

CENTRO UNIVERSITÁRIO FEI
JOÃO VICTOR LOURENÇO AGUIAR

**COMPARAÇÃO DE FUSÃO DE SENSORES PARA ESTIMATIVA DE LOCALIZAÇÃO
DE ROBÔS DE FORMA EMBARCADA NO AMBIENTE DA ROBOCUP SMALL SIZE
LEAGUE**

São Bernardo do Campo

2025

JOÃO VICTOR LOURENÇO AGUIAR

**COMPARAÇÃO DE FUSÃO DE SENSORES PARA ESTIMATIVA DE LOCALIZAÇÃO
DE ROBÔS DE FORMA EMBARCADA NO AMBIENTE DA ROBOCUP SMALL SIZE
LEAGUE**

Qualificação de Mestrado apresentada ao Centro
Universitário FEI, como parte dos requisitos
necessários para obtenção do título de Mestre
em Engenharia Elétrica. Orientado pelo Prof.
Flavio Tonidandel

São Bernardo do Campo

2025

RESUMO

Este projeto propõe o desenvolvimento de um sistema de localização para robôs móveis robôs da categoria *Small Size* da RoboCup de futebol de robôs, a fim de melhorar seu posicionamento ao realizarem jogadas e diminuir a dependência do sistema de câmeras disponibilizados pela liga durante os jogos da categoria. O sistema de localização utiliza um filtro de Kalman para realizar a fusão de sensores, que no caso serão os *encoders* acoplados aos eixos dos motores utilizados para movimentação dos robôs, uma IMU, composta por um giroscópio e um acelerômetro, ambos disponíveis no hardware dos robôs, além do sistema de câmeras da categoria que dispõe a posição e orientação dos robôs e bolinha em campo. Os resultados parciais são resumidos à obtenção dos dados dos diferentes sensores, além da implementação dos cálculos do filtro de Kalman no microcontrolador do robô. A partir dos testes propostos espera-se investigar os pontos positivos e negativos das diferentes combinações a fim de identificar a melhor delas e alcançar um sistema de localização robusto que funciona de maneira embarcada.

Keywords: Localização. Estimativa. Correção. Sensores.

ABSTRACT

This project proposes developing a localization system for the robots of the Small Size League from RoboCup to enhance their positioning while performing plays and decrease their dependence on the global system used in league games. The localization system uses a Kalman Filter to fuse different sensors, in this case, encoders attached to the motors' shafts used to move the robot, a gyroscope, and an accelerometer, both available on the robot's hardware, besides the cameras system used by the league that gives the poses and orientations of the robots and the ball in the field. Partial results are summarized to obtain data from the different sensors used in the project, besides developing the code of the Kalman Filter calculations embedded in the robot. The strengths and weaknesses of each combination will be investigated given the tests proposed to choose the best of them and achieve a robust and embedded localization system.

Keywords: Localization. Estimation. Correction. Sensors

LISTA DE ILUSTRAÇÕES

Figura 1	– Ilustração de uma partida da SSL.	17
Figura 2	– Imagem de um robô SSL da equipe RoboFEI.	18
Figura 3	– Comparação da velocidade angular adquirida pelo sistema de visão global e pelo giroscópio.	20
Figura 4	– Padrões de cores para identificação dos robôs.	24
Figura 5	– Ilustração da montagem e funcionamento de um encoder óptico.	25
Figura 6	– Diagrama representando a diferença entre as fusões complementar, competitiva e cooperativa.	40
Figura 7	– Diagrama representando a diferença entre as fusões baseadas no nível de abstração dos dados.	43
Figura 8	– Diagrama relacionando as classificações três níveis e Dasarathy.	44
Figura 9	– Exemplo de distribuição gaussiana.	47
Figura 10	– Ciclo do filtro de Kalman discreto.	48
Figura 11	– Exemplo de predição e atualização das covariâncias do filtro de Kalman.	51
Figura 12	– Exemplo de predição e atualização das covariâncias do filtro de Kalman.	53
Figura 13	– Visão geral da fusão de sensores da odometria com o sistema de visão da categoria MSL.	58
Figura 14	– Placa de desenvolvimento STM32F411E-Disco utilizada.	65
Figura 15	– Diagrama de implementação da fusão de sensores para localização.	68
Figura 16	– Representação do posicionamento dos lasers em campo.	69
Figura 17	– Ilustração do teste de quadrado menor.	70
Figura 18	– Problema de sobreposição de câmeras.	71
Figura 19	– Ilustração do teste de quadrado maior.	72
Figura 20	– Problema da distorção da imagem de câmeras com lente olho de peixe.	73
Figura 21	– Arquitetura do software.	74
Figura 22	– Duas etapas da arquitetura desenvolvida.	76
Figura 23	– Cenários de testes do sistema de ground truth.	77
Figura 24	– Diferença entre os eixos do campo e do robô.	82

Figura 25	– Diferença das velocidades calculadas pelo acelerômetro e pelos encoders. .	87
Figura 26	– Diferença das velocidades calculadas pelo acelerômetro e pelos encoders. .	87
Figura 27	– Distância média no teste com trajeto do quadrado menor.	92
Figura 28	– Erro máximo no teste com trajeto do quadrado menor.	93
Figura 29	– Erro médio de orientação no teste com trajeto do quadrado menor.	94
Figura 30	– Erro máximo no teste com trajeto do quadrado maior.	95
Figura 31	– Erro máximo no teste com trajeto do quadrado menor.	96

LISTA DE TABELAS

Tabela 1	–	Combinações dos sensores na predição e correção do Filtro de Kalman. . .	66
Tabela 2	–	Especificações de hardware do notebook.	67
Tabela 3	–	Especificações dos LiDARs utilizados.	69
Tabela 4	–	Resultados alcançados para o eixo X no campo de teste.	78
Tabela 5	–	Resultados alcançados para o eixo Y no campo de teste.	79
Tabela 6	–	Erro e desvio padrão médios nos cenários.	79
Tabela 7	–	Resultados dos quadrantes para o eixo X.	80
Tabela 8	–	Resultados dos quadrantes para o eixo Y.	80
Tabela 9	–	Resumo dos resultados obtidos dos cenários no teste do quadrado menor. .	94
Tabela 10	–	Resumo dos resultados obtidos dos cenários no teste do quadrado maior. .	96

LISTA DE ALGORITMOS

Algoritmo 1 – Filtro de Kalman linear	51
Algoritmo 2 – Filtro de Kalman estendido	54

LISTA DE ABREVIATURAS

DC	<i>Direct Current</i>
EKF	Filtro de Kalman Estendido
FP	Filtro de Partículas
FSEV	Filtro Suave de Estrutura Variável
I ² C	<i>Inter-Integrated Circuit</i>
IMU	<i>Inertial Measurement Unit</i>
INS	<i>Inertial Navigation System</i>
KF	Filtro de Kalman
KG	Ganho de Kalman
MEMS	Sistema Microeletromecânico
MSL	<i>Middle Size League</i>
MUV	Movimento Uniformemente Variado
PCB	<i>Printed Circuit Board</i>
SPI	<i>Serial Peripheral Interface</i>
SSL	<i>Small Size League</i>
TDP	<i>Team Description Paper</i>
UKF	Filtro de Kalman <i>Unscented</i>

SUMÁRIO

1	INTRODUÇÃO	12
1.1	OBJETIVO	13
1.2	JUSTIFICATIVA	14
1.3	ESTRUTURA DA DISSERTAÇÃO	14
1.4	CONTRIBUIÇÕES	15
1.4.1	Publicações	15
2	FUTEBOL DE ROBÔS	16
2.1	<i>SMALL SIZE LEAGUE</i>	16
2.1.1	Sistema de visão da categoria SSL	18
2.1.2	Fusão de sensores na <i>Small Size League</i>	20
2.2	SENSORES	21
2.2.1	Tipos de sensores	22
2.2.2	Sistema de câmeras	23
2.2.3	Encoder	23
2.2.4	Giroscópio	25
2.2.5	Acelerômetro	26
2.2.6	<i>Inertial Measurement Unit (IMU)</i>	27
2.2.7	Calibração dos sensores	28
2.2.7.1	<i>Calibração do acelerômetro</i>	29
2.2.7.2	<i>Calibração do giroscópio</i>	31
3	LOCALIZAÇÃO DE ROBÔS MÓVEIS	32
3.1	O PROBLEMA DA LOCALIZAÇÃO	32
3.2	OS DESAFIOS DA LOCALIZAÇÃO	33
3.3	INFORMAÇÃO DISPONÍVEL	34
3.4	FUSÃO DE SENSORES	35
3.4.1	Vantagens e Desvantagens	36
3.4.2	Classificação de técnicas	39
3.4.2.1	<i>Classificação baseada na relação entre as fontes de dados</i>	39
3.4.2.2	<i>Classificação em três níveis</i>	41
3.4.2.3	<i>Classificação baseada na entrada e saída do sistema</i>	41
3.5	FILTRO DE KALMAN	42

3.5.1	Introdução	43
3.5.2	Premissas	45
3.5.2.1	<i>Sistema dinâmico linear</i>	45
3.5.2.2	<i>Características do ruído</i>	46
3.5.2.3	<i>Processo a ser estimado</i>	47
3.5.3	Equações	48
3.5.3.1	<i>Predição</i>	49
3.5.3.2	<i>Correção</i>	49
3.6	FILTRO DE KALMAN ESTENDIDO	52
4	TRABALHOS RELACIONADOS	55
5	METODOLOGIA	64
5.1	DOMÍNIO DE TESTES	64
5.2	COMBINAÇÃO DOS SENSORES	65
5.3	IMPLEMENTAÇÃO	66
5.4	<i>GROUND TRUTH</i>	68
5.5	TESTES	70
5.6	AVALIAÇÃO	71
6	RESULTADOS	74
6.1	AQUISIÇÃO DO GROUND TRUTH	74
6.1.1	Técnica utilizada e visão geral do sistema	74
6.1.2	Resultados do sistema desenvolvido	77
6.2	PARAMETRIZAÇÃO DAS COVARIÂNCIAS	81
6.3	IMPLEMENTAÇÃO DOS CENÁRIOS	81
6.3.1	Cenário 1 - Modelo + Visão	81
6.3.2	Cenário 2 - Modelo + Encoders	83
6.3.3	Cenário 3 - Modelo + IMU	85
6.3.4	Cenário 4 - IMU + Visão	87
6.3.5	Cenário 5 - IMU + Encoders	89
6.3.6	Cenário 6 - Encoders + Visão	89
6.3.7	Cenário 7 - Encoders + IMU	91
6.4	TESTE COM TRAJETO QUADRADO MENOR	91
6.5	TESTE COM TRAJETO QUADRADO MAIOR	93
6.6	CONSIDERAÇÕES FINAIS	95

7	CONCLUSÃO	97
	REFERÊNCIAS	98

1 INTRODUÇÃO

A utilização de robôs móveis cresce cada vez mais em diferentes áreas da sociedade, como na medicina, agricultura, serviços e na indústria de diferentes tipos, como química, automotiva, metalúrgica, alimentícia, entre outras. De acordo com Statista (2023), a indústria global de robótica ultrapassou 37 bilhões de dólares e, se mantida a projeção de crescimento de 3,8% ao ano, até o final de 2028 alcançará o valor de 45 bilhões de dólares, mostrando o crescimento da utilização dos robôs em diversos setores do mercado.

Apesar de os robôs estarem sendo amplamente utilizados na indústria, eles também podem ser encontrados no dia-a-dia do ser humano cada vez mais, como robôs que limpam o chão de casa, que estão cada vez mais acessíveis. Além disso, pode-se citar o desenvolvimento e pesquisa de outras categorias de robôs, como sistemas de entrega por meio de drones (BENARBIA; KYAMAKYA, 2022) e carros autônomos (PAREKH et al., 2022)

A característica que difere os robôs móveis de robôs industriais, como os robôs manipuladores utilizados para montagem de carros na indústria automotiva, é a capacidade de navegação, que acaba sendo um problema desafiador para os robôs móveis. A navegação pode ser dividida em 4 subsistemas basicamente, que são: mapeamento, localização, planejamento e desvio de obstáculos (RAJ; KOS, 2022).

No caso desse projeto, a questão principal é a localização, que é responsável por determinar a posição e orientação do robô no ambiente. Dentro do escopo da localização, os principais pontos são o posicionamento/localização global e o rastreamento da posição de robôs móveis dentro de um mapa conhecido.

De acordo com Panigrahi e Bisoy (2022), o objetivo do rastreamento da posição é acompanhar o posicionamento do robô a cada instância de tempo sabendo sua posição inicial, o que é possível por monitorar continuamente a rota do robô, seja por meio de sensores ou das equações cinemáticas que descrevem o robô. Já na localização global, a localização inicial não é conhecida e, assim, o robô deve se localizar dentro do ambiente global.

Basicamente, também segundo Panigrahi e Bisoy (2022), a localização pode ser dividida em duas etapas: a predição e a percepção. Na etapa de predição o robô faz o rastreamento da posição utilizando sensores proprioceptivos, que medem informações do próprio robô e são atualizados numa alta frequência geralmente, para estimar sua posição. Entretanto, por conta do aumento da incerteza ao longo do tempo desse tipo de sensores, para a localização global é necessário

que o robô corrija na etapa de percepção utilizando seus sensores exteroceptivos, que medem informações do robô em relação ao ambiente e são atualizados numa frequência baixa.

Para unir os dados desses diferentes sensores é utilizada uma técnica conhecida como 'fusão de sensores', cuja ideia geral é unir dados de diferentes sensores levando a uma análise mais profunda e complexa de uma situação, o que não seria possível utilizando esses dados separadamente e/ou de maneira singular (MENDES JR. et al., 2016). Assim, ao realizar a fusão dos diferentes dados, os pontos negativos de cada sensor são minimizados por ter um outro sensor que irá atuar nessa dificuldade.

Um ambiente que possui a característica de um sensor externo com uma taxa de latência alta, onde os robôs precisam se posicionar com uma ótima precisão e navegam de maneira rápida e dinâmica dentro do ambiente, é a categoria *Small Size League* (SSL) da *RoboCup*. Por conta disso, a escolha dos sensores a serem utilizados, análises, testes e conclusões estarão relacionadas com este ambiente. Basicamente, é importante que os robôs tenham um alto controle do seu posicionamento para que possam realizar jogadas e que não causem colisões com outros robôs, além de que na liga há um sistema central de visão por meio de câmeras que rodam a 60 frames por segundo.

Embora a taxa de atualização do sistema de câmeras global da liga tenha uma taxa de atualização de aproximadamente um quarto de segundo, o que aparenta ser suficiente, a dinamicidade em que os jogos ocorrem na SSL, com a bola podendo chegar a uma velocidade de $6.5m/s$, implica numa alta necessidade de que as habilidades do robô estejam embarcadas no seu processamento, tais como a navegação, controle de posição e, por conta disso, a localização. Assim, com a alta taxa de atualização do processador, os robôs têm acesso às informações dos seus diversos sensores de maneira quase instantânea, o que aumenta a performance deles durante uma partida (CHURCHLEY et al., 2015).

1.1 OBJETIVO

Este trabalho tem como objetivo realizar uma análise comparativa do uso de diferentes sensores para resolver o problema de localização e posicionamento de robôs móveis. No caso, os sensores a serem comparados serão a *Inertial Measurement Unit* (IMU), que é composta por um giroscópio e um acelerômetro, *encoders* acoplados às rodas do robô e o sistema de câmeras utilizado na categoria SSL. Com essas combinações será possível verificar as diferenças da utilização desses sensores para um sistema de localização.

1.2 JUSTIFICATIVA

De acordo com Alatise e Hancke (2020), a navegação é um problema fundamental para a robótica que acaba dependendo de outros aspectos para o seu bom funcionamento, como a localização. A tarefa de localização de robôs móveis em um ambiente arbitrário é um desafio por conta da complexidade e diversidade de ambientes, métodos e sensores envolvidos. Além disso, os problemas de localização e navegação são o que acabam limitando a performance de robôs ainda.

Além disso, analisando trabalhos desenvolvidos na Small Size League que citam a questão da fusão de sensores (RYLL et al., 2013b; BAIE et al., 2017; ZOLANVARI et al., 2015; BEHZAD et al., 2019; NAEINI et al., 2020; CHURCHLEY et al., 2015; SALEHI et al., 2023), é possível notar a falta de trabalhos que façam as devidas comparações entre os diferentes sensores usualmente usados num robô da SSL, além de trabalhos que comparem os resultados obtidos por um sistema de localização baseado em fusão de sensores com o sistema global de câmeras da categoria.

Portanto, é importante que os diferentes sensores e algoritmos utilizados para localização de robôs móveis e estimativa de posição sejam testados em um ambiente dinâmico e com uma maior velocidade, a fim de determinar os pontos fracos de cada um, determinando formas de superar esses déficits e, assim, garantir o funcionamento deles de forma adequada em outros ambientes posteriormente.

1.3 ESTRUTURA DA DISSERTAÇÃO

No Capítulo 2 é feito o estudo do referencial teórico do futebol de robôs, onde os robôs da categoria SSL são descritos, os sensores a serem utilizados no trabalho e como estes devem ser calibrados. Já no Capítulo 3 são trazidos os conceitos necessários para entendimento do problema da localização de robôs móveis, com a explicação do método de fusão de sensores, além dos algoritmos para realização do método. No Capítulo 4 são apresentados os trabalhos relacionados, onde serão analisados os trabalhos correlatos que buscam resolver o problema da localização de robôs móveis, ou a estimativa de posição, mas também guiaram a escolha dos sensores, por exemplo, desse trabalho. No Capítulo 5 é apresentada a metodologia deste trabalho, onde estão descritos os testes, o domínio onde os testes serão performados e como serão avaliados os dados retirados dos testes. Já no Capítulo ?? são apresentados os resultados parciais conseguidos até o momento da banca de qualificação deste projeto. O cronograma dos

dois anos de projeto é apresentado no Capítulo ???. O Capítulo ?? traz as considerações finais e expectativas dos resultados para a continuação do projeto.

1.4 CONTRIBUIÇÕES

Nesta seção serão destacadas as contribuições desta dissertação.

1.4.1 Publicações

Os resultados obtidos para obtenção do ground truth do robô durante os testes foram publicados no 1º Congresso Brasileiro de Robótica.

2 FUTEBOL DE ROBÔS

A ideia de robôs que jogam futebol foi proposta pela primeira vez pelo professor Alan Mackworth, em seu artigo *On Seeing Robots* (MACKWORTH, 1982). Independentemente, em outubro de 1992, um grupo de pesquisadores japoneses organizou um workshop sobre os grandes desafios em IA, onde iniciaram-se as primeiras discussões sobre usar o futebol para promoção da ciência e tecnologia. Em junho de 1993 foi organizada uma competição de robótica e, em menos de um mês, pesquisadores de fora do Japão começaram a pedir que essa iniciativa fosse ampliada para um projeto conjunto internacional (ROBOCUP, 2020).

A *RoboCup* busca promover pesquisas na área de robótica e inteligência artificial com um objetivo final de vencer a seleção campeã do mundo em 2050 com uma equipe totalmente autônoma de robôs humanoides (ROBOCUP, 2020).

O atual cenário competitivo da *RoboCup* mostra equipes tanto do ensino superior quanto do ensino básico que disputam diversas categorias, em eventos tanto a nível nacional quanto a nível internacional, tais como *RoboCup Soccer*, *RoboCup Rescue*, *RoboCup@home* e a *RoboCup Junior*.

2.1 SMALL SIZE LEAGUE

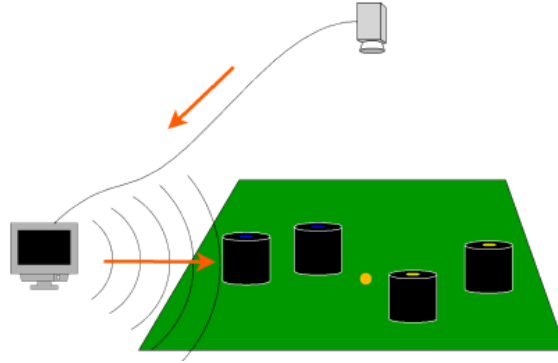
A *Small Size League* (SSL) é uma das ligas mais antigas da *RoboCup Soccer*, tendo o foco em solucionar o problema da cooperação e controle de robôs inteligentes num ambiente altamente dinâmico com um sistema híbrido centralizado/distribuído. A partida ocorre entre duas equipes utilizando seis ou onze robôs totalmente autônomos, que tem um máximo de diâmetro e altura, com algumas outras restrições (ROBOCUP, 2020). Além disso, os robôs da liga são omnidirecionais, o que proporciona jogos dinâmicos com jogadas imprevisíveis.

Os jogos desta categoria são destacados por conta da alta velocidade tanto dos robôs, que podem chegar até 4m/s, quanto da bola utilizada, que pode chegar até 6.5m/s, mas também pela quantidade de robôs numa partida, que acontece entre dois times que podem ter de 6 a 11 robôs num campo de 9m X 6m a 12m X 9m, dependendo da divisão da partida (ROBOCUP-SSL, 2019b).

O tamanho do campo em relação aos robôs, a quantidade de robôs em uma partida e as velocidades dos robôs e da bolinha são algumas características que tornam a SSL um ótimo lugar para testes de algoritmos de localização e posicionamento.

Para realização da partida, um *setup* específico é necessário. No caso, acima do campo são instaladas câmeras, as imagens delas são processadas por um computador central que disponibiliza, a partir de pacotes de rede, as posições x e y e a orientação θ dos robôs em campo, além das posições x e y da bola, como ilustrado na Figura 1.

Figura 1 – Ilustração de uma partida da SSL.



Fonte: Retirado de RoboCup-SSL (2019a)

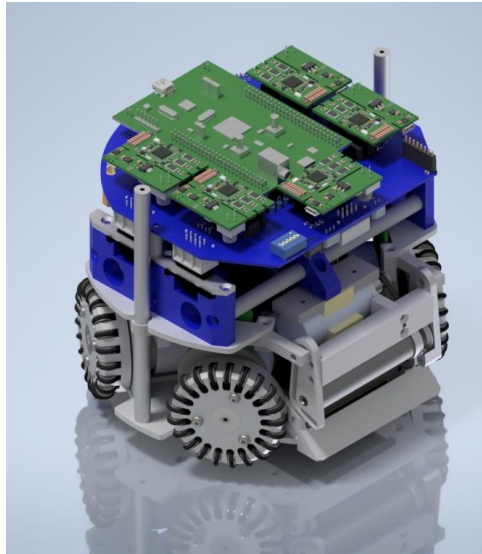
Além disso, os robôs possuem limitações de tamanho, i.e., eles devem caber num diâmetro de 180mm e possuir uma altura máxima de 150mm (ROBOCUP-SSL, 2019b). A fim de lidar com essa limitação de dimensões e possuir robôs ágeis, as equipes utilizam um sistema de deslocamento omnidirecional, que é conseguido utilizando uma adaptação de rodas mecanum, em que os roletes são montados com uma certa angulação em relação ao eixo da roda (AGUIAR, J. V. L. et al., 2022).

Com esse sistema de deslocamento omnidirecional, o robô torna-se um sistema holonômico, i.e., o robô possui controle sobre todos os graus de liberdade da sua movimentação, ou seja, a rotação dele não interfere na translação, o que torna a SSL uma liga muito dinâmica e imprevisível. A Figura 2 mostra um robô SSL da equipe RoboFEI, nela é possível observar a roda omnidirecional utilizada.

No caso, a Equação 1 descreve um robô da categoria SSL, isto é, como teoricamente a posição de um robô propaga ao longo do tempo a partir de comandos de velocidade.

$$\begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \theta_{k-1} \end{bmatrix} + \begin{bmatrix} dt & 0 & 0 \\ 0 & dt & 0 \\ 0 & 0 & dt \end{bmatrix} \times \begin{bmatrix} Vx_{k-1} \\ Vy_{k-1} \\ \omega_{k-1} \end{bmatrix} \quad (1)$$

Figura 2 – Imagem de um robô SSL da equipe RoboFEI.



Fonte: Autor

Já o sistema de visão global da categoria SSL, que faz a captura de todos os robôs em campo e da bola, pode ser caracterizado conforme a Equação 2. É possível notar que a visão fornece diretamente a posição do robô acrescida de um certo ruído σ .

$$\vec{z}_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} \pm \begin{bmatrix} \sigma_x \\ \sigma_y \\ \sigma_\theta \end{bmatrix} \quad (2)$$

2.1.1 Sistema de visão da categoria SSL

Durante a partida o processamento da tomada de decisões é feito num computador central de cada equipe, analisando as posições dos robôs e da bolinha em campo e, assim, enviando por meio de rádio frequência o que cada robô deve realizar, tal como: a velocidade de cada robô, se o robô deve chutar, se o robô deve ligar o dispositivo de drible. Atualmente, na equipe RoboFEI, o cálculo do controle de posicionamento dos robôs é feito junto ao código da equipe e, então, somente a velocidade angular e linear que o robô deve impor é passado para ele no pacote de dados via rádio.

Apesar do sistema de câmeras ser suficiente para o jogo, há problemas com o controle de posição dos robôs por conta da latência consideravelmente alta da atualização do posicionamento deles. Segundo Huang et al. (2020), a câmera envia imagens a cada 15ms, mas por conta da

filtragem realizada pelo sistema a atualização da imagem pode demorar de 3 a 4 ciclos (40 a 60ms), então há uma considerável demora para que o pacote atual com os dados seja recebido pelas equipes.

O problema de alta latência de envio das imagens compromete a realização de jogadas durante as partidas e gera dificuldades no controle de posicionamento dos robôs. Assim, nota-se uma necessidade de aplicação de outras tecnologias para resolver esse problema de posicionamento, como utilização de outros sensores embarcados nos próprios robôs.

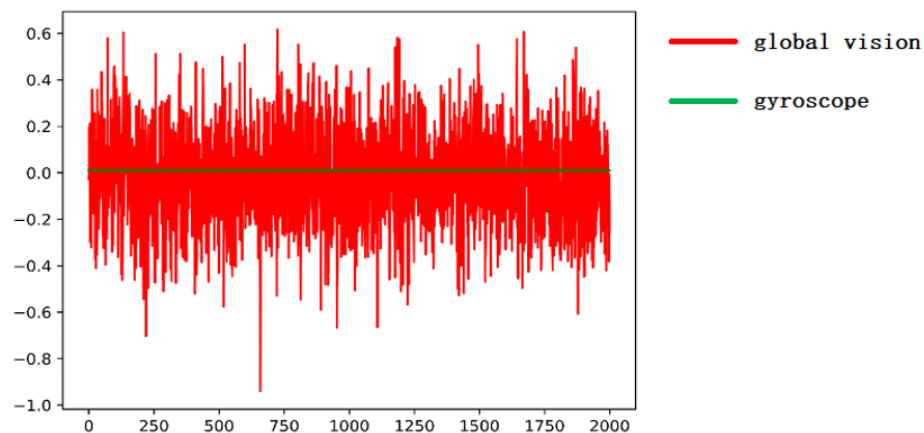
Segundo Huang et al. (2020), há quatro grandes problemas com o sistema de visão global da categoria. Em primeiro lugar, como dito, é o problema de que a frequência de atualização de 75Hz não é o suficiente para um controle de movimentação rápido e acurado. Segundo, a informação das posições que é enviada para as equipes possui uma quantidade alta de ruído, o que compromete altamente o controle de orientação dos robôs. Terceiro, a informação enviada aos times é previamente processada, levando de 3 a 4 frames (40-60ms) entre coletar a informação original da visão até obter a informação da visão. Quarto, a taxa de quadros do sistema de visão é instável, o que pode gerar perda de frames e, conseqüentemente, torna a frequência do controle instável.

A Figura 3 mostra uma comparação das informações de *feedback* do ângulo de um robô. É possível notar um ruído muito alto da informação vinda da visão global, enquanto em relação ao giroscópio mal é possível notar algum ruído. Isso mostra o quão necessário é ter um sistema de controle que não seja totalmente dependente do sistema de câmeras da categoria.

Além da equipe ZJUNlict em seu *Team Description Paper* (TDP), a equipe RoboTeam Twente em seu TDP de 2018 estima que o delay entre enviar um comando para o robô e notar uma resposta nas medições pode levar entre 80 e 150ms, dependendo da câmera sendo utilizada (DOORNKAMP et al., 2018). Por conta disso, faz-se necessário um sistema de localização que possua um delay relativamente menor para que o controle de posição dos robôs seja feito adequadamente.

É notável que o sistema de câmeras da categoria SSL facilita a aquisição dos dados de posição, visto que ele já entrega para os times o posicionamento de todos os robôs e da bolinha. Apesar disso, para realizar o controle dos robôs de maneira robusta e qualificada é necessário a utilização de um sistema de localização embarcado que realize estimativas de posicionamento dele por conta do tempo de atualização dos dados das câmeras ser relativamente alto para a aplicação.

Figura 3 – Comparação da velocidade angular adquirida pelo sistema de visão global e pelo giroscópio.



Fonte: Retirado de Huang et al. (2020)

2.1.2 Fusão de sensores na *Small Size League*

A fusão de sensores aplicada na categoria SSL, principalmente para localização dos robôs em campo numa partida, é justificada pela dinamicidade em que as jogadas acontecem por conta da velocidade máxima imposta na bola de jogo para realização de passes e chutes (ROBOCUP-SSL, 2019b), além do problema já abordado da taxa de atualização das posições capturadas pelo sistema global de câmeras.

Por conta da complexidade de trabalhar com diversas habilidades, como localização, navegação e controle, de forma embarcada, grande parte das equipes prefere executar o controle de posicionamento no computador central de estratégia, utilizando somente os dados adquiridos do sistema global de visão como feedback, e enviando para os robôs via-rádio a velocidade local que cada um deles deve impor (CHURCHLEY et al., 2015; SALEHI et al., 2023). Assim, os robôs precisam somente ter um feedback das velocidades de cada uma das rodas para manter um controle da velocidade local deles.

Apesar de ser um sistema mais simples de implementar, o controle de posição dos robôs no computador central acaba levando a uma menor performance comparado com a utilização dele de forma embarcada (CHURCHLEY et al., 2015), muito por conta do delay inerente para atualização das imagens do sistema global de câmeras da liga. Entre um frame e outro, por conta da alta velocidade da bola em jogo, somado com o delay de envio e recebimento dos dados via

rádio-frequência, aumenta a dificuldade da realização de certas jogadas e habilidades (SALEHI et al., 2023).

Para solucionar o problema do delay entre pacotes do sistema de visão global da liga, a fusão de sensores passou a ser utilizada por equipes que, nos anos seguintes, passaram a demonstrar uma qualidade de jogo superior (CHEN et al., 2018; RYLL et al., 2013a; CHURCHLEY et al., 2015; BAIE et al., 2017; SALEHI et al., 2023; ZOLANVARI et al., 2015), a fim de aumentar a realização de habilidades de forma embarcada no robô, então um sistema de localização preciso e com baixa taxa de atualização é necessário, utilizando dados mais confiáveis na malha de controle.

