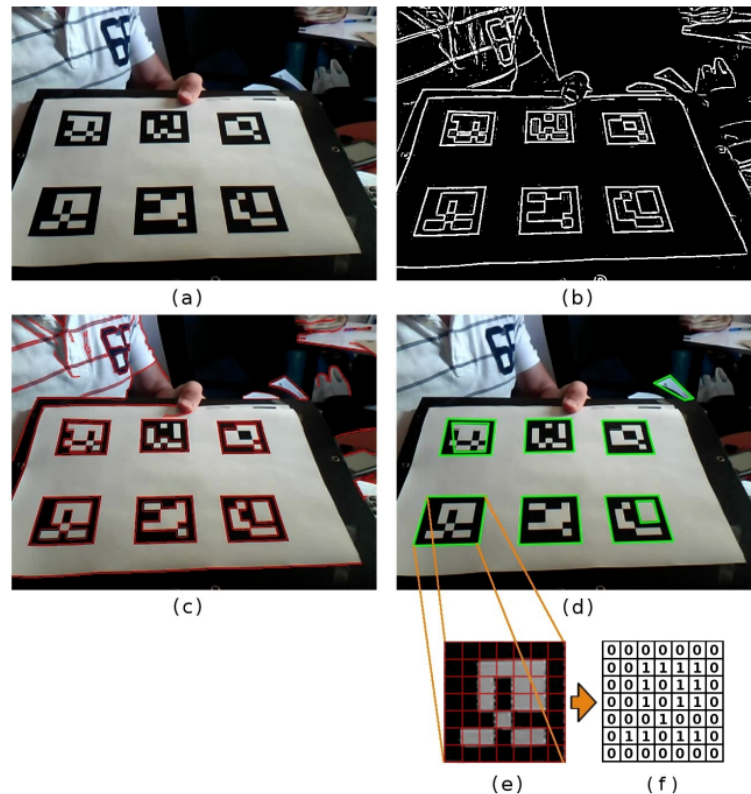


Figura 3.3:

Processos para detecção da tag ArUco. (a) Imagem original. (b) Resultado da segmentação. (c) Detecção de contorno. (d) Aproximação poligonal e remoção de contornos irrelevantes. (e) Exemplo de marcador após transformação de perspectiva. (f) Atribuição de bit.

Fonte: Garrido-Jurado et al. (2014)

3.2.3 Modelagem

No experimento, o robô inicia o seu trajeto a partir de uma posição conhecida. A IMU estará acoplada no eixo superior do robô e abaixo da câmera, que está com sua lente voltada para cima. A tag ArUco está posicionada em um suporte a cima da superfície que o robô trabalhará, de modo que a partir de qualquer ponto da plataforma o robô possa detectá-la.

O sistema do robô foi desenvolvido utilizando o ROS, que permite fácil integração e comunicação entre os componentes do robô, além de permitir a visualização das operações em tempo real.

Movimentação

Os comandos de movimentação são originados no módulo de planejamento, que possui os trajetos que o robô percorrerá. Por meio do ROS [ROS (2021)], o nó de planejamento envia as velocidades desejadas ao nó do driver dos motores, este que contém a cinemática do robô e transforma a velocidade desejada em informações para os motores conectados as rodas.

Para operar o robô é utilizado o modelo cinemático descrito na seção 2.2.2. O raio d , 7 centímetros, é a distância do centro do robô ao ponto médio de uma das rodas holonômicas de raio r , 2 centímetros.

Substituindo nas equações cinemáticas:

$$u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \frac{1}{0.02} \begin{bmatrix} -0.07 & 1 & 0 \\ -0.07 & -\frac{1}{2} & -\frac{\sqrt{3}}{2} \\ -0.07 & -\frac{1}{2} & \frac{\sqrt{3}}{2} \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{x} \\ \dot{y} \end{bmatrix} \quad (3.1)$$

v_{bx} , v_{by} e w_{bz} são obtidos do módulo de planejamento do robô e o resultado u é enviado ao controlador dos motores das rodas.

Transformação das coordenadas

A detecção da tag ArUco retorna a pose da câmera em relação ao marcador como uma transformação 3D do sistema de coordenadas do marcador para o sistema de coordenadas da câmera. Entretanto, para que a pose obtida seja a do robô em relação ao mundo, é necessária realizar uma transformação no sistema de coordenadas da posição retornada.

Para isso são utilizados três pares vetor-quaternion para representar a transformação: $V_{wa} = (T_{wa}, Q_{wa})$, $V_{ra} = (T_{ra}, Q_{ra})$ e $V_{wr} = (T_{wr}, Q_{wr})$.

