



UNIVERSIDADE FEDERAL DO MARANHÃO  
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA  
BACHARELADO INTERDISCIPLINAR EM CIÊNCIAS E TECNOLOGIA

THIAGO MATHEUS SOARES DE SOUSA

**MAPEAMENTO DE AMBIENTE BASEADO EM LiDAR:**

**Introdução a técnicas de estimativa de posição e construção de mapas fixos**

SÃO LUÍS  
2025



THIAGO MATHEUS SOARES DE SOUSA

**MAPEAMENTO DE AMBIENTE BASEADO EM LiDAR:**

**Introdução a técnicas de estimativa de posição e construção de mapas fixos**

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado Interdisciplinar em Ciências e Tecnologia da Universidade Federal do Maranhão, Campus Bacanga, como requisito parcial para obtenção do grau de Bacharel em Ciências e Tecnologia.

**Orientador: Prof. Dr. Denis Fabrício de Sousa Sá**

**Coorientador: Prof. Me. Gustavo Araujo de Andrade**

SÃO LUÍS  
2025

---

THIAGO MATHEUS SOARES DE SOUSA

**MAPEAMENTO DE AMBIENTE BASEADO EM LIDAR:**

**Introdução a técnicas de estimativa de posição e construção de mapas fixos**

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado Interdisciplinar em Ciências e Tecnologia da Universidade Federal do Maranhão, Campus Bacanga, como requisito parcial para obtenção do grau de Bacharel em Ciências e Tecnologia.

Aprovado em 7 de Março de 2025.

**Banca Examinadora**

---

Prof. Dr. Denis Fabrício de Sousa Sá, Engenharia da Computação (UFMA) – Orientador.

---

Prof. Me. Gustavo Araújo de Andrade, Engenharia Elétrica (UFMA) - Coorientador.

---

Prof. Dr. Pedro Baptista Fernandes, Engenharia da Computação (UFMA).

---

Prof. Me. Marcio Mendes Cerqueira, Engenharia da Computação (UFMA).

---

Ficha gerada por meio do SIGAA/Biblioteca com dados fornecidos pelo(a) autor(a).  
Diretoria Integrada de Bibliotecas/UFMA

de Sousa, Thiago Matheus Soares.

MAPEAMENTO DE AMBIENTE BASEADO EM LiDAR : introdução a técnicas de estimativa de posição e construção de mapas fixos / Thiago Matheus Soares de Sousa. - 2025.

71 p.

Coorientador(a) 1: Gustavo Araújo de Andrade.

Orientador(a): Denis Fabrício de Sousa Sá.

Curso de Ciência e Tecnologia, Universidade Federal do Maranhão, São Luís, 2025.

1. Lidar. 2. Mapeamento Em Grade. 3. Filtro de Kalman Estendido. 4. Slam. I. de Andrade, Gustavo Araújo. II. Sá, Denis Fabrício de Sousa. III. Título.

---

*Aos que fizeram parte dessa caminhada... obrigado.*

---

## AGRADECIMENTOS

Primeiramente à minha família, aos meus amigos, a aqueles que, em dado momento, compartilharam da sua vida comigo.

Aos meus Orientadores Prof. Dr. Denis Sá e Prof. Me. Gustavo Andrade, por terem me dado a oportunidade de realizar esse trabalho e terem compartilhado de seu conhecimento e recursos, em especial, ao Professor Denis por ter dedicado seu tempo e esforço na conclusão deste projeto.

À UFMA, em especial às pessoas que compõe o LABSECI, que quando necessário se fizeram disponíveis.

Outros agradecimentos, ficarão para trabalhos futuros, obrigado!!!

---

## RESUMO

Na robótica móvel, um dos principais desafios é a estimativa de pose e na construção de mapas online confiáveis. Este trabalho investiga o mapeamento de ambientes estáticos utilizando um sistema de fusão sensorial baseado em LiDAR de baixo custo e a aplicação do Filtro de Kalman Estendido. O estudo propõe uma abordagem integrada para a estimativa de posição e construção de mapas de ocupação, com introdução em técnicas de Mapeamento e Localização Simultaneos (SLAM). O robô móvel estabelecido envolve um sensor YDLiDAR X4 PRO e uma unidade inercial MPU6050, que enviam dados coletados a partir de um sistema embarcado a um computador via Wi-Fi para processamento. O EKF é utilizado para aprimorar a estimativa da pose do robô ao combinar a odometria com as medições do LiDAR, o mapa do ambiente é gerado por meio de uma grade de ocupação, utilizando o modelo probabilístico *log-odds* para atualizar as células conforme novas medições são incorporadas. Os experimentos demonstram que o erro médio absoluto do LiDAR foi de aproximadamente 24,67 (mm) no cenário A e 16,00 (mm) no cenário B, indicando uma maior estabilidade no segundo cenário, a análise dos resultados evidencia a influência do ajuste do ganho do EKF na precisão do mapa gerado, mostrando que a fusão de dados sensoriais pode reduzir a incerteza da posição do robô e melhorar a representação do ambiente. Conclui-se que a integração entre técnicas de localização probabilística e sensores de baixo custo viabiliza, dado um certo refino, um mapeamento mais robusto e acessível para aplicações em robótica móvel.

**Palavras-chave:** LiDAR. Mapeamento em Grade. Filtro de Kalman Estendido. SLAM.

---

## ABSTRACT

In mobile robotics, one of the main challenges is pose estimation and the construction of reliable online maps. This work investigates the mapping of static environments using a low-cost LiDAR based sensor fusion system and the application of the Extended Kalman Filter (EKF). The study proposes an integrated approach for position estimation and occupancy grid mapping, introducing techniques in Simultaneous Localization and Mapping (SLAM). The developed mobile robot incorporates a YDLiDAR X4 PRO sensor and an MPU6050 inertial unit, which transmit collected data from an embedded system to a computer via Wi-Fi for processing. The EKF is used to enhance the robot pose estimation by combining odometry with LiDAR measurements. The environment map is generated through an occupancy grid, using the probabilistic log-odds model to update the cells as new measurements are incorporated. Experiments demonstrate that the absolute mean error of the LiDAR was approximately 24.67 (mm) in scenario A and 16 (mm) in scenario B, indicating greater stability in the second scenario. The analysis of the results highlights the influence of EKF gain tuning on the accuracy of the generated map, showing that sensor data fusion can reduce the robot position uncertainty and improve environmental representation. It is concluded that the integration of probabilistic localization techniques and low-cost sensors enables, with proper refinement, more robust and accessible mapping for mobile robotics applications.

**Keywords:** LiDAR. Grid Mapping. Extended Kalman Filter. SLAM.

---

## LISTA DE FIGURAS

Figura 1 - Referencial global e local do robô.....	24
Figura 2 - Deslocamento do robô móvel diferencial.....	15
Figura 3 - Representação do sensor IMU.....	27
Figura 4 - LiDAR 2D.....	28
Figura 5 - <i>Encoder</i> Incremental.....	30
Figura 6 - Representação Mapa em Grade.....	32
Figura 7 - Fluxo geral do Filtro de Kalman.....	32
Figura 8 - Robô Móvel utilizado.....	39
Figura 9 - Atuadores.....	40
Figura 10 - Ponte H LN298N.....	40
Figura 11 - YDLIDAR X4 PRO.....	44
Figura 12 - MPU6050.....	42
Figura 13 - LM393.....	42
Figura 14 - ESP32.....	43
Figura 15 - Fluxograma de comunicação do ESP32.....	44
Figura 16 - Arduino UNO.....	44
Figura 17 - Fluxograma de atuação do Arduino.....	45
Figura 18 – Diagrama de conexões elétricas.....	46
Figura 19 - Fluxograma de processamento dos dados.....	47

---

Figura 20 - Cenário A.....	53
Figura 21 - Cenário A parado.....	54
Figura 22 - Trajetória Prevista Cenário A.....	55
Figura 23 - Trajetória Corrigida Cenário A.....	55
Figura 24 - Cenário B.....	56
Figura 25 - Cenário B parado.....	56
Figura 26 - Trajetória Prevista Cenário B.....	58
Figura 27 - Trajetória Corrigida Cenário B.....	58

---

## LISTA DE ABREVIATURAS E SIGLAS

GPS	Sistema de Posicionamento Global
IMU	Unidade de Medida Inercial
SLAM	Localização e Mapeamento Simultâneos
LiDAR	<i>Light Detection and Ranging</i>
2D	Duas Dimensões
3D	Três Dimensões
EKF	Filtro de Kalman Estendido
RPM	Rotações Por Minuto
UART	<i>Universal Asynchronous Receiver/Transmitter</i>
Wi-Fi	<i>Wireless Fidelity</i>
TCP	<i>Transmission Control Protocol</i>
PWM	<i>Pulse Width Modulation</i>
MAE	Erro Absoluto Médio

---

## LISTA DE SÍMBOLOS

$(x, y, z)$	Coordenadas no espaço 3D
$(roll, pitch, yaw)$	Ângulos de Euler
$\theta$	Ângulo
$(X_I, Y_I)$	Referencial global do sistema
$(X_R, Y_R)$	Referencial local do sistema
$P_R$	Ponto local
$(x_R, y_R)$	Conjunto de pontos locais
$P_I$	Ponto Global
$(x_I, y_I)$	Conjunto de pontos globais
$v$	Velocidade
$v_r$	Velocidade da roda direita
$v_l$	Velocidade da roda esquerda
$\omega$	Velocidade angular
$L$	Distância entre as rodas
$x_t$	Posição de x no instante t
$y_t$	Posição de y no instante t

---

$\theta_t$	Ângulo no instante t
$\theta_{gyro}$	Ângulo do giroscópio
$\omega_{gyro}$	Velocidade angular do giroscópio
$w_{bias}$	Erro do giroscópio
$\theta_k$	Ângulo no instante k
$\Delta t$	Varição de tempo
$\theta_{k-1}$	Ângulo no instante anterior a k
$d$	Distância até um objeto
$b$	Ponto de recepção da luz
$\theta_{sensor}$	Ângulo do sensor
$\omega_{roda}$	Velocidade angular da roda
$P(A B)$	Probabilidade do evento A dado o evento B
$P(B A)$	Probabilidade do evento B dado o evento A
$P(A)$	Probabilidade do evento A
$P(B)$	Probabilidade do evento B
$P(m_i z_{1:t}, x_{1:t})$	Probabilidade condicional de uma célula do mapa

---

$l_{t,i}$	Log-odds da ocupação da célula i no instante t
$\log$	Função logaritmo
$x_t$	Vetor de previsão do Filtro Kalman
$A_t$	Matriz de transição de estado
$B_t$	Matriz de controle
$u_t$	Vetor de controle
$\varepsilon_t$	Ruido do processo
$\bar{x}_t$	Vetor de atualização do Filtro de Kalman
$K_t$	Ganho de Kalman
$z_t$	Vetor de observação do instante t
$C_t$	Matriz de observação
$\hat{x}_k$	Previsão do vetor de estado no instante EKF
$f(\hat{x}_{k-1}, u_k)$	Função de transição de estado
$\omega_k$	Ruido do processo
$\hat{P}_k$	Previsão de covariância EKF
$F_k$	Jacobiana da função de transição de estado
$F_k^T$	Transposta de $F_k$

---

$Q_k$	Matriz de covariância do ruído
$\partial f(x_k, u_k)$	Derivada parcial da função estado
$\partial x_k$	Derivada em relação ao vetor de entrada
$\hat{z}_k$	Previsão da observação
$h(\hat{x}_k)$	Função de observação do sensor
$K_k$	Ganho de Kalman EKF
$H_k^T$	Transposta da Jacobiana de previsão
$S_k^{-1}$	Inversa da inovação no instante k
$S_k$	Covariância de inovação
$H_k$	Jacobiana da previsão
$R_k$	Matriz de covariância do ruído
$\hat{x}_k^-$	Atualização da previsão de estado
$\hat{P}_k^-$	Atualização da previsão de covariância
$I$	Matriz identidade
$x_{lidar}$	Ponto em x da observação do sensor LiDAR
$r_{lidar}$	Distância da observação do sensor LiDAR
$y_{lidar}$	Ponto em y da observação do sensor LiDAR

---

$T$	Matriz homogênea 4x4
$T_{acumulado}$	Deslocamento acumulado ICP
$r_{mapa}$	Resolução do mapa em grade
$i$	Índice i
$j$	Índice j
$\Delta x$	Varição no espaço x
$\Delta y$	Varição no espaço y
$x_{final}$	Medição de x final
$x_{inicial}$	Medição de x inicial
$y_{final}$	Medição de y final
$y_{inicial}$	Medição de y inicial
$L_{t+1}$	Confiança no instante t da célula i