Em Salehi et al. (2023) é feita uma fusão de sensores para orientação entre os dados do sistema de visão e os dados da IMU, melhorando a qualidade da orientação predita. Em Behzad et al. (2019) a fusão de sensores também é feita especialmente para a orientação, sendo que a equipe coloca que comparado com a visão é a fusão de sensores é mais acurada.

A partir disso, é notória a melhoria de performance levando as habilidades do robô, como localização e navegação, para o microcontrolador do robô. A obtenção de informação dos sensores presentes nos robôs e alta taxa de processamento permitem que o robô tenha melhores estimativas de localização em campo entre um frame e outro do sistema global de câmeras, aumentando a performance dos robôs.

2.2 SENSORES

A utilização de sensores é parte essencial para o funcionamento correto de um robô. Segundo Oliveira et al. (2017), o mais predominante em robôs industriais são robôs que são projetados para realizarem operações pré-programadas, que acabam não usufruindo de sensores para atingirem seu objetivo. Entretanto, para robôs mais complexos, os sensores acabam introduzindo um maior nível de inteligência para poder interagir com o meio que está inserido por meio de atuadores.

Também de acordo com Oliveira et al. (2017), a utilização de sistemas sensoriais faz com que robôs sejam mais facilmente adaptáveis a uma maior gama de tarefas, atingindo um maior grau de universalidade, diferente dos robôs pré-programados, que acabam realizando apenas uma única função. Um robô que, a partir da leitura dos sensores, possui sensações tal como um humano, é mais facilmente treinado para realização de tarefas complexas.

Portanto, essa seção visa apresentar as diferentes categorias de sensores, tal como explicar o funcionamento geral dos sensores a serem utilizados neste trabalho e como eles devem ser calibrados para conseguir aproveitar o melhor dos dados apresentados por eles. Os sensores descritos nessa seção são os usualmente utilizados pelos robôs da SSL, por conta da configuração do sistema global de visão e do limite de tamanho descrito em regra.

2.2.1 Tipos de sensores

De acordo com Sasiadek e Hartana (2000), os sensores podem ser divididos em duas categorias principais: internos e externos. Essa diferenciação diz respeito a partir de onde vem a informação lida pelo sensor, ou seja, se é do mundo externo ou se é internamente do próprio robô.

Sensores internos fornecem informações sobre parâmetros internos do robô, ou seja, medem variáveis físicas dele, como a velocidade e o sentido de rotação de um motor, ou o ângulo de uma junta, como exemplos. Alguns possíveis sensores que fazem parte desse tipo são: encoder, giroscópio, acelerômetro, bússolas.

Já os sensores externos medem a relação entre o robô e o ambiente em que ele está inserido, que podem ser objetos naturais ou artificiais, como por exemplo a distância do robô até um objeto ou medidas químicas do ambiente. Alguns possíveis sensores que fazem parte desse tipo são: sensores de contato (bumpers), sensores de distância como laser, sonar e radar, entre outros.

Ambos sensores possuem vantagens e desvantagens. Para períodos curtos de tempo, as medições de sensores internos são bem acuradas, embora a longo prazo as medidas normalmente passam a ter desvios e erros. Ao contrário disso, os sensores externos não tem problemas de desvio do sinal ao passar do tempo, mas as medidas deles normalmente não estão sempre disponíveis, ou seja, possuem um período grande para atualizarem suas medidas.

Então, para obter resultados ótimos, ambos sensores são normalmente combinados, juntando as qualidades de ambos e fazendo com que as desvantagens deles sejam superadas. Por conta do erro de ambos os sensores, é realizada uma fusão das medidas dos dois tipos de sensores, o que irá produzir uma estimativa desejada da posição do robô.

No caso deste projeto, os seguintes sensores serão utilizados e, então, explicados de maneira mais aprofundada: câmera, encoder, giroscópio e acelerômetro.

2.2.2 Sistema de câmeras

A câmera é um instrumento cujo uso em aplicações na área da robótica tem crescido bastante. Mapeamento, localização, navegação, desvio de obstáculos, reconhecimento de objetos e inspeção de qualidade são só alguns exemplos de possíveis utilizações de câmeras para realização de tarefas por parte de robôs. No centro dessa ascensão das câmeras está a evolução tanto dos processadores quanto dos algoritmos de visão computacional avançados.

Segundo TechNexion (2023), câmeras são cruciais no campo da robótica guiada por visão por aperfeiçoar as habilidades de percepção. Um robô pode aprender muito sobre seu arredor a partir dos dados visuais que câmeras coletam. Robôs podem obter informações valiosas desses dados utilizando diferentes métodos de processamento, permitindo então que o robô enxergue, compreenda e interaja com o ambiente de uma maneira mais profunda.

No caso deste projeto, a utilização de câmeras se dá externamente aos robôs dentro da categoria SSL da RoboCup, já que, diferente da categoria *Middle Size League*, o sistema de visão não é embarcada nos robôs, embora haja diversos estudos para alocar uma câmera dentro dos robôs da categoria de pequeno porte, como trazido por Melo e Barros (2022).

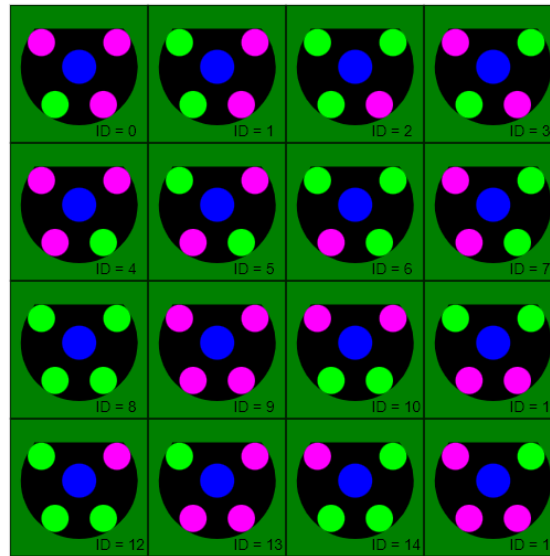
Detalhando um pouco mais o sistema de visão por câmeras da categoria SSL da RoboCup, atualmente utiliza-se uma ou duas câmeras, dependendo se o campo é da divisão A ou da divisão B. Independente do caso, a câmera fica posicionada acima do campo a 6 metros de altura e fica conectada a um computador central. Nele, a imagem é recebida, tratada e processada, identificando a posição da bola e dos robôs a partir do padrão de cor posicionado na parte de cima dos robôs, como é possível observar na Imagem 4 as 16 diferentes combinações possíveis (ZICKLER et al., 2010).

Tanto o computador onde as imagens são processadas quanto o computador de cada uma das equipes estão conectadas numa mesma rede. Assim, após o processamento das imagens, as posições dos robôs e da bola são passadas para as equipes pela rede, por isso que o sistema de visão é dito compartilhado, pois ambas as equipes recebem as mesmas informações.

2.2.3 Encoder

Encoders são dispositivos utilizados a fim de medir o estado interno e a dinâmica de um robô móvel. Eles possuem uma vasta gama de aplicações fora da robótica e, por conta disso, robôs se aproveitaram dos benefícios da alta qualidade e baixo custo de sensores que oferecem

Figura 4 – Padrões de cores para identificação dos robôs.



Fonte: RoboCup-SSL (2019b)

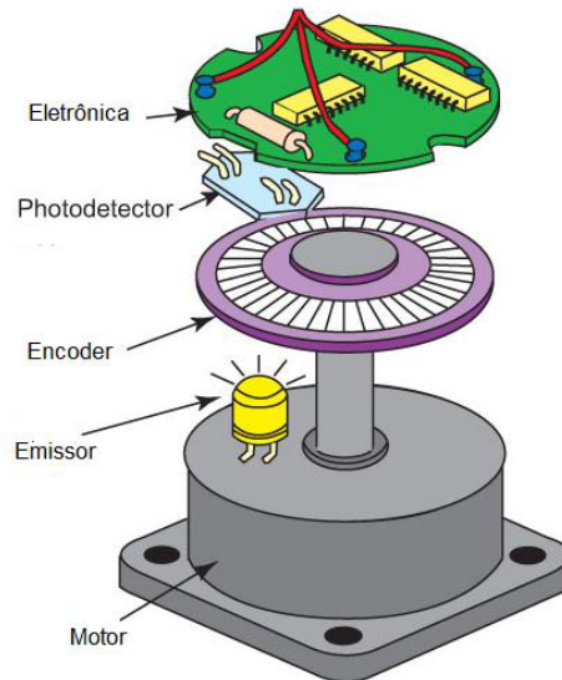
uma excelente resolução de leitura. No mercado existem alguns diferentes tipos de encoders, como os ópticos e magnéticos.

No caso deste projeto, é utilizado um encoder do tipo óptico. Segundo Siegart, Nourbakhsh e Scaramuzza (2011), este tipo de encoder se tornou o dispositivo mais popular para medição de velocidade e posição angulares de um motor, do eixo de uma roda ou mecanismo de direção.

Um encoder óptico é basicamente um picador de luz mecânico que produz uma certa quantidade de pulsos na forma de um seno ou quadrado para cada revolução. No caso, o sensor consiste de uma fonte de iluminação, uma "grade" fixa que mascara a luz, um disco rotor com uma grade óptica fina que gira com o eixo e um detector óptico fixo. Ao passo que o rotor se movimenta, a quantidade de luz atingindo o detector óptico varia baseado no alinhamento das grades fixas e móveis. É possível observar a montagem e ter uma melhor ideia do funcionamento de um encoder óptico com a Figura 5.

O projeto RoboFEI utiliza motores brushless da empresa Maxon®, no caso um motor brushless EC 45 com 50W acoplado em cada uma das rodas (MAXON GROUP, 2019). Já o encoder utilizado é da empresa US Digital®(US DIGITAL, 2024), no caso é utilizado um encoder do modelo E4T por roda também, sendo que eles ficam acoplados diretamente no eixo do motor.

Figura 5 – Ilustração da montagem e funcionamento de um encoder óptico.



Fonte: Adaptado de Luiz R. (2021)

Cada encoder está ligado a um timer do microcontrolador utilizado em cada robô para realizar a contagem de pulsos do sensor. Seguindo a Equação 3, pode-se determinar a velocidade de rotação de cada roda em rpm, conhecendo a contagem de pulsos realizada, o intervalo de tempo (em segundos), a transmissão envolvida no conjunto roda-motor e a resolução de leitura do encoder (quantidade de pulsos por volta).

$$\begin{aligned} \text{Rotação} &= \frac{\text{Pulsos}}{\text{Transmissão} \times \text{Pulsos por volta}} \\ \text{Velocidade} &= \frac{\text{Rotação} \times 60}{\Delta t} \end{aligned} \quad (3)$$

2.2.4 Giroscópio

Giroscópios também são um dos principais sensores utilizados em robôs para realização de tarefas básicas como navegação. De acordo com Dingman (2020), eles são componentes essenciais de sistemas complexos utilizados em todas aplicações aeroespaciais, industriais e na área da robótica. Giroscópios auxiliam desde aviões e barcos até drones e carros autônomos a navegarem.

Segundo Passaro et al. (2017), giroscópios são dispositivos montados em uma estrutura capazes de realizar medidas de velocidade angular caso a ela esteja girando. Esse sensor pode ser utilizado de forma sozinha ou pode estar incluso em um sistema mais complexo, como uma bússola giroscópica, uma IMU (*Inertial Measurement System*, do inglês) ou um INS (*Inertial Navigation System*, do inglês), por exemplo.

No livro '*Introduction to Autonomous Mobile Robot*', Siegwart, Nourbakhsh e Scaramuzza (2011) trazem que giroscópios são sensores de direção que preservam sua orientação em relação a um *frame* de referência fixo. Por isso, eles fornecem uma medida absoluta de orientação de um sistema móvel.

Também de acordo com Siegwart, Nourbakhsh e Scaramuzza (2011), os giroscópios são divididos em duas categorias: mecânicos e ópticos. Os primeiros dependem das propriedades de um rotor de rotação rápida, propriedade chamada de precessão giroscópica. Já os segundos são sensores de velocidade angular que utilizam dois feixes de luz monocromáticos, ou lasers, emitidos de uma mesma fonte

No caso desse projeto, o giroscópio utilizado é o I3G4250D, que é um sensor de velocidade angular de baixo consumo de energia capaz de realizar medidas nos 3 eixos (STMICROELECTRONICS, 2015). Esse componente inclui o sensor e uma interface capaz de fornecer a medida de velocidade angular ao mundo externo por meio de uma interface digital I²C (do inglês, *Inter-Integrated Circuit*) ou SPI (do inglês, *Serial Peripheral Interface*).

2.2.5 Acelerômetro

O acelerômetro é mais um dos sensores que é utilizado para que o robô possua a habilidade de entender sozinho sua localização no espaço, o que é criticamente importante para alcançar com êxito o objetivo determinado para o robô desenvolvido.

Segundo Nistler e Selekwa (2011), grande parte dos sistemas de odometria para aplicações em robótica possuem acelerômetros. Estes continuamente medem a aceleração do veículo, que é integrada para determinada a velocidade dele, e é integrado novamente para ter a medida da posição relativa ao ponto inicial.

Entretanto, por conta da influência da gravidade, da força inercial de Coriolis (PERSSON, 1998) e componentes rotacionais de aceleração, sistemas de odometria baseados em acelerômetros estão sujeitos a diversos erros dependendo do processamento das medidas do sensor. Também segundo Nistler e Selekwa (2011), quando o robô se move numa superfície horizontal, a

velocidade computada irá refletir a velocidade real do robô, mas em superfícies inclinadas, a velocidade medida irá incluir esses componentes, que não fazem parte da velocidade real.

De acordo com Dadafshar (2014), a operação básica de um acelerômetro recai na Segunda Lei de Newton, a qual diz que a aceleração de um corpo é diretamente proporcional, e na mesma direção, a força resultante atuante no corpo, e inversamente proporcional à massa do corpo, descrito na Equação 4.

$$\vec{a}(m/s^2) = \frac{\vec{F}(N)}{m(kg)} \quad (4)$$

Nota-se que a aceleração gera uma força que é capturada pelo mecanismo de detecção de força do acelerômetro. Então, o acelerômetro na verdade realiza medidas de força, e não aceleração, mas ele acaba medindo a aceleração indiretamente por meio da força aplicada em um de seus eixos.

De acordo com Siegwart, Nourbakhsh e Scaramuzza (2011), os acelerômetros são separados dependendo do princípio físico utilizado para realizar a medição da deflexão da massa interna do sensor. Um mecanismo comum de detecção utilizado é a detecção por capacitância, que mede a deflexão ao medir a capacitância entre uma estrutura física e a massa interna. Outra alternativa de medição é a piezoelétrica, que é baseada na propriedade de certos cristais em gerarem tensão quando um estresse mecânico é aplicado neles, no caso a massa interna é posicionada no cristal e, então, quando uma força externa é aplicada a massa induz uma tensão que pode ser medida.

No caso desse projeto, o acelerômetro utilizado é o LSM303AGR, que é um sensor digital de aceleração linear capaz de realizar medidas nos 3 eixos, e é um sensor digital magnético nos 3 eixos também (STMICROELECTRONICS, 2022). O componente inclui uma interface serial I²C, que suporta os modos padrão e rápido com 100kHz e 400kHz, ou uma interface serial padrão SPI.

2.2.6 Inertial Measurement Unit (IMU)

A IMU, é um dispositivo que utiliza giroscópios e acelerômetros para estimar a posição, velocidade e aceleração relativos do veículo em movimento. Este componente se tornou comum em aviões e barcos, por exemplo, por estimar a posição do veículo em seis graus de liberdade, no caso: posição(x, y, z) e orientação (*roll, pitch, yaw*) (SIEGWART; NOURBAKHSH; SCARAMUZZA, 2011).

Além disso, as IMUs comercializadas também estimam velocidade e aceleração. Considerando que a IMU possua 3 acelerômetros ortogonais e 3 giroscópios ortogonais, os dados do segundo são integrados para estimar a orientação do veículo enquanto os dados do primeiro são utilizados para estimar a aceleração instantânea do veículo.

A aceleração é, então, transformada para o frame da navegação local por meio da estimativa atual da orientação do veículo relativo à gravidade. Então, o vetor gravidade pode ser subtraído das medidas, resultando numa aceleração que é integrada para obter a velocidade e, então, integrada novamente para obtenção da posição. Para sobrepor o problema da necessidade de conhecer a velocidade inicial, a integração é tipicamente iniciada no repouso, ou seja, velocidade igual a zero.

IMUs são extremamente sensíveis na questão de erros de medidas tanto em relação ao giroscópio quanto ao acelerômetro. Por exemplo, o desvio no giroscópio inevitavelmente prejudica a estimativa da orientação do veículo relativa à gravidade, o que resulta numa cancelação incorreta do vetor da gravidade. Além disso, por exemplo, os dados do acelerômetro são integrados duplamente para obter a posição, então qualquer resíduo do vetor gravidade gera um erro que é duplamente integrado na posição. Por conta desse problema de desvio, é necessário alguma referência de fonte externa de medida, como um GPS (do inglês, *Global Positioning System*), câmera ou LiDAR (do inglês, *Light Detection And Ranging*).

2.2.7 Calibração dos sensores

O mercado mundial de sensores vem expandindo numa alta taxa ao longo dos últimos anos empurrado pelo desenvolvimento de outras tecnologias que fazem uso desses componentes, como robôs, carros autônomos, tecnologias de energia verde e internet das coisas (*Internet of Things*, do inglês), por exemplo. De acordo com BCC (2024), o mercado global de sensores estima o crescimento de \$179.7 bilhões em 2023 para \$300.5 bilhões até 2029.

Entretanto, apesar do forte avanço do mercado de sensores mundial, de acordo com Jain, Zhang e Jiang (2019), a calibração de sensores é um dos tópicos menos discutidos no desenvolvimento de sistemas autônomos, apesar de ser o bloco de fundação do sistema e de seus sensores, e é uma etapa de processamento necessária antes da implementação de técnicas de fusão sensorial.

De acordo com Lv et al. (2020), a calibração dos sensores é uma parte fundamental para o desenvolvimento de um projeto de fusão multi-sensorial. Isso se dá pelo fato do aumento da

qualidade dos dados lidos pelos sensores e, assim, uma consequente melhoria na confiabilidade do sistema como um todo. Em sistemas como robôs e carros autônomos, isso pode determinar diretamente a segurança e viabilidade deles.

No caso desse projeto, a calibração será realizada para os sensores giroscópio e acelerômetro, que são chamados de sensores inerciais microeletromecânicos (MEMS, do inglês), cujo desenvolvimento foi o protagonista para o crescimento de sistemas de navegação inerciais (INS, do inglês) e superar os pontos negativos de outros sensores, como o GPS, que são consideravelmente lentos para atualizar a informação.

Segundo Papafotis e Sotiriadis (2020), sensores inerciais MEMS são utilizados preferencialmente por conta tanto do seu baixo custo quanto do seu reduzido tamanho. Entretanto, uma grande desvantagem deles é a sua característica de grande erro. Por conta disso, a calibração desses sensores é necessária para garantir seu bom funcionamento num INS, compensando a parte determinística de seu erro.

2.2.7.1 Calibração do acelerômetro

A calibração do componente acelerômetro pode ser dividida em duas partes diferentes: compensação do *bias* da leitura de cada um dos eixos e a calibração da inclinação da IMU em relação ao frame do robô, este que acontece por conta da solda do componente ou de algum fator mecânico, o que faz com que os eixos do robô e do componente provavelmente não sejam compatíveis.

A calibração é realizada pelo método proposto por Menezes Filho et al. (2020), que é baseado numa estimativa pelo método dos mínimos quadrados. No caso, o método é uma adaptação de uma técnica utilizada para calibração de magnetômetros (dispositivos que medem a força do campo magnético), alterando que ao invés de medir o campo magnético da Terra é medida a aceleração local da gravidade. Além disso, o método para acelerômetros deve ser realizado enquanto o componente esteja estacionário ou submetido a rotações que não produzam forças centrípetas detectáveis.

De acordo com Menezes Filho et al. (2020), há 4 componentes que compõe o erro de um acelerômetro, que são:

- a) **Desvio:** Chamado de *bias* em inglês, o desvio é o componente mais comum na calibração de acelerômetro. Esse tipo de erro adiciona um deslocamento nas leituras do sensor.

- b) **Fatores de escala:** Esse componente corrompe as medições ao escalar incorretamente elas.
- c) **Desalinhamentos:** Também conhecido como não-ortogonalidades ou erros de acoplamento cruzado, esse componente diz respeito à disposição angular entre os eixos do corpo e os eixos reais de sensibilidade. O efeito prático desses desalinhamentos é que um sensor acaba sentindo (leia-se, medindo) a aceleração dos outros eixos, então a leitura não é completamente relacionada ao seu respectivo eixo.
- d) **Ruídos aleatórios:** Este componente é assumido como ruído branco e com média zero seguindo uma distribuição Gaussiana. Os autores colocam que como a aceleração é realizada no domínio dela própria, então nenhuma integração numérica é realizada, então esses processos aleatórios podem ser desconsiderados.

De acordo com Hassan e Bao (2020), apenas os três primeiros itens são considerados durante a modelagem de erros sistemáticos. Assim, a leitura de um acelerômetro pode ser descrita conforme a Equação 5, em que a é o vetor que representa as acelerações calibradas nos 3 eixos, S é a matriz que representa os erros de fator de escala e desalinhamento (no caso, na diagonal principal estão os dados do erro de fator de escala, enquanto o restante dos valores são os erros de desalinhamento), enquanto \tilde{a} representa os dados crus do sensor nos 3 eixos.

$$a = S(\tilde{a} - B)$$

$$\begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} S_x & S_{xy} & S_{xz} \\ S_{xy} & S_y & S_{yz} \\ S_{xz} & S_{yz} & S_z \end{bmatrix} \left(\begin{bmatrix} \tilde{a}_x \\ \tilde{a}_y \\ \tilde{a}_z \end{bmatrix} + \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \right) \quad (5)$$

Como dito anteriormente, o método se baseia que durante o repouso do sensor, o módulo da aceleração nos três eixos deve ser igual à aceleração da gravidade, como descrito na Equação 6.

$$a_x^2 + a_y^2 + a_z^2 = g^2 \quad (6)$$

A partir disso, devem ser recolhidas amostras do sensor em diferentes posições para que a calibração tenha uma maior eficiência, já que em Menezes Filho et al. (2020) é mostrado que o método consegue melhores resultados ao aumentar a quantidade de amostras, sendo que os autores colocam que 9 é a quantidade mínima nesse caso.

2.2.7.2 Calibração do giroscópio

A calibração do giroscópio segue as mesmas ideias da calibração do acelerômetro. Entretanto, em Hassan e Bao (2020) somente o fator de escala e os desvios são considerados no estudo como fontes de erro.

Assim, a relação entre a velocidade angular real e a velocidade angular medida pode ser observada com a Equação 7, em que ω é o vetor que representa as velocidades angulares calibradas nos 3 eixos, enquanto $\tilde{\omega}$ representa o vetor com os dados crus do sensor nos 3 eixos.

$$\omega = S(\tilde{\omega} - B)$$

$$\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} = \begin{bmatrix} S_x & S_{xy} & S_{xz} \\ S_{xy} & S_y & S_{yz} \\ S_{xz} & S_{yz} & S_z \end{bmatrix} \left(\begin{bmatrix} \tilde{\omega}_x \\ \tilde{\omega}_y \\ \tilde{\omega}_z \end{bmatrix} + \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \right) \quad (7)$$

Entretanto, como nesse caso as amostras também devem ser tomadas em repouso, então a aceleração real deve ser considerada nula. Portanto, os erros de fator de escala podem ser desconsiderados no caso desse projeto, sendo necessário apenas descobrir os desvios do sensor.

3 LOCALIZAÇÃO DE ROBÔS MÓVEIS

Nessa seção serão comentadas as questões que envolvem como um todo a localização de robôs móveis. No caso, serão discutidos a questão geral da localização de robôs e seus principais problemas, suas diferentes instâncias e a questão da informação disponível para localização de robôs.

3.1 O PROBLEMA DA LOCALIZAÇÃO

A navegação é uma das competências mais desafiadoras necessárias em um projeto de robô móvel. De acordo com Siegwart, Nourbakhsh e Scaramuzza (2011), o sucesso da navegação depende do sucesso de 4 pilares principais: percepção, localização, cognição e controle de movimento. O primeiro é como o robô interpreta os dados dos sensores para extrair dados significativos. O segundo é a determinação da posição do robô no ambiente. O terceiro é sobre como o robô deve decidir suas ações para atingir seus objetivos. O quarto define como o robô deve modelar as saídas dos seus motores para atingir a trajetória desejada.

Este trabalho se debruça no pilar da localização basicamente. O problema da localização consiste em responder a pergunta "Onde estou?" do ponto de vista do robô, o que quer dizer que o robô precisa descobrir sua localização relativa ao ambiente em que ele se encontra. Quando fala-se sobre posição, quer dizer sobre as coordenadas x e y do robô, tal como sua orientação no sistema de coordenadas global.

Como dito em Thrun et al. (2001), o problema de localização de um robô é algo muito importante, sendo um componente chave em diversos sistemas robóticos autônomos de sucesso. Se um robô não sabe onde está relativamente ao ambiente, a tomada de decisão do que fazer é praticamente impossível, o robô precisa ter pelo menos uma certa noção de onde ele está para poder operar e agir de maneira certa. Segundo Borenstein et al. (1997), saber exatamente a posição de um robô é um problema fundamental em aplicações de robôs móveis para prover realmente capacidades autônomas.

Problemas de localização são caracterizados pelo tipo de conhecimento que está disponível inicialmente e durante a execução. Basicamente, há três tipos de problemas de localização com diferentes graus de dificuldade, que são:

- a) **Rastreo de posição:** Assume que a posição inicial do robô é conhecida, então a localização do robô pode ser conseguida ao acomodar o ruído na movimentação do

robô, geralmente o efeito desse ruído sendo pequeno. Esse problema é dito como local, já que a incerteza é local e restrita a uma região perto da posição verdadeira do robô, além de que a incerteza é geralmente aproximada por uma distribuição unimodal, como uma gaussiana.

- b) **Localização Global:** Aqui a posição inicial do robô é desconhecida, já que o robô é inicialmente colocado em algum local do ambiente, mas há a falta do conhecimento de onde é o local, ou seja, ele precisa se localizar do zero. As abordagens para localização global não podem assumir limite no erro da posição, assim como assumir distribuição probabilística unimodal é geralmente inapropriado.
- c) **Problema do sequestro de robô:** É uma variante do problema de localização global, mas nesse caso o robô sabe onde está localizado e de repente é "sequestrado" para outra localização sem que o robô esteja ciente disso. O problema é o robô detectar que foi sequestrado e, em seguida, descobrir sua nova localização. A importância prática disso, apesar de ser algo que não aconteça frequentemente, decorre que grande parte dos algoritmos de localização não garantem que nunca falharão.

3.2 OS DESAFIOS DA LOCALIZAÇÃO

Ao falar dos desafios da localização, Siegwart, Nourbakhsh e Scaramuzza (2011) falam sobre a situação hipotética de utilizar um GPS (do inglês, *Global Positioning System*) em um robô móvel e como o problema de localização estaria evitado, já que o sensor informaria a posição exata interna e externamente e, então, a questão "Onde estou?" sempre estaria respondida. Porém, infelizmente, esse sensor não é prático, já que o GPS atual tem uma acurácia de alguns metros, o que é inaceitável para localizar robôs móveis, além de não funcionar em ambientes internos.

Indo mais a fundo nas limitações do GPS, a localização é mais do que saber a posição absoluta do robô em relação à Terra, é também saber a sua localização relativa em respeito a, por exemplo, humanos, considerando a situação de um robô que tem que interagir com pessoas. Além do mais, se o robô planeja atingir uma localização específica, talvez seja necessário adquirir um modelo do ambiente (um mapa) e, então, identificar a posição relativa do robô nele.

Por conta da falta de acurácia e imperfeição de sensores e atuadores que a localização enfrenta desafios difíceis. Os principais aspectos que tornam o funcionamento de sensores e atuadores sub-ótimos são: ruído e *aliasing* em sensores e ruído em atuadores.

Sensores são a entrada fundamental do robô para o processo de percepção e, portanto, o grau em que sensores podem discriminar o estado em relação ao mundo que o robô se encontra é crítico. O ruído induz uma limitação na consistência das leituras de um sensor em um mesmo estado do ambiente. Geralmente, a fonte de problemas com ruídos em sensores é que algumas características não são capturadas pelo robô e, então, ignoradas. Resumindo, o ruído em sensores reduzem a informação útil da leitura deles, uma saída para isso é levar várias leituras em conta, aplicando fusão temporal ou fusão de diversos sensores para aumentar a qualidade geral da informação de entrada de robôs.

Outra deficiência em relação aos sensores é a questão do *aliasing*, que os leva a colherem pouco conteúdo informativo, o que acaba agravando o problema da percepção e, assim, dificultando a localização de robôs móveis. Um exemplo que mostra bem a questão do *aliasing*, trazido em Siegwart, Nourbakhsh e Scaramuzza (2011), é que a utilização de um sonar em um robô móvel não traz a informação se algo que foi detectado é um humano que o robô deveria dizer "com licença" ou um objeto inanimado que o computador deveria recalcular o trajeto para ultrapassar. Ou seja, a quantidade de informações é geralmente insuficiente para identificar a posição do robô a partir de uma leitura de percepção única.

Já o ruído em atuadores cai na questão de que uma única ação tomada por um robô móvel pode ter diferentes resultados possíveis, mesmo que da perspectiva do robô o estado inicial antes da ação tomada é bem conhecido. Em resumo, atuadores em robôs móveis introduzem incerteza sobre o estado futuro, como por exemplo, o ato de andar tende a aumentar a incerteza de um robô. A maior fonte de erro geralmente reside em um modelo incompleto do ambiente, como por exemplo o fato de o modelo não levar em conta que as rodas de um robô podem escorregar ou que um humano pode empurrar o robô, ou seja, não leva em conta possíveis fontes de erros que não podem ser modeladas, resultando numa falta de acurácia entre o movimento físico do robô, a movimentação pretendida pelo robô e a estimativa de movimentação pelo sensor.

3.3 INFORMAÇÃO DISPONÍVEL

Para determinar sua localização, um robô tem acesso a dois tipos de informação, primeiro por meio de uma compreensão a priori obtida pelo próprio robô ou suprida por uma fonte externa numa fase chamada de inicialização, segundo o robô obtém informação sobre o ambiente por meio de cada observação e ação realizadas durante a fase chamada navegação.