O vetor-quaternion V_{wa} é composto pelo vetor translação e quaternion que armazena a posição dos marcadores ArUco no mundo. O par V_{ra} é atualizado com as leituras obtidas a partir da detecção dos marcadores. E por fim, V_{wr} é resultado da transformação do sistema de coordenadas, obtido por meio da equação [Corke (2011)]:

$$\begin{aligned} (T_{wr}, Q_{wr}) &= (T_{wa}, Q_{wa}) \oplus (T_{ra}, Q_{ra}) \\ (T_{wr}, Q_{wr}) &= (T_{wa}, Q_{wa}) \oplus (-Q_{ra}^{-1} \cdot T_{ra}, Q_{ra}^{-1}) \\ (T_{wr}, Q_{wr}) &= (T_{wa} + Q_{wa} \cdot (-Q_{ra}^{-1} \cdot T_{ra}), Q_{wa} \cdot Q_{ra}^{-1}) \\ (T_{wr}, Q_{wr}) &= (T_{wa} - Q_{wa} \cdot Q_{ra}^{-1} \cdot T_{ra}, Q_{wa} \cdot Q_{ra}^{-1}) \end{aligned} \quad (3.2)$$

Aplicação do Filtro de Kalman

Para a estimação do estado, o filtro de Kalman foi utilizado em conjunto da IMU e leitura das tags ArUco por meio da câmera.

Fazendo uso da equações descritas na seção 2.4:

Predição

$$\begin{aligned}\hat{x}(k+1|k) &= F(k)\hat{x}(k|k) + G(k)u(k) \\ P(k+1|k) &= F(k)P(k|k)F(k)^T + V(k)\end{aligned}$$

Para o Omnino, seu vetor de estado possui as informações de sua posição x , posição y e rotação em torno do seu eixo-z θ :

$$\hat{x} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$$

Definindo as matrizes que foram utilizadas:

- F : Como a nova predição sempre estará no mesmo espaço do vetor de estado \hat{x} , a matriz F será uma matriz identidade 3×3 (I_3);
- G : Matriz que transforma as leituras da IMU (u) para o mesmo espaço que o vetor de estados. Será uma matriz diagonal com valores $[\frac{dt^2}{2} \ \frac{dt^2}{2} \ dt]$, com intervalo de amostra $dt = 0.5$;
- u : Vetor $[\ddot{x} \ \ddot{y} \ \dot{\theta}]^T$, contendo as acelerações nos eixos x-y e a velocidade no eixo z, obtidas da IMU;
- P : Inicialmente uma matriz identidade 3×3 (I_3), dado que nossa estimação inicia na coordenada $(0, 0)$;
- V : Matriz de covariância para as leituras da IMU. Como não foi fornecida no *datasheet* do módulo GY-521, foi criada a partir da variância de uma amostra de 1000 leituras enquanto imóvel. Será uma matriz diagonal com valores $[0.001 \ 0.0027 \ 0.00148]$.

Atualização

$$\begin{aligned}\hat{x}(k+1|k+1) &= \hat{x}(k+1|k) + Rv \\ P(k+1|k+1) &= P(k+1|k) - RH(k+1)P(k+1|k) \\ v &= y(k+1) - H(k+1)x(k+1|k) \\ S &= H(k+1)P(k+1|k)H(k+1)^T + W(k+1) \\ R &= P(k+1|k)H(k+1)^T S^{-1}\end{aligned}$$

Definindo as matrizes que foram utilizadas:

- y : Vetor $[x \ y \ \theta]^T$ com as leituras dos marcadores ArUco possuindo um intervalo de amostra $dt = 1.0$;
- H : Como o vetor de observação da câmera está no mesmo espaço que o vetor de estados \hat{x} , a matriz H será uma matriz identidade 3×3 (I_3);
- W : Matriz de covariância para as leituras da câmera, com valor $I_3 \cdot 0.1$, representando uma maior confiança na observação.

3.3 Cenários

3.3.1 Cenário 1

O primeiro cenário trata-se de fazer o robô seguir uma trajetória reta (figura 3.4) e rastrear o deslocamento no eixo-x. Na primeira execução será usado somente as leituras da IMU, na segunda execução as leituras da IMU após passagem pelo filtro de Kalman e na última execução a adição da leitura da *tag* ArUco a partir da câmera no passo de correção do filtro.

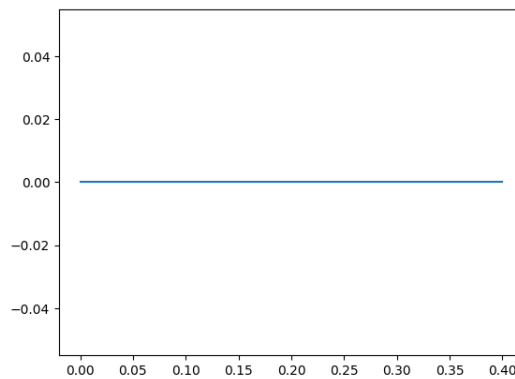


Figura 3.4:
Trajeto linear (em metros)
Fonte: Autor

3.3.2 Cenário 2

Assim como no primeiro cenário, serão realizadas três execuções com a diferença no trajeto. Agora a avaliação ocorrerá em um caminho com formato de L, como na figura 3.5.