---

## SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>19</b>
<b>1.1 Objetivos.....</b>	<b>21</b>
<b>1.1.1 Objetivo Geral.....</b>	<b>21</b>
<b>1.1.2 Objetivos Específicos.....</b>	<b>21</b>
<b>1.2 Justificativa.....</b>	<b>21</b>
<b>1.3 Estrutura do Trabalho .....</b>	<b>22</b>
<b>2 REPRESENTAÇÃO ESPACIAL E SENSORIAMENTO.....</b>	<b>23</b>
<b>2.1 Representação no espaço: Posição e Orientação .....</b>	<b>23</b>
<b>2.2 Cinemática.....</b>	<b>25</b>
<b>2.3 Sensores.....</b>	<b>26</b>
<b>2.3.1 Sensores de Unidade de Medida Inercial (IMU) .....</b>	<b>27</b>
<b>2.3.2 LiDAR.....</b>	<b>28</b>
<b>2.3.3 Encoders.....</b>	<b>29</b>
<b>3 MÉTODOS PARA MAPEAMENTO E POSICIONAMENTO.....</b>	<b>31</b>
<b>3.1 Teorema de Bayes.....</b>	<b>31</b>
<b>3.1.2 Mapeamento em Grade.....</b>	<b>31</b>
<b>3.2 Filtro Gaussiano.....</b>	<b>34</b>
<b>3.2.1 Filtro de Kalman.....</b>	<b>34</b>
<b>3.2.2 Filtro de Kalman Estendido.....</b>	<b>36</b>
<b>4 HARDWARE.....</b>	<b>39</b>
<b>4.1 Estrutura e locomoção.....</b>	<b>40</b>
<b>4.2 Sensores.....</b>	<b>41</b>
<b>4.2.1 YDLiDAR X4 PRO.....</b>	<b>41</b>
<b>4.2.2 MPU6050.....</b>	<b>42</b>
<b>4.2.3 LM393.....</b>	<b>42</b>
<b>4.2.4 Microcontroladores .....</b>	<b>43</b>
<b>4.2.5 Conexão Elétrica.....</b>	<b>45</b>
<b>5 MODELAGEM DO SISTEMA.....</b>	<b>47</b>
<b>5.1 Tratamento de Pontos.....</b>	<b>47</b>
<b>5.2 Iterative Closest Point (ICP) .....</b>	<b>48</b>
<b>5.3 Modelo EKF.....</b>	<b>49</b>

---

<b>5.4 Configuração do Mapa.....</b>	<b>50</b>
<b>5.5 Atualização do Mapa .....</b>	<b>51</b>
<b>5.6 Construção do Mapa .....</b>	<b>52</b>
<b>6 EXPERIMENTOS E RESULTADOS.....</b>	<b>53</b>
<b>6.1 Cenário A.....</b>	<b>53</b>
<b>6.2 Cenário B.....</b>	<b>56</b>
<b>6.3 Análise de resultados .....</b>	<b>59</b>
<b>7 CONCLUSÃO .....</b>	<b>60</b>
<b>REFERENCIAS</b>	
<b>ANEXO A - YDLiDAR X4 PRO</b>	
<b>APÊNDICES A - TRANSFORMAÇÕES DE PONTOS NO PLANO</b>	
<b>APÊNDICES B - INTERATIVE CLOSEST POINT</b>	
<b>APÊNDICES C - EXTENDED KALMAN FILTER</b>	
<b>APÊNDICES D - GRID MAPPING</b>	
<b>APÊNDICES E - <i>LOG-ODDS</i></b>	
<b>APÊNDICES F - REPRESENTAÇÃO NO ESPAÇO</b>	

---

## 1 INTRODUÇÃO

No decorrer dos anos a robótica demonstrou extenso potencial em atuar nas mais diversas áreas da sociedade, dentre elas, aplicações industriais focadas na otimização, logística e transporte de materiais (Niku, 2013), no setor agrícola, para monitoramento e manutenção de plantações (Jiang et al., 2022), na melhor eficiência e segurança de veículos autônomos (Van Brummelen et al., 2018) e até a atuação em ambientes totalmente desconhecidos, como exploração planetária (Rekleitis et al., 2009).

Na robótica móvel a eficácia em perceber e interpretar o ambiente a qual se está inserido se torna primordial, uma vez que um dos seus principais desafios é a capacidade de operar em ambientes dinâmicos e pouco estruturados. Isto ocorre porque a incerteza associada as medições dos sensores e limitações dos atuadores acabam por interferir na confiança a respeito do espaço, impactando diretamente a precisão das ações realizadas (Thrun et al., 2006).

Em ambientes em que o acesso é restrito, com ausência e limitação de recursos, como Sistema de Posicionamento Global (GPS) e interferência em Sensores de Unidade Inercial (IMU), dificultam a navegação de robôs móveis (Cruz Júnior, 2021). Para tratar com essas limitações, o estudo de algoritmos e técnicas de Mapeamento e Localização Simultâneos (SLAM) surge como uma solução importante, permitindo a construção de um mapa do ambiente enquanto estima sua própria posição (Thrun et al., 2006).

Abordagens fundamentais em SLAM, são amplamente usadas com técnicas de fusão sensorial, baseadas em lasers, câmeras, *encoders*, giroscópios, acelerômetros, ultrassônicos etc. (Xiang et al., 2020). Do ponto de vista teórico o SLAM pode-se considerar resolvido, entretanto, em questões práticas, o método enfrenta desafios devido a sensibilidade que esses sensores podem oferecer, e o acúmulo de erros ao longo do tempo, aplicação em cenários reais ainda apresenta empecilhos com a necessidade de construir mapas ricos em detalhes, a escolha e qualidade dos sensores e algoritmos de filtragem tornam-se cruciais para garantir melhores resultados (Durrant-Whyte; Bailey, 2006).

Dentre os sensores usados para a percepção do ambiente, o *Light Detection and Ranging* (LiDAR) possui relevância devido a sua capacidade de fornecer medições bidimensionais (2D) ou tridimensionais (3D) com uma certa precisão da superfície de um objeto. Este sensor opera por intermédio da emissão e recepção de pulsos laser, que servem para mensurar a distância, a

---

partir do tempo decorrido ou o ângulo formado, após serem refletidos por um objeto (McManamon, 2019).

Modelos de filtragem probabilística são tradicionalmente utilizados na construção de mapas, como nas técnicas de Mapeamento em Grade, que se baseiam no Filtro de Bayes para representar o ambiente detectado. Nesse método, o ambiente é modelado por meio de uma matriz, em que cada célula armazena a probabilidade de ocupação de um conjunto de pontos, indicando se a região está livre, ocupada ou em estado desconhecido (Thrun et al., 2006).

Para determinar sua orientação e posição com precisão, o robô depende de métodos robustos, como o Filtro de Kalman Estendido (EKF), ajuste do modelo original para comportar sistema não lineares (Thrun et al., 2006). Esta técnica beneficia-se da fusão sensorial de sensores como LiDAR, IMU e de dados referentes a odometria, a fim de compensar as limitações individuais de cada sensor, aumentando a confiança de sua pose no ambiente (Cruz Júnior, 2021).

Segundo Nehmzow (2003) a construção de um mapa e a autolocalização são dois dos três pilares fundamentais da estrutura de um robô móvel. Nesse contexto, o trabalho em questão, propõe-se a desenvolver um sistema de mapeamento móvel utilizando técnicas de LiDAR-SLAM, com introdução a aplicação do filtro de Kalman Estendido, para estimativa de posição, integração das informações do LiDAR, IMU, odometria das rodas para verificar a construção de um mapa 2D em um ambiente estático.

## 1.1 Objetivos

### 1.1.1 Objetivo Geral

- Validar o mapeamento de ambientes estáticos, empregando técnicas LiDAR-SLAM, com o Filtro de Kalman Estendido para a fusão de sensores ruidosos.

### 1.1.2 Objetivos Específicos

- Validar o desempenho de sensores de baixo custo, como YDLiDAR X4 PRO, na coleta e processamento de dados para a construção de mapas do ambiente.
- Desenvolver um sistema integrado entre robô e computador para o mapeamento de ambiente.
- Desenvolver um robô de baixo custo para mapeamento.
- Desenvolver um ambiente virtual de integração com o sistema robótico.

## 1.2 Justificativa

As técnicas de mapeamento móvel e localização simultânea (SLAM) são abordagens amplamente utilizadas na robótica (Thrun et al., 2006), mas a sua implementação muitas vezes esbarra nos altos custos dos sensores e sistemas necessários. Este trabalho propõe uma aplicação acessível, utilizando sensores de baixo custo, como YDLiDAR X4 PRO e a unidade inercial MPU6050, para realizar mapeamento e estimativas de posição. A escolha dos componentes com preços reduzidos torna este sistema viável para pesquisas acadêmicas.

De acordo com Brynjolfsson e McAfee (2014), a redução contínua dos custos das tecnologias tem permitido o acesso crescente a soluções avançadas, tornando-as mais acessíveis e aplicáveis a uma gama maior de usuários. Ao optar por componentes de baixo custo, busca-se demonstrar que é possível implementar soluções fundamentadas em SLAM com recursos limitados, tornando essas tecnologias mais viáveis na prática.

---

Na Tabela 1 é destacado os principais componentes utilizados neste trabalho e seus valores aproximados (Informações obtidas no dia 8/03/2025).

Tabela 1 – Custo dos componentes

<i>Produto</i>	<i>Preço (R\$)</i>
<i>YDLiDAR X4 PRO</i>	617,15
<i>Esp32</i>	61,72
<i>Arduino Uno</i>	43,86
<i>MPU 6050</i>	18,32
<i>Outros</i>	300

Autor (2025)

### 1.3 Estrutura do Trabalho

O seguinte trabalho é estruturado em 7 Capítulos, a qual o Capítulo 2 apresenta os conceitos fundamentais de representação espacial e o detalhamento dos sensores utilizados. O Capítulo 3 aborda os principais métodos de mapeamento e posicionamento, incluindo o Teorema de Bayes, Filtros Gaussianos e Filtro de Kalman Estendido. No Capítulo 4, é descrita a arquitetura do hardware utilizado no robô, incluindo motores, sensores e microcontroladores. O Capítulo 5 trata da modelagem do sistema, abordando a transformação de pontos, alinhamento com ICP e a construção do mapa de ocupação. Os experimentos e resultados são discutidos no Capítulo 6, e as considerações finais são apresentadas no Capítulo 7.

---

## 2 REPRESENTAÇÃO ESPACIAL E SENSORIAMENTO

A capacidade de um robô móvel se deslocar de maneira eficiente depende diretamente de uma modelagem precisa de sua posição e orientação no espaço, juntamente com coleta constante de informações sobre o ambiente. Para isso, a representação espacial e o uso de sensores são fundamentais para estabelecer a sua localização e interação eficaz com o meio (Siegwart; Nourbakhsh, 2004).

### 2.1 Representação no espaço: Posição e Orientação

Uma das definições primordiais conhecidas na robótica, é o de pose. A pose pode ser detalhada como um conjunto de pontos que traduz a posição e a orientação de um corpo rígido em um sistema de coordenadas, comumente definidos por seis variáveis, sendo elas, as coordenadas cartesianas  $(x, y, z)$  e os ângulos de Euler (*roll, pitch, yaw*) (Thrun et al., 2006).

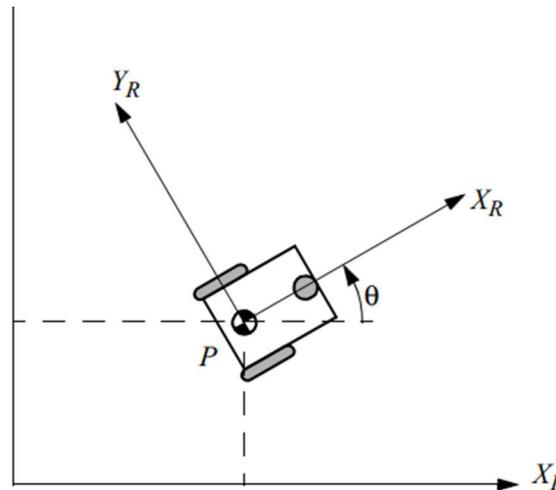
Essa representação é essencial em aplicações como drones e manipuladores robóticos, que operam no espaço tridimensional e usufruem principalmente dos ângulos que definem a orientação (Niku, 2013). Todavia, para robôs que se deslocam ao longo de superfícies no espaço bidimensional, tal como um carro motorizado, pode-se simplificar a representação da pose, definindo-a como um conjunto de coordenadas  $(x, y)$  e a sua orientação  $(\theta)$  (Thrun et al., 2006) como descrito na Equação (1):

$$Pose = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (1)$$

Para Siegwart e Nourbakhsh (2004) descrever com precisão o movimento de um robô no ambiente a qual está inserido, assim como, definir sua pose em diferentes tipos de referenciais, seja em relação a um sistema de coordenadas locais, globais ou a um objeto específico no mapa, permite melhor entender a interação que o sistema tem com o ambiente que o cerca.

Como demonstrado na Figura 1, o dado conjunto de coordenadas  $(X_I, Y_I)$  é a expressão que define o referencial global no ambiente, utilizada para definir a posição absoluta do robô, já o conjunto  $(X_R, Y_R)$  reflete a posição relativa a respeito do mundo, o referencial local, sendo este preso ao seu movimento. A conversão entre os referenciais é realizada utilizando os princípios da transformação homogênea, que incorpora o modelo das transformações de rotação e translação.

Figura 1 - Referencial global e local do robô



Siegwart; Nourbakhsh (2004, p. 49)

Dessa forma, através da Equação (2) é demonstrada a aplicação de um ponto local  $P_R = (x_R, y_R)$  a fim de obter seu referencial global, o ponto  $P_I = (x_I, y_I)$ .

$$\begin{bmatrix} x_I \\ y_I \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x_R \\ y_R \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix} \quad (2)$$

Expandido a Equação (2), tem-se que:

$$\begin{cases} x_I = \cos \theta \cdot x_R - \sin \theta \cdot y_R + x \\ y_I = \sin \theta \cdot x_R + \cos \theta \cdot y_R + y \end{cases} \quad (3)$$

No contexto de robótica móvel, a simplificação da pose permite diminuir custos computacionais, tornando o processo mais direto, sem perder a precisão de sua posição, garantindo que o robô tenha sua localização de forma mais eficiente.