Em geral, a informação a priori fornecida ao robô descreve o ambiente pelo qual o robô está navegando, ou seja, especifica algumas características que são variantes no tempo e assim podem ser utilizadas para determinar a localização. Alguns exemplos desse tipo de informação podem ser mapas e relações causa-efeito.

Robôs podem ter acesso a um mapa que descreve o ambiente em que está localizado. Os mapas podem ser topológicos ou geométricos (MENDES, 2017), o primeiro tipo descreve o ambiente em termos métricos, como por exemplo mapas de rodovias, já o segundo tipo descreve o ambiente em termos de características específicas em localizações e maneiras de ir de um local para outro. O mapa pode ser aprendido pelo robô previamente, ou fornecido por uma fonte externa, ou aprendido enquanto navega pelo ambiente. Já as relações causa-efeito fornecem informações a priori ao robô por meio de uma dada entrada de observação, dizendo ao robô onde ele está a partir delas.

Robôs também tem acesso à chamada informação de navegação, que é o tipo de informação que o robô reúne de diferentes sensores enquanto navega pelo ambiente. Um robô tipicamente performa dois tipos de ações ao navegar: ele anda ou age no ambiente por um lado, e sente o ambiente por outro lado.

Um sistema de locomoção consiste de rodas, pernas ou trilhos, ou qualquer coisa que faça o veículo se movimentar pelo ambiente. A maneira na qual um sistema de deslocamento muda a localização contém informação valiosa para realizar a estimativa da própria localização, ou seja, saber os efeitos de ações executadas pelo sistema indica diretamente a localização do veículo depois da execução dessas ações.

O robô sente o ambiente por meio de sensores, que indicam a informação de uma situação momentânea, chamada de observação ou medição, ou seja, essa informação descreve uma situação do ambiente do robô em um certo momento. Observações feitas do ambiente providenciam informação sobre a localização do robô que é independente de uma estimativa de localização anterior, dando ênfase que a informação dessas medições vem da observação do ambiente ao invés do próprio robô.

3.4 FUSÃO DE SENSORES

Alatise e Hancke (2020) trazem que robôs móveis autônomos estão se tornando mais proeminentes nos últimos tempos por conta do aumento de sua relevância e aplicações em diversas áreas, como em empresas, indústrias, hospitais, setor agrícola, realizando funções como

carregamento de objetos pesados, monitoramento e busca. Por conta disso, a fusão de sensores vem sendo utilizada para solução de problemas como localização, mapeamento e navegação.

A fusão de sensores é um tema que envolve uma grande multidisciplinaridade de áreas, por conta disso existem diversas definições do que é fusão de sensores na literatura, como a definição de Castanedo et al. (2013) e Nagla, Uddin e Singh (2014), que definem como o uso cooperativo de informação providenciada por diversos sensores a fim de ajudar no desempenho de uma determinada função. Trazendo mais para o campo da robótica, Luo, Chang e Lai (2011) trazem que a fusão de multi sensores é uma tecnologia que realiza a combinação sinérgica de dados sensoriais de múltiplos sensores a fim de atingir inferências que não são possíveis com os sensores operando separadamente.

A ideia de unir sensores não é recente na história da humanidade, um exemplo muito bom para mostrar esse fato é que, de acordo com Hall e Llinas (1997), humanos e animais desenvolveram a capacidade de utilizar múltiplos sentidos para melhorar suas habilidades de sobrevivência, como no caso de um animal que não consegue ver ao redor de cantos ou por meio da vegetação, então o sentido de audição pode prover bons avisos de perigos. Assim, a fusão de sensores é naturalmente realizada por animais e humanos para uma melhor abordagem do ambiente ao redor e para identificação de ameaças.

De acordo com MÁRTON e GYÖRGY (2013), a fusão de sensores é um método efetivo para solucionar o problema de localização precisa de robôs móveis. Nessa técnica, mais de um sensor é utilizado para obter a posição do robô e para uma combinação efetiva de diferentes medições a fim de gerar os estados estimados do sistema. Assim, a fusão de sensores permite a mitigação das limitações de diferentes sensores, obtendo uma posição mais precisa do robô.

A seguir, nessa seção serão comentadas as vantagens e desvantagens de utilizar fusão de sensores em sistemas inteligentes, além de apresentar três tipos de classificação dos diferentes métodos e técnicas de fundir dados de múltiplos sensores em um sistema para melhor entendimento da técnica utilizada neste projeto.

3.4.1 Vantagens e Desvantagens

Fung, Chen e Chen (2017) trazem que a maioria dos sensores não geram diretamente um sinal de um fenômeno externo, mas sim através de diversas etapas de conversão. Por conta disso, o dado sensorial lido pelo usuário pode desviar da entrada real. O autor também coloca que existem algumas características de sensores que são inevitáveis, como velocidade e frequência de

resposta, atraso e tempo de acomodação, e que acabam levando a diversas complicações, que são enfrentadas pela fusão de sensores. Além disso, existem outras características estáticas, como acurácia, precisão, resolução e sensibilidade, que podem ser facilmente geridas antes do processo de fusão.

Fung, Chen e Chen (2017) trazem também que a maior parte dos sensores não são ideais e possuem desvios que podem vir junto da informação necessária, alguns deles podem ser considerados de uma fonte aleatória de ruído, que precisam de um processamento para redução, já outros são considerados sistemáticos correlacionados com o tempo, estes também podem ser melhorados se o erro é conhecido.

Como comentado anteriormente, o principal propósito de sensores externos é prover ao sistema informação útil no que diz respeito a informações de interesse do ambiente. A fusão de dados de diferentes sensores traz diversas vantagens relacionadas a obtenção de informações mais precisas, que no caso são impossíveis de perceberem somente com os dados individuais. Segundo Alatise e Hancke (2020), os seguintes itens são as principais vantagens da fusão de dados de sensores.

- a) **Redução da incerteza:** os dados providenciados por sensores estão, por vezes, sujeitos a um nível de incerteza e discrepância. Assim, a fusão de dados de diferentes sensores reduz a incerteza ao combinar dados de inúmeras fontes. É, assim, imperativo compensar usando outros sensores ao fundir seus dados utilizando algoritmos de fusão.
- b) **Aumento na acurácia e confiabilidade:** integração de múltiplos sensores vai permitir que o sistema providencie informação inerente mesmo em caso de falha parcial em algum de seus módulos sensoriais.
- c) **Cobertura temporal e espacial estendida:** a área coberta por um sensor pode não ser coberta por outro sensor, portanto a medição de um é dependente do outro e ambos se complementam. Um exemplo em que ocorre isso é um sensor inercial, como acelerômetro e giroscópio, e visão, nesse caso a cobertura da câmera como sensor de visão não pode ser comparada com o uso do acelerômetro, que apenas pega medidas sobre a rota de navegação.
- d) **Resolução aprimorada:** o valor da resolução resultante de múltiplas medições independentes fundidas é melhor que a medição singular de um sensor.
- e) **Complexidade reduzida do sistema:** um sistema em que os dados do sensor são pré-processados por algoritmos de fusão, a entrada para a aplicação de controle

pode ser padronizada de forma autônoma dos tipos de sensores empregados, assim simplificando a implementação e providenciando a opção de modificações no sistema de sensor relativo ao número e tipo dos sensores sem alterações do software aplicado.

Embora seja provado a qualidade da fusão de sensores, de acordo com Fung, Chen e Chen (2017), existem alguns problemas associados com a criação de uma metodologia geral para fusão de diferentes sensores e eles se concentram em torno dos métodos utilizados para modelagem do erro ou incertezas no processo de integração dos dados, na informação sensorial e na operação do sistema em geral incluindo os sensores. Sendo assim, os seguintes itens são colocados pelo autor como potenciais problemas.

- a) **Registro dos dados:** sensores individuais possuem seus próprios frames de referência do qual os dados são calculados. Para que a fusão ocorra, os conjuntos de dados diferentes devem ser convertidos para um frame de referência comum, e assim alinhados juntos. Erro de calibração de sensores individuais deve ser abordado durante este estágio. Este problema acaba sendo crítico na determinação se a fusão funcionará ou não.
- b) **Incerteza nos dados sensoriais:** Diversos formatos de dados podem criar ruídos e ambiguidade no processo de fusão. Dados competitivos ou conflitivos podem ser resultados desses erros. A redundância dos dados de diversos sensores precisa estar engajada em reduzir a incerteza e aprender a rejeitar valores discrepantes se dados conflitivos são encontrados.
- c) **Dados incompletos, inconsistentes e falsos:** dados são considerados incompletos se os dados observados permanecem os mesmos independente do número de interpretações. Sensores inconsistentes são definidos como dois ou mais conjuntos de dados completos mas que possuem diferentes interpretações.
- d) **Associação de dados/Correspondência:** um aspecto da fusão de sensores é estabelecer se duas faixas de cada sensor representam o mesmo objeto, sendo isto necessário para saber como características de dados combinam de diferentes sensores, além de saber se podem ser discrepantes.
- e) **Granularidade:** o nível de detalhes de diferentes sensores são dificilmente similares. Os dados podem ser esparsos ou densos, relativos a outros sensores. O nível dos dados pode ser diferente e isso deve ser abordado no processo de fusão.

- f) **Escala de tempo:** sensores podem medir o mesmo ambiente em taxas diferentes. O tempo de chegada ao nó de fusão pode não coincidir por conta de atrasos de propagação no sistema. Em casos em que o algoritmo de fusão necessita do histórico dos dados, o quão rápido o sensor consegue prover o dado é diretamente relacionado à validade dos resultados.

3.4.2 Classificação de técnicas

Após o entendimento do que é a fusão de sensores, como ela pode ajudar diferentes sistemas e alguns pontos dela que merecem certa atenção para evitar problemas, é necessário diferenciar as diversas técnicas que realizam essa função de unir dados de sensores. De acordo com Castanedo et al. (2013), esse tema é uma área multidisciplinar que envolve diferentes campos do conhecimento, então é difícil estabelecer uma classificação clara e estrita das diferentes técnicas. Por isso, foram escolhidas 3 maneiras para classificação dos diferentes métodos de fusão sensorial.

3.4.2.1 Classificação baseada na relação entre as fontes de dados

De acordo com Castanedo et al. (2013), a relação entre as fontes de dados é uma maneira de dividir as diferentes técnicas de fusão de sensores, separando nas seguintes três categorias.

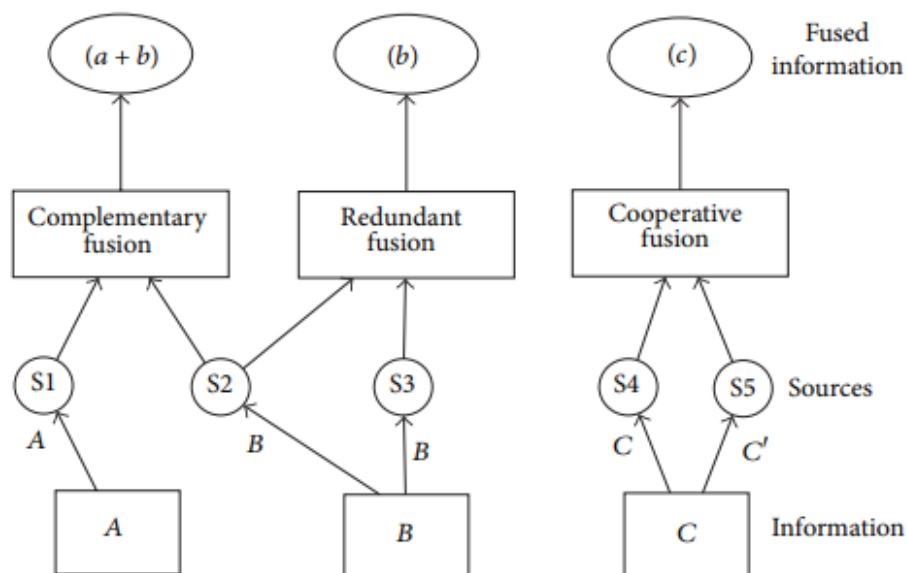
- a) **Complementar:** é o caso de quando os sensores não dependem diretamente entre si, mas podem ser combinados de uma maneira que entreguem uma visão mais completa do fenômeno sendo observado. Ou seja, a informação providenciada pelas diferentes fontes representam diferentes partes do cenário. Um exemplo que pode ser colocado são câmeras em uma sala sendo que cada uma acaba observando partes disjuntas.
- b) **Competitiva:** também chamada de redundante, é o caso em que cada sensor entrega medidas independentes de uma mesma propriedade e, assim, as informações podem ser utilizadas a fim de obter uma informação global mais confiável. Visser e Groen (1999) ainda separam essa categoria em dois - a fusão de dados de diferentes sensores ou a fusão de medições de um mesmo sensor tomadas em diferentes instantes. Um exemplo é o caso de dados vindo de áreas sobrepostas em redes de sensores visuais.

- c) **Cooperativa:** é quando as informações fornecidas por dois sensores independentes são utilizadas para conseguir alguma informação que não estaria disponível com os sensores funcionando sozinhos. De acordo com Brooks e Iyengar (1998), é a fusão mais difícil de projetar, já que o dado resultante está suscetível a problemas de todos os sensores sendo fundidos, o que geralmente diminui a acurácia e confiabilidade em relação às outras categorias. Um exemplo é uma fusão de dados multi-modal entre áudio e vídeo para gerar uma informação mais complexa.

De acordo com essa classificação, esse projeto transitará entre os tipos complementar e competitiva, já que dependerá da combinação de sensores a ser empregada. Por exemplo, o encoder e o giroscópio podem prover observações sobre a velocidade angular, o que é uma fusão competitiva, mas há combinações que estarão no caso de fusão complementar por apresentarem dados diferentes de uma mesma cena.

A Figura 6 representa claramente a diferença entre as três categorias da classificação proposta por Castanedo et al. (2013).

Figura 6 – Diagrama representando a diferença entre as fusões complementar, competitiva e cooperativa.



Fonte: Retirado de Castanedo et al. (2013)

3.4.2.2 *Classificação em três níveis*

A fusão de dados normalmente aborda três níveis de abstração: medidas, características e decisões. Essa maneira de classificar os diferentes métodos de fusão de sensores se baseia nessa ideia, dividindo-os em baixo, intermediário e alto nível, como é descrito a seguir (CASTANEDO et al., 2013).

- a) **Nível baixo:** também chamada de fusão de dados crus, essa categoria combina diferentes fontes de dados crus para produzir novos dados que espera-se que sejam mais informativos do que os de entrada.
- b) **Nível intermediário:** também chamada de fusão a nível de características, essa categoria combina diversos aspectos, como bordas, linhas, texturas ou posições em um mapa de características que pode então ser utilizado para segmentação ou detecção.
- c) **Nível alto:** também chamada de fusão de decisões, essa categoria combina decisões de diversos especialistas para obter uma decisão ainda mais precisa. Normalmente métodos bayesianos são empregados neste nível.
- d) **Nível múltiplo:** esse nível aborda dados providenciados de diferentes níveis de abstração. Um exemplo é a união de uma medição com uma característica para obter uma decisão.

Segundo essa classificação, esse projeto utiliza uma fusão em baixo nível, visto que utiliza dados crus de sensores a fim de prever a localização de um robô móvel, que é um dado mais informativo do que os dados de entrada.

3.4.2.3 *Classificação baseada na entrada e saída do sistema*

Esse sistema de classificação proposto por Dasarathy (1997) (por isso também chamado de modelo de Dasarathy) refinou o modelo de classificação em três níveis, dividindo as diferentes técnicas de fusão de sensores em 5 categorias baseado no nível de abstração dos dados de entrada e saída do sistema. As características das 5 categorias são resumidas de acordo com a natureza da entrada e os resultados de saída do processo de fusão (LI et al., 2021; VAKIL et al., 2021).

- a) **Entrada de dados-Saída de dados (DAI-DAO):** é o tipo mais básico de fusão de dados, nele há o processo de entradas e saídas de dados crus, os resultados

são tipicamente mais confiáveis e acurados. Nesse caso a fusão é conduzida imediatamente depois da coleta de dados dos sensores.

- b) **Entrada de dados-Saída de características (DAI-FEO):** nesse nível o processo de fusão emprega dados crus das fontes para extrair características ou aspectos que descrevem a entidade do ambiente.
- c) **Entrada de características-Saída de características (FEI-FEO):** nesse nível tanto a saída quanto a entrada dos dados do processo de fusão são características. Assim, o processo aborda um conjunto de aspectos a fim de melhorar, refinar ou obter novas características.
- d) **Entrada de características-Saída de decisões (FEI-DEO):** este nível recebe um conjunto de características e, a partir delas, fornece um conjunto de decisões como saída do sistema. A maior parte dos sistemas que realizam uma decisão baseada no recebimento de dados de sensores entram nessa categoria.
- e) **Entrada de decisões-Saída de decisões (DEI-DEO):** esse tipo de classificação é também conhecida como fusão de decisão, já que funde decisões de entrada para obtenção de melhores ou novas decisões.

A partir dessa classificação, pode-se dizer que a fusão de sensores desse projeto funcionará com entrada de dados e saída de dados, já que serão utilizados dados de posição, aceleração e velocidade para conseguir melhores dados de posição e orientação.

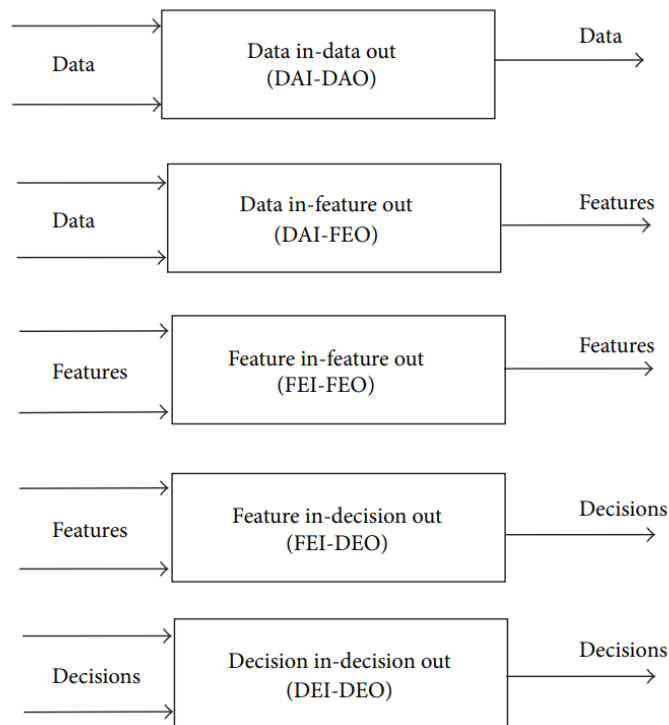
A Figura 7 representa claramente a diferença entre as cinco categorias de classificação proposta por Dasarathy (1997). Já a Figura 8 relaciona e mostra as diferenças entre os modelos de classificação em três níveis e Dasarathy.

3.5 FILTRO DE KALMAN

Nesta seção serão abordados os conceitos teóricos necessários para entendimento do tão divulgado e utilizado filtro de Kalman. Nela, serão apresentadas suas equações, as premissas para desenvolvimento das equações, as etapas do algoritmo, assim como a apresentação do filtro de Kalman estendido, utilizado em situações cujo sistema e/ou medição são não-lineares.

No caso desse trabalho, o filtro de Kalman é o algoritmo utilizado para realizar a fusão de diferentes sensores a fim de estimar a localização de robôs móveis, por isso a importância da descrição detalhada do seu funcionamento ao longo dessa seção.

Figura 7 – Diagrama representando a diferença entre as fusões baseadas no nível de abstração dos dados.



Fonte: Retirado de Castanedo et al. (2013)

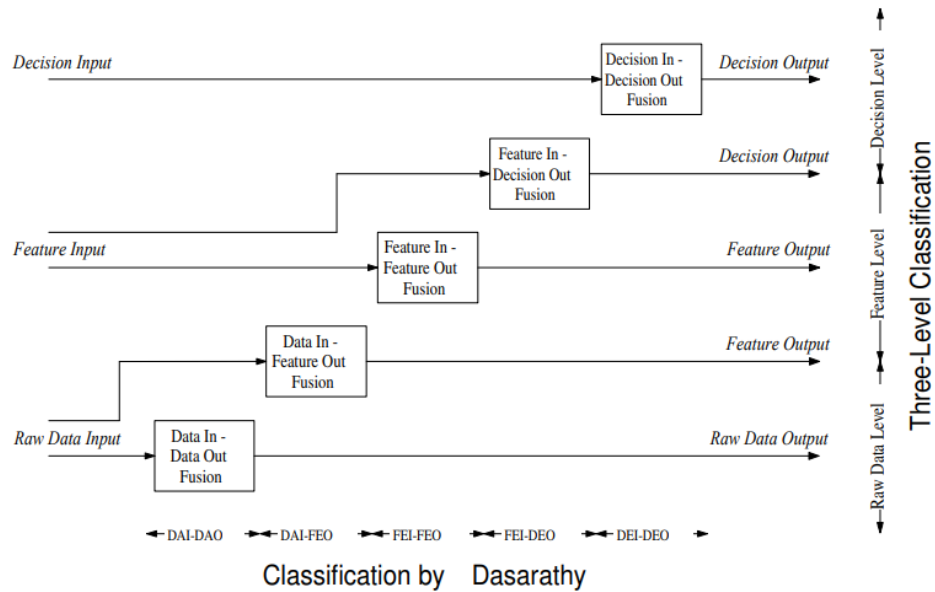
3.5.1 Introdução

O filtro de Kalman foi inventado durante a década de 50 por Rudolph Emil Kalman como uma técnica para filtragem e predição em sistemas lineares. Desde então, por conta dos avanços na área de computação digital, o KF é objeto de extenso estudo e aplicações, particularmente na área de navegação autônoma ou assistida.

O filtro de Kalman é um algoritmo que já foi utilizado em uma vasta gama de aplicações, principalmente na área de controle e na predição de sistemas dinâmicos, sendo a base para o desenvolvimento da teoria do controle moderno e processamento de sinais em tempo real. Nos dias de hoje, segundo Khodarahmi e Maihami (2023), o KF evoluiu de um simples estimador de estados ótimo e possui aplicações na automação, posicionamento, rastreamento de alvo, processamento de sinais, imagens digitais, sinais de voz e previsão de terremotos.

Focando mais no campo da robótica, o filtro de Kalman é aplicado no rastreamento de trajetória, estimativa de posição para robôs manipuladores, SLAM (do inglês, *Simultaneous Localization and Mapping*) e detecção de objetos (URREA; AGRAMONTE, 2021). Além de que

Figura 8 – Diagrama relacionando as classificações três níveis e Dasarathy.



Fonte: Retirado de Elmenreich (2002)

sua flexibilidade permitiu a integração da informação de diferentes tipos de sensores e técnicas, tornando possível responder as questões fundamentais da navegação de robôs: Onde estou? Para onde estou indo? E como eu chego no meu destino?

Em suma, o filtro de Kalman é um conjunto de equações matemáticas que serve para estimar o estado de um sistema dinâmico linear com ruídos de tal maneira que a média do erro quadrático diminui de forma eficiente computacionalmente por ser um algoritmo recursivo. Ou seja, o KF precisa de pouca memória já que precisam de memória apenas para salvar informação de estados passados, sendo adequado para problemas de tempo real e sistemas embarcados (KHODARAHMI; MAIHAMI, 2023).

Quando fala-se sobre o estado de um sistema, coloca-se um vetor x que consiste de n variáveis que descrevem importantes propriedades de um sistema. Um exemplo de estado é a localização de um robô, que consiste das coordenadas x e y e a orientação θ de um robô.

Como colocado anteriormente, robôs normalmente utilizam uma grande quantidade de sensores, cada um deles provendo a posição do robô, mas também cada um sendo sujeito a erros ou falhas no funcionamento. Então, a obtenção da localização ótima de um robô móvel deve levar em conta a informação gerada por todos sensores. Segundo Siegwart, Nourbakhsh e Scaramuzza (2011), o filtro de Kalman é uma técnica poderosa para atingir essa fusão de sensores por ser

eficiente ao representar a função de densidade probabilística da crença do robô e até das leituras individuais dos sensores, resultando num algoritmo de processamento de dados recursivo ótimo.

Entretanto, segundo Negenborn (2003), o fato de que as variáveis de um estado podem conter ruídos e não serem diretamente observáveis dificultam a estimação do estado. O KF possui acesso às medições do sistema para poder realizar a estimativa do estado, estas medições estão linearmente relacionadas ao estado e estão corrompidas por ruídos. Caso as fontes desses ruídos possuam uma distribuição gaussiana, então a estimativa do KF é estatisticamente ótima para qualquer medida razoável de otimização.

Também segundo Negenborn (2003), o KF processa todas medidas disponíveis de sensores para estimar o estado, tanto as medidas precisas quanto as imprecisas. Ele utiliza conhecimento do sistema e dinâmica dos sensores, descrição probabilística do próprio sistema e dos ruídos das medidas, e qualquer dado disponível sobre os valores iniciais do estado.

3.5.2 Premissas

A utilização do filtro de Kalman para prever e corrigir a crença do estado presume a necessidade de um modelo tanto do sistema quanto das medições. O KF assume uma descrição de sistema dinâmico linear do sistema que está estimando o estado. O sistema dinâmico pode ser corrompido por fontes de ruídos, os quais o KF assume que podem ser modelados por distribuições independentes, brancas, média zero e gaussianas (URREA; AGRAMONTE, 2021).

3.5.2.1 Sistema dinâmico linear

Falando sobre o modelo do sistema, ele descreve como o verdadeiro estado do sistema evolui ao longo do tempo, utilizado pelo filtro para realizar previsões sobre o estado. Basicamente, o KF assume que o estado do sistema evolui de acordo com a Equação (8), onde o verdadeiro estado x_k do sistema no tempo k depende do estado um passo antes x_{k-1} e algum ruído, a matriz A tem tamanho $n \times n$ e relaciona os estados passado e atual, enquanto o vetor $w_k - 1$ modela o ruído no sistema, adicionando os efeitos de influências não modeladas no estado (URREA; AGRAMONTE, 2021).

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \quad (8)$$

O modelo de medição descreve como medições se relacionam com os estados. O filtro de Kalman necessita do modelo das medições a fim de corrigir a previsão do estado quando uma

medição está disponível. Basicamente, o KF assume que as medições podem ser modeladas por um equação linear que relaciona o estado do sistema para uma medição, como a Equação (9), em que z_k depende linearmente do estado do sistema x_k , já a matriz H possui tamanho $m \times n$ e relaciona a medição com o estado, enquanto v_k é o ruído nas medições (KHODARAHMI; MAIHAMI, 2023).

$$z_k = Hx_k + v_k \quad (9)$$

Ambas Equações (8) e (9) mostram que o estado x_k no tempo k não depende de todos os outros estados e medições dado x_{k-1} e que a medição z_k não depende de qualquer estado ou medida, o que torna o sistema um processo Markoviano.

3.5.2.2 Características do ruído

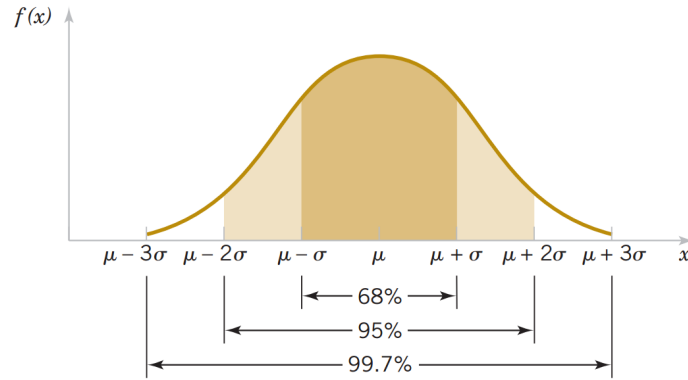
Uma característica necessária do ruído para o filtro de Kalman é a independência, que torna o cálculo envolvido na estimativa de estado mais fácil. De acordo com Negenborn (2003) em geral é justo assumir que os ruídos no sistema e medição são independentes.

Outra característica que simplifica a matemática envolvida no filtro de Kalman é o ruído branco, este tem poder em todas frequências do espectro e é completamente não correlacionado com ele mesmo em qualquer momento exceto o presente. Ou seja, os erros não se correlacionam pelo tempo, saber a quantidade de ruído neste momento não ajuda em prever qual será a quantidade de ruído em outro momento.

Uma terceira característica que é assumida é que o ruído possui média zero, o que implica que o erro no sistema e medição é aleatório. Um ruído aleatório significa que ele não é sistemático, ou seja, ele não possui um *bias* constante, algumas horas ele é positivo, outras negativo, mas sempre média zero.

A última característica importante que é assumida pelo filtro de Kalman é que o ruído é gaussiano, que é uma característica que lida com amplitude do ruído, colocando que a quantidade de ruído envolvida pode ser modelada por uma curva conforme a Figura 9, em que o centro do gráfico representa a média μ dos valores, já a dispersão (ou largura) do gráfico é representada pelo desvio padrão σ (ou pela variância, que é o desvio padrão elevado ao quadrado). Esta premissa é justificada ao assumir que os ruídos do sistema e medição são causados por diversas fontes pequenas de ruídos que, independente de suas distribuições, a soma delas será distribuída conforme uma gaussiana.

Figura 9 – Exemplo de distribuição gaussiana.



Fonte: Retirado de Zibetti (2022)

Com as premissas da média zero e a distribuição gaussiana, os ruídos podem ser descritos de acordo com $N(\mu, \Sigma)$, que denota uma função gaussiana de média μ e covariância Σ .

3.5.2.3 Processo a ser estimado

O filtro de Kalman aborda o problema geral de tentar estimar o estado $x \in \mathbb{R}^n$ de um processo controlado em tempo discreto que é governado pela equação diferencial estocástica linear descrita pela Equação (10) com medição $z \in \mathbb{R}^m$, que é representada pela Equação (11). No caso, as variáveis aleatórias w_k e v_k representam os ruídos do processo e das medições, respectivamente.

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \quad (10)$$

$$z_k = Hx_k + v_k \quad (11)$$

Assume-se que as variáveis w_k e v_k são independentes uma das outras, brancas, e com uma distribuição probabilística normal, segundo as probabilidades definidas na Equação (12).

$$\begin{aligned} p(w) &\sim N(0, Q) \\ p(v) &\sim N(0, R) \end{aligned} \quad (12)$$

Segundo Welch, Bishop et al. (1995), a matriz de covariância do ruído do processo Q e a matriz de covariância do ruído das medições R podem variar a cada passo de tempo ou a cada medição, embora nesse caso seja considerado constante.