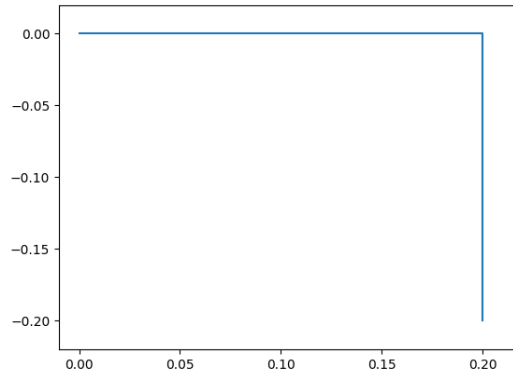


Figura 3.5:
Trajeto em L (em metros)
Fonte: Autor

3.3.3 Cenário 3

Por fim, agora em um trajeto mais complicado, o objetivo é percorrer um caminho formado pela Lemniscata de Bernoulli, figura 3.6.

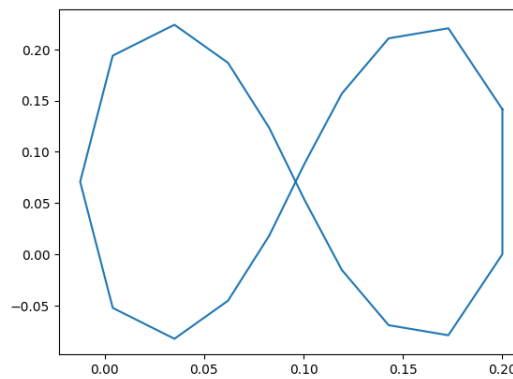


Figura 3.6:
Trajeto Lemniscata de Bernoulli (em metros)
Fonte: Autor

Capítulo 4

Resultados

4.1 Cenário 1 - Percurso linear

Apesar do objetivo do cenário ter sido executar um percurso linear, foi utilizada uma das execuções que não seguiu o trajeto com fidelidade, ilustrando o efeito do movimento irregular causado pela disposição e tipo de roda usada em adição a falta de um sistema de controle.

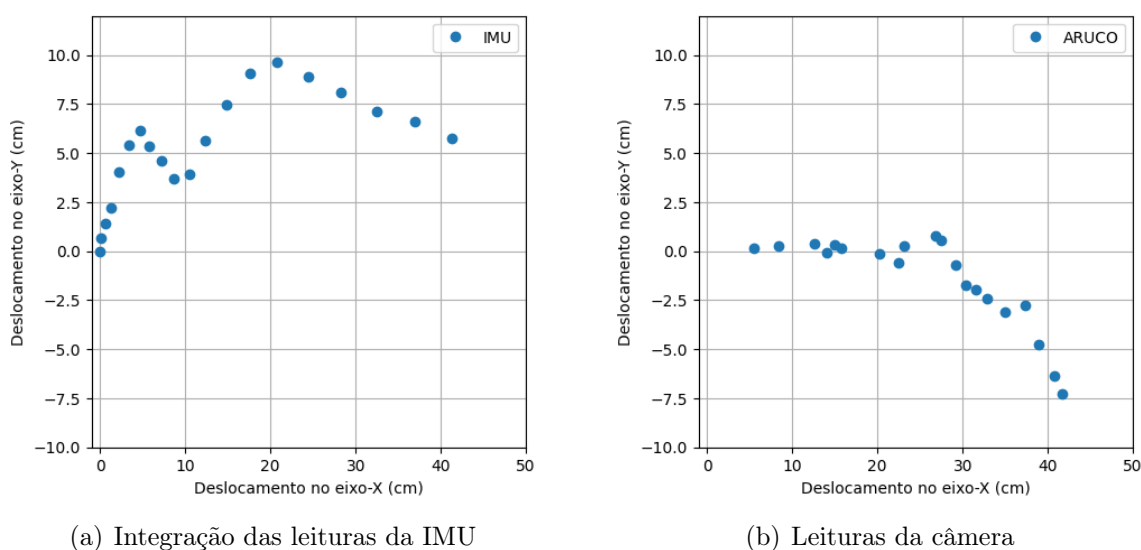


Figura 4.1:
Leituras da IMU e câmera durante o deslocamento em linha reta

Fonte: Autor

Ao analisar a imagem (a) da figura 4.1, é possível notar como o cálculo da integração das acelerações são afetados pelo ruído e o quão mais nítido fica por se tratar de um percurso mais longo em uma única direção, quando se comparado com os outros percursos. Ainda assim, os dados obtidos da câmera foram suficientes para corrigir o deslocamento. Um fator que contribuiu para esta correção foi a utilização de uma mesma frequência de observação para a IMU e câmera que permitiu a comparação exata entre os dados, embora

esse processo acaba sendo mais custoso por conta do cálculo envolvendo as transformações do sistema de coordenadas da câmera.