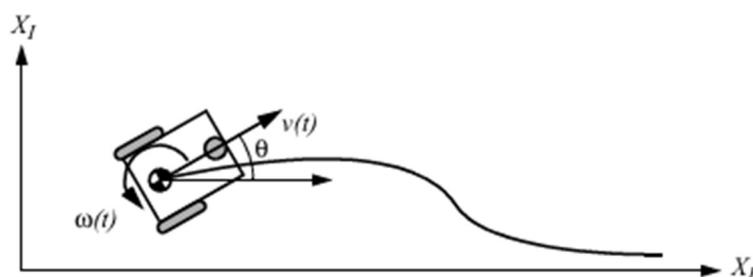
## 2.2 Cinemática

O estudo do comportamento de plataformas móveis, conhecido como cinemática, é fundamental para compreender a relação entre os comandos de controle e o comportamento do robô no ambiente (Craig, 2012).

Dentro da robótica móvel, a construção de modelos cinemáticos leva em consideração o papel que cada roda tem na locomoção do robô, uma vez que cada uma contribui de maneira independente para o deslocamento e, ao mesmo tempo, impõe restrições ao seu movimento (Siegwart; Nourbakhsh, 2004).

De acordo com Siegwart e Nourbakhsh (2004) o comportamento de um robô móvel diferencial (Figura 2), caracterizado por possuir duas ou mais rodas motrizes com controle individuais, são aplicados em robôs móveis devido a sua eficiência no controle de movimentação.

Figura 2 - Deslocamento do robô móvel diferencial



Siegwart; Nourbakhsh (2004, P. 49)

---

O comportamento de um sistema diferencial pode ser relacionado como, velocidade linear  $v$  é a média da soma entre a velocidade do conjunto de rodas da direita e esquerda,  $v_r$  e  $v_l$ , respectivamente, de acordo com a Equação (4):

$$v = \frac{v_r + v_l}{2} \quad (4)$$

Já a velocidade angular do robô  $\omega$ , Equação (5), é obtida pela diferença das velocidades  $v_r$  e  $v_l$ , dividido pela distância entre as rodas  $L$ .

$$\omega = \frac{v_r - v_l}{L} \quad (5)$$

De maneira simplificada, considera-se a ausência de movimentos curvos,  $\omega \neq 0$ , e assumir somente presença de movimento retilíneo,  $\omega = 0$ , para  $v \neq 0$ , com isso define-se o modelo de monociclo na sua forma discreta, expressa pela Equação (6):

$$\begin{bmatrix} x_{t+1} \\ y_{t+1} \\ \theta_{t+1} \end{bmatrix} = \begin{bmatrix} x_t \\ y_t \\ \theta_t \end{bmatrix} + \begin{bmatrix} v\Delta t \cos \theta \\ v\Delta t \sin \theta \\ \omega\Delta t \end{bmatrix} \quad (6)$$

### 2.3 Sensores

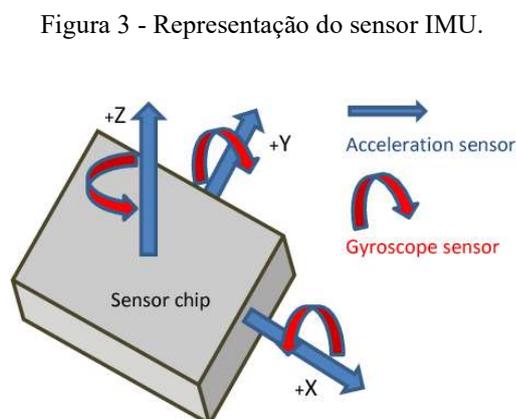
De acordo com Matarić, (2014, p. 43) “Sensores são dispositivos físicos que permitem a um robô perceber seu ambiente físico, a fim de obter informações sobre si mesmo e sobre os objetos que o cercam”. Dessa forma, os sensores desempenham um papel fundamental na autonomia dos robôs móveis, pois permitem que eles adquiram dados do ambiente e os utilizem para navegação e mapeamento.

Sensores podem ser classificados de acordo com sua função e aplicação, podendo medir grandezas como distância, velocidade, temperatura, campo magnético etc. (Niku, 2013). Além disso, sensores podem ser divididos como passivos ou ativos, dependendo pela forma como captam as informações do ambiente. Sensores passivos não necessitam de uma fonte de energia

adicional, convertendo diretamente um estímulo externo em sinal elétrico, já sensores ativos adicionam energia ao ambiente para interagir com o meio (Balbinot; Brusamarello, 2010; Siegwart; Nourbakhsh, 2004).

### 2.3.1 Sensores de Unidade de Medida Inercial (IMU)

Representado na Figura 3, IMUs são sistemas caracterizados pela composição de sensores inerciais, como *acelerômetros*, que são capazes de medir a aceleração linear ao longo dos eixos  $(x, y, z)$ , e *giroscópios* responsáveis por mensurar a taxa de rotação angular nos mesmos eixos, tornando-se essenciais em aplicações que necessitam de informações de navegação e localização, como a robótica móvel (EPSON, 2025).



Kardóš et al., (2019, P. 72)

Estes sensores estão sujeitos a interferências e imprecisões nos dados, provenientes do meio a qual estão inseridos, em particular, o giroscópio apresenta o fenômeno de deriva (*drift*), que é resultado dos acúmulos de erros obtidos no decorrer do tempo (Ahmad et al., 2013).

Esse efeito é descrito na Equação (7), no qual  $\omega_{gyro}$  é a representação da taxa de variação do ângulo  $\theta$  em relação ao tempo,  $w_{bias}$  é o erro, e o ângulo de rotação  $\theta_{gyro}$  é obtido.

$$\theta_{gyro} = \int (\omega_{gyro} + w_{bias}) dt \quad (7)$$

A Equação (7) é expressa no tempo discreto através da Equação (8), a qual a obtenção do  $\theta_k$  é influenciada pelo erro  $w_{bias}$ .

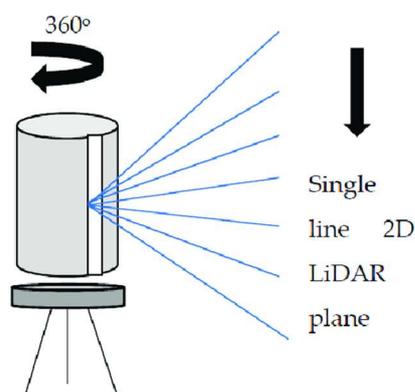
$$\theta_k = \theta_{k-1} + (\omega_{gyro} + w_{bias})\Delta t \quad (8)$$

### 2.3.2 LiDAR

Para McManamon (2019) o LiDAR desempenha papel fundamental em projetos de robótica móvel. Dentre as vantagens, as principais é a capacidade de realizar varreduras em até  $360^\circ$ , a obtenção de dados de forma precisa de obstáculos, e sua capacidade de operar em locais independente da iluminação, torna-o uma alternativa robusta para mapeamento.

Sensores LiDAR são dispositivos que usam a luz em forma pulsos de laser para mensurar a distância e outras propriedades do objeto detectado (Niku, 2013). Existem diferentes tipos de LiDAR, os capazes de fazerem medidas unidimensionais (1D), medindo apenas a distância, a obtenção de medidas no espaço 2D, sendo elas informações da distância e do ângulo relativo, e do plano 3D, que fornece uma construção mais detalhada do espaço (McManamon, 2019). A Figura 4 representa as vantagens e o funcionamento de sensores do tipo LiDAR.

Figura 4 - LiDAR 2D



Researchgte (2025)

---

De acordo com Niku (2013), essa tecnologia é baseada em três métodos diferentes, dentre eles, o método de triangulação que fornece uma resolução mais precisa quando trabalhada em medições de curta distância.

O método de triangulação envolve a emissão de um feixe e a captação da luz refletida por meio de uma câmera ou sensor de posição. O ângulo  $\theta_{sensor}$  formado entre a fonte emissora e o ponto da recepção da luz,  $b$ , permite determinar a distância  $d$ , conforme expresso na Equação (9):

$$d = \frac{b}{\tan \theta_{sensor}} \quad (9)$$

### 2.3.3 Encoders

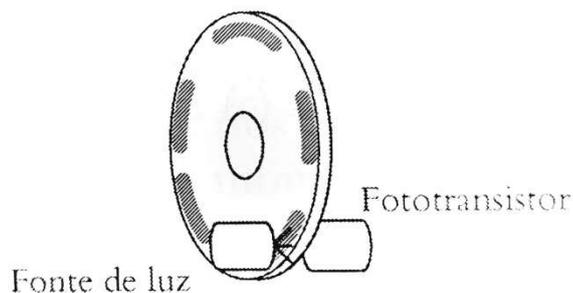
Os *encoders* desempenham papel fundamental na robótica móvel, sendo amplamente usados para estimar a posição e medir velocidade angular de motores e rodas (Siegwart; Nourbakhsh, 2004).

Entre as diversas aplicações, destacam-se sua utilização em técnicas de odometria, um método prático para estimar a posição de robôs móveis ao longo do tempo, baseado na integração do deslocamento do sistema (Siegwart; Nourbakhsh, 2004). No entanto, a odometria, quando realizada exclusivamente com *encoders*, está sujeita a erros acumulativos, pois pequenas imprecisões nas medições podem resultar em desvios significativos ao longo do tempo (Nehmzow, 2003).

Dentre os tipos de *encoders*, os de funcionamento incremental são os mais comuns, operam por meio da interrupção de um feixe de luz através de um disco rotativo (Siegwart; Nourbakhsh, 2004). A variação da luz captada pelos detectores gera sinais analógicos, que podem ser convertidos em sinais digitais, permitindo, assim, a medição de rotações por minuto (RPM).

Na Figura 5 é representado o princípio de funcionamento do dispositivo em questão.

Figura 5 - *Encoder Incremental*



Niku (2013, P. 264)

A Equação (10) permite obter valor de RPM a partir da frequência dos pulsos gerados pelo *encoder*, levando em conta o número de pulsos por rotação. Esse cálculo é fundamental, pois a estimativa de RPM influencia diretamente na precisão da odometria.

$$RPM = \frac{\text{Frequência de Pulsos}}{\text{Pulsos por volta}} \cdot 60 \quad (10)$$

É possível determinar a velocidade angular  $\omega_{roda}$  das rodas a partir do RPM obtido, essa relação pode ser verificada pela Equação (11):

$$\omega_{roda} = \frac{RPM \cdot 2\pi}{60} \quad (11)$$

A velocidade linear  $v_{roda}$  é calculada multiplicando  $\omega_{roda}$  pelo raio  $r$  das rodas, conforme a Equação (12):

---

$$v_{roda} = \omega_{roda} \cdot r \quad (12)$$

### 3 MÉTODOS PARA MAPEAMENTO E POSICIONAMENTO

A elaboração de mapas e a estimativa de posição envolvem incertezas que exigem abordagens probabilísticas para uma representação confiável do ambiente. De acordo com Thrun et al., (2006) métodos baseados em inferência estatística possibilitam atualizar informações à medida que novos dados são coletados, enquanto técnicas de filtragem minimizam os impactos de ruídos e erros sensoriais. A combinação dessas estratégias possibilita refinamentos contínuos, garantindo maior precisão na interpretação do espaço e na correção de estimativas ao longo do tempo.

#### 3.1 Teorema de Bayes

O Teorema de Bayes é um dos princípios elementares da teoria da probabilidade, ele permite atualizar a probabilidade de uma hipótese com base em novas evidências. É amplamente usado em robótica, aplicado a problemas de interferência probabilística, como o mapeamento (Thrun et al., 2006). A fórmula bayesiana permite que um robô refine sua percepção do ambiente a partir de dados sensoriais coletados ao longo do tempo. (Thrun et al., 2006).

A representação da probabilidade a posteriori do evento  $A$  acontecer dado a evidência  $B$  é descrita na Equação (13), como:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)} \quad (13)$$

A Equação (13) descreve como uma nova evidencia pode ser utilizada para atualizar a crença sobre um determinado evento.

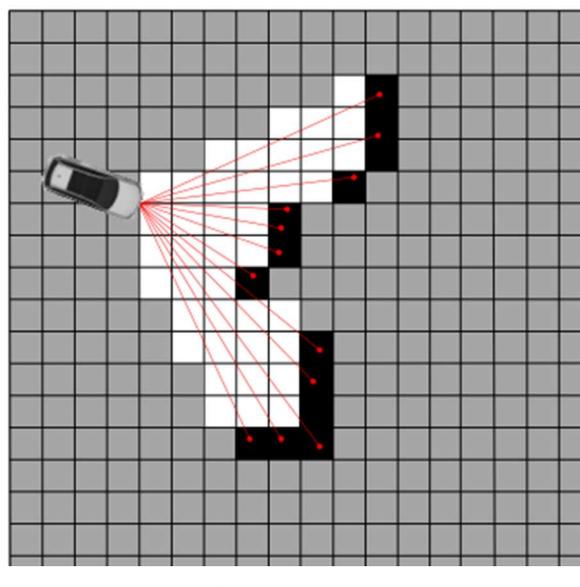
#### 3.1.2 Mapeamento em Grade

A construção de um mapa é um dos desafios centrais da robótica móvel, parte dos ambientes não possuem mapas pré-estabelecidos ou estão sofrendo constantes mudanças

(Thrun et al., 2006). Outro desafio relevante é a alta dimensionalidade do espaço de hipótese, pois um ambiente pode ser representado por milhares de células, tornando inviável o cálculo de distribuição de probabilidade para cada uma delas (Thrun et al., 2006).