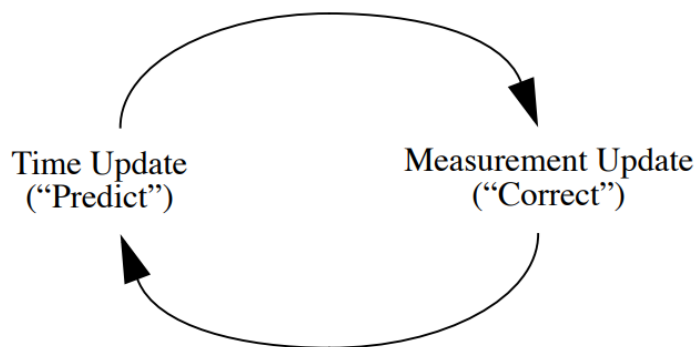
A matriz A da Equação (10) possui tamanho $n \times n$ e relaciona o estado no passo de tempo anterior $k - 1$ com o estado no passo de tempo atual k na ausência de uma função ou ruído de processo. Já a matriz B possui tamanho $n \times l$ e relaciona a entrada de controle $u \in \mathbb{R}^l$ ao estado x . A matriz H possui tamanho $m \times n$ na Equação (11) e relaciona o estado com a medição z_k .

3.5.3 Equações

De acordo com Khodarahmi e Maihami (2023), o filtro de Kalman estima um processo utilizando uma forma de controle por meio de feedback, nele o filtro estima o estado do processo em um dado instante e então obtém feedbacks na forma de medições, no caso ruidosas. Como tal, as equações do filtro de Kalman podem ser divididas em dois grupos: as equações de atualização de tempo e as equações de atualização de medições. O primeiro grupo é responsável por projetar a frente no tempo as estimativas do estado atual e a covariância do erro para obter a estimativa a priori do próximo período de tempo. Já o segundo grupo é responsável pelo feedback, isto é, por incorporar uma nova medição na estimativa a priori a fim de obter uma melhor estimativa a posteriori.

As equações de atualização no tempo podem ser chamadas como equações de predição, enquanto as equações de atualização de medição podem ser chamadas de equações de correção. Basicamente o algoritmo de estimativa final se assemelha com um algoritmo predição-correção para solução de problemas numéricos. A Figura 10 mostra o ciclo do filtro de Kalman, em que a predição projeta a estimativa do estado atual a frente no tempo, enquanto a correção ajusta a estimativa projetada por uma medição real naquele instante.

Figura 10 – Ciclo do filtro de Kalman discreto.



Fonte: Retirado de Welch, Bishop et al. (1995)

3.5.3.1 Predição

A cada instante de tempo o sistema pode estar em um estado diferente. Portanto, o KF calcula uma nova crença anterior a cada passo de tempo. As equações de predição (também chamada de atualização por tempo ou propagação) predizem o novo estado do sistema projetando à frente a crença mais recente, ou seja, calculando a crença $bel(x_k)$ a partir da crença do estado anterior $bel(x_{k-1})$.

No caso, de acordo com Thrun (2002), $bel(x_k) = N(\hat{x}_k^-, P_k^-)$, em que a média \hat{x}_k^- e a covariância P_k^- são definidos segundo a Equação (13).

$$\begin{aligned}\hat{x}_k^- &= A\hat{x}_{k-1} + Bu_k \\ P_k^- &= AP_{k-1}A^T + Q_k\end{aligned}\tag{13}$$

O KF calcula a estimativa de estado \hat{x}_k^- baseado tanto na última estimativa de estado \hat{x}_{k-1} quanto no modelo disponível do sistema. A melhor hipótese que o KF pode fazer sobre o estado do sistema depois dele progredir um passo a frente no tempo é a melhor hipótese propagada pelo modelo que o KF possui do sistema.

Além disso, o filtro de Kalman também reconhece que a evolução do sistema está sujeita a ruídos e, assim, possui uma incerteza aumentada P_k^- na estimativa do estado. O primeiro termo da covariância do erro $AP_{k-1}A^T$ propaga a incerteza da última estimativa à frente para a estimativa atual do estado. Já o segundo termo Q_k é o ruído do sistema que corrompe o estado do sistema a cada passo de tempo.

3.5.3.2 Correção

As equações de correção (ou atualização da medição) lidam com as medições dos sensores. Elas são utilizadas apenas quando há a atualização da medição dos sensores. As medições providenciam informação direta sobre o estado atual do sistema. As equações desta etapa corrigem a previsão da crença mais recente ao incorporar a informação recebida das medições. Segundo Thrun (2002), as equações dessa fase calculam a crença posterior $bel(x_k) = N(\hat{x}_k, P_k)$, em que \hat{x}_k e P_k são definidos segundo a Equação (14).

$$\begin{aligned}
\hat{x}_k &= \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \\
P_k &= (I - K_k H)P_k^- \\
K_k &= P_k^- H^T (H P_k^- H^T + R_k)^{-1}
\end{aligned} \tag{14}$$

A nova crença posterior \hat{x}_k é utilizada no próximo passo de tempo para calcular a predição de uma nova crença. A natureza recursiva do filtro de Kalman permite implementações práticas, já que nem todos os dados são necessários para estimar os estados.

O filtro de Kalman calcula a estimativa do estado posterior combinando a predição da estimativa de estado com o ganho de Kalman K_k vezes a diferença entre a medição z_k e a previsão de medição $H\hat{x}_k^-$, chamada de inovação.

O termo $H\hat{x}_k^-$ na Equação (14) é chamado de previsão de medição. Dadas a estimativa do estado anterior x_k^- e a matriz de medições H do modelo de medição na Equação (11), o filtro de Kalman prediz qual medição irá receber. Assim, denota-se a previsão de medição segundo a Equação (15).

$$\hat{z}_k = H\hat{x}_k^- + \hat{v}_k \tag{15}$$

No caso, o ruído de medição \hat{v}_k é zero e a previsão de medição é uma variável aleatória que segue uma distribuição gaussiana, podendo notar isso ao analisar que ela depende linearmente da estimativa anterior do estado \hat{x}_k^- e do ruído de medição, sendo que ambos são variáveis aleatórias gaussianas. Então, facilmente deriva-se que a predição de medida \hat{z}_k segue a distribuição descrita na Equação (16).

$$\hat{z}_k = N_z(H\hat{x}_k^-, H P_k^- H^T + R_k) \tag{16}$$

A diferença entre a medição z_k e a medição prevista x_k^- é chamada de inovação da medição ou \tilde{z}_k residual. A inovação diz quanto uma medida prevista difere de uma medição real, sendo definida segundo a Equação (17). Caso a inovação seja igual a zero, então a medida prevista reflete exatamente a medição real, o que implica que o estado estimado com o qual a predição da medição foi realizada estava muito perto do verdadeiro estado que a medição foi feita. Entretanto, se existir uma diferença entre as medições prevista e observada, então a estimativa do estado anterior precisa ser atualizada com um certo valor.

$$\tilde{z}_k = z_k - \hat{z}_k \tag{17}$$

O fator K_k na Equação (14) é chamado de ganho de Kalman (KG, do inglês *Kalman Gain*), que é o fator que determina até que ponto a inovação deve ser levada em conta na estimativa de estado posterior. Isso é determinado ao olhar a incerteza relativa entre a estimativa de estado anterior e a inovação da medição, como descrito na Equação (14).

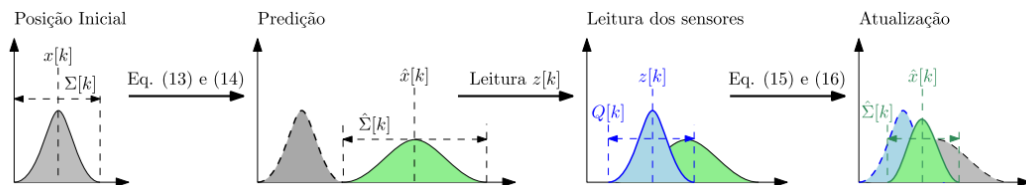
A fim de comparar a incerteza da estimativa do estado anterior no espaço de estados com a incerteza da inovação no espaço de medição, o KF converte a incerteza no espaço de medição para o espaço de estados por meio da matriz H^T .

Em resumo, o ciclo do filtro de Kalman pode ser entendido conforme o Algoritmo 1, em que há o detalhamento das Equações de cada um dos passos do KF, a predição e correção. Já a Figura 11 representa um exemplo em uma dimensão de como o filtro de Kalman realiza a predição e correção em termos de média e covariância.

Algoritmo 1 – Filtro de Kalman linear

- 1 **Entrada:** Estado anterior x_{k-1} ; Covariância anterior P_{k-1} ; Entrada de controle u_k ;
Entrada de medição z_k
- 2 **Saída:** Estado atual \hat{x}_k ; Covariância atual P_k
- 3 **início**
- 4 $\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k$
- 5 $P_k^- = AP_{k-1}A^T + Q_k$
- 6 **se medição disponível então**
- 7 $K_k = P_k^- H^T (H P_k^- H^T + R_k)^{-1}$
- 8 $\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$
- 9 $P_k = (I - K_k H) P_k^-$
- 10 **fim**
- 11 **fim**
- 12 **retorna** \hat{x}_k, P_k

Figura 11 – Exemplo de predição e atualização das covariâncias do filtro de Kalman.



Fonte: Retirado de Costa e Tonidandel (2023)

3.6 FILTRO DE KALMAN ESTENDIDO

Para sua descrição, o filtro de Kalman possui algumas premissas, tal como a de que as observações são funções lineares dos estados e que o próximo estado é uma função linear do estado anterior, o que é crucial para a correção do filtro (THRUN, 2002). Não só isso, para o desenvolvimento do KF, observa-se que qualquer transformação linear de uma variável aleatória gaussiana resulta em outra variável aleatória gaussiana.

Embora o filtro de Kalman tenha provado sua eficiência e qualidade ao longo dos anos com sua vasta utilização em diversas áreas, infelizmente sistemas mais complicados podem ser não-lineares (KHODARAHMI; MAIHAMI, 2023). Por exemplo, um robô que se move com velocidade de translação e rotação constantes tipicamente realizam uma trajetória circular, que não pode ser descrita por uma transição de estado linear (THRUN, 2002).

Então, a fim de resolver o problema da linearidade para o filtro de Kalman, foi desenvolvida uma versão dele que leva em conta a não-linearidade dos sistemas, medições e ruídos, que é o filtro de Kalman estendido (EKF, do inglês).

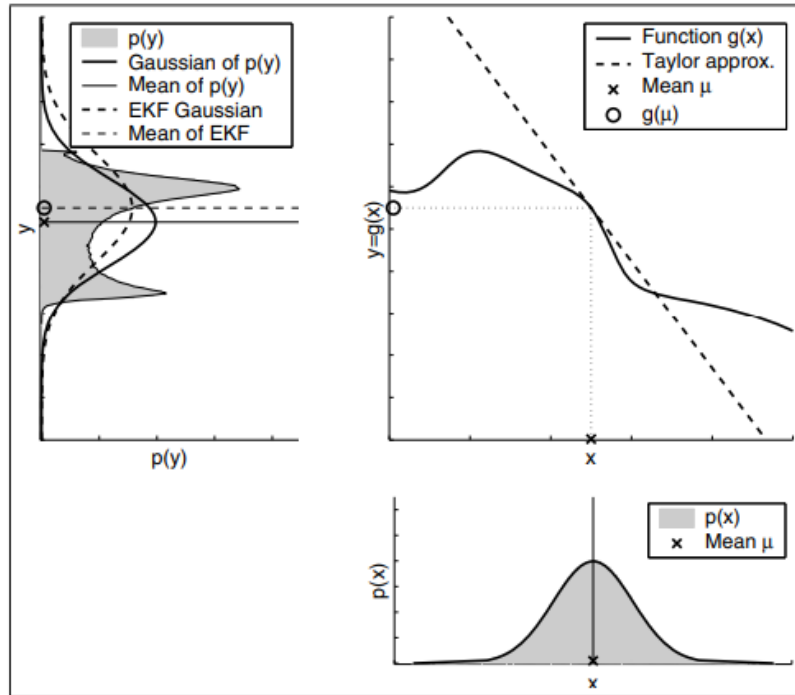
O EKF segue a mesma ideia do filtro de Kalman linear, isto é, com a separação nas etapas de predição, que projeta o sistema a frente para obter uma estimativa no próximo período de tempo, e correção, que incorpora uma nova medição na estimativa da predição a fim de obter uma melhor estimativa. A diferença entre ambos recai na particularidade de que o EKF utiliza séries de Taylor para linearizar o sistema não-linear.

De acordo com Thrun (2002), a ideia da linearização é aproximar uma função não-linear g por uma função linear que é tangente a g na média da gaussiana. Assim, projetar a gaussiana por meio dessa aproximação linear resulta em uma densidade gaussiana, como é demonstrado na Figura 12. O autor coloca que a principal vantagem da linearização recai na sua eficiência.

As predições lineares no filtro de Kalman são substituídas pelas generalizações não-lineares no filtro de Kalman estendido. Além de que o EKF utiliza de Jacobianas G_k e C_k ao invés das matrizes lineares do sistema A_k , B_k e H_k no KF.

Assim, a etapa de predição no filtro de Kalman estendido é descrita conforme a Equação 18. É possível notar que a estimativa do estado é dada por uma função não-linear que depende do estado anterior x_{k-1} e da entrada de controle naquele instante u_k . Além disso, como dito anteriormente, G_k é a matriz Jacobiana com relação ao estado. No caso, uma matriz Jacobiana é formada pelas derivadas parciais de primeira ordem de uma função.

Figura 12 – Exemplo de predição e atualização das covariâncias do filtro de Kalman.



Fonte: Retirado de Thrun (2002)

$$\begin{aligned}
 \hat{x}_k^- &= f(x_{k-1}, u_k) \\
 P_k^- &= G_k P_{k-1} G_k^T + Q_k \\
 G_k &= \frac{\partial f(x_{k-1}, u_k)}{\partial x_{k-1}}
 \end{aligned} \tag{18}$$

A etapa de correção no filtro de Kalman estendido é, então, descrita conforme a Equação 19. No caso, a estimativa do sensor também é uma função não-linear. Assim, C_k representa a matriz Jacobiana do sensor com relação ao estado, já $h(x_k^-)$ representa a linearização do sensor.

$$\begin{aligned}
 z_k &= h(x_k) \\
 \hat{x}_k &= \hat{x}_k^- + K_k(z_k - h(x_k^-)) \\
 P_k &= (I - K_k C_k) P_k^- \\
 K_k &= P_k^- C_k^T (C_k P_k^- C_k^T + R_k)^{-1} \\
 C_k &= \frac{\partial h(x_k)}{\partial x_k}
 \end{aligned} \tag{19}$$

As equações que representam o filtro de Kalman estendido podem ser entendidas em conjunto segundo o Algoritmo 2.

Algoritmo 2 – Filtro de Kalman estendido

```

1 Entrada: Estado anterior  $x_{k-1}$ ; Covariância anterior  $P_{k-1}$ ; Entrada de controle  $u_k$ ;
   Entrada de medição  $z_k$ 
2 Saída: Estado atual  $\hat{x}_k$ ; Covariância atual  $P_k$ 
3 início
4    $\hat{x}_k^- = f(x_{k-1}, u_k)$ 
5    $P_k^- = G_k P_{k-1} G_k^T + Q_k$ 
6   se medição disponível então
7      $K_k = P_k^- C_k^T (C_k P_k^- C_k^T + R_k)^{-1}$ 
8      $\hat{x}_k = \hat{x}_k^- + K_k (z_k - h(\hat{x}_k^-))$ 
9      $P_k = (I - K_k C_k) P_k^-$ 
10  fim
11 fim
12 retorna  $\hat{x}_k, P_k$ 

```

4 TRABALHOS RELACIONADOS

Diversos estudos foram realizados na área de localização de robôs móveis e estimativa de posição a fim de obter o conhecimento necessário para o desenvolvimento do projeto em questão. Entretanto, uma boa gama dos trabalhos encontrados utilizam robôs com uma dinâmica diferente de um robô omnidirecional da *Small Size League* da RoboCup, como o modelo de duas rodas ou o modelo *car-like*, ou também os estudos levam em conta diferentes sensores dos que são utilizados nesse projeto.

A seguir serão descritos os trabalhos relacionados nas áreas de fusão de sensores, estimativa de posição de robôs móveis e filtro de Kalman. Os termos utilizados na pesquisa dos trabalhos foram "*position estimation*", "*position estimation kalman filter*", "*sensor fusion for position estimation*" e "*position estimation sensor fusion*".

Em Eman e Ramdane (2020), um filtro de Kalman estendido é utilizado para resolver o problema de localização de um robô móvel num ambiente *indoor*. Os autores estudam a eficiência do filtro em três casos distintos na questão do ruído presente no sistema, que são: sem ruído, ruído Gaussiano, ruído não-Gaussiano.

Sobre o sistema, o modelo utilizado é de um robô de duas rodas, no qual as equações da cinemática do modelo estão descritas na Equação (20), que os autores definem (x, y) sendo a posição e θ a orientação do robô, μ a velocidade linear e ω a velocidade angular.

$$\begin{cases} \dot{x} = \mu \cos \theta \\ \dot{y} = \mu \sin \theta \\ \dot{\theta} = \omega \end{cases} \quad (20)$$

A partir disso, utilizando aproximação de Taylor, a posição e a orientação do robô em qualquer momento futuro $k + 1$ são descritas segundo a Equação (21), em que T_s é o período de amostragem.

$$\begin{cases} x(k+1) = x(k) + \mu(k)T_s \cos(\theta(k)) \\ y(k+1) = y(k) + \mu(k)T_s \sin(\theta(k)) \\ \theta(k+1) = \theta(k) + \omega(k)T_s \end{cases} \quad (21)$$

Já sobre o EKF, o autor determina que o vetor de estados a serem estimados X e o vetor de controle U são o que está definido na Equação (22). Assim, as matrizes Jacobianas do sistema podem ser definidas conforme a Equação (23).

$$\begin{cases} X = [x & y & \theta]^T \\ U = [\mu & \omega]^T \end{cases} \quad (22)$$

$$\begin{cases} A_k = \frac{\delta f(X_k, U_k)}{\delta X_k} = \begin{bmatrix} 1 & 0 & -\mu(k)T_s \sin(\theta(k)) \\ 0 & 1 & \mu(k)T_s \cos(\theta(k)) \\ 0 & 0 & 1 \end{bmatrix} \\ H_k = \frac{\delta f(X_k, U_k)}{\delta X_k} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{cases} \quad (23)$$

Para realização dos testes, os códigos foram desenvolvidos utilizando MATLAB num cenário 2D, com T_s sendo $0.1s$, μ sendo $5m/s$ e ω sendo $1rad/s$, além de que o vetor de estados inicial e a estimativa inicial são matrizes nulas. Como dito anteriormente, foram considerados 3 cenários de ruídos a fim de avaliar a atuação do EKF.

Os resultados mostraram que o EKF é uma boa ferramenta de estimativa para sinais ruidosos em todos os casos, embora os resultados tenham confirmado que o filtro funciona melhor no caso de ruídos Gaussianos comparado com ruídos não-Gaussianos. Essa conclusão é de suma importância para a realização desse trabalho, visto que os sinais dos sensores e do sistema deste projeto são assumidos como Gaussianos, além da importância do uso de diferentes tipos de ruídos para estudo.

Já em Korotaj, Novoselnik e Baotić (2021) é descrita a utilização de fusão de sensores para um sistema mecatrônico omnidirecional. As expressões são dadas para um filtro de Kalman linear discreto que junta dados de um magnetômetro e um giroscópio, e um filtro de Kalman estendido discreto que estima a posição e orientação da plataforma com dados de um acelerômetro também.

Sobre o sistema, a plataforma é composta de 2 níveis e possui 4 rodas acionadas para movimentação, cada uma acionada por um motor DC (do inglês, *Direct Current*). Os sensores presentes na plataforma são um acelerômetro de 3 eixos, um giroscópio de 3 eixos e um magnetômetro.

A estimativa de estado é separada em duas partes, a primeira é a estimativa da orientação do sistema a partir da fusão do giroscópio e do magnetômetro utilizando KF. Essa orientação estimada é utilizada como uma observação melhorada da orientação para estimativa tanto da

posição quanto da orientação na segunda parte por meio de um EKF discreto, já que o modelo cinemático da plataforma é não-linear, além de utilizar as velocidades de cada uma das rodas e as leituras do acelerômetro também.

Os resultados simulados da resposta de estimativa de estado mostraram satisfatórias acurácia e velocidade do procedimento de estimativa selecionado. Apesar das limitações do layout utilizado das rodas e imperfeições das ranhuras que conectam a roda o seu eixo do motor, os experimentos em tempo real confirmaram a eficiência da fusão de sensores.

O trabalho demonstrou uma maneira diferente da utilização do filtro de Kalman, já que ele é usado em um primeiro passo para estimar a orientação com os dados do magnetômetro para corrigir o erro da integração do giroscópio. Então, a partir dessa estimativa, utilizar o EKF para prever tanto a posição quanto a orientação do veículo.

Em Ismail, Purwanto e Arifin (2022) é feito um estudo sobre a localização baseada em fusão de sensores de um robô de futebol da categoria *Middle Size League* (MSL) da *RoboCup*. A MSL é como se fosse uma categoria acima da SSL, em que os robôs possuem carcaças maiores (de aproximadamente 1,30m X 30cm), além de que não há um sistema global de visão, ou seja, a visão deve estar embarcada em cada robô, assim cada robô deve estimar sua posição e orientação de forma automática e independente.

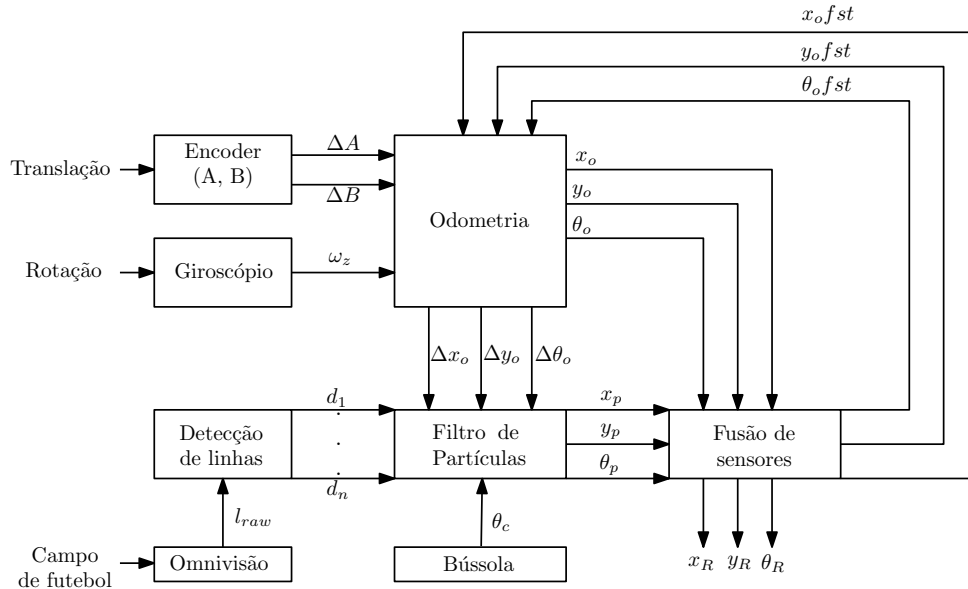
A maior parte dos robôs dessa categoria utilizam câmeras omnivisão, permitindo que os robôs tenham visão 360°. A partir dessas imagens é utilizado filtro de partículas (FP) para estimar a posição e orientação do robô em campo, embora o filtro necessite de um processamento computacional pesado.

O sistema de fusão de sensores aplicada no trabalho pode ser visualizado na Figura 13, em que há a representação em um diagrama de blocos da fusão de sensores aplicada nos robôs da categoria MSL em Ismail, Purwanto e Arifin (2022).

A partir da Figura 13 é possível notar que o encoder e o giroscópio são utilizados para reconhecer a disposição do robô, sendo que são 2 encoders para detectar a translação e o giroscópio para detectar a rotação. Já a omnivisão detecta linhas brancas dentro do campo e, a partir das imagens cruas, há um processamento para gerar dados dessas linhas (i.e., cálculo da distância do robô até essas linhas por meio de regressão) antes de entrar no FP, que é ajudado por um bússola a fim de encontrar a posição .

A fusão de sensores atua combinando dados de posição e orientação do sistema de odometria e do FP. Os autores colocam que existem diferentes maneiras de realizar essa fusão (como comparação, junção, votação inteligente), mas no caso do trabalho são utilizados os

Figura 13 – Visão geral da fusão de sensores da odometria com o sistema de visão da categoria MSL.



Fonte: Retirado de Ismail, Purwanto e Arifin (2022)

métodos de fusão competitiva, onde os dados são tomados de forma independente, e complementar, em que há a utilização dos dados de ambos os sistemas de localização.

O teste realizado para validação do sistema de localização é o seguinte: o robô é movido manualmente para 20 coordenadas num campo de 9m x 6m, sendo que o movimento do robô é rastreado a fim de comparar com os dados de uma câmera global alocada em cima do campo. Depois de percorrer essas 20 coordenadas, são retirados a média, o desvio padrão do erro e o máximo erro dos sistemas de odometria, filtro de partículas e fusão de sensores.

A partir dos resultados, nota-se que no sistema de localização por odometria há um erro que cresce ao longo do tempo, enquanto para o FP e a fusão de sensores o erro não aumenta, enquanto a fusão dos dados acaba suavizando a saída quando comparado ao FP.

Além disso, um outro teste comparando ambos tipos de fusão de sensores foi realizado. O objetivo era movimentar o robô formando um zig-zag de maneira retangular. Os dados mostraram que o modo odometria é mais dominante, sendo que o sistema possui um tempo de resposta de aproximadamente 1.6ms para cada atualização de dados de localização.

Os autores concluem colocando que a fusão de sensores pode produzir dados ótimos ao combinar a odometria e a omnivisão com o FP. Os erros resultantes dos testes foram em x igual a $10.5 \pm 7.8\text{cm}$, em y igual a $7.6 \pm 6.8\text{cm}$ e em θ igual a $1.9 \pm 1.2^\circ$. Esses resultados são

melhores do que a utilização dos dados da odometria sozinhos, além de serem mais suaves do que o FP sozinho.

Este trabalho, por estar inserido na lógica do futebol de robôs, tem grande importância por utilizar um sistema de visão, mesmo que seja embarcado, diferente do sistema de visão da SSL. Além disso, o trabalho também compara duas maneiras de realizar a fusão de sensores com sensores que são semelhantes aos que serão utilizados nesse projeto.

Em Coito et al. (2014) é realizado um estudo de fusão de sensores utilizando um sistema de visão e IMU por meio de um filtro de Kalman de múltiplas taxas. No caso, a IMU possui um processamento bem mais rápido do que o sistema de visão, além de que a taxa de atualização da primeira é constante enquanto o da segunda costuma ter oscilações.

No caso, o robô utilizado é do tipo direcional, sendo 4 rodas que são unidas em pares por 2 eixos. A IMU é alocada no centro do robô, sendo que o giroscópio é alinhado com o eixo vertical.

Para lidar com as diferentes taxas de amostragem dos sistemas de visão e medição inercial é utilizado um KF discreto de múltiplas taxas, que segue a Equação 24, em que Z é o vetor de estados $\begin{bmatrix} v_x & x & v_y & y & \theta \end{bmatrix}$, Y_k é o vetor de valores medidos, n_k e w_k são ruídos. Já A_k e G_k são as matrizes do sistema. No caso, as matrizes A_k e G_k são descritas conforme a Equação 25.

$$Z_{k+1} = A_k Z_k + G_k n_k \quad (24)$$

$$Y_k = C_k Z_k + w_k$$

$$A_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ \delta t_k & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \delta t_k & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (25)$$

$$G_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Os testes simulados foram realizados com o sistema inercial atualizando a uma taxa de 100Hz, enquanto o sistema de visão é simulado e fornece 2 medidas de posição por segundo. Os

resultados mostraram que a trajetória do robô não apresenta um erro crescente ao longo do tempo, permanecendo abaixo de 3cm na maior parte do teste, sendo que o desvio aumenta quando o robô performa uma curva rápida com um raio pequeno, o que poderia ser mitigado com um sistema de visão mais rápido, segundo os autores.

Já nos resultados experimentais, o robô realizou uma trajetória com 4 seções em que se movia com velocidade linear de 10cm/s e uma velocidade angular de 0.2rad/s. As estimativas do filtro de Kalman eram calculadas a cada 25ms. Os resultados demonstraram uma boa estimativa da posição real sendo que, de acordo com os autores, a precisão e a taxa de amostragem se adequam para boa parte de aplicações de controle.

O trabalho traz informações importantes para fusão de sensores com diferentes taxas de atualização, principalmente por conta do uso de um sistema de visão, que normalmente possui um delay consideravelmente alto em relação aos outros sensores, como IMU e encoder. Os autores colocam que a cada momento de fusão o conjunto de sensores disponíveis varia, então a Equação 24 leva em conta a taxa de atualização.

Continuando com aplicações na competição do futebol de robôs, em Luis Aguiar et al. (2017) é feito um estudo comparando o filtro de Kalman estendido, o filtro de Kalman *unscented* (do inglês, UKF) e o filtro de Kalman linear para rastreamento de posição para a categoria IEEE *Very Small Size*, que é como uma categoria abaixo em relação a SSL, já que os robôs devem caber num cubo de 7.5cm de lado.

Os robôs da categoria, em suma, são diferenciais, que possuem 2 motores acoplados em 2 rodas normalmente. Os autores colocam que para o rastreamento da posição podem ser utilizados 2 modelos estocásticos para um robô desse tipo, cada um com uma representação de estados diferente.

O primeiro modelo é chamado de modelo *unicycle*, em que o estado é dado por $x_k = \begin{bmatrix} x_k & y_k & \theta_k & v_k & \omega_k \end{bmatrix}^T$. No caso, os três primeiros são as coordenadas e orientação de um ponto fixo entre as rodas do robô em relação a um *frame* de referência. No problema de rastreamento do robô oponente não se tem acesso às entradas de controle, mas a formulação do estado e a cinemática do robô são suficientes para extrair as estimativas da velocidade. Isso pode ser observado na Equação 26. É possível notar que as acelerações linear e angular são modeladas como um vetor aleatório Gaussiano com covariância Q .