4.2 Cenário 2 - Percurso em L

Para o segundo cenário foi escolhido a melhor execução. Mais uma vez percebe-se o acúmulo do erro na integração das leituras da IMU, além do efeito que os dois desvios na leitura da câmera (próximos ao centímetro 15 da imagem (b) na figura 4.2) causam no resultado após a passagem pelo filtro. Ademais, diferente do primeiro cenário onde foi utilizado uma frequência igual para os dois sensores, para o segundo e terceiro cenários a frequência para a câmera foi menor, representando melhor uma situação real em que o sistema mais robusto utilizado para correção pode nem sempre estar disponível ou é muito custoso de se manter operante durante todo o trajeto.

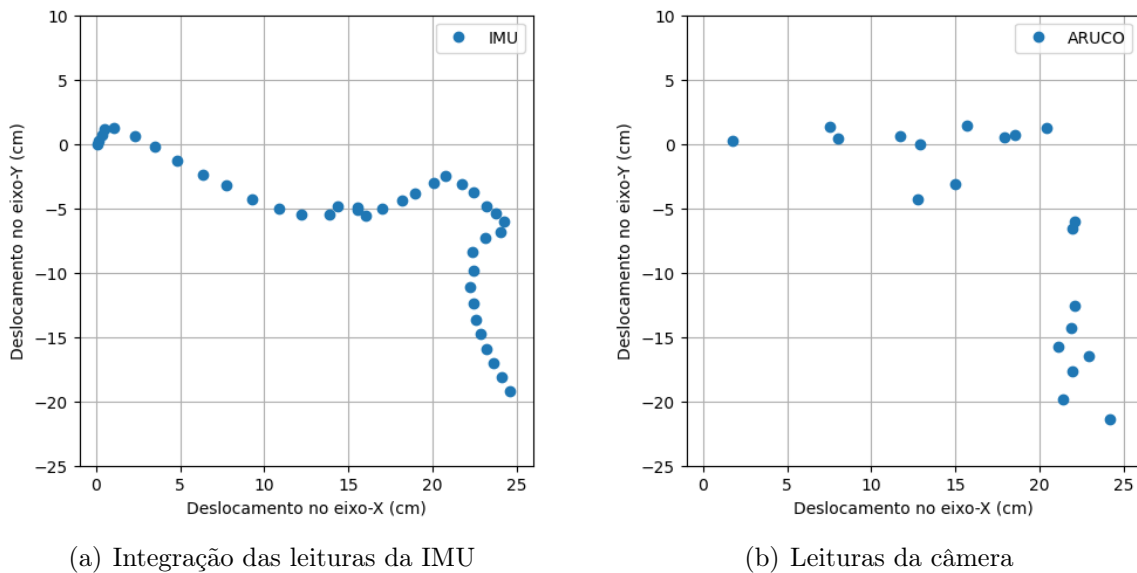


Figura 4.2:
Leituras da IMU e câmera durante o deslocamento em L
Fonte: Autor

4.3 Cenário 3 - Percurso Lemniscata de Bernoulli

O terceiro cenário se apresentou o mais complicado de executar, como previsto anteriormente. O fato do robô se deslocar nos eixos $x - y$ durante todo o caminho causava maior variação nas leituras da IMU. Além disso, como o intuito do experimento foi fazer o uso da capacidade do robô se locomover em qualquer direção, se deslocar com o eixo de rotação z fixo resultou em uma movimentação mais instável.