Neste contexto, Thrun et al., (2006) sugere o mapeamento em grade como solução, (Figura 6). A ideia é representar o ambiente através de uma grade de ocupação, em que cada célula é determinada por um conjunto de pontos que representa a probabilidade de está ocupada ou livre.

Figura 6 - Representação Mapa em Grade



Nuß (2017, P. 29)

O modelo que tem como objetivo estimar a distribuição de ocupação de cada célula do mapa, baseando-se em medições sensoriais, é representado pela expressão na Equação (14), no qual a probabilidade de uma dada célula específica da grade  $m_i$ , é o conjunto de todas as medições do sensor  $z_{1:t}$  até o instante  $t$  e de posições ao longo do tempo  $x_{1:t}$ .

$$P(m_i | z_{1:t}, x_{1:t}) \quad (14)$$

A representação em *log-odds* é utilizada no processamento computacional, para simplificar os cálculos probabilísticos e evitar instabilidades relacionadas a aproximações

numéricas. A equação geral que descreve a atualização de  $l_{t,i}$  (*log-odds* da ocupação da célula  $m_i$  no instante  $t$ ) é dada pela Equação (15):

$$l_{t,i} = \log \frac{P(m_i | z_{1:t}, x_{1:t})}{1 - P(m_i | z_{1:t}, x_{1:t})} \quad (15)$$

A probabilidade ocupação, Equação (14), pode ser recuperada pela representação em *log-odds* mostrada na Equação (16):

$$P(m_i | z_{1:t}, x_{1:t}) = 1 - \frac{1}{1 + e^{l_{t,i}}} \quad (16)$$

A lógica de construção do mapa de ocupação é representada pelo Algoritmo 1, a partir das leituras feitas de um sensor, o LiDAR por exemplo. Esse processo atualiza continuamente a crença sobre a ocupação de cada célula da grade com base nas medições realizadas, verificando se a célula foi ou não detectada pelo sensor.

Algoritmo 1: Representação do Algoritmo do Mapa de Ocupação

- 1: Algoritmo de mapeamento\_por\_grade\_de\_ocupação( $\{l_{t-1,i}\}, x_t, z_t$ ):
- 2:     para todas as células  $m_i$  faça:
- 3:         se  $m_i$  estiver no campo perceptivo de  $z_t$ , então:
- 4:              $l_{t,i} = l_{t-1,i} + \log \frac{P(m_i | z_{1:t}, x_{1:t})}{1 - P(m_i | z_{1:t}, x_{1:t})} - l_o$
- 5:         senão:
- 6:              $l_{t,i} = l_{t-1,i}$
- 7:         Fim se
- 8:     Fim para
- 9:     Retorna  $\{l_{t,i}\}$

Adaptado de Thrun et al., (2006, P. 98)

---

## 3.2 Filtro Gaussiano

Filtros Gaussianos são métodos matemáticos usados para estimar informações de um sistema, de forma precisa, mesmo contendo incertezas e ruídos associados. Eles funcionam baseando-se em duas características, a média, uma estimativa do valor real, e a covariância, que é uma indicação do nível de confiança dessa estimativa (Zou et al., 2022).

### 3.2.1 Filtro de Kalman

O Filtro de Kalman é uma aplicação prática dos filtros Gaussianos, pois utiliza o conceito de que a melhor forma de estimar um estado incerto é representá-lo por uma média e uma covariância. De acordo com Thrun et al. (2006), o Filtro de Kalman representa a crença sobre o estado, e a covariância.

Para que o filtro de Kalman funcione corretamente, a forma como o sistema evolui ao longo do tempo e as medições feitas para acompanhar essa evolução, seguem um padrão previsível, mas com pequenas variações aleatória. (Thrun et al., 2006). Ou seja, o sistema tem um comportamento linear com ruído Gaussiano, isto significa que as mudanças no sistema ocorrem de forma previsível, mas com pequenas variações aleatórias, e que as medições feitas para corrigir essa estimativa também contêm incertezas.

Dividido em duas etapas principais, primeiro ele faz a previsão, representada através da Equação (17), de um próximo estado com base nos dados anteriores, em seguida novas informações são recebidas, então ele faz a atualização, conforme a Equação (18), dessa previsão, ajustando para que fique mais próximo do valor esperado. (Thrun et al., 2006).

$$x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t \quad (17)$$

O estado do sistema  $x_t$  é calculado com base nas somas do estado anterior  $x_{t-1}$ , da ação de controle  $u_t$  e de um ruído  $\varepsilon_t$ . O estado e o controle são multiplicados por uma matriz que define como o sistema evolui com o tempo e uma que define como o controle influencia o estado, respectivamente,  $A_t$  e  $B_t$ .

$$\bar{x}_t = x_t + K_t(z_t - C_t x_t) \quad (18)$$

Na Equação (18) observa-se que o estado corrigido  $\bar{x}_t$  é obtido somando o estado previsto, Equação (17), a diferença entre a medição real  $z_t$  e a medição estimada  $C_t x_t$ . O ganho de Kalman  $K_t$  ajusta a atualização, garantindo que a estimativa seja melhorada (Thrun et al., 2006).

A Figura 7 ilustra o processo de previsão e atualização do Filtro de Kalman. Na etapa de previsão, o estado futuro é estimado com base no estado anterior e nas ações de controle, considerando a incerteza associada. Na etapa de atualização, os valores medidos são utilizados para corrigir essa estimativa, minimizando o erro. Esse ciclo se repete a cada iteração, garantindo uma estimativa cada vez mais precisa do estado do sistema

Figura 7 – Fluxo geral do Filtro de Kalman



Elaboração: Análise Macro com base em Alex Becker.

### 3.2.2 Filtro de Kalman Estendido

Segundo Ribeiro (2004), o Filtro de Kalman Estendido é uma adaptação do Filtro de Kalman tradicional para sistemas dinâmicos e não lineares. Para possibilitar sua aplicação, o EKF realiza uma linearização do modelo dinâmico em torno de uma estimativa de estado mais recente, permitindo a utilização de equações do filtro linearizado.

Thrun et al. (2006) enfatiza que o EKF não calcula diretamente a distribuição posterior do estado, mas busca uma aproximação eficiente de seus momentos de média e covariância, tornando-se uma alternativa viável quando a modelagem exata do sistema é inviável.

O EKF é uma das ferramentas mais utilizadas na robótica para estimar a posição de um robô, isso se dá pela capacidade de responder a sistemas estocásticos e pela eficiência computacional. O filtro permite integrar múltiplas fontes de dado, permitindo obter uma estimativa mais aproximada da posição real do sistema, aplicação essa que se torna fundamental em abordagens de localização e mapeamento (Thrun et al. 2006).

O EKF segue as mesmas etapas do KF, mas substitui as matrizes constantes por uma Jacobiana de cada interação.

$$\hat{x}_k = f(\hat{x}_{k-1}, u_k) + \omega_k \quad (19)$$

A Equação (19) representa a previsão do vetor de estado  $\hat{x}_k$  no instante  $k$  e a entrada de controle  $u_k$  no instante  $k$ . A previsão de covariância é dada pela Equação (20):

$$\hat{P}_k = F_k P_{k-1} F_k^T + Q_k \quad (20)$$

Os elementos da Equação (20) são a Jacobiana  $F_k$  da função de transição de estado  $f(x_k, u_k)$  definida pela Equação (21), e  $F_k^T$  e  $Q_k$ , são, respectivamente, a matriz transposta dessa Jacobiana e a matriz de covariância que define a dispersão do ruído  $\omega_k$ .

---


$$F_k = \frac{\partial f(x_k, u_k)}{\partial x_k} \quad (21)$$

A previsão da medição  $\hat{z}_k$  é modelada de acordo como observado pelo sensor no estado previsto  $h(\hat{x}_k)$ , Equação (22):

$$\hat{z}_k = h(\hat{x}_k) \quad (22)$$

O novo ganho de Kalman  $K_k$ , agora, é definido pela Equação (23):

$$K_k = \hat{P}_k H_k^T S_k^{-1} \quad (23)$$

A covariância de inovação  $S_k$  é definida pela Equação (24),  $H_k$  é a Jacobiana da previsão de medição e  $R_k$  é a matriz de ruído associado a observação dos sensores.

$$S_k = H_k \hat{P}_k H_k^T + R_k \quad (24)$$

Por último, o modelo de atualização, Equação (25), é a soma do estado previsto e o produto do ganho  $K$  e a diferença ( $z_k - \hat{z}_k$ ), sendo  $z_k$  a observação real do sensor associada a um ruído  $v_k$ .

$$\hat{x}_k^- = \hat{x}_k + K_k(z_k - \hat{z}_k) \quad (25)$$

A atualização da covariância  $\hat{P}_k^-$ , definida na Equação (26), no qual  $I$  é uma matriz identidade.

$$\hat{P}_k^- = (I - K_k H_k) \hat{P}_k \quad (26)$$

---

O Fluxo do EKF é representado pela Figura 7:

Algoritmo 2: Algoritmo do Filtro de Kalman Estendido

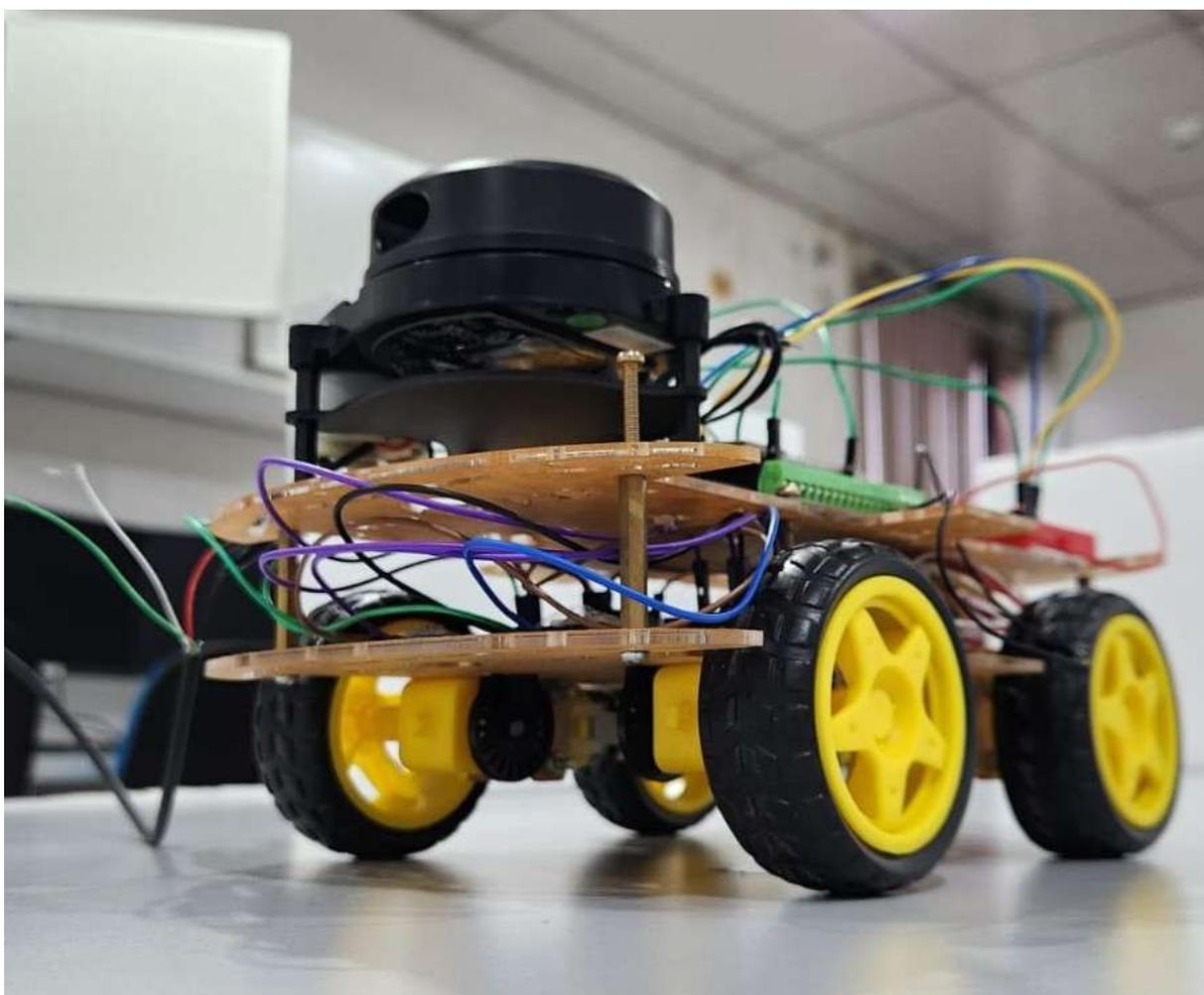
- 1: Algoritmo Filtro\_Kalman\_Estendido( $\hat{x}_{k-1}, P_{k-1}, u_k, z_k$ ):
- 2:      $\hat{x}_k = f(\hat{x}_{k-1}, u_k)$
- 3:      $\hat{P}_k = F_k P_{k-1} F_k^T + Q_k$
- 4:      $K_k = \hat{P}_k H_k^T (H_k \hat{P}_k H_k^T + R_k)^{-1}$
- 5:      $\hat{x}_k^- = \hat{x}_k + K_k (z_k - \hat{z}_k)$
- 6:      $\hat{P}_k^- = (I - K_k H_k) \hat{P}_k$
- 7:     retorna  $\hat{x}_k^-, \hat{P}_k^-$

Adaptado de Thrun et al., (2006, P. 59)

## 4 HARDWARE

Neste capítulo é descrito os componentes necessários para o funcionamento físico do robô móvel utilizado, conforme Figura 8, para a realização do experimento proposto. Os componentes fundamentais de hardware são abordados em sessões que abrangem a estrutura mecânica, sensores, microcontroladores, sistema de comunicação e fornecimento da energia elétrica.