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} v \times \cos \theta \\ v \times \sin \theta \\ \omega \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} w_a \\ w_\alpha \end{bmatrix} \quad (26)$$

Já o modelo de observação é dado pelo sistema de visão, também posicionado acima do campo, provendo observações de posição e orientação. Entretanto, esses dados são ruidosos, assumidos aditivos e Gaussiano com covariância R , como é possível ver na Equação 27.

$$z_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} + \begin{bmatrix} v_{x,k} \\ v_{y,k} \\ v_{\theta,k} \end{bmatrix} \quad (27)$$

O segundo modelo é chamado de dupla integração, que é demonstrado na Equação 28. No caso, o estado do robô é $x_k = [x_k \ y_k \ v_{x,k} \ v_{y,k}]^T$ e o modelo de observação é obtido conforme a Equação 29. Na categoria VSS, pode não ser possível extrair de maneira fácil a orientação do oponente a partir do sistema de visão, já que o sistema de cores não é unificado, diferente da categoria SSL.

$$\begin{bmatrix} x_k \\ y_k \\ v_{x,k} \\ v_{y,k} \end{bmatrix} = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ v_{x,k-1} \\ v_{y,k-1} \end{bmatrix} + \begin{bmatrix} \frac{T^2}{2} & 0 \\ 0 & \frac{T^2}{2} \\ T & 0 \\ 0 & T \end{bmatrix} \begin{bmatrix} w_{a,x} \\ w_{a,y} \end{bmatrix} \quad (28)$$

$$z_k = \begin{bmatrix} x_k \\ y_k \end{bmatrix} + \begin{bmatrix} v_{x,k} \\ v_{y,k} \end{bmatrix} \quad (29)$$

Para validar as técnicas de rastreamento foram realizados testes com os robôs reais e na simulação. Os testes reais foram realizados em um campo da categoria VSS e foram extraídos as posições cartesianas e a orientação providas pelo sistema de visão.

Os dados reais mostraram que o EKF e o UKF obtiveram resultados iguais, indicando que não há necessidade da utilização de uma técnica mais complexa como a segunda, já que o modelo do VSS não é um sistema que representa grandes não-linearidades, então o EKF seria preferível por conta da simplicidade e custo computacional. Além disso, os autores colocam que com o modelo *unicycle* se consegue um estado mais completo por incluir a orientação e velocidade angular do robô.

Já para a simulação, com o modelo de dupla integração utilizando o filtro de Kalman linear, apesar da baixa precisão para representar a dinâmica de um robô diferencial, os autores colocam que a performance de estimativa foi quase a mesma em relação ao EKF e UKF com o modelo *unicycle*, visto que o EKF apresentou um erro quadrático médio apenas 1.5% menor que o KF para estimativa de velocidade. Isso mostra também que o modelo consegue estimar com sucesso a velocidade no caso em que não se possui a orientação do robô.

Este trabalho é mais um inserido na realidade do futebol de robôs e numa categoria tão dinâmica quanto a SSL. Além disso, o autor mostra duas maneiras possíveis para representar o modelo dinâmico de um sistema, mesmo que os robôs dessa categoria sejam diferenciais.

Em Pacheco, Silva e Farrell (2023) é feito um estudo para atingir um posicionamento acurado de robôs inteligentes para agricultura por meio da integração de um GPS e sensores de odometria de baixo custo, justamente para que seja acessível aos pequenos e médios agricultores, utilizando o EKF para conseguir essa integração.

Para realizar a etapa de predição do EKF os autores definem os estados como sendo os erros na solução de navegação utilizando odometria. No caso, os estados são o erro na orientação do robô pela odometria, os erros nas medições da posição norte e leste da odometria, o desvio associado ao raio das rodas direita e esquerda do robô, o erro no tempo de recepção do sinal do GPS e o desvio do GPS. O processo do modelo é estabelecido pelas equações que governam a evolução dos erros de estado da odometria ao longo do tempo.

Já a etapa de atualização é dividida a partir da topologia utilizada. Como nesse trabalho não haverá a utilização do sensor GPS como atualização do EKF, então dessa etapa basta falar que na topologia fracamente acoplada a inovação da medição é determinada como a diferença entre a posição cartesiana fixa e centrada na Terra medida pelo GPS e a estimativa correspondente da odometria. Já na topologia fortemente acoplada a inovação da medição é calculada como a diferença entre as pseudo-distâncias medidas pela antena receptora do GPS para todos satélites em vista e a solução de navegação odométrica.

O trabalho demonstrou bons resultados, com uma melhoria significativa da precisão e acurácia do robô no que diz respeito às estimativas de posição e orientação, com melhorias que podem chegar à 97% comparadas com os dados tanto da navegação utilizando odometria somente quanto os dados utilizando só a navegação por GPS.

Os autores concluem que, apesar dos resultados positivos alcançados no trabalho, o sistema de navegação odométrica auxiliada por GPS não alcançou as especificações de acurácia de posicionamento exigidas pela norma SAE J2945 (SAE INTERNATIONAL, 2016).

Apesar dos resultados não serem satisfatórios para os autores por não cumprirem o determinado pela norma SAE J2945 (SAE INTERNATIONAL, 2016), o modo que a etapa de predição é descrita no trabalho é uma ideia diferente que pode ser utilizada neste trabalho.

Já em Dyer et al. (2020) é realizado o teste de uma estratégia diferente para utilização de filtros em um robô omnidirecional com 4 rodas. No caso, é utilizada uma estratégia multi-filtro em que um KF estima a velocidade da roda a partir da dinâmica do motor e tem como medidas a leitura de velocidade de um encoder e o valor de corrente no motor, além de um EKF e um filtro suave de estrutura variável (FSEV) para estimativa do estado do robô.

Os testes conduzidos buscaram realizar diferentes combinações e encontrar a melhor, isto é, utilizando ou não o KF e, em seguida, utilizando ou não um dos outros 2 filtros para estimativa do robô. As simulações foram conduzidas utilizando MATLAB, apesar de que as constantes físicas do motor foram conseguidas experimentalmente por meio de um osciloscópio, um multímetro e um gerador de ondas.

Os resultados mostraram que apenas a utilização do KF nas rodas melhorou a precisão da estimativa em relação a utilização de nenhum filtro (erro médio quadrático de 10.02cm para 9.28cm no eixo X, por exemplo). Já a utilização de um filtro para estimativa do robô, mesmo sem um filtro para as rodas, acabou dobrando a precisão da estimativa, sendo que o FSEV se mostrou ligeiramente mais acurado (4.63cm de erro médio quadrático para ambos no eixo X). Já a utilização de ambos em conjunto aumenta a precisão duplamente, gerando um erro médio quadrático menor que 2cm.

O trabalho demonstra a qualidade da utilização do filtro de Kalman para predição do estado do robô, além de mostrar uma maneira interessante para estimar a velocidade das rodas do veículo a partir do modelo do motor, mostrando que a junção de ambos traz resultados melhores estimando o estado atual.

5 METODOLOGIA

Neste capítulo serão apresentados o domínio de teste do projeto proposto, os testes a serem realizados, como os sensores serão combinados, a calibração dos sensores e quais métricas serão analisadas para avaliação dos testes propostos.

5.1 DOMÍNIO DE TESTES

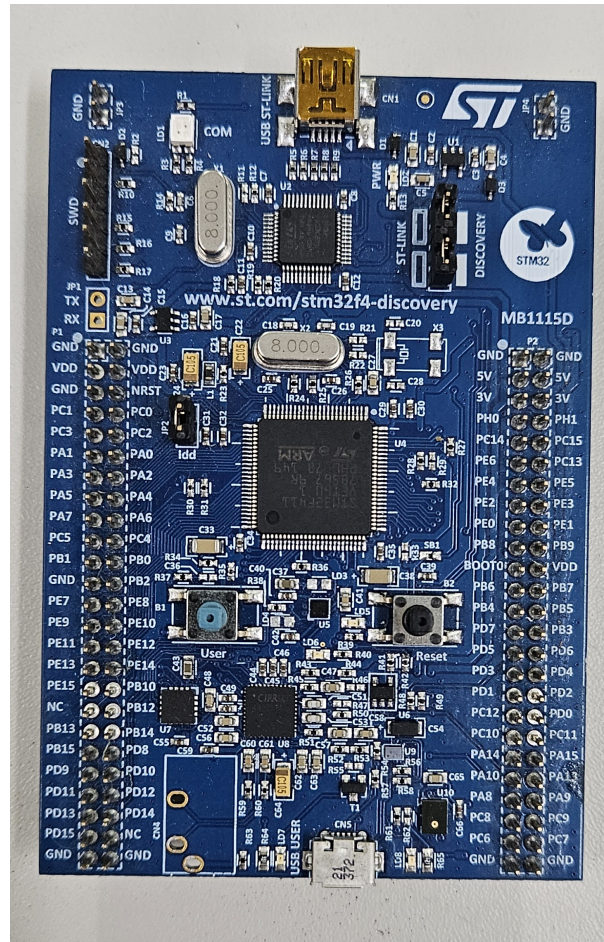
Neste trabalho será utilizado o domínio de teste da categoria SSL de futebol de robôs da RoboCup (ROBOCUP, 2020), descrito na Seção 2.1. Nele, as combinações de sensores serão testadas e comparadas. Para a realização dos testes propostos, será utilizado o campo do laboratório da equipe RoboFEI, que possui 4,3 X 3,6m, o que possibilita uma boa estimativa da movimentação do robô em um campo oficial da competição.

O que torna a SSL um interessante campo de teste para as combinações dos sensores são algumas características dos robôs e das partidas, como:

- a) **Velocidade alta dos robôs:** A dinamicidade das partidas se dá muito por conta da alta velocidade atingida pelos robôs, sendo necessário um bom sistema de controle de posição, o que requer um estimador de posição utilizando sensores além das câmeras da liga;
- b) **Altas penalidades para colisões:** as regras da competição punem severamente os times que causam muitas colisões, podendo levar a diversos cartões amarelos (um robô deve ficar fora do campo por 2 minutos), e até cartões vermelhos (o robô é expulso pelo resto da partida);
- c) **Alta precisão necessária para realização de jogadas:** por conta do pequeno tamanho tanto dos robôs e da bolinha, para realizar jogadas rápidas que evitem a chegada de robôs adversários, faz-se necessário uma alta precisão de posicionamento dos robôs.

Sobre o robô utilizado na equipe RoboFEI, os sensores giroscópio e acelerômetro estão presentes na placa de desenvolvimento STM32F411E-Disco (STMICROELECTRONICS, 2020), que é a placa onde está alocado o microcontrolador do robô. É possível observar uma imagem deste componente na Figura 14. Esta placa fica alocada numa placa eletrônica chamada de principal, que também contém os *drivers* de ativação dos motores do robô e o módulo de comunicação via rádio frequência.

Figura 14 – Placa de desenvolvimento STM32F411E-Disco utilizada.



Fonte: Autor

5.2 COMBINAÇÃO DOS SENSORES

Nesta seção serão definidos os sensores a serem utilizados e como eles foram combinados para realizar a predição e correção do filtro de Kalman.

Os sensores utilizados nos testes são os descritos ao longo da Seção 2.2, que são: IMU (Giroscópio + Acelerômetro), encoders, sistema de câmeras utilizado na SSL, além do modelo do robô omnidirecional (apresentado na Seção 2.1). Eles foram combinados entre as etapas de predição e correção a fim de encontrar a melhor configuração para realizar a estimativa de posição utilizando filtro de Kalman, como é descrito na Tabela 1.

No caso, os três primeiros cenários foram realizados utilizando o modelo do sistema na fase de predição, variando qual sensor era utilizado na fase de correção. Estes primeiros cenários serviram como uma introdução para o desenvolvimento do filtro de Kalman. No caso, o modelo do sistema leva em conta a dinâmica de um robô da categoria SSL (ver Seção 2.1) e o

Tabela 1 – Combinações dos sensores na predição e correção do Filtro de Kalman.

Teste	Predição	Correção
Cenário 1	Modelo do sistema	Câmera
Cenário 2	Modelo do sistema	Encoders
Cenário 3	Modelo do sistema	IMU
Cenário 4	IMU	Câmera
Cenário 5	IMU	Encoders
Cenário 6	Encoders	Câmera
Cenário 7	Encoders	IMU
Cenário 8	-	Câmeras

Fonte: o Autor

comando de velocidade enviado para ele. Os dois próximos cenários foram realizados com a IMU na fase de predição, enquanto na fase de correção foram utilizados os outros dois sensores, respectivamente. Os dois cenários seguintes tiveram a mesma ideia dos testes 4 e 5, mas na fase de predição foram utilizados os encoders e na fase de correção os outros dois sensores. Já o cenário 8 é o sistema de localização atual da equipe, que utiliza apenas os dados vindos do sistema de câmeras, ou seja, não há uma fusão de sensores propriamente dita, e serve como comparação para os outros cenários.

5.3 IMPLEMENTAÇÃO

Esta seção descreverá a visão geral da implementação dos algoritmos deste trabalho, isto é, a linguagem utilizada para programá-los, o hardware utilizado e o fluxo geral do sistema de localização proposto.

Em relação à linguagem de programação, para codificar os algoritmos de fusão de sensores no microcontrolador descrito na Seção 5.1 foi utilizada linguagem C. A biblioteca CMSIS-DSP foi usada principalmente para auxiliar nos cálculos matriciais necessários nas equações do KF, assim como para auxílio em cálculos que envolvam trigonometria (ARM, 2024).

O hardware utilizado no trabalho pode ser dividido em dois: o que é interno ao robô, e o que é externo ao robô. Internamente ao robô têm-se os sensores, que foram descritos na Seção 2.2, além do microcontrolador utilizado para cálculo de fusão de sensores para localização de maneira embarcada. De modo externo ao robô deve-se considerar o notebook que é utilizado tanto para recebimento dos dados vindos do sistema de câmeras da categoria, mas também para

cálculo da trajetória do robô em campo e para registro de dados a fim de análise dos resultados obtidos com os algoritmos. As especificações técnicas do notebook estão descritas na Tabela 2.

Tabela 2 – Especificações de hardware do notebook.

Sistema Operacional	Ubuntu 22.04.5 LTS x86_64
CPU	Intel i7-7700HQ CPU @ 2.80GHz
GPU	NVIDIA Geforce MX150
Memória RAM	16GB

Fonte: o Autor

Além disso, as câmeras utilizadas pelo SSL-Vision para detecção dos robôs utilizando o padrão colorido em cima deles são as seguintes:

- a) Stingray F046C (TECHNOLOGIES, 2011) + Lente Tamron 12VM412ASIR
- b) Logitech BRIO 4K UHD (LOGITECH, 2021)

Durante os testes, a transferência dos dados processados pelo SSL-Vision para o robô é feita via rádio-frequência, que a equipe já utiliza para enviar outros comandos para o robô, como velocidades locais, ativação do sistema de chute e acionamento do sistema de drible.

O transceptor utilizado para comunicação entre a estação do rádio conectada ao computador da equipe e o robô é o módulo nRF24L01, que no caso opera numa frequência de 2.4GHz e necessita de poucos componentes para ser projetado como um sistema de rádio (NORDIC SEMICONDUCTOR, 2007). O módulo é configurado e operado via SPI acessando o mapa de registradores que servem para configurar o componente.

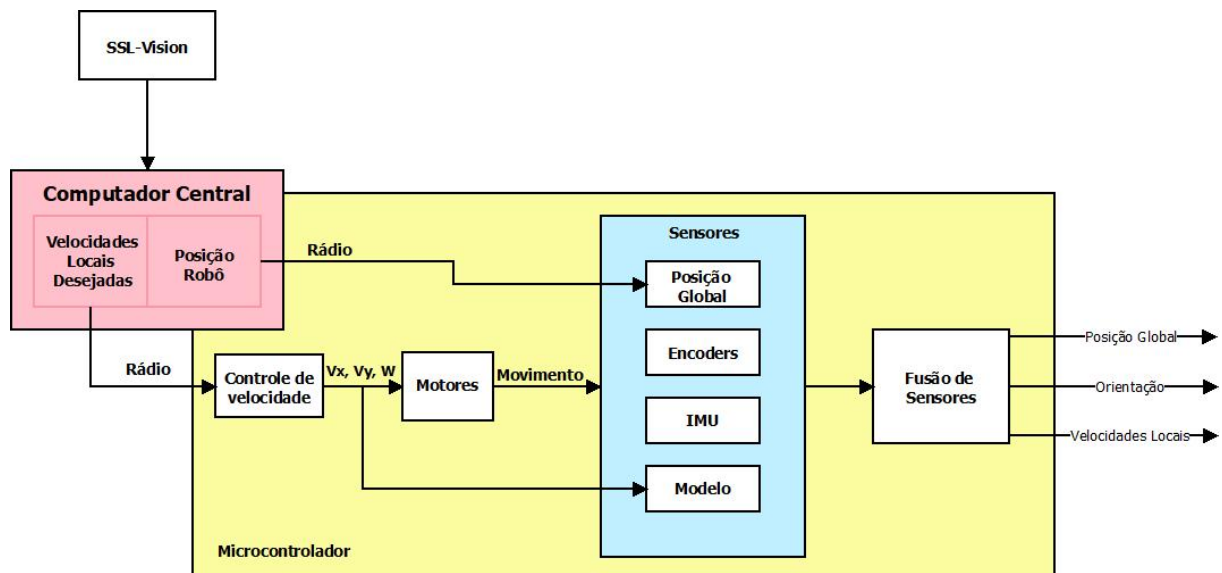
O computador que roda a estratégia da equipe faz a formatação do pacote a ser enviado via rádio para o robô, sendo que o pacote pode ter um comprimento máximo de 32 bytes por conta da limitação do próprio nRF24L01. Assim, o pacote é transmitido para o transceptor por meio de comunicação serial via USB e, em seguida, o pacote é enviado para o robô, que verifica se o pacote pertence a ele pelo ID que é transmitido. O tamanho do pacote pode ser um gargalo, visto que há outras informações que são transmitidas via rádio, como as velocidades linear, angular e do sistema de drible, ID do robô e tipo de chute (já que o robô pode fazer chutes retos ou "cavadinhas"), além da posição do robô em campo.

As posições lidas e transmitidas para os robôs vindas do sistema global de câmeras da liga necessitam de 2 bytes cada, totalizando assim 6 bytes para transmissão dessas informações ao robô. Isto porque apenas 1 byte não é suficiente para transferir esses dados, já que seriam

apenas 256 possibilidades (2^8 pela quantidade de bits num byte) e, por exemplo, a posição linear em X do robô pode ser de $\pm 4500\text{mm}$, totalizando 9000 posições, logo 65536 possibilidades (2^{16} utilizando 2 bytes) já é o suficiente.

O diagrama geral do funcionamento do sistema de localização por fusão de sensores de forma embarcada ao robô pode ser observada na Figura 15. Nela, é possível entender de maneira clara a diferença entre o que é calculado no computador central da equipe (recebimento dos dados do SSL-Vision e cálculo das velocidades locais desejadas) e o que é feito no microcontrolador do robô (recebimento dos dados do computador central, implementação dos sensores e cálculo da fusão de sensores).

Figura 15 – Diagrama de implementação da fusão de sensores para localização.



Fonte: Autor

5.4 GROUND TRUTH

O *ground truth* é parte determinante desse projeto, visto a necessidade de comparar o sistema de localização desenvolvido com a posição real do robô em campo. Por conta disso, um sistema para calcular a posição real do robô foi desenvolvido utilizando dois LiDARs a fim de encontrar o formato arredondado dos robôs da categoria SSL.

A Seção 6.1 traz o método desenvolvido em detalhes, assim como os resultados obtidos em comparação ao sistema de câmeras da categoria SSL, mas também em relação à medição manual utilizando uma trena.

O sistema roda em conjunto ao recebimento dos dados e ao cálculo de trajetória do robô no notebook descrito na Tabela 2. Para performar a detecção, foram utilizados os lasers RPLIDAR A1 (SLAMTEC, 2016) e RPLIDAR S2 (SLAMTEC, 2021), os modos de operação deles estão descritos na Tabela 3. Além disso, ambos os sensores operaram com frequência de 10Hz.

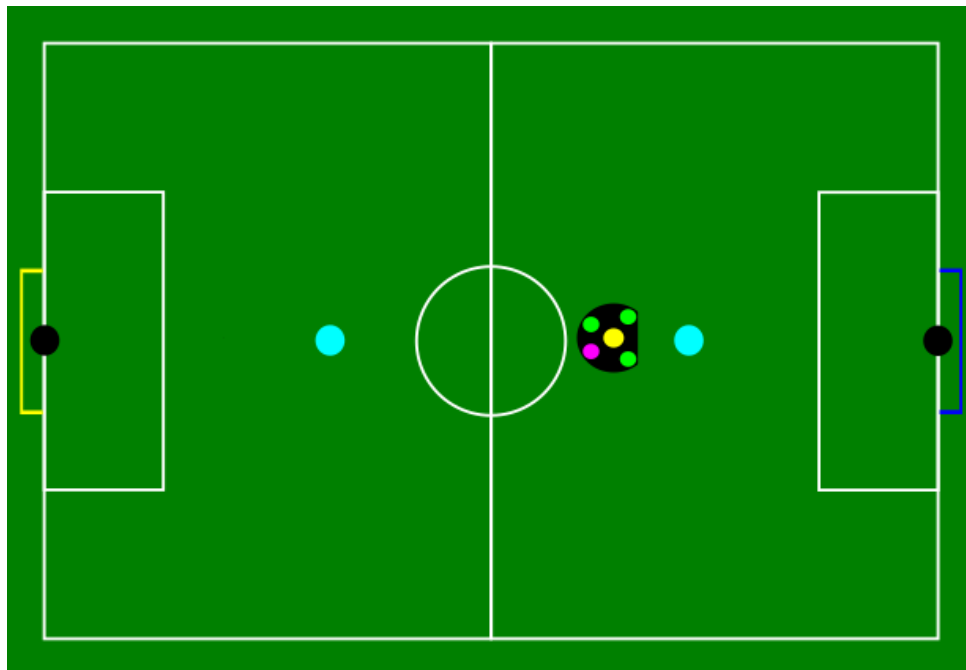
Tabela 3 – Especificações dos LiDARs utilizados.

Sensor	Princípio	Modo de operação	Pontos capturados
RPLIDAR A1	Triangulação	Boost	787
RPLIDAR S1	Time-of-Flight	Standard	1612

Fonte: o Autor

Os LiDARs foram posicionados no meio de cada um dos gols a fim de obter uma melhor detecção, visto que o laser A1 tem um alcance máximo de 6 metros, e o erro de detecção aumenta gradualmente com a distância. Uma representação da posição dos lasers (pontos pretos), posição das câmeras (pontos azuis) e do robô pode ser vista na Figura 16.

Figura 16 – Representação do posicionamento dos lasers em campo.



Fonte: Autor

Apesar de os LiDARs estarem posicionados aproximadamente numa mesma altura, é importante ressaltar que por conta da modulação da frequência do sinal enviado por cada um

deles durante a varredura realizada, além do próprio funcionamento rotativo da varredura, que faz a janela de interferência ser uma fração de segundo, não há interferência entre ambos os LiDARs.

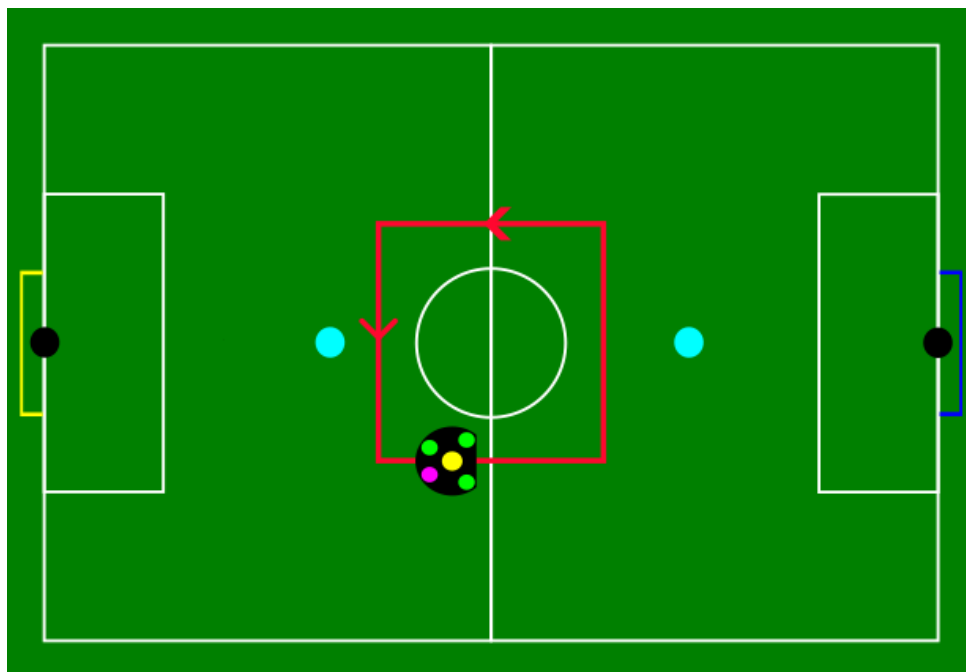
5.5 TESTES

Nesta seção serão definidos os testes a serem realizados a fim de obter dados relevantes para análise posterior das técnicas e combinações de sensores implementadas neste trabalho.

A fim de verificar qual a melhor combinação de sensores para realização da estimativa de posição utilizando filtro de Kalman de robôs da SSL, serão realizados testes em dois cenários a fim de verificar pontos fortes e fracos de cada combinação em ambientes distintos do campo.

O primeiro teste é um trajeto em forma de quadrado localizado na região central do campo, como é mostrado na Figura 17, com 70cm de lado. Como o trajeto está centralizado no campo, será possível analisar os resultados das diferentes combinações numa região em que há sobreposição das duas câmeras utilizadas, que no caso gera um problema de que o mesmo objeto tem diferentes posições por ser visto por câmeras distintas, problema que é demonstrado na Figura 18.

Figura 17 – Ilustração do teste de quadrado menor.



Fonte: o Autor

Figura 18 – Problema de sobreposição de câmeras.



Fonte: Retirado de Pauli e Tonidandel (2019)

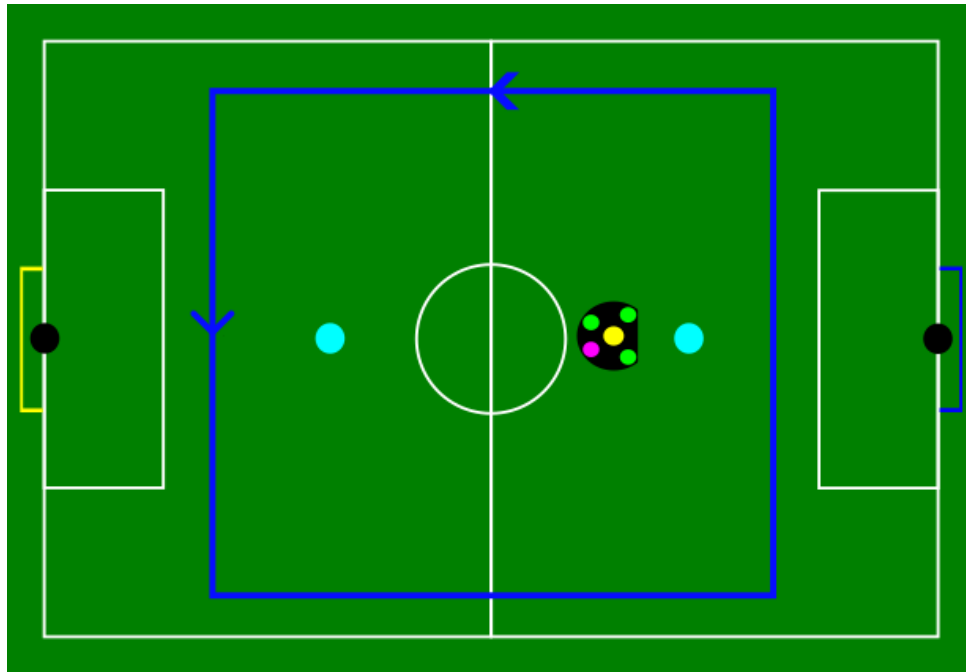
O segundo teste é um trajeto em forma de quadrado também, mas com um tamanho maior (150cm de lado) e passando por regiões do campo localizadas mais para as bordas, como é mostrado na Figura 19. Este trajeto tem o objetivo de analisar as diferentes combinações propostas de sensores em regiões localizadas nas bordas do campo, que apresentam problemas de distorção maiores causadas pela utilização de lentes olho de peixe nas câmeras.

5.6 AVALIAÇÃO

Nesta seção serão descritos os critérios de avaliação que serão utilizados para validar as diferentes combinações de sensores a fim de encontrar a melhor entre elas para realizar a estimativa de posição de um robô móvel.

O primeiro critério de avaliação será a comparação da posição (x, y) e orientação θ preditas e a real, que será medida pela câmera utilizada. A fim de diminuir o erro de posição da câmera, a ideia é fazer os testes logo abaixo da câmera a fim de evitar problemas, como distorção focal da imagem ao ir para as bordas do campo e *overlap* entre duas câmeras. A segunda avaliação é o erro médio das coordenadas (x, y) e da orientação θ . No caso, será analisada a diferença do

Figura 19 – Ilustração do teste de quadrado maior.