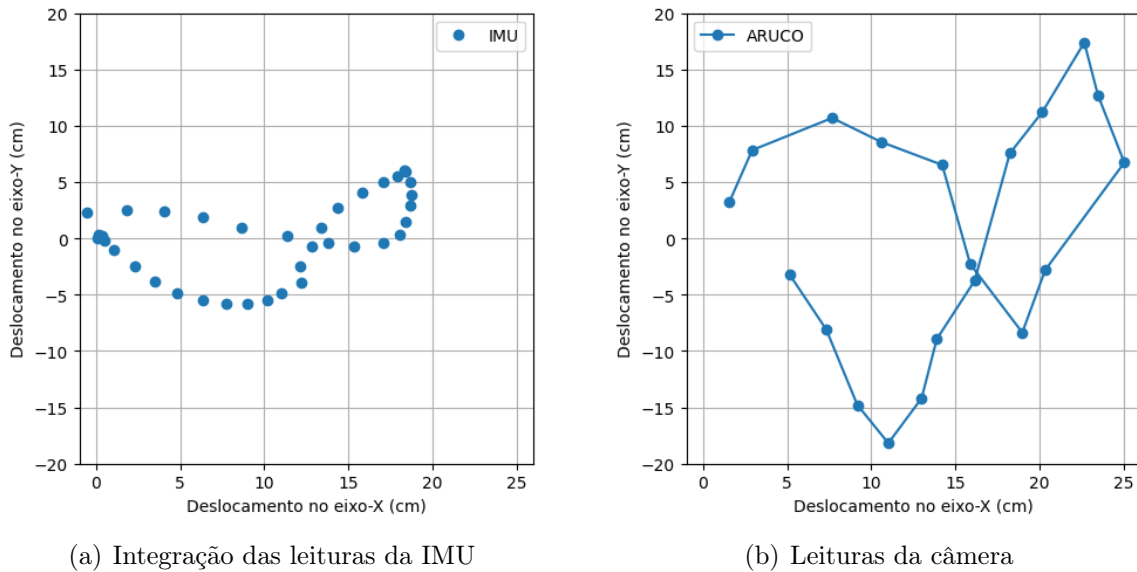


Figura 4.3:
Leituras da IMU e câmera deslocamento seguindo a Lemniscata de Bernoulli
Fonte: Autor

Um percurso mais complexo, como o terceiro cenário, exige um maior refinamento das configurações do robô. Na imagem (a) da figura 4.3, os erros na leitura da IMU são visualizados em sua integração, esta que é cumulativa, e necessita de dados mais lineares para melhor representar o deslocamento. Como os dados obtidos a partir da MPU-6050 são muito irregulares (como exemplo na figura 4.4), um estudo maior na área de calibração ou a substituição por um sensor mais preciso seria imprescindível para trajetos complexos.

Quanto aos dados obtidos pela câmera, imagem (b) figura 4.3, são mais precisos quanto ao real movimento realizado pelo robô, apesar do deslocamento não ser suave em detrimento da menor frequência de leitura e possuir algumas variações no posicionamento devido a sua precisão, visto que um pequeno desvio de centímetros se torna prejudicial ao rastreamento em razão da escala em que o experimento é realizado.

4.4 Discussões

A princípio, é preciso ressaltar que o estudo e exercício deste trabalho foi em analisar os efeitos do filtro de Kalman, portanto, as execuções dos percursos não foram exatas e isto ocorreu pela falta de um sistema de controle capaz de corrigir o deslocamento do robô durante o trajeto, tendo em vista que as rodas holonômicas causam uma movimentação mais irregular, por conta do seus rolamentos perpendiculares ao eixo de rotação da roda.

Como a IMU, MPU-6050, utilizada no projeto é muito sensível e resulta em dados muito ruidosos foi necessário fazer uma calibração, porém, para trabalhos mais avançados é preciso a aplicação de algum filtro para melhorar o resultado. A figura 4.4 exemplifica

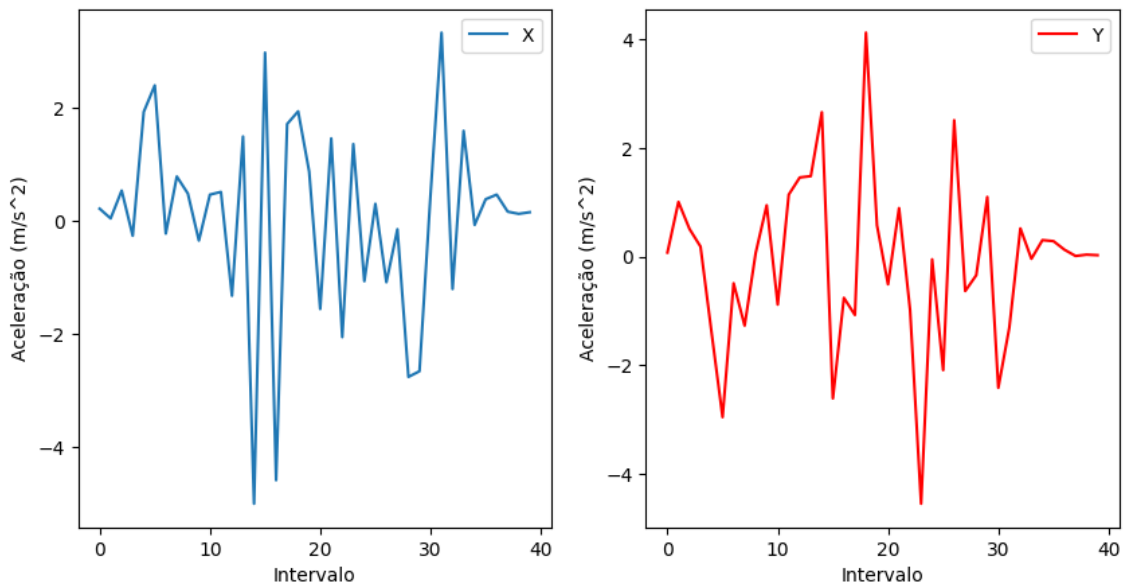


Figura 4.4:
 Acelerações no eixo x e y captadas da IMU MPU-6050 durante o percurso em L.
Fonte: Autor

bem esse problema, comparando com imagem *a* da figura 4.2, era esperado que no intervalo $[0 - 20]$ da aceleração no eixo y apresentasse valores mais próximos a 0, o mesmo se repete na aceleração em x mas no intervalo $[20 - 40]$.