Figura 8: Robô Móvel utilizado



Autor (2025)

#### 4.1 Estrutura de locomoção

Para este experimento, o robô foi constituído de quatro rodas com raio de 35 mm e distribuídas a uma distância de 135 mm, cada roda foi acoplada a motores de corrente contínua (DC) com faixa de operação em uma tensão de 5V, estes atuadores são integrados a uma caixa de redução, como demonstrado na Figura 9.

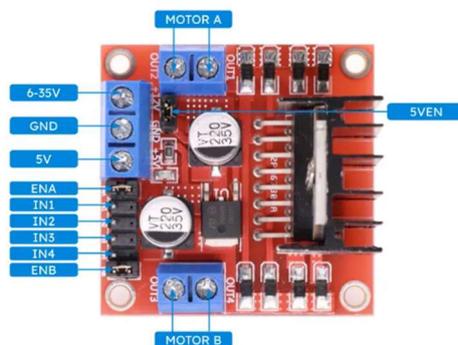
Figura 9 – Atuadores



Autor (2025)

O robô possui controle de rotação para cada par de rodas, o fornecimento de energia para os atuadores é controlado pelo driver Ponte H L298N (Figura 10). Este driver permite ajustar individualmente a velocidade e a direção de cada par de rodas.

Figura 10 – Ponte H LN298N



Guse (2013)

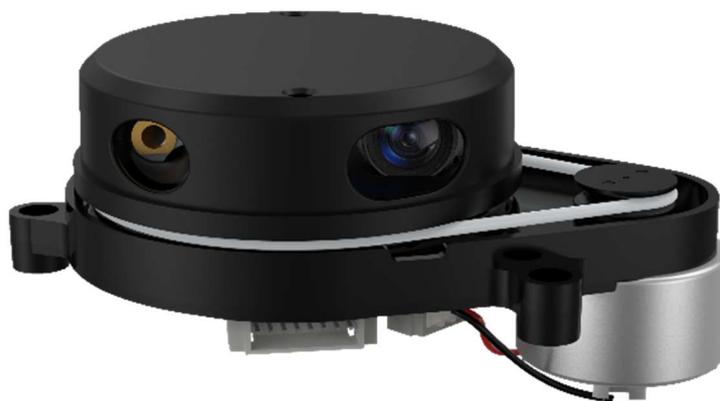
---

## 4.2 Sensores

### 4.2.1 YDLiDAR X4 PRO

Neste trabalho, o principal sensor utilizado foi o YDLiDAR X4 PRO (Figura 11), trata-se de um sensor LiDAR de baixo custo, endereçado para aplicações em aspiradores de pó, robôs de entrega e inspeção de segurança, atuação industrial e principalmente para pesquisa e desenvolvimento. Este sensor é desenvolvido pela empresa Shenzhen EAI Technology Co., Ltd. (EAI Technology).

Figura 11 – YDLiDAR X4 PRO

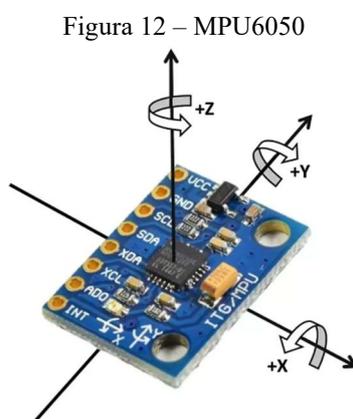


EAI Technology (2025)

O sensor utiliza do princípio da triangulação para a obtenção de dados, coletando-os em um campo de visão de 360° operado em um plano 2D. o dispositivo dispõe sua operação elétrica e parâmetros de varredura dentro dos parâmetros especificados Anexo A – YDLiDAR X4 PRO

### 4.2.2 MPU6050

O modulo IMU integrado ao desenvolvimento foi MPU6050 (Figura 12), este é um modulo de baixo custo, que possui a combinação de sensores inerciais, como giroscópios, acelerômetros e sensores de temperatura. Neste trabalho, o modulo foi utilizado para a obtenção apenas da orientação do sistema, dado a alta sensibilidade a ruídos que ele possui.



MV Electrónica (2025)

### 4.2.3 LM393

O driver LM393, ilustrado na Figura 13, é um *encoder* óptico de funcionamento incremental, este foi responsável por realizar o monitoramento dos RPM do conjunto de rodas, a fim de obter a velocidade média das rodas, necessário para a estimativa de predição estável do sistema.

Figura 13 – LM393



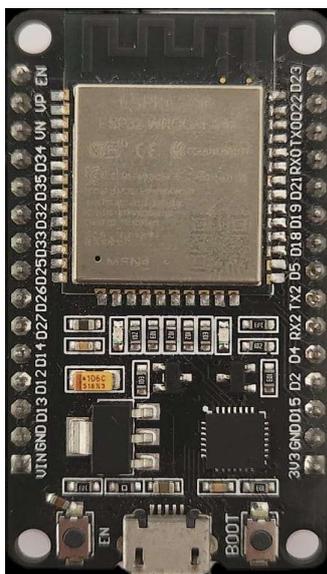
Autor (2025)

#### 4.2.4 Microcontroladores

O projeto contou com a operação de dois microcontroladores ESP32 e ARDUINO UNO, eles são responsáveis pela aquisição e processamento de dados dos sensores.

O ESP32, representado através da Figura 14, é responsável pela comunicação com o sensor LiDAR, através da interface *UART* (Universal Asynchronous Receiver-Transmitter), recebendo os dados de varredura do ambiente, e traduzindo essa informação para medições de ângulo e distância. Além da aquisição de dados, o ESP32 é responsável pela comunicação com o PC via Wi-Fi, utilizando o protocolo TCP/IP. Essa comunicação permite o envio contínuo dos dados coletados para processamento mais refinado, e recebimento de comandos, que são endereçados para o ARDUINO UNO.

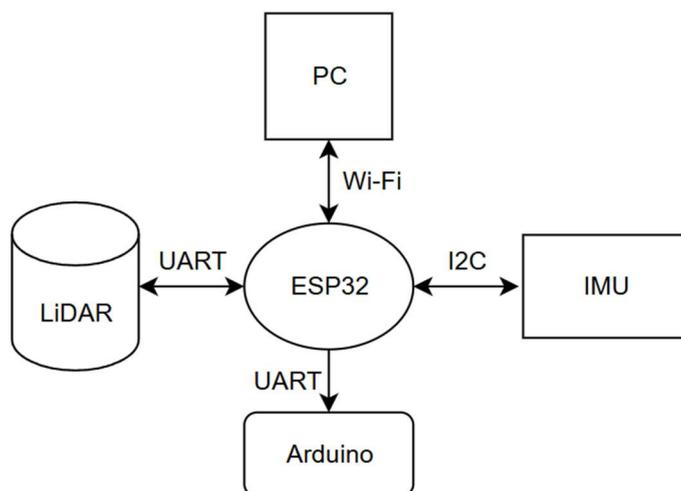
Figura 14 – ESP32



Autor (2025)

O ESP32 é o microcontrolado central, como representado na Figura 15, mostrando os protocolos estabelecidos de comunicação.

Figura 15 – Fluxograma de comunicação do ESP32



Autor (2025)

Arduino UNO, Figura 16, é responsável exclusivamente pela parte de atuação mecânica do robô, realizando os controles dos motores a partir da aquisição de comandos do ESP32, comunicação essa estabelecida via *UART*.

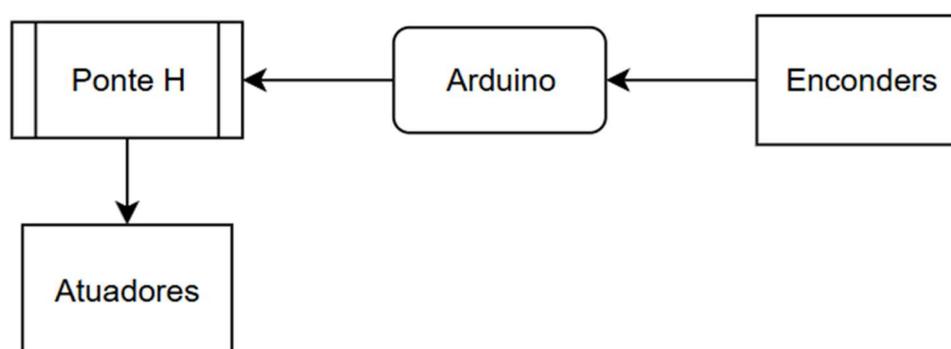
Figura 16 – Arduino UNO



Autor (2025)

A Figura 17 apresenta o fluxograma de atuação do Arduino UNO no sistema a qual demonstra a interação do Arduino com os *encoders*, a Ponte H e os atuadores. O Arduino recebe as informações de locomoção do Esp32, com base nesses dados o Arduino envia comandos à Ponte H que modula a potência fornecida aos atuadores, controlando a velocidade e o sentido de rotação dos motores.

Figura 17 – Fluxograma de atuação do Arduino



Autor (2025)

#### 4.2.5 Conexão Elétrica

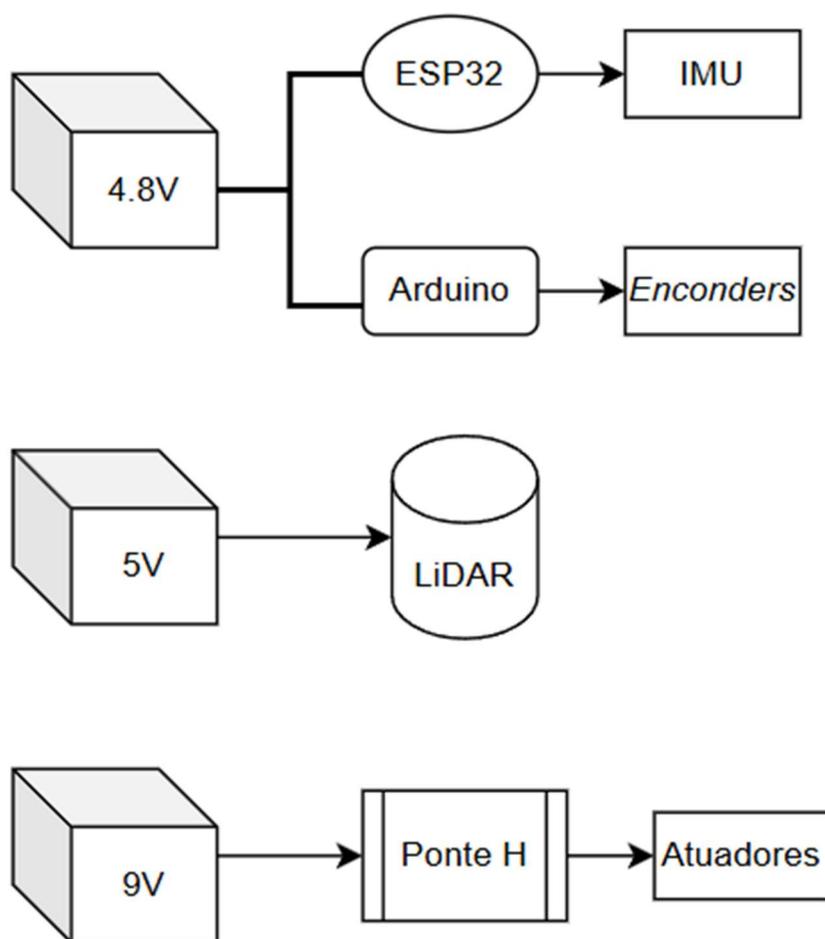
A plataforma móvel desenvolvida utiliza três circuitos de alimentação elétrica, Figura 18, garantindo que cada componente receba a tensão necessária para seu funcionamento adequado.

A alimentação dos módulos de controle e aquisição de dados é fornecida por uma fonte de 4,8V, que distribui energia para os dois principais microcontroladores do sistema: ESP32 e Arduino.

O sensor YDLiDAR X4 PRO, opera com uma fonte exclusiva de 5V, devido aos seus requisitos específicos de alimentação, garantindo estabilidade no funcionamento do sensor.

O sistema de locomoção do robô utiliza uma fonte de 9V, que alimenta a Ponte H, responsável por controlar a velocidade e direção dos motores através de sinais PWM gerados pelo Arduino.