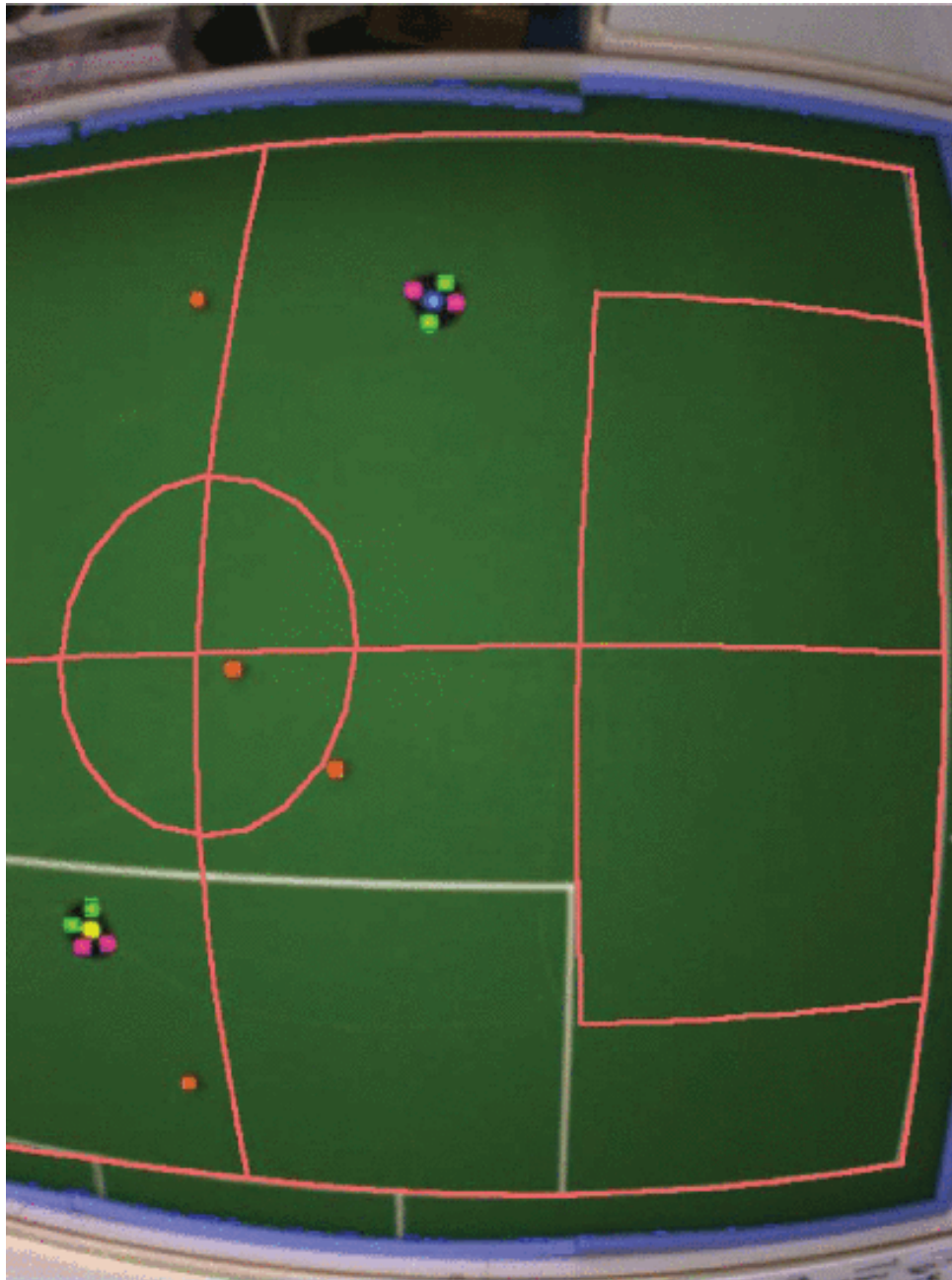
Fonte: o Autor

que é medido pelo sistema de câmeras da SSL e o trajeto proposto em cada um dos testes. O terceiro critério de avaliação será o tempo gasto para completar os percursos propostos.

A fim de aumentar a confiabilidade dos dados a serem extraídos, cada teste foi realizado 10 vezes. A partir deles, serão extraídos a média, desvio padrão e o valor máximo de cada uma das métricas para uma análise concisa. Em cada um dos testes foram obtidas, em média, 700 amostras no trajeto quadrado menor e 1500 amostras no trajeto quadrado maior, quantidade suficiente para análise das combinações de sensores propostas.

Ao fim dos experimentos, os dados serão avaliados e, então, o melhor sistema de estimativa de posição para os robôs da categoria *Small Size* de futebol de robôs da RoboCup será escolhido avaliando os testes comentados acima. Assim, um sistema de estimativa de posição ótimo é o que possui o menor erro médio nas coordenadas (x, y) e na orientação θ e que realiza o percurso no menor tempo possível.

Figura 20 – Problema da distorção da imagem de câmeras com lente olho de peixe.



Fonte: o Autor

6 RESULTADOS

Nesta seção serão apresentados os resultados obtidos neste projeto através dos experimentos descritos na Seção 5.5. Além disso, também será descrita como foi feita a aquisição do *ground truth*, que servirá para comparação dos algoritmos. A fim de melhor entendimento, os resultados serão analisados graficamente ou por meio de tabelas.

6.1 AQUISIÇÃO DO GROUND TRUTH

Como descrito na Seção 5.4, foi necessário o desenvolvimento de um sistema utilizando 2 LiDARs para obter o ground truth do posicionamento do robô a fim de realizar a comparação dos sistema de localização desenvolvido nesse projeto com a posição real do robô. Nessa seção serão descritos os detalhes do sistema desenvolvido, como a técnica utilizada e a visão geral do sistema, além dos resultados alcançados pelo sistema de ground truth desenvolvido a fim de mostrar sua eficiência em comparação com a posição real do robô, medida utilizando uma trena, e com o sistema de câmeras da categoria SSL.

6.1.1 Técnica utilizada e visão geral do sistema

A Figura 21 mostra a arquitetura do software do sistema proposto, que é baseado num pacote do ROS2 que aplicará um *scan matching* para encontrar a posição relativa de ambos os lasers. Em seguida, os dados de ambos os LiDARs são combinados em uma única imagem, em que diferentes operações morfológicas são aplicadas para melhorar a qualidade dela. Na sequência a etapa de detecção acontece para reconhecimento do padrão circular na imagem e, por fim, realiza-se o cálculo da coordenada global do robô, já que o sistema em um primeiro momento calcula a posição em relação ao posicionamento dos lasers.

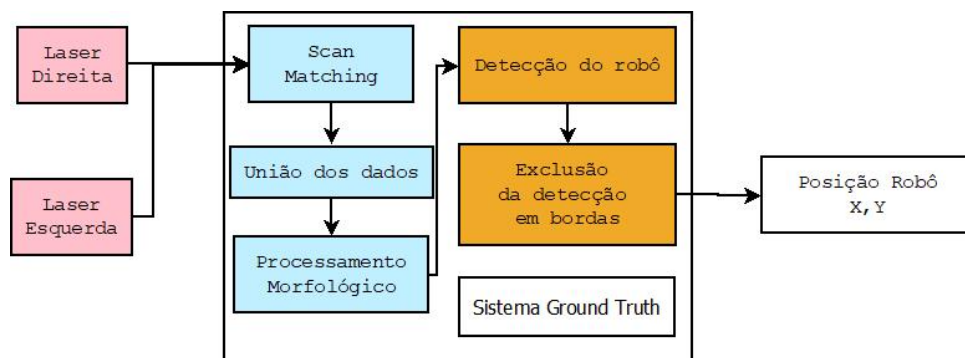


Figura 21 – Arquitetura do software.

O primeiro estágio do sistema é a utilização do *scan matching* para encontrar a transformação (translação, rotação e escala) entre as nuvens de pontos criadas pelo lasers que melhor encaixa os dados de ambos (NIETO; BAILEY; NEBOT, 2007). Essa etapa é decisiva no sistema visto que qualquer centímetro de diferença entre ambos os dados dos lasers pode levar a uma maior dificuldade para encontrar o padrão circular do robô ou pode adicionar um desvio na posição detectada.

O *scan matching* funciona a partir de duas nuvens de pontos, a fonte P e a alvo S. O objetivo do algoritmo é encontrar a transformação $T = (R, t)$, composta pela rotação R e translação t, que minimiza a diferença entre ambas as nuvens de pontos P e S, como descrito na Equação 30, em que r_i é o ponto mais perto na nuvem de ponto alvo S.

$$T^* = \arg \min \sum_{i=1}^n \|(Rp_i + t) - r_i\|^2 \quad (30)$$

Então, o estágio seguinte do software é a união dos dados de ambos os lasers em uma imagem, chamada de imagem de pontos do laser, aplicando a transformação encontrada na etapa de *scan matching*. Para a obtenção dos dados dos sensores, o pacote do ROS2 desenvolvido pela SLAMTEC® para ambos os lasers utilizados no sistema, já que é uma maneira mais fácil para comunicar com os sensores, inicializá-los, e receber os dados deles. O nó do ROS2 publica a mensagem do tipo *laser scan* que contém as distâncias de detecção pelo sensor e seus respectivos ângulos. A partir desses dados, é possível construir a imagem de pontos do laser seguindo a Equação (31), que demonstra como determinar as coordenadas de cada objeto detectado naquele *scan*, no caso r é a distância em metros e θ é o ângulo respectivo à distância detectada em radianos.

$$\begin{cases} x = r * \cos(\theta) \\ y = r * \sin(\theta) \end{cases} \quad (31)$$

Depois de unir dos dados dos sensores, há uma camada para processar a imagem de pontos dos lasers antes do reconhecimento do formato circular no próximo estágio. O processamento morfológico de uma imagem é essencial para reginar a qualidade de imagens, redução de ruídos contidos nelas, e melhorar a análise estrutural delas, o que leva a uma extração de propriedades mais precisa em imagens (LOTUFO et al., 2023). Primeiro, o algoritmo aplica uma leve dilatação dos pontos a fim de conectar pontos próximos, então há uma etapa de erosão para uma melhor detecção de formatos na imagem. Finalmente, um borrão delicado é aplicado na imagem de

pontos dos lasers para suavização, então reduzindo o ruído na imagem e a detecção de falsos círculos, assim como ajudando na detecção do padrão.

A próxima etapa do software é a detecção do robô. Esse estágio realiza a detecção do padrão circular, ou semi-circular, na imagem de pontos dos lasers. O método utilizado para detecção do padrão é a transformada de Hough (HT). A técnica, em um primeiro momento, foi criada para detecção de linhas em imagens binárias, mas outras variações foram sugeridas para detectar diferentes formatos, como os circulares (MUKHOPADHYAY; CHAUDHURI, 2015).

A HT para círculos utiliza a equação do círculo demonstrada na Equação (32), que o sistema calcula para cada ponto de borda na imagem pré-processada de acordo com os limites máximo e mínimo de raio definidos. Cada ponto de borda contribui com votos para possíveis círculos que ele pode fazer parte. Então, os votos são acumulados e o máximos locais irão representar as localizações dos centros e os raios dos círculos nas imagens (HASSANEIN et al., 2015).

$$(x - a)^2 + (y - b)^2 = r^2 \quad (32)$$

Na Figura 22 é possível observar dois momentos da arquitetura. A Figura 22a mostra os pontos dos lasers após os estágios de *scan matching* e união dos dados, mas antes da etapa de processamento morfológico. No caso, os pontos azuis são do laser à direita do campo, enquanto os pontos verdes são do laser à esquerda. Já a Figura 22b mostra a etapa de detecção, em que a HT é aplicada na imagem de pontos dos lasers para detecção do padrão circular.

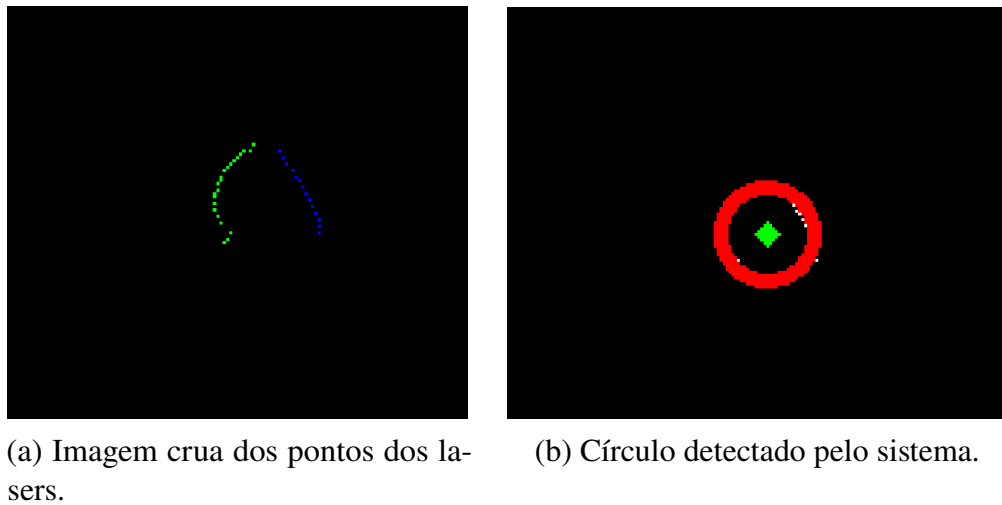


Figura 22 – Duas etapas da arquitetura desenvolvida.

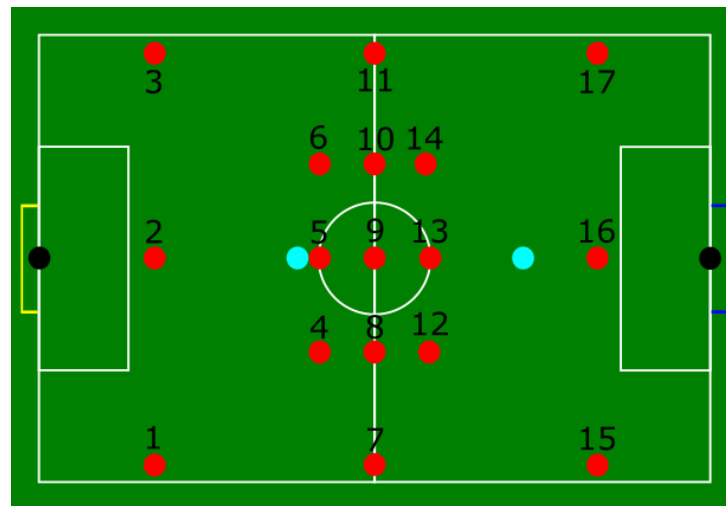
A última etapa de pós-processamento é realizada para detectar e excluir falsos positivos nas bordas do campo, já que a maior parte dos pontos detectados pelo laser são os pontos da

borda do campo. Basicamente, as coordenadas de fronteira do campo são excluídos e a detecção é focada dentro do campo.

6.1.2 Resultados do sistema desenvolvido

Para realizar a comparação, foram decididas 17 posições arbitrárias em campo, que estão demonstrados na Figura 23. Esses cenários foram escolhidos visando observar o comportamento do sistema de lasers desenvolvido em diferentes situações, no caso em posições mais centralizadas no campo (chamados de cenários centralizados, e são as posições 4-6, 8-10 e 12-14), que são cenários praticamente equidistantes de ambos os lasers, e também em posições mais dispersas no campo (chamados de cenários dispersos, são as posições 1-3, 7, 11 e 15-17), onde os pontos estão consideravelmente mais perto de um laser do que outro, mas também nas beiradas do campo. Além disso, os cenários dispersos servirão para comparação com o sistema de câmeras da SSL, visto que por conta da distorção na utilização de lentes olho de peixe, há uma distorção maior nos cantos do campo.

Figura 23 – Cenários de testes do sistema de ground truth.



Fonte: o Autor

Além dessa separação dos cenários, eles também foram separados em 3 conjuntos, que serão chamados de quadrantes, que visam observar as diferenças de captura de ambos os sistemas na região de sobreposição das câmeras, localizada ao longo da região central do campo. O primeiro quadrante (posições 1-6) são os cenários em que somente a câmera da esquerda consegue capturar. O segundo quadrante (posições 7-14) são os cenários posicionados onde ambas câmeras

conseguem capturar. Por fim, o terceiro quadrante (posições 15-17) representam os cenários em que somente a câmera à direita pode capturar. Essa separação também ajudará a entender como o sistema câmeras do SSL lidam com a distorção separadamente. Espera-se que as coordenadas do SSL-Vision tenha um erro substancial com o distanciamento do centro das câmeras.

Para poder comparar os sistemas com a posição real medida por uma trena, em cada posição serão capturadas 200 amostras do sistema de lasers, enquanto a saída do SSL-Vision também é adquirida. A média das amostras de ambos sistemas será calculada para cada posição, calculando o erro em relação ao medido com a trena, assim como o desvio padrão para cada cenário a fim de analisar a estabilidade dos sistemas.

As Tabelas 4 e 5 resumem os resultados alcançados pelo SSL-Vision, sistemas de lasers e a medição real para os eixos X e Y, respectivamente, nos 17 cenários de teste. A coluna Real indica as coordenadas reais do robô com o auxílio de uma trena, enquanto as colunas Laser e Visão mostram a média e o desvio padrão das amostras medidas pelos sistemas de lasers e câmeras, respectivamente.

Tabela 4 – Resultados alcançados para o eixo X no campo de teste.

Cenário	Real [cm]	Laser [cm]	Visão [cm]
1	-91,5	-91.797 +- 0.584	-92.945 +- 0.026
2	-88,1	-87.594 +- 0.437	-89.099 +- 0.023
3	-93	-92.510 +- 0.391	-93.397 +- 0.049
4	-55	-54.584 +- 0.703	-55.389 +- 0.037
5	-51	-50.321 +- 0.655	-51.323 +- 0.017
6	-54,5	-54.504 +- 0.926	-55.363 +- 0.037
7	0	1.092 +- 0.674	0.488 +- 0.012
8	0	0.390 +- 0.624	0.370 +- 0.073
9	0	0.552 +- 0.513	-0.158 +- 0.061
10	0	0.786 +- 0.478	-0.505 +- 0.036
11	0	0.972 +- 0.953	0.178 +- 0.025
12	61,3	61.859 +- 0.649	61.588 +- 0.017
13	52	51.964 +- 0.519	52.379 +- 0.011
14	53	53.742 +- 1.036	52.868 +- 0.046
15	101	101.653 +- 1.450	100.866 +- 0.015
16	99	99.683 +- 0.554	98.785 +- 0.017
17	97	97.526 +- 1.509	96.679 +- 0.018

A Tabela 6 mostra o erro e desvio padrão médios em todos os cenários, mas também separadamente para os conjuntos centralizados e dispersos, nos eixos X e Y. Todas as medidas estão em centímetros. É possível analisar que o erro médio para o eixo X em ambos sistemas são comparáveis, com um diferença de aproximadamente 0.1cm entre ambos. Já para o eixo Y é

Tabela 5 – Resultados alcançados para o eixo Y no campo de teste.

Cenário	Real [cm]	Laser [cm]	Visão [cm]
1	-138,5	-137.784 +- 0.726	-139.464 +- 0.047
2	0	0.844 +- 1.475	0.365 +- 0.018
3	132	131.688 +- 0.625	134.255 +- 0.022
4	-85	-84.104 +- 0.851	-85.172 +- 0.016
5	0	-0.518 +- 1.424	0.690 +- 0.036
6	81,5	80.801 +- 1.435	83.025 +- 0.018
7	-141	-140.488 +- 0.742	-139.829 +- 0.014
8	-82,5	-82.172 +- 0.978	-82.306 +- 0.061
9	0	0.668 +- 1.052	0.963 +- 0.041
10	83,5	84.431 +- 1.781	85.154 +- 0.028
11	125,5	125.017 +- 0.925	126.895 +- 0.083
12	-92,3	-92.848 +- 0.900	-91.665 +- 0.015
13	0	-0.568 +- 0.879	0.810 +- 0.014
14	85,3	85.046 +- 1.317	86.171 +- 0.053
15	-135,5	-135.906 +- 0.813	-135.158 +- 0.022
16	0	0.048 +- 0.598	1.236 +- 0.016
17	123,5	123.125 +- 1.015	124.690 +- 0.023

possível observar uma diferença considerável entre eles, perto de 0.43cm. Essa diferença pode ser explicada pela distorção ocorrida nas bordas da imagem da câmera, o que aumenta o erro nesses pontos, já que eles possuem uma distância maior das câmeras no eixo Y do que no X.

Tabela 6 – Erro e desvio padrão médios nos cenários.

	X - Laser	X - Visão	Y - Laser	Y - Visão
Erro médio	0,5519	0,4461	0,5356	0,9665
Erro - Centralizado	0,4626	0,3785	0,6011	0,8348
Erro - Disperso	0,6523	0,5221	0,4620	1,1147
Des. Padrão médio	0,7444	0,0305	1,0315	0,0310
Des. Padrão - Centralizado	0,6781	0,0372	1,1796	0,0313
Des. Padrão - Disperso	0,8190	0,0231	0,8648	0,0306

Os erros entre os conjuntos de cenários centralizado e dispersos mostram que a câmera lida melhor com posições no centro do campo, o que é explicado pela distorção nas bordas das imagens das câmeras. Esse efeito não pode ser percebido analisando o sistema com lasers, mostrando que a detecção por eles é regular em todas as partes do campo, e prova que há vantagem na utilização de dois lasers para melhorar a detecção do padrão circular do robô em campo.

Ao analisar o desvio padrão na Tabela 6, é possível notar que o sistema de câmeras da glxtrshortssl apresenta resultados bem mais expressivos do que o sistema com LiDARs visto

que a detecção foi aproximadamente 0.3mm dispersa da média analisando todos os cenários no eixo X, e menos que 0.4mm e 0.3mm nos conjuntos centralizado e disperso, respectivamente. Já para o eixo Y os resultados foram praticamente os mesmos, comparando com o sistema do laser, este foi pior também, mas os resultados continuam aceitáveis. A resolução da imagem formada pelos dados dos lasers podem explicar essa diferença no desvio padrão. Essa imagem formada é de 1000x1000 pixels, e cada pixel representa 0.533cm no campo por conta da escala utilizada para alocar os pontos na imagem. Então, a menor diferença na detecção leva a aproximadamente 0.5cm na posição detectada do robô.

Analisando os quadrantes, as Tabelas 7 e 8 mostram os resultados resumidos para eles nos eixos X e Y, respectivamente. Notavelmente o quadrante pertencente à câmera esquerda é pior para o eixo X quando analisando o erro médio, com mais que o dobro de erro comparando com os quadrantes 2 e 3. A utilização de lente olho de peixe pode explicar esse efeito já que leva a uma distorção comparável à câmera direita. Para o eixo Y em todos os quadrantes, é possível observar que o erro médio é pior quando comparado com o sistema de lasers em pelo menos 50%. As medidas nas Tabelas 7 e 8 também estão em centímetros.

Tabela 7 – Resultados dos quadrantes para o eixo X.

	Erro X - Laser	Erro X - Vis.	Des. Padrão X - Laser	Des. Padrão X - Vis.
Quad. 1	0,3986	0,7360	0,6224	0,0315
Quad. 2	0,6411	0,3122	0,6807	0,0351
Quad. 3	0,6206	0,2233	1,1710	0,01667

Tabela 8 – Resultados dos quadrantes para o eixo Y.

	Erro Y - Laser	Erro Y - Vis.	Des. Padrão Y - Laser	Des. Padrão Y - Vis.
Quad. 1	0,6641	0,9951	1,0893	0,0261
Quad. 2	0,5844	1,0754	1,0956	0,0454
Quad. 3	0,3665	0,8473	0,9203	0,0238

Ao analisar o desvio padrão, não é notável qualquer influência do posicionamento das câmeras em ambos os eixos. Além disso, o SSL-Vision continua melhor que o sistema de lasers em todos os quadrantes para ambos os eixos. Quando analisado para ambos os eixos, não é possível analisar diferenças significativas entre os quadrantes para o sistema de lasers, mostrando que a detecção realizada por ele é regular, e os erros acumulados em longas distâncias para um laser são compensados pelo outro laser.

6.2 PARAMETRIZAÇÃO DAS COVARIÂNCIAS

A parametrização das variâncias dos sensores é parte importante do projeto do Filtro de Kalman Estendido, principalmente para a estimativa das matrizes de covariância dos ruídos de processo e medições, respectivamente, Q_k e R_k , visto que elas impõe quanta confiança se coloca nos valores do estado estimado.

A partir disso, para definir o desvio padrão das posições globais X e Y e da orientação a partir dos sensores foram capturadas 1000 amostras dessas variáveis com o robô parado, já para definir o desvio padrão das velocidades locais X e Y, foram capturadas 1000 amostras dessas variáveis com o robô andando numa velocidade constante de $0.2m/s$. Então, a partir dessas amostras é calculado o desvio padrão de cada variável dita.

6.3 IMPLEMENTAÇÃO DOS CENÁRIOS

6.3.1 Cenário 1 - Modelo + Visão

O primeiro cenário utiliza do modelo cinemático de um robô omnidirecional na fase de predição e os dados do sistema global de visão na fase de correção.

A Equação 34 mostra a atualização dos estados na fase de predição do sistema de localização nesse cenário. As entradas do sistema para esse cenário são as velocidades locais do robô nos eixos X e Y e a angular (v_{x-t} , v_{y-t} e $v_{\theta-t}$, respectivamente) calculadas pelo sistema de planejamento de trajetória da equipe, representada pelo vetor \mathbf{u}_k , conforme a Equação 33.

$$\mathbf{u}_k = \begin{bmatrix} v_{x-t} \\ v_{y-t} \\ v_{w-t} \end{bmatrix} \quad (33)$$

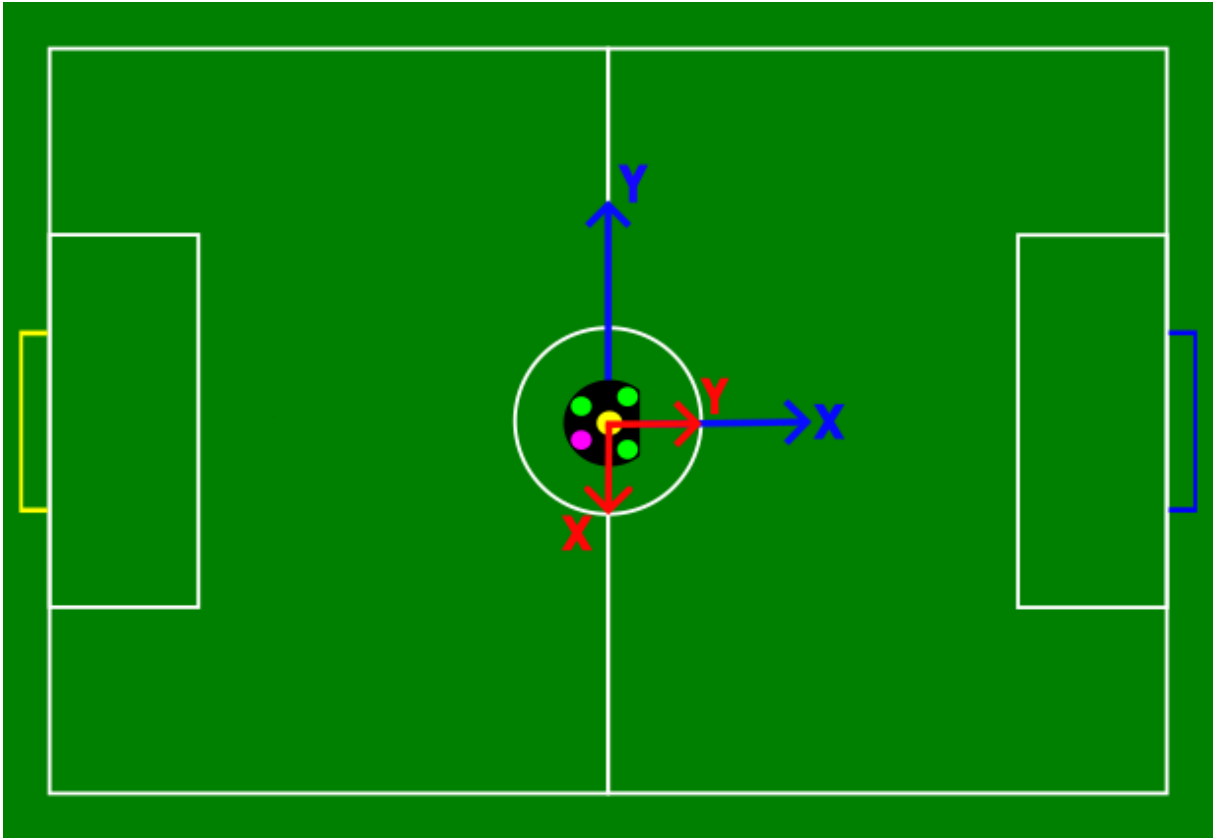
$$\begin{cases} x[k] = x[k-1] + V_{x-G} \times dt \\ y[k] = y[k-1] + V_{y-G} \times dt \\ \theta[k] = \theta[k-1] + v_{w-t}[k] \times dt \\ v_x[k] = v_{x-t}[k] \\ v_y[k] = v_{y-t}[k] \end{cases} \quad (34)$$

Para o cálculo das posições globais do robô é necessário que as velocidades estejam descritas em relação ao eixo do campo, no caso descritas como V_{x-G} e V_{y-G} , e são calculadas

conforme a Equação 35. Além disso, é importante notar a transformação feita subtraindo $\frac{\pi}{2}$ do ângulo calculado do robô para alinhar os eixos locais do robô com os eixos global do campo, conforme a Figura 24, em que o eixo em azul representa a convenção adotado pelo sistema de visão da SSL, enquanto o eixo em vermelho representa a convenção adotada pela equipe para o robô. Esse conceito de transformação será utilizado tanto nesse quanto nos próximos cenários.

$$\begin{cases} V_{x-G} = \cos(\theta[k-1] - \frac{\pi}{2}) \times v_x[k-1] - \sin(\theta[k-1] - \frac{\pi}{2}) \times v_y[k-1] \\ V_{y-G} = \sin(\theta[k-1] - \frac{\pi}{2}) \times v_x[k-1] + \cos(\theta[k-1] - \frac{\pi}{2}) \times v_y[k-1] \end{cases} \quad (35)$$

Figura 24 – Diferença entre os eixos do campo e do robô.



Fonte: o Autor

A partir disso, a matriz jacobiana G_k dos estados previstos em relação ao estado anterior segue a Equação 36.

$$G_k = \begin{bmatrix} 1 & 0 & (\cos(\theta)v_x\Delta_t - \sin(\theta)v_y\Delta_t) & \sin(\theta)\Delta_t & \cos(\theta)\Delta_t \\ 0 & 1 & (\sin(\theta)v_x\Delta_t + \cos(\theta)v_y\Delta_t) & -\cos(\theta)\Delta_t & \sin(\theta)\Delta_t \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (36)$$

Por fim, a matriz de covariância dos ruído do processo Q_k é definida na Equação 37.

$$Q_k = \begin{bmatrix} 1.53 \cdot 10^{-2} & 0 & 0 & 0 & 0 \\ 0 & 1.53 \cdot 10^{-2} & 0 & 0 & 0 \\ 0 & 0 & 4.72 \cdot 10^{-3} & 0 & 0 \\ 0 & 0 & 0 & 1 \cdot 10^{-3} & 0 \\ 0 & 0 & 0 & 0 & 1 \cdot 10^{-3} \end{bmatrix} \quad (37)$$

Quanto à fase de correção, nesse cenário são utilizados os dados de posição do sistema global de câmeras da SSL. Conforme a Equação ??, o sistema de visão providencia observações diretas tanto das posições globais dos robôs, quanto da orientação deles, com um certo ruído. O vetor de medições é apresentado pelo vetor \mathbf{z}_k . A partir disso, a matriz de transformação do estado real no estado observado C_k é definida na Equação 39, já a matriz de covariância das medições do sistema global de câmeras é definida na Equação 40.

$$\mathbf{z}_k = \begin{bmatrix} x_k \\ y_k \\ \theta_k \end{bmatrix} \quad (38)$$

$$C_k = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (39)$$

$$R_k = \begin{bmatrix} 1.16 \cdot 10^{-3} & 0 & 0 \\ 0 & 1.16 \cdot 10^{-3} & 0 \\ 0 & 0 & 9.87 \cdot 10^{-3} \end{bmatrix} \quad (40)$$

6.3.2 Cenário 2 - Modelo + Encoders

O segundo cenário também utiliza do modelo cinemático de um robô omnidirecional na fase de predição do EKF e os dados dos encoders acoplados às rodas na fase de correção.