Outro ponto a se analisar, é o erro de estimação da câmera em relação aos marcadores fiduciais. Assim como acontece com o GPS, que possui uma precisão de 5 a 10 metros nos modelos comerciais, ao detectar a posição do robô com as *tags* ArUco pode ocorrer dessa posição estar deslocada em alguns centímetros, evento perceptível nas imagens (b) das figuras 4.2 e 4.3. Os erros podem acontecer por diversos fatores, como exemplo: oclusão, movimento da câmera muito rápido, condições de iluminação ruim, entre outros. Para extrair dados mais precisos dos marcadores, um foco no aprimoramento dos parâmetros da câmera poderia ser realizado. Outra solução, é o uso de uma malha de *tags* ArUco, aumentando o seu campo de visão, mas impactando no desempenho do sistema, como descrito por Garrido-Jurado et al. (2014).

Através desses experimentos, foi possível observar a importância da modelagem do problema. Encontrar as matrizes de covariância das incertezas de predição e medição - no experimento, representado pelos sensores IMU e câmera - é o ponto chave, dado que o filtro de Kalman, em sua versão linear, trabalha somente com ruídos gaussianos.

O primeiro cenário ilustra bem esse efeito. Para as configurações definidas na seção 3.2.3, o resultado obtido mantém relação com o deslocamento gerado pela integração das acelerações do sensor inercial, possível ver na imagem (a) da figura 4.5 no ponto 20 do eixo-x, sendo o exato momento em que o robô sofre uma variação na direção. Ainda, o rastreamento foi corrigido pela linearidade da detecção da câmera, que para este cenário

funcionou na mesma frequência da IMU, apesar de ter sua dimensões reduzidas pelo fato da detecção da *tag* ArUco ocorrer um pouco depois do início do movimento do robô.