Figura 18 – Diagrama de conexões elétricas

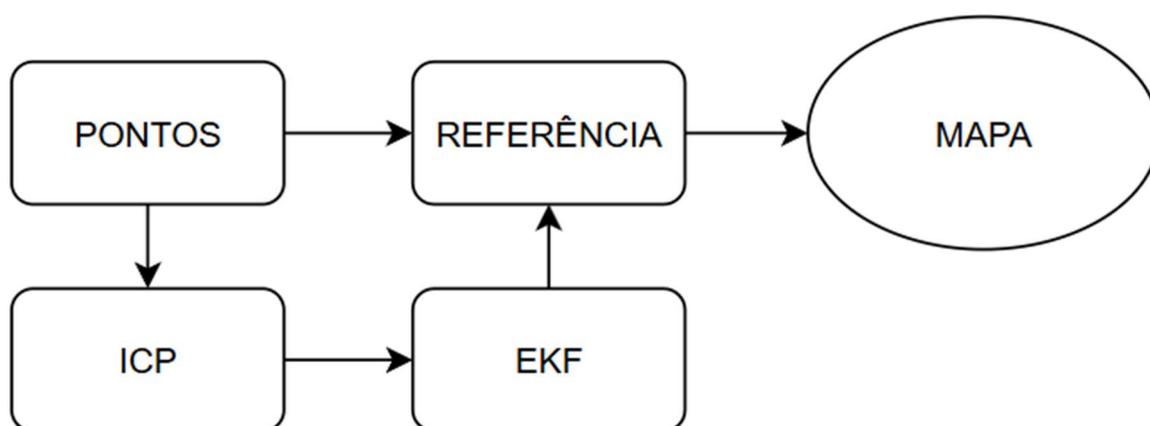


Autor (2025)

## 5 MODELAGEM DO SISTEMA

Esta sessão irá abordar o processo de construção e atualização do mapa de ocupação do ambiente, se dedicando exclusivamente a modelagem realizada no computador, utilizando a linguagem de programação *python* (versão 3.11.0). Através da Figura 19 é representado o fluxograma de interação entre os principais processos, que serão discutidos a seguir.

Figura 19 – Fluxo de processamento dos dados



Autor (2025)

### 5.1 Tratamento de Pontos

Na etapa de pontos (Apêndice A – Transformações de Pontos no Plano) inicialmente foi tratado a forma com que os dados são recebidos, o YDLiDAR X4 PRO ao detectar um objeto retorna a informação em coordenadas polares que é uma representação do plano em distância  $r_{lidar}$  e ângulo  $\theta_{lidar}$ , a fim de prosseguir com o processamento foi necessário definir a representação em coordenadas cartesianas  $(x_{lidar}, y_{lidar})$ , estas coordenadas representam o referencial local do sistema, expressa através da Equação (26):

$$\begin{cases} x_{lidar} = r_{lidar} \cdot \cos \theta_{lidar} \\ y_{lidar} = r_{lidar} \cdot \sin \theta_{lidar} \end{cases} \quad (26)$$

## 5.2 Iterative Closest Point (ICP)

Esta etapa utilizou-se a biblioteca *Open3D* (Zhou et al., 2018), que é uma ferramenta para registro de nuvens de pontos, permitindo a realização de alinhamentos, e transformações geométricas.

Para que a biblioteca crie nuvem de pontos é exigido um sistema de coordenadas 3D no espaço, para isso os pontos obtidos (sessão 5.1) passam por um tratamento (Apêndice B – Iterative Closest Point) que adiciona o eixo  $z = 0$  no espaço de coordenadas, com as nuvens de pontos preparadas, a estrutura ICP foi adaptada para alinhar a nuvem de pontos em um instante  $t$  com uma no instante  $t - 1$  para realizar o alinhamento de forma iterativa, para isso foi usado a frequência de varredura e rotação do sensor para estimar a quantidade pontos coletados, a fim de estruturar o intervalo de operação do algoritmo.

O resultado do ICP é uma matriz homogênea  $T$  4x4 no espaço tridimensional, conforme a Equação (27), como o movimento observado é apenas no plano 2D, as variáveis  $t_x$  e  $t_y$  representam, respectivamente, as coordenadas  $(x, y)$ .

$$T = \begin{bmatrix} \cos \theta & -\sin \theta & 0 & t_x \\ \sin \theta & \cos \theta & 0 & t_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (27)$$

A transformação permite o deslocamento acumulado  $T_{acumulado}$  a partir da expressão representada na Equação (28):

$$T_{acumulado} = T_n * T_{n-1} * \dots * T_1 \quad (28)$$

---

Essa transformação integra o modelo de observação  $\hat{z}_k$  na execução do Filtro de Kalman Estendido.

### 5.3 Modelo EKF

Nessa etapa utilizou-se do Filtro de Kalman Estendido para estimar a posição do robô ao longo do tempo a partir das informações do modelo de movimento (Sessão 2.2) e dos dados obtidos de observação (Sessão 5.2).

A implementação do algoritmo (Apêndice C – *Extended Kalman Filter*) segue a estrutura do EKF, conforme descrito na Sessão 3.2.2. Esse modelo baseia-se em um sistema dinâmico não linear, no qual a predição  $\hat{x}_k$  (Sessão 3.2.2, Equação 16) recebe como entrada o vetor de estado  $[x_r, y_r, \theta_r]$ , que representa a pose do robô no espaço (Sessão 2.2, Equação 6) e o vetor de controle  $u_k$ , que inclui a velocidade linear  $v$  e angular do robô  $\omega$ , modeladas a partir dos valores médios de RPM (Sessão 2.3.3).

A Jacobiana  $F_k$  (Sessão 3.2.2, Equação 20) que representa a linearização do vetor de estado é definida pelas derivas parciais de  $f(x_k, u_k)$ , em relação ao vetor de estado, é representada pelas Equações (29 e 29.1).

$$F_k = \frac{\partial}{\partial x_k} \begin{bmatrix} x + v \cos \theta \Delta t \\ y + v \sin \theta \Delta t \\ \theta_t + \omega \Delta t \end{bmatrix} \quad (29)$$

$$F_k = \begin{bmatrix} 1 & 0 & -v \sin \theta \Delta t \\ 0 & 1 & v \cos \theta \Delta t \\ 0 & 0 & 1 \end{bmatrix} \quad (29.1)$$

A predição de covariância  $\hat{P}_k$  (Sessão 3.2.2, Equação 19) pode ser atualizada, então é necessário realizar a atualização da covariância de inovação  $S_k$ , está por sua vez, se torna

necessário a linearização do vetor de observação  $\hat{z}_k$ , representada pela entrada  $[t_x, t_y, \theta_{gyro}]$ , assim, a Jacobiana  $H_k$  é definida conforme a Equação (30):

$$H_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (30)$$

Além disso, os outros parâmetros do modelo, como os ruídos de processo e de medição,  $Q_k$  e  $R_k$ , respectivamente, foram ajustados empiricamente por meio de testes práticos. Esses valores foram determinados de forma experimental, buscando otimizar a convergência do EKF a fim de obter o equilíbrio dado o comportamento da odometria e as correções baseadas nos sensores.

#### 5.4 Configuração do Mapa

A abordagem definida para construção do mapa (Apêndice D – *Grid Mapping*) representa o ambiente do robô em uma matriz de células, no qual cada célula armazena um valor de probabilidade de ocupação, inicialmente em estado desconhecido, representada por 0,5 e atualizada com base nas leituras do sensor.

A princípio a grade é uma matriz de resolução  $r_{mapa}$  definida pela razão entre o tamanho do mapa e o tamanho de cada célula, conforme a Equação (31).

$$r_{mapa} = \frac{\textit{Tamanho do Mapa}}{\textit{tamanho da célula}} \quad (31)$$

Cada célula da matriz corresponde a uma região do espaço mapeado, e a posição de um ponto no ambiente deve ser convertida para um índice dentro da matriz  $(i, j)$ . Para isso, utilizamos a relação entre as coordenadas do mundo real e os índices da matriz. Dado um ponto  $(x, y)$ , a conversão em índices da matriz pode ser apresentada pela Equação (32).

$$i = \frac{y + \frac{\text{tamanho do mapa}}{2}}{r_{\text{mapa}}}; j = \frac{x + \frac{\text{tamanho do mapa}}{2}}{r_{\text{mapa}}} \quad (32)$$

Essa transformação garante que o sistema de coordenadas do mapa, seja centrado na origem (0, 0).

## 5.5 Atualização do Mapa

Como discutido na sessão anterior (Sessão 5.4) o mapa é criado com todos os seus índices em estado desconhecido, a atualização (Apêndice E – *Log-Odds*) que permite trabalhar a incerteza das medições do sensor é dada pelo comportamento definido de *Log Odds* (Sessão 3.1.2 Equação 14 & Algoritmo 1), em que  $P(\text{ocupado})$  é estimado pela confiança a respeito das medições do sensor, dado um ponto observado a confiança ao longo do tempo é atualizado conforme a Equação (33).

$$L_{t+1} = L_{t-1} + \log \frac{P(\text{ocupado})}{P(\text{livre})} \quad (33)$$

Contudo, a confiança a respeito do objeto não é a única a sofrer atualização ao longo do tempo (Sessão 3.1.2), a confiança a respeito das células ao longo do percurso também são acrescidas de confiança, dado isso, foi adaptado uso da biblioteca *Bresenham* (ENCUKOU, 2025), ela é responsável por desenhar linhas em uma grade discreta, ou seja, determina quais células pertencem a trajetória entre dois pontos.

Dessa forma, é possível estabelecer o percurso entre um ponto observado ( $x_{\text{final}}, y_{\text{final}}$ ) e o sensor ( $x_{\text{inicial}}, y_{\text{inicial}}$ ) dado suas posições, conforme a Equação (34),

$$\begin{cases} \Delta x = x_{\text{final}} - x_{\text{inicial}} \\ \Delta y = y_{\text{final}} - y_{\text{inicial}} \end{cases} \quad (34)$$

Esses valores permitem determinar a inclinação da linha de trajetória entre os pontos.

Assim, a o espaço definido entre os pontos pode ser atualizado conforme a Equação (35), a qual estabelece a chance de o percurso ser livre se um é objeto observado.

---

$$L_{t+1} = L_{t-1} + \log \frac{P(\text{livre})}{P(\text{ocupado})} \quad (35)$$

A confiança geral a respeito das medições  $L(x)$  é definido pelos parâmetros:

- $L(x) = 0$ ; representa uma probabilidade de 50%, logo o estado é incerto.
- $L(x) > 0$ ; representa a probabilidade de que o evento ocorreu.
- $L(x) < 0$ ; representa a probabilidade de que o evento não ocorreu.

## 5.6 Construção do Mapa

A construção do mapa (Apêndice F – Representação no Espaço) consiste em representar o Mapa de Ocupação em relação ao referencial global (Sessão 2.1, Equação 2), dado as estimativas de posição do EKF e os pontos coletados.

Após a atualização da matriz de ocupação com *log-odds*, os valores são convertidos para probabilidades reais, permitindo uma representação intuitiva do ambiente (Sessão 3.1.2 Equação 15).

Assim, o valor a priori da ocupação, estabelecido como 0.5, é atualizado em que a representação de:

- Valores próximos de 1 indicam alta probabilidade de ocupação;
- Valores próximos de 0 indicam baixa probabilidade de ocupação;

## 6 EXPERIMENTOS E RESULTADOS

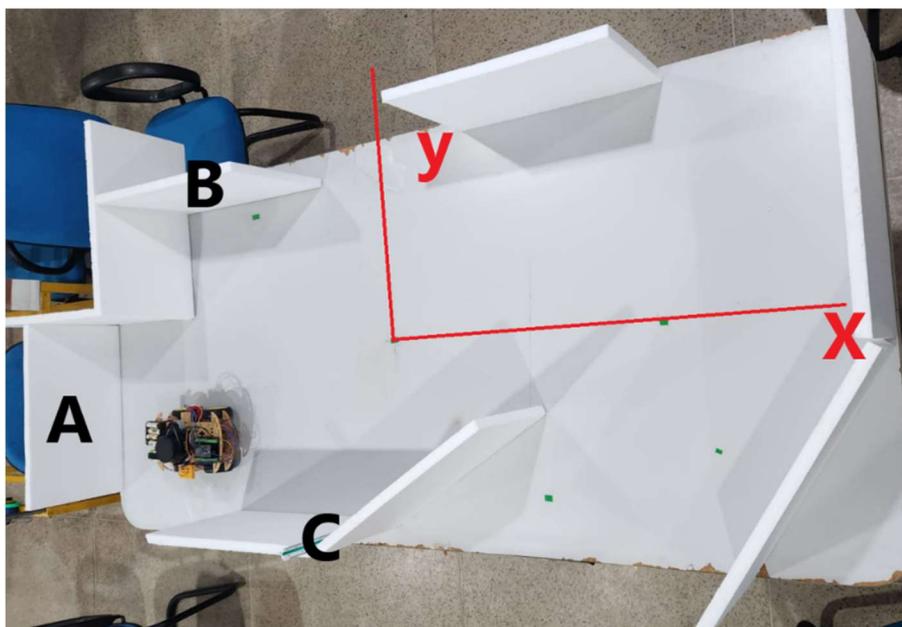
Neste capítulo, será descrito o ambiente de testes, assim como, os resultados obtidos do Sensor e da aplicação do Filtro de Kalman Estendido. Para isso, foi feita a comparação do ambiente real com a do Sensor de maneira estática, e a comparação entre o mapa gerado e as trajetórias previstas e corrigidas do robô com ganho de Kalman  $K$  diferentes para cada cenário.

O ambiente de testes é caracterizado por uma mesa de aproximadamente 2000mm x 920mm e folhas de isopor que foram dispostos em cada cenário, para representar obstáculos, cada folha possui 470mm x 20mm, as medidas foram tiradas manualmente. O sensor foi configurado para realizar varreduras no limite de 1000mm.

### 6.1 Cenário A

A Figura 20 ilustra o cenário A qual o robô foi disposto e a sua orientação, destacando os obstáculos A, B, C.