A atualização dos estados na fase de predição do sistema de localização segue a Equação 34 também, sendo as entradas definidas na Equação 33.

Para a fase de correção, são utilizados os dados dos encoders acoplados às rodas na etapa de correção do EKF para ter uma estimativa melhor das velocidades locais X e Y do robô, e assim das posições globais e da orientação, como demonstrado pelo vetor de medições \mathbf{z}_k na Equação 41. A Equação 42 demonstra como os estados para a fase de correção são atualizados com as leituras dos encoders, sendo que as posições globais, orientação e velocidades locais são calculadas a partir das leituras dos encoders.

$$\mathbf{z}_k = \begin{bmatrix} x_{enc,k} \\ y_{enc,k} \\ \theta_{enc,k} \\ v_{x,enc,k} \\ v_{y,enc,k} \end{bmatrix} \quad (41)$$

$$\begin{cases} x_{enc,k} = x_{enc,k-1} + V_{x,enc,G} \Delta_t \\ y_{enc,k} = y_{enc,k-1} + V_{y,enc,G} \Delta_t \\ \theta_{enc,k} = \theta_{enc,k-1} + \omega_{enc,k} \Delta_t \\ v_{x,enc,k} = v_{x,enc,k} \\ v_{y,enc,k} = v_{y,enc,k} \end{cases} \quad (42)$$

O cálculo das velocidades globais para determinar as posições globais segue a ideia da Equação 35. Porém, na Equação 43 as velocidades locais utilizadas são providas a partir dos encoders.

$$\begin{cases} V_{x,enc,G} = \cos(\theta_{enc,k-1} - \frac{\pi}{2}) \times v_{x,enc,k} - \sin(\theta_k - \frac{\pi}{2}) \times v_{y,enc,k} \\ V_{y,enc,G} = \sin(\theta_{enc,k-1} - \frac{\pi}{2}) \times v_{x,enc,k} + \cos(\theta_k - \frac{\pi}{2}) \times v_{y,enc,k} \end{cases} \quad (43)$$

A partir disso, a matriz C de transformação do estado real no estado observado segue a Equação 44

$$C_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (44)$$

A matriz de covariância dos ruídos das medições R_k desse cenário é definida na Equação 45.

$$R_k = \begin{bmatrix} 5 \cdot 10^{-4} & 0 & 0 & 0 & 0 \\ 0 & 5 \cdot 10^{-4} & 0 & 0 & 0 \\ 0 & 0 & 7 \cdot 10^{-3} & 0 & 0 \\ 0 & 0 & 0 & 5 \cdot 10^{-4} & 0 \\ 0 & 0 & 0 & 0 & 5 \cdot 10^{-4} \end{bmatrix} \quad (45)$$

6.3.3 Cenário 3 - Modelo + IMU

O terceiro cenário também utiliza do modelo cinemático de um robô omnidirecional na fase de predição do EKF, enquanto na fase de correção são utilizados os dados da IMU presente na placa principal do robô.

Para a fase de predição, o conceito segue o que foi descrito nas Seções 6.3.1 e 6.3.2. Já para a fase de correção, são utilizados os conceitos da IMU (giroscópio + acelerômetro) para obter uma estimativa de posição, orientação e velocidade do robô, conforme o que é descrito na Equação 46 no vetor \mathbf{z}_k . A Equação 47 demonstra como os estados para a fase de correção são atualizados com as leituras do giroscópio e acelerômetro.

$$\mathbf{z}_k = \begin{bmatrix} x_{imu,k} \\ y_{imu,k} \\ \theta_{imu,k} \\ v_{x,imu,k} \\ v_{y,imu,k} \end{bmatrix} \quad (46)$$

$$\begin{cases} x_{imu,k} = x_{imu,k-1} + V_{x,IMU,G} \times dt + \frac{1}{2} A_{x,IMU,G} \Delta_t^2 \\ y_{imu,k} = y_{imu,k-1} + V_{y,IMU,G} \times dt + \frac{1}{2} A_{y,IMU,G} \Delta_t^2 \\ \theta_{imu,k} = \theta_{imu,k-1} + \omega_{imu,k} \Delta_t \\ v_{x,imu,k} = v_{x,imu,k-1} + a_{x,k} \Delta_t \\ v_{y,imu,k} = v_{y,imu,k-1} + a_{y,k} \Delta_t \end{cases} \quad (47)$$

No caso da IMU, tanto as velocidades globais quanto as leituras de aceleração precisam ser transformadas para os eixos globais do campo. Sendo assim, as Equações 48 e 49 demonstram a transformação feita para ambos os componentes.

$$\begin{cases} V_{x,imu,G} = \cos(\theta_{imu,k-1} - \frac{\pi}{2}) \times v_{x,imu,k-1} - \sin(\theta_{imu,k-1} - \frac{\pi}{2}) \times v_{y,imu,k-1} \\ V_{y,imu,G} = \sin(\theta_{imu,k-1} - \frac{\pi}{2}) \times v_{x,imu,k-1} + \cos(\theta_{imu,k-1} - \frac{\pi}{2}) \times v_{y,imu,k-1} \end{cases} \quad (48)$$

$$\begin{cases} A_{x,imu,G} = \cos(\theta_{imu,k-1} - \frac{\pi}{2}) \times a_{x,k} - \sin(\theta_{imu,k-1} - \frac{\pi}{2}) \times a_{y,k} \\ A_{y,imu,G} = \sin(\theta_{imu,k-1} - \frac{\pi}{2}) \times a_{x,k} + \cos(\theta_{imu,k-1} - \frac{\pi}{2}) \times a_{y,k} \end{cases} \quad (49)$$

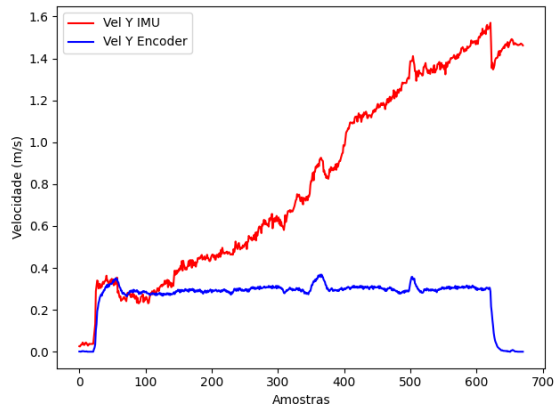
A partir disso, a matriz C de transformação do estado real no estado observado segue a Equação 50, já que a correção a partir dos dados da IMU provê observações diretas sobre todos os estados do sistema.

$$C_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (50)$$

A matriz de covariância dos ruídos das medições R_k desse cenário é definida na Equação 51.

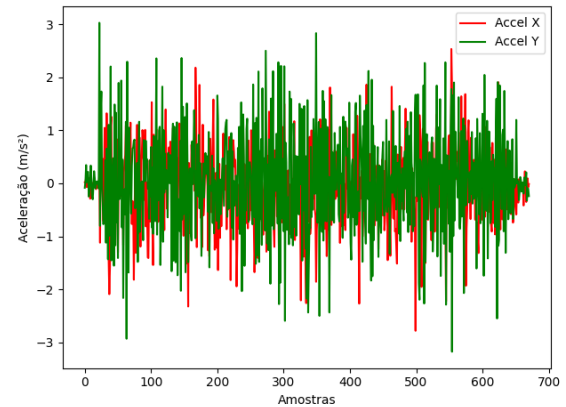
$$R_k = \begin{bmatrix} 8.39 \cdot 10^{-3} & 0 & 0 & 0 & 0 \\ 0 & 8.39 \cdot 10^{-3} & 0 & 0 & 0 \\ 0 & 0 & 2.01 \cdot 10^{-2} & 0 & 0 \\ 0 & 0 & 0 & 5 \cdot 10^{-2} & 0 \\ 0 & 0 & 0 & 0 & 5 \cdot 10^{-2} \end{bmatrix} \quad (51)$$

Figura 25 – Diferença das velocidades calculadas pelo acelerômetro e pelos encoders.



Fonte: o Autor

Figura 26 – Diferença das velocidades calculadas pelo acelerômetro e pelos encoders.



Fonte: o Autor

É importante pontuar que ao utilizar a IMU para a fase de correção, tanto nesse quanto nos outros cenários, é feita a integração da aceleração provida pelo acelerômetro a fim de estimar a velocidade e posição do robô. Entretanto, ao fazer essa integração, por conta do alto ruído presente nos dados do sensor, acaba-se acumulando um grande erro em um curto espaço de tempo. Esse efeito pode ser observado na Figura 25, que compara a velocidade local Y obtida a partir tanto da IMU (vermelho) quanto dos encoders (azul). É notável que durante o começo do movimento a integração da aceleração tem resultados parecidos com a velocidade dos encoders, mas depois de um tempo os erros da integração começam a crescer desenfreadamente.

Para impedir esse acúmulo de erro desenfreado a longo prazo, decidiu-se que as velocidades locais X e Y calculadas a partir da IMU, a cada 5 ciclos de correção realizados pelo EKF, são definidas como as velocidades locais X e Y estimadas pelo filtro, seguindo a Equação 52.

$$v_{x,imu,k} = v_{x,k} \quad (52)$$

$$v_{y,imu,k} = v_{y,k} \quad (53)$$

6.3.4 Cenário 4 - IMU + Visão

Nesse cenário 4, os dados da IMU são utilizados na fase de predição do EKF, enquanto na fase de correção são utilizados os dados do sistema global de câmeras da categoria SSL.

Para a fase de predição do EKF, os dados do acelerômetro são utilizados para prever tanto as posições globais X e Y quanto as velocidades locais X e Y do robô, enquanto os dados do giroscópio são utilizados para prever a orientação do robô em campo. A Equação 55 mostra como é feita a atualização de cada um dos estados com os dados da IMU. O vetor de entrada do sistema \mathbf{u}_k é composto pelas acelerações locais X e Y do robô, assim como a velocidade angular do robô, como está descrito na Equação 54.

$$\mathbf{u}_k = \begin{bmatrix} a_x \\ a_y \\ \omega \end{bmatrix} \quad (54)$$

$$\begin{cases} x_k = x_{k-1} + V_{x,G} \times dt + \frac{1}{2} A_{x,IMU,G} \Delta_t^2 \\ y_k = y_{k-1} + V_{y,G} \times dt + \frac{1}{2} A_{y,IMU,G} \Delta_t^2 \\ \theta_k = \theta_{k-1} + \omega_k \Delta_t \\ v_{x,k} = v_{x,k-1} + a_{x,k} \Delta_t \\ v_{y,k} = v_{y,k-1} + a_{y,k} \Delta_t \end{cases} \quad (55)$$

O cálculo das velocidades globais desse cenário segue o que é descrito na Equação 35, enquanto a transformação das acelerações locais para globais é demonstrada na Equação 56.

$$\begin{cases} A_{x,imu,G} = \cos(\theta_{k-1} - \frac{\pi}{2}) \times a_{x,k} - \sin(\theta_{k-1} - \frac{\pi}{2}) \times a_{y,k} \\ A_{y,imu,G} = \sin(\theta_{k-1} - \frac{\pi}{2}) \times a_{x,k} + \cos(\theta_{k-1} - \frac{\pi}{2}) \times a_{y,k} \end{cases} \quad (56)$$

A partir disso, a matriz jacobiana G_k dos estados previstos em relação ao estado anterior segue a Equação 57.

$$G_k = \begin{bmatrix} 1 & 0 & \cos(\theta) \left(v_x \Delta t + a_x \frac{\Delta t^2}{2} \right) - \sin(\theta) \left(v_y \Delta t + a_y \frac{\Delta t^2}{2} \right) & \sin(\theta) \Delta t & \cos(\theta) \Delta t \\ 0 & 1 & \sin(\theta) \left(v_x \Delta t + a_x \frac{\Delta t^2}{2} \right) + \cos(\theta) \left(v_y \Delta t + a_y \frac{\Delta t^2}{2} \right) & -\cos(\theta) \Delta t & \sin(\theta) \Delta t \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (57)$$

A matriz das covariâncias dos ruídos do modelo é definida conforme a Equação 58.

$$Q_k = \begin{bmatrix} 8.39 \cdot 10^{-3} & 0 & 0 & 0 & 0 \\ 0 & 8.39 \cdot 10^{-3} & 0 & 0 & 0 \\ 0 & 0 & 2.01 \cdot 10^{-2} & 0 & 0 \\ 0 & 0 & 0 & 5 \cdot 10^{-2} & 0 \\ 0 & 0 & 0 & 0 & 5 \cdot 10^{-2} \end{bmatrix} \quad (58)$$

Quanto à fase de correção, nesse cenário são utilizados os dados de posição do sistema global de câmeras da SSL, conforme foi descrito na Seção 6.3.1 nas Equações 38 e 39, que descrevem o vetor de medições \mathbf{z}_k e a matriz de transformação do estado real no estado observado C_k , respectivamente. A matriz de covariâncias das medições no caso dos encoders é definida na Seção 6.3.1 na Equação 40.

6.3.5 Cenário 5 - IMU + Encoders

No quinto cenário, continua-se utilizando os dados da IMU (acelerômetro + giroscópio) para a fase de predição do Filtro de Kalman Estendido, enquanto os dados de velocidade dos encoders são utilizados na fase de correção do EKF.

A predição segue o que foi descrito na Seção 6.3.4, especialmente as Equações 54, que define o vetor de controle \mathbf{u}_k , 55, que define como os estados do sistema são atualizados na fase de predição, 56, que mostra como as acelerações locais são transformadas em globais para o cálculo das posições, e 57, que define a matriz das jacobianas dos estados em relação ao estado anterior.

Para fase de correção, os conceitos utilizados seguem o que foi descrito na Seção 6.3.2, especialmente as Equações 41, que define o vetor de medições do sistema, 42, que demonstra como as medições são atualizadas com as leituras dos encoders, e 44, que define a matriz de transformação do estado real no estado observado. A matriz de covariâncias das medições no caso dos encoders é definida na Seção 6.3.2 na Equação 45.

6.3.6 Cenário 6 - Encoders + Visão

No sexto cenário, os dados dos encoders acoplados às rodas do robô providenciam leituras das velocidades locais lineares e angular na fase de predição, enquanto na fase de correção são utilizados os dados do sistema global de câmeras da SSL.

Para a fase de predição do EKF, utilizam-se os dados provenientes dos encoders acoplados às rodas, transformando a leitura nas velocidades locais X e Y e angular, como descrito na Equação 59, que descreve o vetor das variáveis de controle do sistema nesse cenário. Os estados do sistema são atualizados conforme a Equação 60.

$$\mathbf{u}_k = \begin{bmatrix} v_{x,enc} \\ v_{y,enc} \\ \omega_{enc} \end{bmatrix} \quad (59)$$

$$\begin{cases} x_k = x_{k-1} + V_{x,G} \times dt \\ y_k = y_{k-1} + V_{y,G} \times dt \\ \theta_k = \theta_{k-1} + \omega_{enc,k} \Delta t \\ v_{x,k} = v_{x,enc,k-1} \\ v_{y,k} = v_{y,enc,k-1} \end{cases} \quad (60)$$

O cálculo das velocidades globais desse cenário segue o que foi descrito na Equação 35.

A partir disso, a matriz jacobiana G_k dos estados previstos em relação ao estado anterior segue a Equação 61.

$$G_k = \begin{bmatrix} 1 & 0 & \cos(\theta)(v_x \Delta t) - \sin(\theta)(v_y \Delta t) & \sin(\theta) \Delta t & \cos(\theta) \Delta t \\ 0 & 1 & \sin(\theta)(v_x \Delta t) + \cos(\theta)(v_y \Delta t) & -\cos(\theta) \Delta t & \sin(\theta) \Delta t \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (61)$$

A matriz das covariâncias dos ruídos do modelo é definida conforme a Equação 62.

$$R_k = \begin{bmatrix} 5 \cdot 10^{-4} & 0 & 0 & 0 & 0 \\ 0 & 5 \cdot 10^{-4} & 0 & 0 & 0 \\ 0 & 0 & 7 \cdot 10^{-3} & 0 & 0 \\ 0 & 0 & 0 & 5 \cdot 10^{-4} & 0 \\ 0 & 0 & 0 & 0 & 5 \cdot 10^{-4} \end{bmatrix} \quad (62)$$

Na fase de correção do EKF é utilizada a informação da posição global provida pelo sistema global de câmeras da SSL, conforme foi descrito na Seção 6.3.1, especialmente nas Equações 38 e 39, que descrevem o vetor de medições \mathbf{z}_k e a matriz de transformação do estado real no estado observado C'_k , respectivamente. A matriz de covariâncias das medições no caso dos encoders é definida na Seção 6.3.1 na Equação 40.

6.3.7 Cenário 7 - Encoders + IMU

No sétimo, e último, cenário de fusão de sensores, também utiliza os dados providos das leituras dos encoders ligados às rodas, transformando os dados nas velocidades locais lineares e angular do robô, na fase de predição do EKF. Já na fase de correção são utilizados dados do acelerômetro e giroscópio disponíveis no hardware do robô.

Quanto à fase de predição do EKF, segue o que foi descrito na Seção 6.3.6, especialmente nas Equações 59, que define o vetor de controle \mathbf{u}_k , 60, que define a atualização dos estados do sistema na fase de predição, e 61, que define a matriz das jacobianas dos estados em relação ao estado anterior.

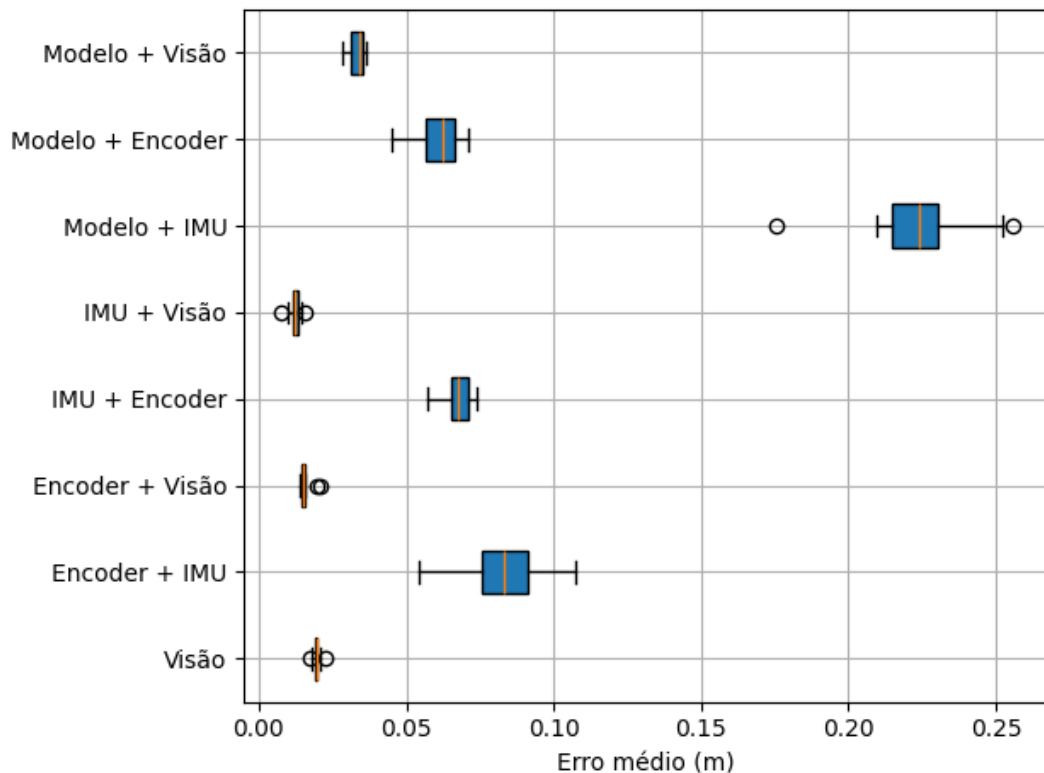
Na fase de correção segue o que foi descrito na Seção 6.3.3, especialmente as Equações 46, que define o vetor de medições do sistema \mathbf{z}_k , 47, que define como as medições são atualizadas conforme as leituras da IMU, 48, que mostra o cálculo das velocidades globais com os dados da IMU, 49, que demonstra o cálculo das acelerações globais, e 50, que define a matriz de transformação do estado real no estado observado C'_k . A matriz de covariâncias das medições no caso dos encoders é definida na Seção 6.3.3 na Equação 51.

6.4 TESTE COM TRAJETO QUADRADO MENOR

Nesta seção será feita a análise do teste do trajeto do quadrado menor em campo. O intuito desse teste é analisar o desempenho das diferentes combinações de sensores propostas para localização do robô numa região do campo de sobreposição de duas câmeras, que acaba levando a certas oscilações.

Na Figura 27 são mostrados os resultados obtidos no trajeto com quadrado menor em relação à distância média entre o que é calculado pelo sistema de localização proposto e o sistema de ground truth desenvolvido. Nela, é possível notar que os cenários que utilizam do sistema global de câmeras da liga conseguem resultados mais satisfatórios, até em comparação com o que é utilizado atualmente pela equipe somente com o sistema global de câmeras (Cenário 8).

Figura 27 – Distância média no teste com trajeto do quadrado menor.



Fonte: o Autor

Além disso, é possível notar que a utilização de sensores na fase de predição do Filtro de Kalman Estendido traz melhores resultados que a utilização do modelo teórico cinemático do robô omnidirecional da categoria SSL, visto que as leituras de velocidade e aceleração adquiridas pelos sensores encoder e IMU provém estimativas melhores do que a utilização da velocidade teórica enviada ao robô.

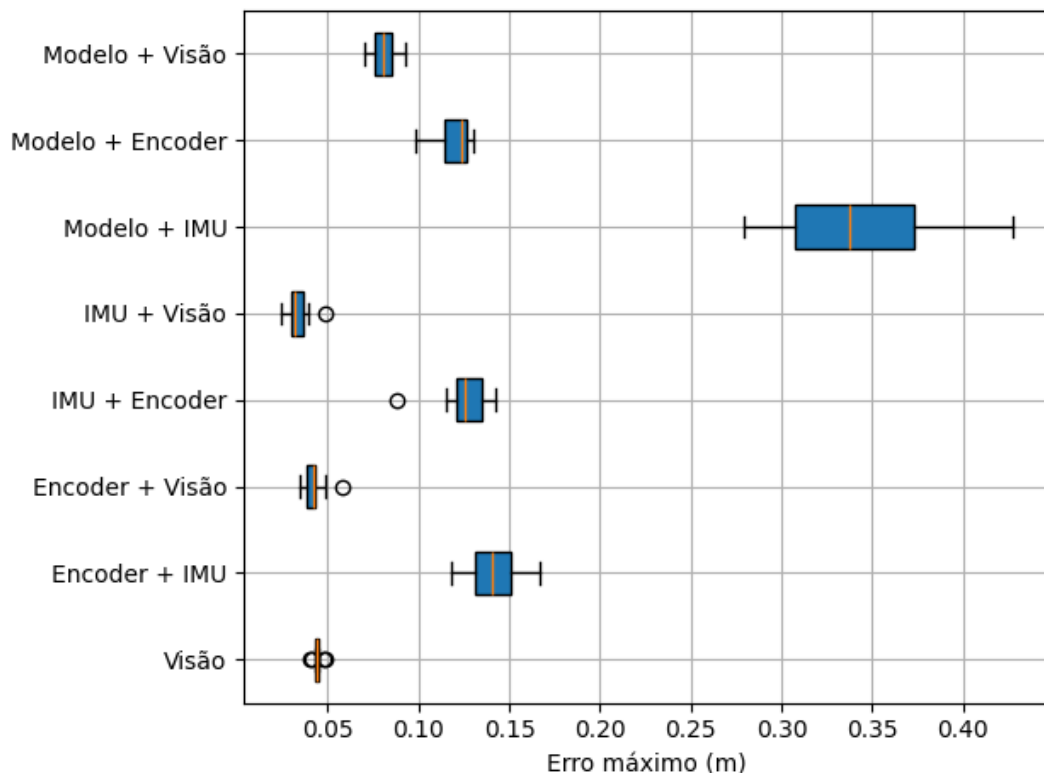
Sobre os cenários que não utilizam do sistema global de câmeras da SSL, é possível notar que os erros médios desses cenários são maiores que 5cm, sendo o Cenário 3 (modelo + IMU) o pior deles, o que indica um problema na utilização do acelerômetro na fase de correção. Apesar disso, para escopos fora da SSL, os erros apresentados nos Cenários 2, 5 e 7 são erros considerados aceitáveis.

Além disso, também a partir da Figura 27, é possível notar que os cenários que utilizam do sistema global de câmeras da SSL possuem menos variações em relação ao erro do que os cenários que utilizam apenas os sensores embarcados no robô. Isso demonstra que o erro médio

ao longo dos testes nas combinações que utilizam a visão é mais estável do que nas outras combinações.

A Figura 28 mostra a distribuição do erro máximo em metros nos testes do trajeto do quadrado menor em cada combinação. O gráfico corrobora com o que foi discutido anteriormente sobre a utilização do sistema global de câmeras na fase de correção do EKF, em que os cenários que utilizam disso possuem os menores erros máximos ao longo dos trajetos. Além disso, também é possível notar que a utilização dos sensores na fase de predição é superior à utilização do modelo cinemático do robô omnidirecional.

Figura 28 – Erro máximo no teste com trajeto do quadrado menor.



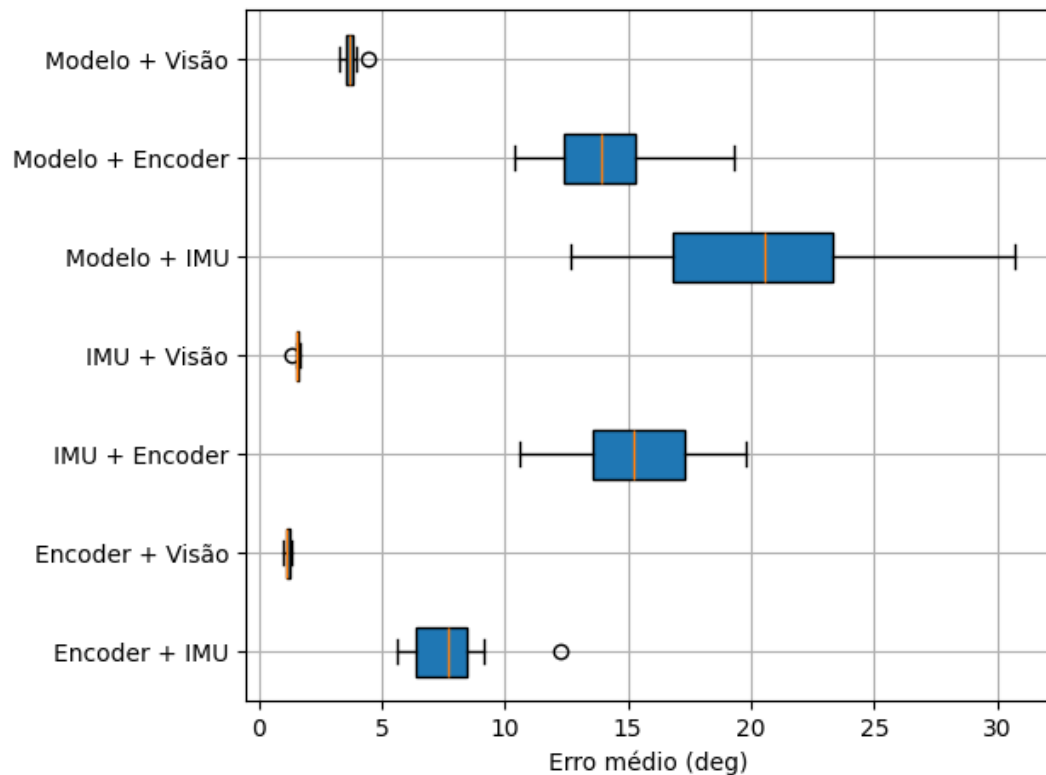
Fonte: o Autor

A Tabela 9 mostra a média dos resultados obtidos em cada cenário

6.5 TESTE COM TRAJETO QUADRADO MAIOR

Nesta seção será feita a análise do teste do trajeto do quadrado maior em campo. O intuito desse teste é analisar o desempenho das diferentes combinações de sensores propostas para localização do robô numa região do campo em que a distorção da imagem do sistema global

Figura 29 – Erro médio de orientação no teste com trajeto do quadrado menor.



Fonte: o Autor

Tabela 9 – Resumo dos resultados obtidos dos cenários no teste do quadrado menor.

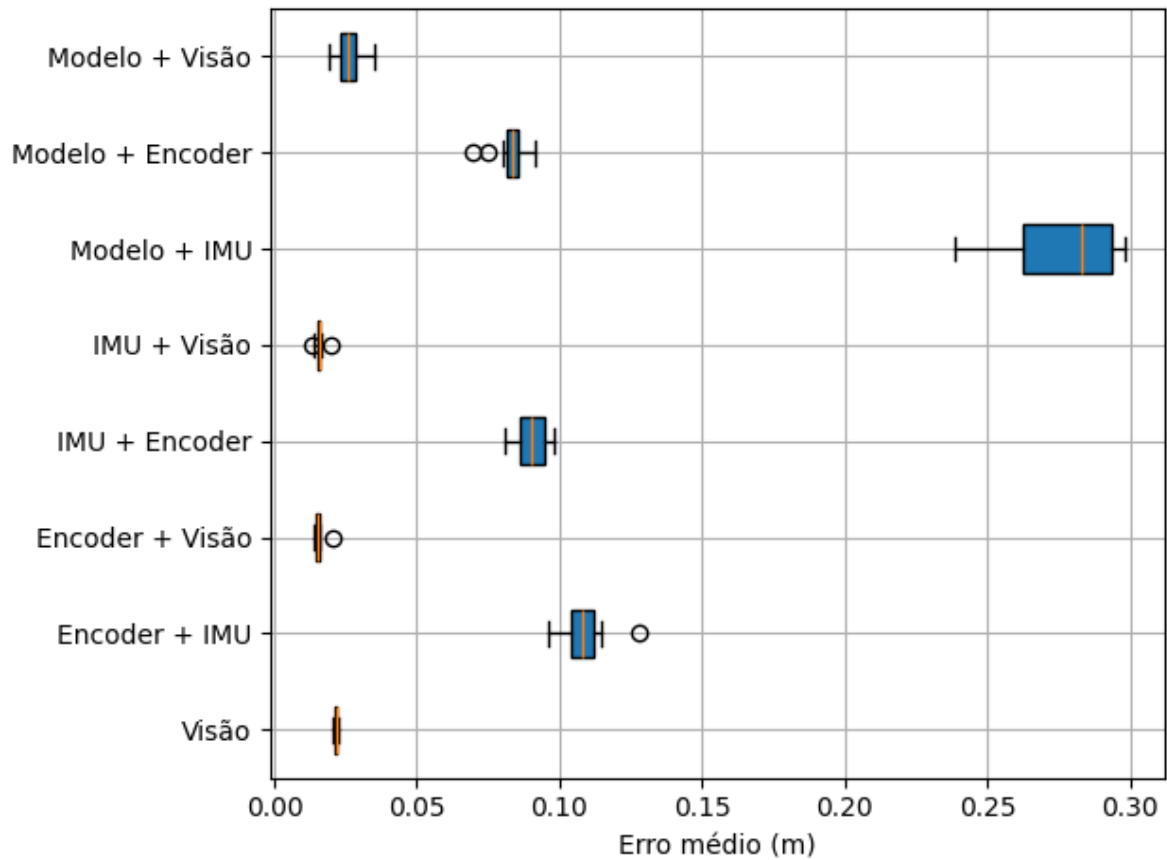
Cenário	Distância - Erro			Orientação - Erro		T_{att} [ms]
	Média [m]	Máximo [m]	Desv. Pad. [m]	Média [deg]	Máximo [deg]	
Modelo + Visão	0,033067	0,080706	0,037178	3,725156	9,467714	3,45970
Modelo + Encoders	0,060239	0,119910	0,061663	13,88283	31,08587	3,58013
Modelo + IMU	0,223313	0,343750	0,199025	20,679713	57,796915	3,59125
IMU + Visão	0,012167	0,033837	0,014022	1,541681	4,788489	3,51205
IMU + Encoders	0,067586	0,125119	0,061642	15,263899	30,86708	3,59578
Encoders + Visão	0,015852	0,042467	0,018073	1,146865	4,148206	3,45462
Encoders + IMU	0,083122	0,140408	0,076724	7,710782	18,938697	3,59147
Visão	0,019585	0,044054	0,018842	-	-	16,2879

de câmeras da SSL é maior, o que pode levar a maiores erros na detecção da posição do robô em campo.