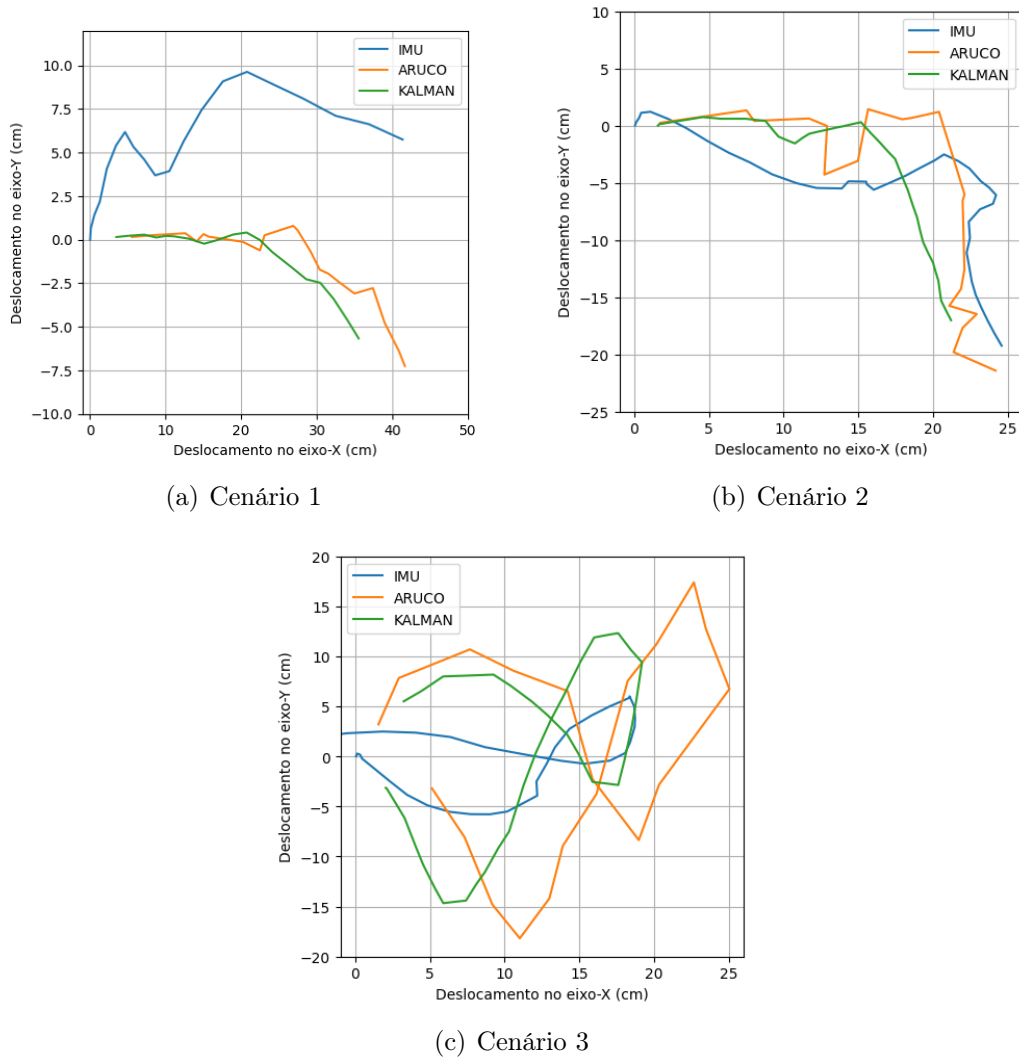


Figura 4.5:
Comparação entre IMU, câmera e aplicação do filtro
Fonte: Autor

Para executar o filtro de Kalman, em específico para o terceiro cenário, foi necessário alterar os valores da matriz de covariância de observação de $I_3 \cdot 0.1$ para $I_3 \cdot 0.01$, isto pelo fato de que os valores utilizados nos experimentos anteriores não foram suficientes para corrigir as leituras da IMU. Já valores menores que 0.01 tornavam o resultado do filtro muito próximo ao que era apresentado pela leitura dos marcadores ArUco.

Apesar da dificuldade em realizar esse cenário, a resposta obtida foi positiva. O rastreamento da posição foi corrigido com os dados da câmera durante a etapa de atualização do filtro de Kalman, enquanto que a predição usando a IMU foi capaz de amenizar os pontos de imprecisão, devido a sua maior frequência, como pode ser visto na imagem (c) da figura 4.5.

Observando a figura 4.5, é possível inferir que a execução do caminho em L foi a que obteve melhores resultados (imagem (b)). O cenário 2 une a simplicidade do trajeto linear, como visto no cenário 1, com uma mudança no sentido da direção, diminuindo o erro cumulativo.

Os resultados, em geral, se revelaram bem satisfatórios e se mostraram bem próximos da realidade, dado que o objetivo do estudo foi aplicar o filtro de Kalman utilizando sensores de baixo custo permitindo a sua integração em qualquer dispositivo compatível. Sua facilidade de implementação e robustez quanto a observações incertas se mostram pontos fortes e refletem o bom desempenho em sua execução.

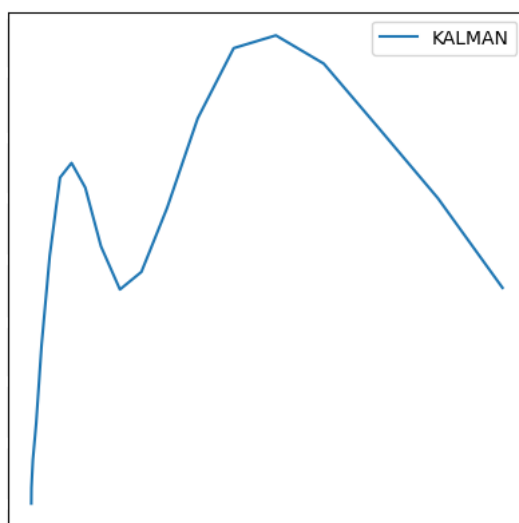


Figura 4.6:
Aplicação do filtro de Kalman utilizando somente a IMU

Fonte: Autor

Entretanto, utilizar somente as leituras da IMU sem o passo de observação do filtro de Kalman acaba não sendo tão eficiente como visto na figura 4.6, em que para o caminho linear houve somente uma suavização do traçado quando comparado com a imagem (a) da figura 4.5, comprovando que a técnica de rastreamento de posição não é indicada para longos percursos, ainda mais quando está atuando sem uma técnica auxiliar.

Capítulo 5

Considerações finais

Neste trabalho, foi apresentado uma implementação do Filtro de Kalman em um robô holonômico, realizando uma fusão dos sensores unidade de medida inercial MPU-6050 e câmera Raspiberry Pi, com objetivo de estimar a sua posição ao longo de percursos predeterminados.

Os resultados obtidos se mostraram muito satisfatórios. Nos cenários 1 (3.3.1) e 2 (3.3.2), ficou evidente a facilidade de rastrear a posição quando se possui trajetos mais simples de se executar. No cenário 3 (3.3.3), apesar do filtro não resultar em uma saída tão precisa quanto nos outros cenários, ainda se mostrou bem eficaz. O experimento, como um todo, se mostra muito interessante quando se faz necessário pequenos ajustes nas matrizes de covariância de medição e observação, representando bem como seria a modelagem para situações diferentes, apesar da utilização dos mesmos sensores, como exemplo a atuação de um robô em diferentes tipos de solo.