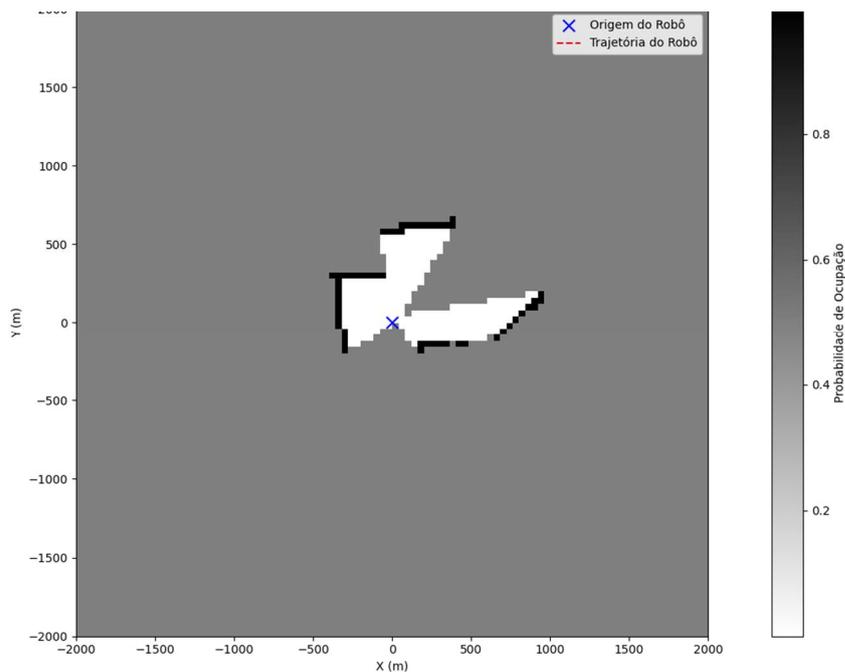
Figura 20 – Cenário A



Autor (2025)

A Figura 21 contém a representação da percepção do robô, em que dado os parâmetros de varredura, ele percebe os obstáculos A, B e C

Figura 21 – Cenário A parado



Autor (2025)

A Tabela 2 dispõe dos valores reais medidos e pelos valores gerados pelo algoritmo, levando em consideração o ponto médio de cada obstáculo, do sensor até o obstáculo.

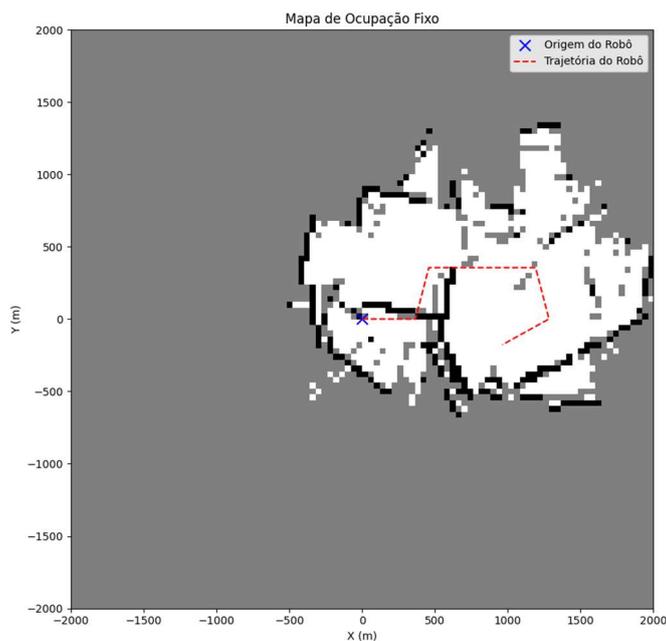
Tabela 2 – Dados do cenário A

<i>Obstáculo</i>	<i>Mapa real (mm)</i>	<i>Mapa gerado (mm)</i>	<i>Erro (mm)</i>
<i>A</i>	380	336	44
<i>B</i>	627	614	13
<i>C</i>	802	785	17

Autor (2025)

As Figuras 22 e 23, respectivamente, representam a trajetória prevista e corrigida do mapa gerado pelo robô, onde o deslocamento foi planejado para que a plataforma ultrapassasse o obstáculo C

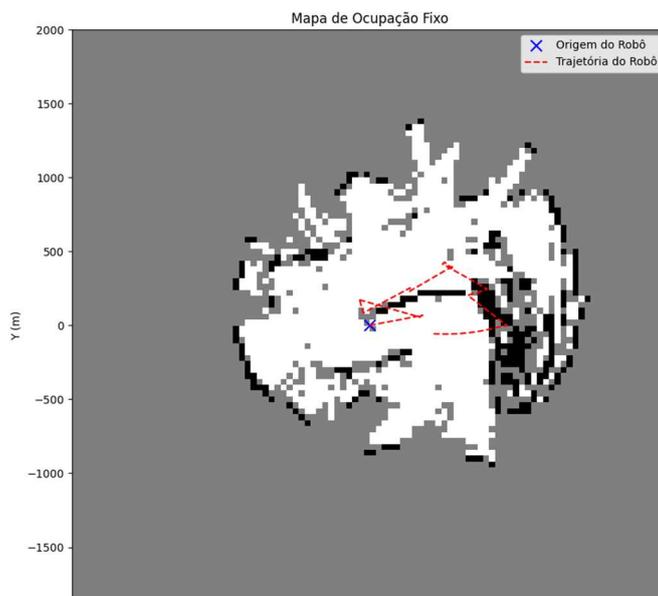
Figura 22 – Trajetória Prevista Cenário A



Autor (2025)

O ganho de Kalman foi escolhida de forma que existisse um equilíbrio entre o modelo de observação e a entrada de controle.

Figura 23 – Trajetória Corrigida Cenário A

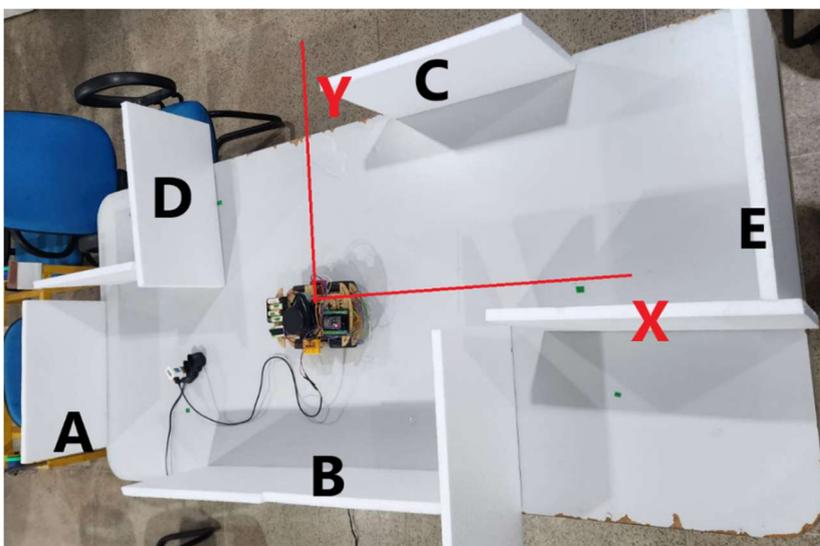


Autor (2025)

## 6.2 Cenário B

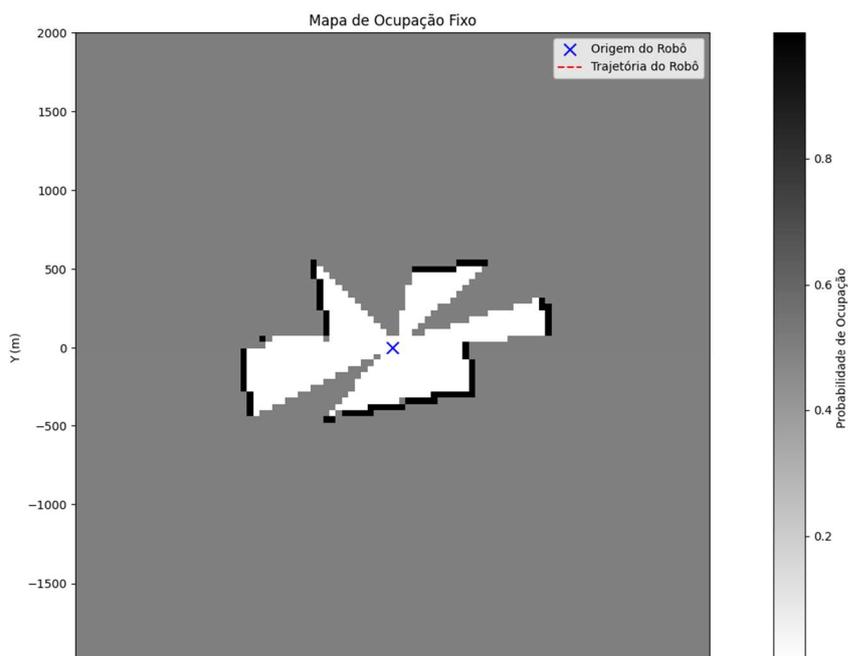
As Figuras 24 e 25, respectivamente, representam o novo cenário disposto dos obstáculos A, B, C, D e E, e o novo mapa gerado.

Figura 24– Cenário B



Autor (2025)

Figura 25 – Cenário B parado



Autor (2025)

A Tabela 3, representa os novos valores medidos do sensor até os obstáculos, e dos valores obtidos do mapa gerado.

Tabela 3 – Dados do cenário B

<i>Obstáculo</i>	<i>Mapa real (mm)</i>	<i>Mapa Gerado (mm)</i>	<i>Erro (mm)</i>
<i>A</i>	972	951	21
<i>B</i>	380	358	22
<i>C</i>	470	457	13
<i>D</i>	520	512	8
<i>E</i>	982	966	16

Autor (2025)

As novas trajetórias, prevista e corrigida, Figuras 26 e 27, respectivamente, mostram a diferença de percepção do robô, dado um novo ganho  $K$  de Kalman, sendo esse, ajustado agora para confiar mais no deslocamento da entrada de controle, e na entrada de orientação do modelo de observação. O deslocamento planejado na nova situação era que o robô móvel realizasse uma volta em torno da sua origem.

Figura 26 – Trajetória Prevista Cenário B

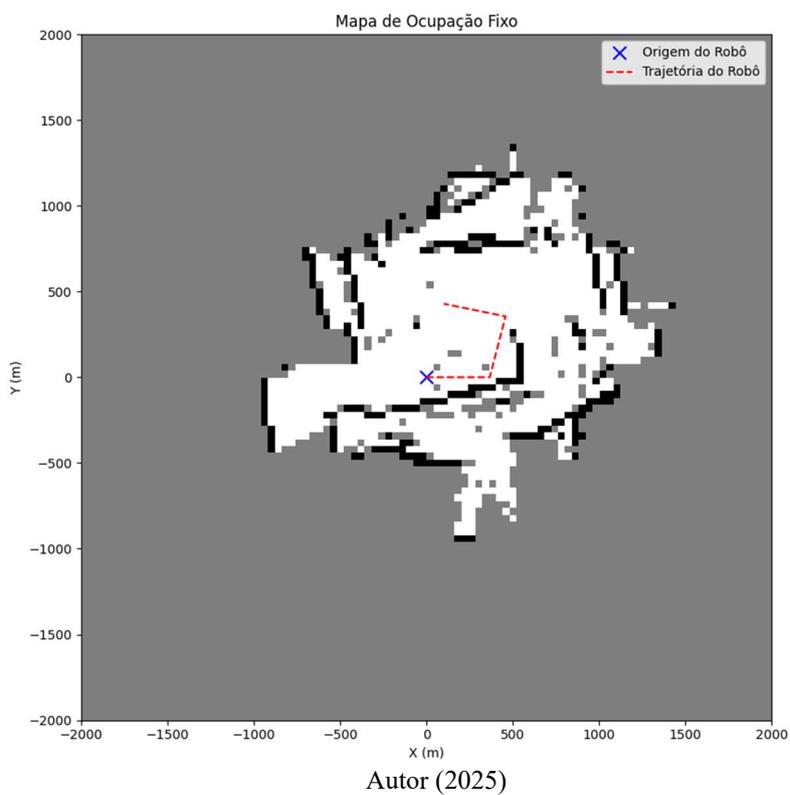
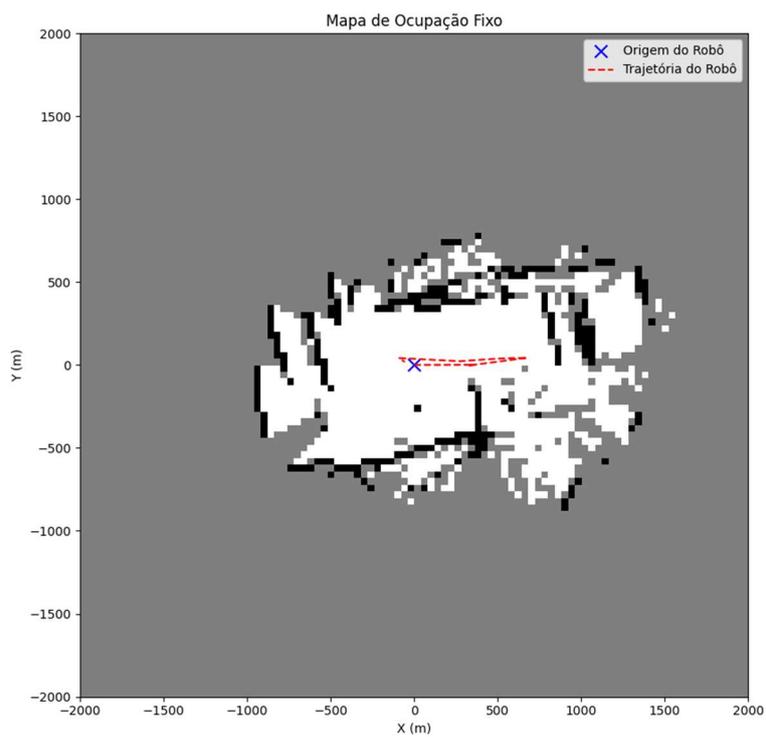


Figura 27 – Trajetória Corrigida Cenário B



---

### 6.3 Análise dos Resultados

Dado as informações obtidas, conforme apresentado anteriormente (Sessão 6.1 e 6.2) foi possível avaliar as medições realizadas pelo sensor YDLiDAR X4 PRO em um ambiente estático. A análise considerou as discrepâncias entre as distâncias reais e as distâncias mapeadas pelo sensor, permitindo a quantificação dos erros e a identificação de padrões de comportamento do sensor.