Na Figura 30

Já na Figura 31

Figura 30 – Erro máximo no teste com trajeto do quadrado maior.



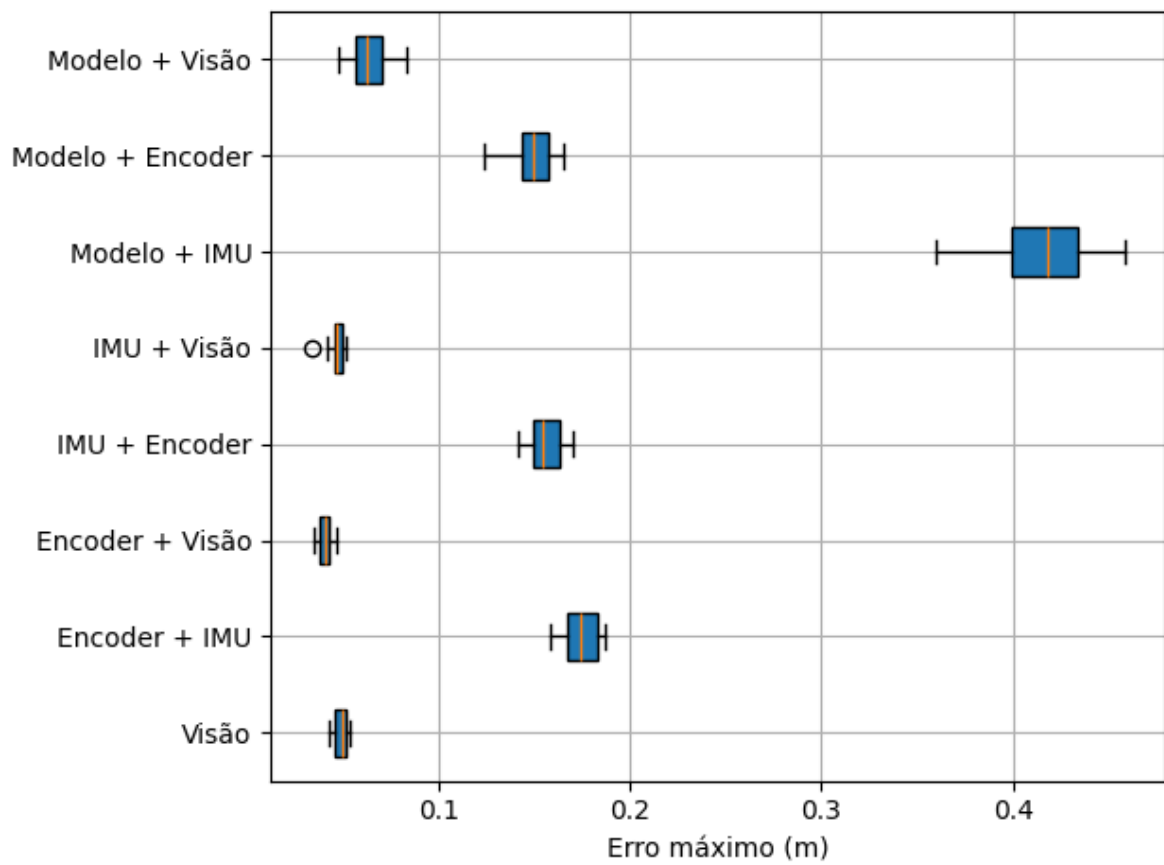
Fonte: o Autor

6.6 CONSIDERAÇÕES FINAIS

A partir do que foi analisado com os gráficos em cada um dos trajetos propostos para análise dos resultados, é possível notar que a utilização de um sensor que te dê informações absolutas do seu robô em relação ao ambiente é essencial para a fusão de sensores. Especialmente na SSL, que a precisão necessária para realização de jogadas é alta por conta da velocidade e tamanho tanto dos robôs, quanto da bola utilizada, utilizar o sistema de visão da liga é essencial para realizar a fusão de sensores para localização de forma embarcada.

Caso não haja a possibilidade de utilização de um sensor que forneça informações globais, como LiDAR, câmera ou GPS, apenas a utilização de IMU e encoder podem trazer resultados aceitáveis (abaixo de 10cm na média de um trajeto).

Figura 31 – Erro máximo no teste com trajeto do quadrado menor.



Fonte: o Autor

Tabela 10 – Resumo dos resultados obtidos dos cenários no teste do quadrado maior.

Cenário	Distância - Erro			Orientação - Erro		T_{att} [ms]
	Média [m]	Máximo [m]	Desv. Pad. [m]	Média [deg]	Máximo [deg]	
Modelo + Visão	0,02586	0,062953	0,028476	3,824521	15,458832	3,47707
Modelo + Encoders	0,082803	0,149021	0,073858	16,23678	38,790607	3,58073
Modelo + IMU	0,277591	0,413911	0,222238	17,841146	45,34764	3,59174
IMU + Visão	0,015806	0,046374	0,019426	1,470293	5,637516	3,52665
IMU + Encoders	0,089906	0,155431	0,07952	18,94846	46,190327	3,59629
Encoders + Visão	0,015518	0,040223	0,01688	1,128545	3,986355	3,46823
Encoders + IMU	0,108431	0,174239	0,093185	8,268624	23,724647	3,59208
Visão	0,021699	0,048287	0,020637	-	-	16,4081

7 CONCLUSÃO

REFERÊNCIAS

AGUIAR, João Victor Lourenço; DA SILVA COSTA, Leonardo; TONIDANDEL, Flavio. Linear Quadratic Regulator Path Tracking for Omnidirectional Robots in High-Dynamic Environments. In: 2023 Latin American Robotics Symposium (LARS), 2023 Brazilian Symposium on Robotics (SBR), and 2023 Workshop on Robotics in Education (WRE). [S.l.: s.n.], 2023. P. 266–271. DOI: 10.1109/LARS/SBR/WRE59448.2023.10332958.

AGUIAR, João Victor Lourenço et al. Reformulação do Sistema de Deslocamento de Robôs Omnidirecionais. **Simpósio de Iniciação Científica, Didática e de Ações Sociais de Extensão da FEI**, p. 48, 2022.

AGUIAR, Luis et al. Kalman filtering for differential drive robots tracking. In: XIII Simposio Brasileiro de Automacao Inteligente,(October 1-4, 2017, Porto Alegre, 1520. [S.l.: s.n.], 2017.

ALATISE, Mary B; HANCKE, Gerhard P. A review on challenges of autonomous mobile robot and sensor fusion methods. **IEEE Access**, IEEE, v. 8, p. 39830–39846, 2020.

ARM. **CMSIS-DSP Library**. Acesso em 4 jan. 2025 as 19:06. Jul. 2024. Disponível em: https://arm-software.github.io/CMSIS_6/main/DSP/index.html.

BAIE, Chuan Yu et al. **2017 Team Description Paper: UBC Thunderbots**. [S.l.: s.n.], 2017. Team Description Paper, RoboCup Small Size League. Disponível em: <https://ssl.robocup.org/team-description-papers/>.

BCC. **Sensors: Technologies and Global Markets**. Acesso em 06 jun. 2024 as 16:18. Mar. 2024. Disponível em: <https://www.bccresearch.com/market-research/instrumentation-and-sensors/sensors-technologies-markets-report.html>.

BEHZAD, Kian et al. **PARSIAN 2019: Extended Team Description Paper**. [S.l.: s.n.], 2019. Extended Team Description Paper, RoboCup Small Size League. Disponível em: <https://ssl.robocup.org/team-description-papers/>.

BENARBIA, Taha; KYAMAKYA, Kyandoghere. A Literature Review of Drone-Based Package Delivery Logistics Systems and Their Implementation Feasibility. **Sustainability**, v. 14, n. 1, 2022. ISSN 2071-1050. DOI: 10.3390/su14010360. Disponível em: <https://www.mdpi.com/2071-1050/14/1/360>.

BORENSTEIN, Johann et al. Mobile robot positioning: Sensors and techniques. **Journal of robotic systems**, Wiley Online Library, v. 14, n. 4, p. 231–249, 1997.

BROOKS, Richard R; IYENGAR, Sundararaja S. **Multi-sensor fusion: fundamentals and applications with software**. [S.l.]: Prentice-Hall, Inc., 1998.

CASTANEDO, Federico et al. A review of data fusion techniques. **The scientific world journal**, Hindawi, v. 2013, 2013.

CHEN, Lingyun et al. **ZJUNlict Extended Team Description Paper for RoboCup 2018**. [S.l.: s.n.], 2018. Extended Team Description Paper, RoboCup Small Size League. Disponível em: <https://ssl.robocup.org/team-description-papers/>.

CHURCHLEY, Scott et al. **2015 Team Description Paper: UBC Thunderbots**. [S.l.: s.n.], 2015. Team Description Paper, RoboCup Small Size League. Disponível em: <https://ssl.robocup.org/team-description-papers/>.

COITO, Francisco et al. Tracking a Mobile Robot Position Using Vision and Inertial Sensor. In: CAMARINHA-MATOS, Luis M.; BARRENTO, Nuno S.; MENDONÇA, Ricardo (Ed.). **Technological Innovation for Collective Awareness Systems**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. P. 201–208. ISBN 978-3-642-54734-8.

COSTA, Leonardo da Silva; TONIDANDEL, Flavio. **Análise de técnicas de navegação de robôs autônomos em ambientes dinâmicos e incertos**. Set. 2023. Dissertação de Mestrado – Centro Universitário FEI, São Bernardo do Campo. <https://repositorio.fei.edu.br/handle/FEI/5194>.

DADAFSHAR, Majid. Accelerometer and gyroscopes sensors: operation, sensing, and applications. **Maxim Integrated [online]**, 2014.

DASARATHY, Belur V. Sensor fusion potential exploitation-innovative architectures and illustrative applications. **Proceedings of the IEEE**, IEEE, v. 85, n. 1, p. 24–38, 1997.

DINGMAN, Jeremy. **What Is A Gyroscope?** Acesso em 16 mar. 2024 as 15:32. Set. 2020. Disponível em: https://aerospace.honeywell.com/us/en/about-us/blogs/what-is-a-gyroscope?utm_source=google%5C&utm_medium=cpc%5C&utm_campaign=23-aero-ww-dsa-blogs%5C&utm_content=dyn-en-lp%5C&gad_source=1%5C&gclid=Cj0KCQjwwMqvBhCtARIsAIXsZpb80B5mORQKLaGBbWRu0O5pSudNu1Xfje7I3ofgDTiBVRPTMROtyR4aAiK-EALw_wcB.

DOORNKAMP, Cas et al. RoboTeam Twente 2018 Team Description Paper. In.

DURRANT-WHYTE, Hugh F. Sensor models and multisensor integration. **The international journal of robotics research**, Sage Publications Sage CA: Thousand Oaks, CA, v. 7, n. 6, p. 97–113, 1988.

DYER, Benjamin M. et al. Filtering Strategies for State Estimation of Omniwheel Robots. In: 2020 IEEE International Conference on Mechatronics and Automation (ICMA). [S.l.: s.n.], 2020. P. 186–191. DOI: 10.1109/ICMA49215.2020.9233826.

ELMENREICH, Wilfried. An introduction to sensor fusion. **Vienna University of Technology, Austria**, v. 502, p. 1–28, 2002.

EMAN, Alhamdi; RAMDANE, Hedjar. Mobile robot localization using extended Kalman filter. In: IEEE. 2020 3rd International Conference on Computer Applications & Information Security (ICCAIS). [S.l.: s.n.], 2020. P. 1–5.

FUNG, Man Lok; CHEN, Michael ZQ; CHEN, Yong Hua. Sensor fusion: A review of methods and applications. In: IEEE. 2017 29th Chinese Control And Decision Conference (CCDC). [S.l.: s.n.], 2017. P. 3853–3860.

GONÇALVES, José; LIMA, José; COSTA, Paulo Gomes da. Real-time tracking of an omnidirectional robot: an extended kalman filter approach. In: INTERNATIONAL Conference on Informatics in Control, Automation and Robotics. [S.l.: s.n.], 2008.

HALL, David L; LLINAS, James. An introduction to multisensor data fusion. **Proceedings of the IEEE**, IEEE, v. 85, n. 1, p. 6–23, 1997.

HASSAN, Mahmood ul; BAO, Qilian. A field calibration method for low-cost mems accelerometer based on the generalized nonlinear least square method. **Multiscale Science and Engineering**, Springer, v. 2, p. 135–142, 2020.

HASSANEIN, Allam Shehata et al. A survey on Hough transform, theory, techniques and applications. **arXiv preprint arXiv:1502.02160**, 2015.

HUANG, Zheyuan et al. ZJUNliet Extended Team Description Paper for Robocup 2020. In.

ISMAIL, Muhammad Azhar; PURWANTO, Djoko; ARIFIN, Achmad. Soccer Robot Localization Based on Sensor Fusion From Odometry and Omnivision. In: IEEE. 2022 International Seminar on Intelligent Technology and Its Applications (ISITIA). [S.l.: s.n.], 2022. P. 273–278.

JAIN, A.; ZHANG, L.; JIANG, L. **High-Fidelity Sensor Calibration for Autonomous Vehicles**. Acesso em 06 jun. 2024 as 16:31. Ago. 2019. Disponível em: <https://medium.com/wovenplanetlevel5/high-fidelity-sensor-calibration-for-autonomous-vehicles-6af06eba4c26>.

KHODARAHMI, Masoud; MAIHAMI, Vafa. A review on Kalman filter models. **Archives of Computational Methods in Engineering**, Springer, v. 30, n. 1, p. 727–747, 2023.

KOROTAJ, Blaž; NOVOSELENIK, Branimir; BAOTIĆ, Mato. Kalman filter based sensor fusion for omnidirectional mechatronic system. In: IEEE. 2021 International Conference on Electrical Drives & Power Electronics (EDPE). [S.l.: s.n.], 2021. P. 183–188.

LI, Xianzhi et al. Data fusion for intelligent crowd monitoring and management systems: A survey. **IEEE Access**, IEEE, v. 9, p. 47069–47083, 2021.

LOGITECH. **BRIO ULTRA HD BUSINESS WEBCAM**. [S.l.], mar. 2021.

LOTUFO, Roberto A et al. Morphological image processing. In: MICROSCOPE image processing. [S.l.]: Elsevier, 2023. P. 75–117.

LUIZ R., José. **Como funciona um Encoder**. Acesso em 26 mar. 2024 as 17:13. Mar. 2021. Disponível em: <https://como-funciona.co/un-encoder/>.

LUO, Ren C; CHANG, Chih Chia; LAI, Chun Chi. Multisensor fusion and integration: Theories, applications, and its perspectives. **IEEE Sensors Journal**, IEEE, v. 11, n. 12, p. 3122–3138, 2011.

LUO, Ren C; KAY, Michael G. A tutorial on multisensor integration and fusion. In: IEEE. [PROCEEDINGS] IECON'90: 16th Annual Conference of IEEE Industrial Electronics Society. [S.l.: s.n.], 1990. P. 707–722.

LV, Jiajun et al. Targetless calibration of lidar-imu system based on continuous-time batch estimation. In: IEEE. 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). [S.l.: s.n.], 2020. P. 9968–9975.

MACKWORTH, Alan K. On Seeing Robots, 1982.

MÁRTON, Lőrinc; GYÖRGY, Katalin. Two-Stage Kalman Filtering for Indoor Localization of Omnidirectional Robots. **Electrical and Mechanical Engineering**, v. 5, p. 44–60, 2013.

MAXON GROUP. **EC 45 brushless 50W with sensor hall**. [S.l.], abr. 2019.

MELO, João G; BARROS, Edna. An embedded monocular vision approach for ground-aware objects detection and position estimation. In: ROBOT World Cup. [S.l.]: Springer, 2022. P. 100–111.

MENDES, Jorge Miguel Ferreira da Silva. Percepção visual semântica de uma vinha para auxílio à navegação de robôs, 2017.

MENDES JR., José Jair Alves et al. Sensor Fusion and Smart Sensor in Sports and Biomedical Applications. **Sensors**, v. 16, n. 10, 2016. ISSN 1424-8220. DOI: 10.3390/s16101569. Disponível em: <https://www.mdpi.com/1424-8220/16/10/1569>.

MENEZES FILHO, Rogério P et al. Triaxial accelerometer calibration using an extended two-step methodology. In: IEEE. 2020 Latin American Robotics Symposium (LARS), 2020 Brazilian Symposium on Robotics (SBR) and 2020 Workshop on Robotics in Education (WRE). [S.l.: s.n.], 2020. P. 1–6.

MUKHOPADHYAY, Priyanka; CHAUDHURI, Bidyut B. A survey of Hough Transform. **Pattern Recognition**, Elsevier, v. 48, n. 3, p. 993–1010, 2015.

NAEINI, Meisam Kasaeian et al. **MRL Extended Team Description 2020**. [S.l.: s.n.], 2020. Extended Team Description Paper, RoboCup Small Size League. Disponível em: <https://ssl.robocup.org/team-description-papers/>.

NAGLA, KS; UDDIN, Moin; SINGH, Dilbag. Multisensor data fusion and integration for mobile robots: A review. **IAES International Journal of Robotics and Automation**, IAES Institute of Advanced Engineering e Science, v. 3, n. 2, p. 131, 2014.

NEGENBORN, R.R. **Robot Localization and Kalman Filters. On finding your position in a noisy world**. Jan. 2003. Tese (Doutorado).

NIETO, Juan; BAILEY, Tim; NEBOT, Eduardo. Recursive scan-matching SLAM. **Robotics and Autonomous Systems**, v. 55, n. 1, p. 39–49, 2007. Simultaneous Localisation and Map Building. ISSN 0921-8890. DOI: <https://doi.org/10.1016/j.robot.2006.06.008>. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0921889006001461>.

NISTLER, Jonathan R.; SELEKWA, Majura F. Gravity compensation in accelerometer measurements for robot navigation on inclined surfaces. **Procedia Computer Science**, v. 6, p. 413–418, 2011. Complex adaptive systems. ISSN 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2011.08.077>. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1877050911005424>.

NORDIC SEMICONDUCTOR. **nRF24L01 Single Chip 2.4GHz Transceiver - Product Specification**. [S.l.], jul. 2007. Rev. 2.0.

OLIVEIRA, Bruno Queres de et al. Tipos e aplicações de sensores na robótica. **Caderno de Graduação-Ciências Exatas e Tecnológicas-UNIT-ALAGOAS**, v. 4, n. 1, p. 223–223, 2017.

PACHECO, Marcus Vinicius O.; SILVA, Felipe O.; FARRELL, Jay A. GPS-Aided Odometry Navigation for IARs: Comparison Between Loosely and Tightly Coupled Integrations Under Restricted Satellite Visibility Conditions. In: 2023 Latin American Robotics Symposium (LARS), 2023 Brazilian Symposium on Robotics (SBR), and 2023 Workshop on Robotics in Education (WRE). [S.l.: s.n.], 2023. P. 278–283. DOI: 10.1109/LARS/SBR/WRE59448.2023.10333060.

PANIGRAHI, Prabin Kumar; BISOY, Sukant Kishoro. Localization strategies for autonomous mobile robots: A review. **Journal of King Saud University - Computer and Information Sciences**, v. 34, 8, Part B, p. 6019–6039, 2022. ISSN 1319-1578. DOI: <https://doi.org/10.1016/j.jksuci.2021.02.015>. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1319157821000550>.

PAPAFOTIS, Konstantinos; SOTIRIADIS, Paul P. Exploring the Importance of Sensors' Calibration in Inertial Navigation Systems. In: 2020 IEEE International Symposium on Circuits and Systems (ISCAS). [S.l.: s.n.], 2020. P. 1–4. DOI: 10.1109/ISCAS45731.2020.9181212.

PAREKH, Darsh et al. A review on autonomous vehicles: Progress, methods and challenges. **Electronics**, MDPI, v. 11, n. 14, p. 2162, 2022.

PASSARO, Vittorio M. N. et al. Gyroscope Technology and Applications: A Review in the Industrial Perspective. **Sensors**, v. 17, n. 10, 2017. ISSN 1424-8220. DOI: 10.3390/s17102284. Disponível em: <https://www.mdpi.com/1424-8220/17/10/2284>.

PAULI, Guilherme; TONIDANDEL, Flavio. Solving the Time Lapse from Vision System in a Robot Soccer Game Using Kalman Filter. In: 2019 Latin American Robotics Symposium (LARS), 2019 Brazilian Symposium on Robotics (SBR) and 2019 Workshop on Robotics in Education (WRE). [S.l.: s.n.], 2019. P. 7–12. DOI: 10.1109/LARS-SBR-WRE48964.2019.00010.

PERSSON, Anders. How do we understand the Coriolis force? **Bulletin of the American Meteorological Society**, American Meteorological Society, v. 79, n. 7, p. 1373–1386, 1998.

RAJ, Ravi; KOS, Andrzej. A Comprehensive Study of Mobile Robot: History, Developments, Applications, and Future Research Perspectives. **Applied Sciences**, v. 12, n. 14, 2022. ISSN 2076-3417. DOI: 10.3390/app12146951. Disponível em: <https://www.mdpi.com/2076-3417/12/14/6951>.

RIGATOS, Gerasimos G. Extended Kalman and particle filtering for sensor fusion in motion control of mobile robots. **Mathematics and computers in simulation**, Elsevier, v. 81, n. 3, p. 590–607, 2010.

ROBOCUP. **A Brief History of RoboCup**. Acesso em 13 mar. 2024 as 14:00. 2020. Disponível em: https://www.robocup.org/a_brief_history_of_robocup.

ROBOCUP-SSL. **About the Small Size League**. Acesso em 28 abr. 2024 as 18:18. Set. 2019. Disponível em: <https://ssl.robocup.org/about/>.

ROBOCUP-SSL. **Página das regras da SSL**. Acesso em 1 mar. 2024 as 21:33. Set. 2019. Disponível em: <https://ssl.robocup.org/rules/>.

ROSTAMI, Vahid et al. Localization and Navigation Omni-directional Robots based on Sensors Fusion and Particle Filter. In: IEEE. 2018 9th Conference on Artificial Intelligence and Robotics and 2nd Asia-Pacific International Symposium. [S.l.: s.n.], 2018. P. 65–73.

RYLL, Andre et al. **TIGERS Mannheim (Team Interacting and Game Evolving Robots) Team Description for RoboCup 2013**. [S.l.: s.n.], 2013. Team Description Paper, RoboCup Small Size League. Disponível em: <https://ssl.robocup.org/team-description-papers/>.

RYLL, Andre et al. TIGERS Mannheim Team Description for RoboCup 2013. In. Disponível em: <https://api.semanticscholar.org/CorpusID:202684288>.

SAE INTERNATIONAL. On-Board System Requirements for V2V Safety Communications. **SAE J2945/1**, 2016.

SALEHI, Ali et al. **Immortals 2023 Extended Team Description Paper**. [S.l.: s.n.], 2023. Extended Team Description Paper, RoboCup Small Size League. Disponível em: <https://ssl.robocup.org/team-description-papers/>.

SANTINI, A; NICOSIA, S; NANNI, V. Trajectory estimation and correction for a wheeled mobile robot using heterogeneous sensors and Kalman filter. **IFAC Proceedings Volumes**, Elsevier, v. 30, n. 20, p. 11–16, 1997.

SASIADEK, J.Z.; HARTANA, P. Sensor data fusion using Kalman filter. In: PROCEEDINGS of the Third International Conference on Information Fusion. [S.l.: s.n.], 2000. v. 2, wed5/19–wed5/25 vol.2. DOI: 10.1109/IFIC.2000.859866.

SIEGWART, R.; NOURBAKHS, I.R.; SCARAMUZZA, D. **Introduction to Autonomous Mobile Robots, second edition**. [S.l.]: MIT Press, 2011. (Intelligent Robotics and Autonomous Agents series). ISBN 9780262015356. Disponível em: <https://books.google.com.br/books?id=4of6AQAAQBAJ>.

SINGHAL, Amit. Issues in autonomous mobile robot navigation. **Computer Science Dept, U. of Rochester**, Citeseer, p. 74, 1997.

SLAMTEC. **RPLIDAR A1 - Low Cost Degree Laser Range Scanner**. [S.l.], jul. 2016. Rev. 10.

SLAMTEC. **RPLIDAR S1 - Low Cost 360 Degree Laser Range Scanner**. [S.l.], mar. 2021. Rev. 1.7.

SLAMTEC. **RPLIDAR S1 - Low Cost Degree Laser Range Scanner**. [S.l.], jan. 2018. Rev. 10.

STATISTA. **Robotics - Worldwide**. Acesso em 09 jun. 2024 as 22:54. Ago. 2023. Disponível em: <https://www.statista.com/outlook/tmo/robotics/worldwide?currency=USD#revenue>.

STMICROELECTRONICS. **Discovery kit with STM32F411VE MCU - User manual**. [S.l.], set. 2020. Rev. 2.

STMICROELECTRONICS. **MEMS motion sensor: three-axis digital output gyroscope**. [S.l.], abr. 2015. Rev. 2.

STMICROELECTRONICS. **Repositório do driver do componente I3G4250D**. Acesso em 20 mar. 2024 as 20:07. Dez. 2023. Disponível em: <https://github.com/STMicroelectronics/stm32-i3g4250d>.

STMICROELECTRONICS. **Repositório do driver do componente LSM303AGR**. Acesso em 20 mar. 2024 as 18:58. Dez. 2023. Disponível em: <https://github.com/STMicroelectronics/stm32-lsm303agr>.

STMICROELECTRONICS. **Ultra-compact high-performance eCompass module: 3D accelerometer and 3D magnetometer**. [S.l.], ago. 2022. Rev. 11.

SULIMAN, Caius; CRUCERU, Cristina; MOLDOVEANU, Florin. Mobile robot position estimation using the Kalman filter. **Acta Marisiensis. Seria Technologica**, De Gruyter Poland, v. 6, p. 75, 2009.

TECHNEXION. **Vision-guided Robotics - How Cameras are Transforming Robotics**. Acesso em 19 mar. 2024 as 20:14. Abr. 2023. Disponível em: <https://www.technexion.com/resources/vision-guided-robotics-how-cameras-are-transforming-robotics/#:~:text=Role%20of%20a%20Camera%20in%20Perception%20Enhancement&text=Robots%20can%20learn%20much%20about,their%20environment%20in%20greater%20depth..>

TECHNOLOGIES, Allied Vision. **Technical Manual**. [S.l.], abr. 2011. V4.4.2.

THRUN, Sebastian. Probabilistic robotics. **Communications of the ACM**, ACM New York, NY, USA, v. 45, n. 3, p. 52–57, 2002.

THRUN, Sebastian et al. Robust Monte Carlo localization for mobile robots. **Artificial intelligence**, Elsevier, v. 128, n. 1-2, p. 99–141, 2001.

URREA, Claudio; AGRAMONTE, Rayko. Kalman filter: historical overview and review of its use in robotics 60 years after its creation. **Journal of Sensors**, Wiley Online Library, v. 2021, n. 1, p. 9674015, 2021.

US DIGITAL. **E4T Miniature Optical Kit Encoder**. [S.l.], jan. 2024.

VAKIL, Asad et al. A survey of multimodal sensor fusion for passive RF and EO information integration. **IEEE Aerospace and Electronic Systems Magazine**, IEEE, v. 36, n. 7, p. 44–61, 2021.

VISSER, A; GROEN, FCA. Organisation and design of autonomous systems. Textbook, Faculty of Mathematics. **Computer Science, Physics and Astronomy, University of Amsterdam, Kruislaan**, v. 403, 1999.

WANG, Li et al. An Efficient Calibration Method for Triaxial Gyroscope. **IEEE Sensors Journal**, v. 21, n. 18, p. 19896–19903, 2021. DOI: 10.1109/JSEN.2021.3100589.

WELCH, Greg; BISHOP, Gary et al. An introduction to the Kalman filter. Chapel Hill, NC, USA, 1995.

WHITE, Franklin E et al. Data fusion lexicon. **Joint Directors of Laboratories, Technical Panel for C**, v. 3, p. 19, 1991.

ZIBETTI, Andre. **Distribuição Normal (Gaussiana)**. Acesso em 23 jun. 2024 as 15:45. Jul. 2022. Disponível em: <https://www.inf.ufsc.br/~andre.zibetti/probabilidade/normal.html>.

ZICKLER, Stefan et al. SSL-Vision: The Shared Vision System for the RoboCup Small Size League. In: BALTES, Jacky et al. (Ed.). **RoboCup 2009: Robot Soccer World Cup XIII**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010. P. 425–436. ISBN 978-3-642-11876-0.

ZOLANVARI, Alireza et al. **PARSIAN Team Description for RoboCup 2015**. [S.l.: s.n.], 2015. Team Description Paper, RoboCup Small Size League. Disponível em: <https://ssl.robocup.org/team-description-papers/>.