Em relação às limitações, a falta de uma matriz de covariância fornecida diretamente pelo fabricante do módulo GY-521 [InvenSense (2012)] pode ter sido um fator negativo durante as medições da IMU, tendo em vista que suas leituras possuíam muitos ruídos. Uma outra dificuldade, foi a falta de um sistema de controle para gerenciar o módulo de planejamento do robô, corrigindo seu deslocamento, e conseqüentemente captar leituras as mais próximas possíveis do caminho planejado, devido aos erros de locomoção como o deslizar e derrapar das rodas, já que o foco do trabalho foi em analisar os efeitos do filtro de Kalman,

Como trabalhos futuros, um sistema de controle deve ser adicionado para corrigir o trajeto que o robô percorra, fazendo com que as observações da IMU e da câmera sejam as mais reais possíveis. Em seguida, deve ser estudado técnicas de localização e mapeamento que façam proveito dos sensores disponíveis no Omnino com objetivo de implementar o desvio de obstáculo e torná-lo autônomo.

Bibliografía

- Alatise, M. B. and Hancke, G. P. (2020). A review on challenges of autonomous mobile robot and sensor fusion methods. *IEEE Access*, 8:39830–39846.
- Borenstein, J., Everett, H. R., Feng, L., and Wehe, D. K. (1997). Mobile robot positioning - sensors and techniques. *Journal of Robotic Systems*, 14:231–249.
- Choset, H., Lynch, K. M., Hutchinson, S., Kantor, G. A., and Burgard, W. (2005). *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press.
- Corke, P. (2011). *Robotics, Vision and Control: Fundamental Algorithms in MATLAB*. Springer.
- Francisco Rubio, F. V. and Llopis-Albert, C. (2019). A review of mobile robots: Concepts, methods, theoretical framework, and applications. *International Journal of Advanced Robotic Systems*, 16.
- Garcia, E., Jimenez, M. A., De Santos, P. G., and Armada, M. (2007). The evolution of robotics research. *IEEE Robotics Automation Magazine*, 14(1):90–103.
- Garrido-Jurado, S., Muñoz-Salinas, R., Madrid-Cuevas, F., and Marín-Jiménez, M. (2014). Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292.
- Hashemi, M. and Karimi, H. A. (2014). A critical review of real-time map-matching algorithms: Current issues and future directions. *Computers, Environment and Urban Systems*, 48:153–165.
- InvenSense (2012). Mpu-6000 and mpu-6050 product specification revision 3.3. <https://www.haoyuelectronics.com/Attachment/GY-521/>.
- Lynch, K. M. and Park, F. C. (2017). *Modern Robotics: Mechanics, Planning, and Control*. Cambridge University Press.
- Nava Rodriguez, N., Cabás Ormaechea, L. M., Cabás, R., Gonzalez, A., Ottaviano, E., and JARDÓN HUETE, A. (2011). *Advanced Mechanics in Robotic System*.

- OpenCV (2012). Detection of aruco markers. https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html.
- Panigrahi, P. K. and Bisoy, S. K. (2022). Localization strategies for autonomous mobile robots: A review. *Journal of King Saud University - Computer and Information Sciences*, 34(8, Part B):6019–6039.
- Pi, R. (2013). Raspberry pi camera module. <https://docs.rs-online.com/2888/0900766b8127db0a.pdf>.
- Rahmatullah, B., Zaidan, A. A., Mohamed, F., and Sali, A. (2017). Multi-complex attributes analysis for optimum gps baseband receiver tracking channels selection. In *2017 4th International Conference on Control, Decision and Information Technologies (CoDIT)*, pages 1084–1088.
- Raj, R. and Kos, A. (2021). Fiducial markers for pose estimation. *Journal of Intelligent Robotic Systems*, 101.
- Raj, R. and Kos, A. (2022). A comprehensive study of mobile robot: History, developments, applications, and future research perspectives. *Applied Sciences*, 12(14).
- Ramalho, J. H. V. (2020). Omnino: plataforma aberta de robô omnidirecional para enxame.
- Roland Siegwart, I. R. N. and Scaramuzza, D. (2011). *Introduction to Autonomous Mobile Robots*. MIT Press.
- ROS (2021). Ros - robot operating system. <https://www.ros.org>.
- Siciliano, B., Sciavicco, L., Villani, L., and Oriolo, G. (2009). *Robotics, Modelling, Planning and Control*. Springer.
- Wang, Y., Jiang, S., Yan, F., Gu, L., and Chen, B. (2017). A new redundancy resolution for underwater vehicle–manipulator system considering payload. *International Journal of Advanced Robotic Systems*, 14(5):1729881417733934.
- Waslander, S. and Kelly, J. (2019). State estimation and localization for self-driving cars. <https://www.coursera.org/learn/state-estimation-localization-self-driving-cars>.