Os valores obtidos indicam que o sensor apresenta um erro médio absoluto (MAE) de aproximadamente 24.67 mm no Cenário A e 16.00 mm no Cenário B. Essa diferença sugere que o sensor possui um desempenho mais estável no Cenário B, isso pode ser relacionado a fatores como:

- Erro sistemático nas medidas realizadas dos obstáculos.
- Melhor posicionamento do sensor em relação aos obstáculos.

Apesar da variação entre os cenários, o erro absoluto médio permaneceu dentro dos parâmetros especificados pelo fabricante, que indica um erro típico de **20 mm** para o sensor. Assim, os valores obtidos confirmam que o YDLiDAR X4 PRO manteve uma precisão compatível com suas especificações.

Além disso, ao analisar visualmente os trajetos percorridos pela plataforma móvel e à acomodação do mapa gerado, é possível afirmar que, quando o ganho  $K$  do filtro de Kalman, foi equilibrado entre as confianças da observação e da entrada de controle (Figura 23), o mapa foi construído de maneira totalmente ruidosa, e incoerente com o ambiente previamente estabelecido, entretanto, quando o ganho  $K$  foi acomodado para confiar mais no deslocamento do vetor de estado, e na observação de orientação fornecida pelo IMU, a trajetória corrigida, mesmo que ruidosa, tendeu a manter a proporção do mapa previamente gerado (Figura 27).

---

## 7 CONCLUSÃO

O presente trabalho teve como objetivo validar o mapeamento de ambientes estáticos utilizando técnicas LiDAR-SLAM, com ênfase na fusão de sensores ruidosos por meio do Filtro de Kalman Estendido. Para isso, foi desenvolvido um sistema integrado entre um robô móvel e um computador, permitindo a coleta, processamento e análise dos dados de sensores de baixo custo, como o YDLiDAR X4 PRO e a unidade inercial MPU6050.

Os experimentos realizados indicam que o sensor apresentou um erro médio absoluto de aproximadamente 24,67 mm no Cenário A e 16,00 mm no Cenário B, sugerindo um desempenho mais estável no segundo cenário. Essa diferença pode ser atribuída a fatores como erros sistemáticos do cenário e um melhor posicionamento do sensor em relação aos obstáculos.

A avaliação do Filtro de Kalman Estendido demonstrou a influência do ganho K na construção do mapa. Quando o ganho foi ajustado para equilibrar a confiança entre as observações e os comandos de controle, a representação do ambiente tornou-se inconsistente e ruidosa. Por outro lado, ao favorecer a confiança no deslocamento do vetor de estado e na orientação fornecida pela IMU, a trajetória corrigida apresentou maior coerência com o ambiente previamente estabelecido, ainda que com algum ruído residual.

Desse modo, é possível afirmar que a implementação correta dos modelos de observação é essencial para a acurácia em Mapeamentos e Localização Simultânea (SLAM), pois influencia diretamente na coerência da trajetória estimada e na qualidade do mapa gerado. Modelos bem calibrados permitem uma melhor fusão dos dados sensoriais, reduzindo dados ruidosos e aprimorando a confiança acerca do mapeamento, mesmo em cenários com sensores de baixo custo.

---

## REFERÊNCIAS

THRUN, Sebastian; BURGARD, Wolfram; FOX, Dieter. **Probabilistic Robotics**. Cambridge, MA: MIT Press, 2006. ISBN 978-0-262-20162-9.

NEHMZOW, Ulrich. **Mobile robotics: a practical introduction**. 2. ed. London: Springer-Verlag, 2003. ISBN 1-85233-726-5.

MATARIC, Maja J. **Introdução à robótica**. Tradução de Humberto FeraSoli Filho, José Reinaldo Silva, Silas Franco dos Reis Alves. 1. ed. São Paulo: Editora Unesp/Blucher, 2014. ISBN 978-85-212-0853-2.

NIKU, Saeed Benjamin. **Introdução à robótica: análise, controle, aplicações**. Tradução e revisão técnica de Sérgio Gilberto Taboada. Reimpressão. Rio de Janeiro: LTC, 2013. ISBN 978-85-216-2237-6

BALBINOT, A.; BRUSAMARELLO, V. J. **Instrumentação e Fundamentos de Medidas**. 2. ed. Rio de Janeiro: LTC, [2010].

SIEGWART, Roland; NOURBAKHSI, Illah Reza. **Introduction to autonomous mobile robots**. Cambridge: MIT Press, 2004. ISBN 0-262-19502-X.

CRAIG, John J. **Robótica**. Tradução de Heloísa Coimbra de Souza; revisão técnica de Reinaldo Augusto da Costa Bianchi. 3. ed. São Paulo: Pearson Education do Brasil, 2012.

BRYNJOLFSSON, Erik; MCAFEE, Andrew. **The Second Machine Age: Work, Progress, and Prosperity in a Time of Brilliant Technologies**. New York: W. W. Norton & Company, 2014.

CRUZ JÚNIOR, Gilmar Pereira da. **Localização e Mapeamento para Robôs Móveis em Ambientes Confinados Baseado em Fusão de LiDAR com Odometria de Rodas e Sensor Inercial**. 2021. Dissertação (Mestrado em Computação e Robótica) – Universidade Federal de Minas Gerais, Belo Horizonte, 2021.

---

JIANG, Saike; WANG, Shilin; YI, Zhongyi; ZHANG, Meina; LV, Xiaolan. **Autonomous Navigation System of Greenhouse Mobile Robot Based on 3D Lidar and 2D Lidar SLAM.** *Frontiers in Plant Science*, v. 13, 2022. DOI: 10.3389/fpls.2022.815218.

BRUMMELEN, Jessica Van; O'BRIEN, Marie; GRUYER, Dominique; NAJJARAN, Homayoun. **Autonomous vehicle perception: The technology of today and tomorrow.** *Transportation Research Part C: Emerging Technologies*, v. 89, p. 384-406, Apr. 2018.

REKLEITIS, Ioannis; BEDWANI, Jean-Luc; DUPUIS, Erick. **Autonomous Planetary Exploration using LIDAR data.** In: **IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION**, 2009, Kobe, Japão. *Proceedings of the IEEE International Conference on Robotics and Automation*. Kobe: IEEE, 2009.

XIANG, Y.; YANG, X. Q.; YANG, W. W.; MIAO, W. H. **Localization and Mapping Algorithm for the Indoor Mobile Robot Based on LIDAR.** In: **CMAME 2019. IOP Conference Series: Materials Science and Engineering**, v. 831, 2020. DOI: 10.1088/1757-899X/831/1/012021.

DURRANT-WHYTE, Hugh; BAILEY, Tim. **Simultaneous Localisation and Mapping (SLAM): Part I – The Essential Algorithms.** [2006].

KARDÓŠ, Slavomir; BALOG, Peter; SLOSARCIK, Stanislav. *Gait dynamics sensing using IMU sensor array system.* **AECE – Advances in Electrical and Electronic Engineering**, v. 15, n. 1, p. 71-76, 2019. DOI: 10.15598/aece.v15i1.2019.

AHMAD, Norhafizan; GHAZILLA, Raja Ariffin Raja; KHAIRI, Nazirah M.; KASI, Vijayabaskar. *Reviews on Various Inertial Measurement Unit (IMU) Sensor Applications.* *International Journal of Signal Processing Systems*, v. 1, n. 2, p. 256-262, dez. 2013.

NUß, Dominik. *A random finite set approach for dynamic occupancy grid maps.* In: *Proceedings*. 2017. Disponível em: <https://api.semanticscholar.org/CorpusID:265040087>. Acesso em: [10 fev 2025].

---

ZOU, Q.; SUN, Q.; CHEN, L.; NIE, B.; LI, Q. **A comparative analysis of LiDAR SLAM-based indoor navigation for autonomous vehicles.** *IEEE Transactions on Intelligent Transportation Systems*, v. 23, n. 7, p. 6907-6921, jul. 2022. DOI: 10.1109/TITS.2021.3063477.

RIBEIRO, Maria Isabel. ***Kalman and Extended Kalman Filters: Concept, Derivation and Properties.*** Lisboa: Instituto Superior Técnico, 2004.

ZHOU, Q.; PARK, J.; KOLTUN, V. **Open3D: A Modern Library for 3D Data Processing.** In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, p. 1–8, 2018. Disponível em: <http://www.open3d.org>. Acesso em: 11 fev. de 25.

GUSE, Rosana. **Módulo Ponte H L298N para controle de motor DC.** [Imagem]. Disponível em: <https://www.makerhero.com/blog/motor-dc-arduino-ponte-h-l298n/>. Acesso em: 18 fev. 2025.

YDLIDAR. Site oficial. Disponível em: <https://www.ydlidar.com/index.html>. Acesso em: 18 fev. 2025.

EPSON. What are IMUs? Epson Global. Disponível em: [https://global.epson.com/products\\_and\\_drivers/sensing\\_system/what\\_are\\_imus/](https://global.epson.com/products_and_drivers/sensing_system/what_are_imus/). Acesso em: 01 fev. 2025.

RESEARCHGATE. **Schematic diagram of rotating 2D LiDAR.** Disponível em: [https://www.researchgate.net/figure/Schematic-diagram-of-rotating-2D-LiDAR-2D-LiDAR-plane-perpendicular-to-the-360-degree\\_fig5\\_371091420](https://www.researchgate.net/figure/Schematic-diagram-of-rotating-2D-LiDAR-2D-LiDAR-plane-perpendicular-to-the-360-degree_fig5_371091420). Acesso em: 23 jan. 2025.

MV ELECTRÓNICA. **Acelerômetro e giroscópio MPU6050 GY-521.** [Imagem]. Disponível em: <https://mvelectronica.com/producto/acelerometro-y-giroscopio-mpu6050-gy-521#4>. Acesso em: 20 fev. 2025.

---

**EAIBOT.** YDLidar: Library for the Arduino YDLidar shield. Versão 1.2. [S.l.]: Eaibot, [20--]. Disponível em: <http://www.eaibot.com>. Acesso em: 4 de nov. 2024.

**ENCUKOU.** *Bresenham*. Disponível em: <https://github.com/encukou/bresenham>. Acesso em: 15 dez 2024.

**ANÁLISE MACRO. Filtro de Kalman.** Disponível em: <https://analisemacro.com.br/mercado-financeiro/filtro-de-kalman/>. Acesso em: 10 mar. 2025.

## ANEXOS

### ANEXO A – YDLIDAR X4 PRO

Item	Min	Typical	Max	Unit	Remarks
Ranging frequency	/	5000	/	Hz	5000 times per second
Scanning frequency	6	/	12	Hz	PWM or voltage speed regulation
Ranging distance	0.12	/	10	m	Indoor environment with 80% Reflectivity
Field of view	/	0-360	/	Deg	/
Absolute error	/	2	/	cm	Distance $\leq$ 1m
Relative error	/	3.5%	/	/	1m<Distance $\leq$ 6m
Tilt angle	0.25	1	1.75	Deg	/
Angle resolution	0.43 (frequency@ 6Hz)	0.50 (frequency@ 7Hz)	0.86 (frequency@ 12Hz)	Deg	Different motor frequency
Service life	/	1500	/	h	/

Item	Min	Typical	Max	Unit	Remarks
Supply voltage	4.8	5	5.2	V	Excessive voltage might damage the Lidar while low voltage could affect normal performance
Starting current	/	800	1000	mA	Instantaneous current, device needs higher current at startup
Working current	/	330	380	mA	Normal working current, at low temperature, the rotational resistance of the lidar increases, and the current will rise

---

## APÊNDICE

### APÊNDICE A – TRANSFORMAÇÕES DE PONTOS NO PLANO

Código disponível < <https://github.com/ThiagoMatheusK3/TCC-Mapeamento-LiDAR/blob/main/Pontos.py>>

---

## APÊNDICE B – ITERATIVE CLOSEST POINT

Código disponível < <https://github.com/ThiagoMatheusK3/TCC-Mapeamento-LiDAR/blob/main/icp.py>>

---

### APENDICE C – *EXTENDED KALMAN FILTER*

Código disponível < <https://github.com/ThiagoMatheusK3/TCC-Mapeamento-LiDAR/blob/main/EKF.py>>

---

## APÊNDICE D – *GRID MAPPING*

Código disponível <https://github.com/ThiagoMatheusK3/TCC-Mapeamento-LiDAR.git>

### GRID MAPPING ###

---

## APÊNDICE E – *LOG-ODDS*

Código disponível < <https://github.com/ThiagoMatheusK3/TCC-Mapeamento-LiDAR/blob/main/log-odds.py>>

---

## APÊNDICE F – REPRESENTAÇÃO NO ESPAÇO

Código disponível < <https://github.com/ThiagoMatheusK3/TCC-Mapeamento-LiDAR/blob/main/mapa.py>>