

CENTRO UNIVERSITÁRIO FEI  
JOÃO VICTOR LOURENÇO AGUIAR

**FUSÃO DE SENSORES COM FILTRO DE KALMAN PARA ESTIMATIVA DE  
POSIÇÃO DE ROBÔS MÓVEIS**

São Bernardo do Campo

2025

JOÃO VICTOR LOURENÇO AGUIAR

**FUSÃO DE SENSORES COM FILTRO DE KALMAN PARA ESTIMATIVA DE  
POSIÇÃO DE ROBÔS MÓVEIS**

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. . . . .	14
Figura 2	– Imagem de um robô SSL da equipe RoboFEI. . . . .	15
Figura 3	– Comparação da informação recebida pelo sistema de visão global e pelo giroscópio. . . . .	16
Figura 4	– Padrões de cores para identificação dos robôs. . . . .	19
Figura 5	– Ilustração da montagem e funcionamento de um encoder óptico. . . . .	21
Figura 6	– Diagrama representando a diferença entre as fusões complementar, competitiva e cooperativa. . . . .	36
Figura 7	– Diagrama representando a diferença entre as fusões baseadas no nível de abstração dos dados. . . . .	39
Figura 8	– Diagrama relacionando as classificações três níveis e Dasarathy. . . . .	39
Figura 9	– Exemplo de distribuição gaussiana. . . . .	43
Figura 10	– Ciclo do filtro de Kalman discreto. . . . .	44
Figura 11	– Exemplo de predição e atualização das covariâncias do filtro de Kalman. .	47
Figura 12	– Exemplo de predição e atualização das covariâncias do filtro de Kalman. .	49
Figura 13	– Visão geral da fusão de sensores da odometria com o sistema de visão da categoria MSL. . . . .	54
Figura 14	– Placa de desenvolvimento STM32F411E-Disco utilizada. . . . .	61
Figura 15	– Representação do posicionamento dos lasers em campo. . . . .	65
Figura 16	– Ilustração do primeiro teste. . . . .	66
Figura 17	– Ilustração do segundo teste. . . . .	66
Figura 18	– Ilustração do terceiro teste. . . . .	67
Figura 19	– Arquitetura do software. . . . .	69
Figura 20	– Duas etapas da arquitetura desenvolvida. . . . .	71
Figura 21	– Cenários de testes do sistema de ground truth. . . . .	72
Figura 22	– Boxplot dos erros por combinação de sensores . . . . .	78

## LISTA DE TABELAS

Tabela 1	–	Combinações dos sensores na predição e correção do Filtro de Kalman. . . . .	62
Tabela 2	–	Especificações de hardware do notebook. . . . .	63
Tabela 3	–	Especificações dos LiDARs utilizados. . . . .	64
Tabela 4	–	Resultados alcançados para o eixo X no campo de teste. . . . .	73
Tabela 5	–	Resultados alcançados para o eixo Y no campo de teste. . . . .	74
Tabela 6	–	Erro e desvio padrão médios nos cenários. . . . .	74
Tabela 7	–	Resultados dos quadrantes para o eixo X. . . . .	75
Tabela 8	–	Resultados dos quadrantes para o eixo Y. . . . .	75

## LISTA DE ALGORITMOS

Algoritmo 1 – Filtro de Kalman linear . . . . .	47
Algoritmo 2 – Filtro de Kalman estendido . . . . .	50

## LISTA DE ABREVIATURAS

DC	<i>Direct Current</i>
EKF	Filtro de Kalman Estendido
FP	Filtro de Partículas
FSEV	Filtro Suave de Estrutura Variável
I <sup>2</sup> 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

<b>1</b>	<b>INTRODUÇÃO</b>	10
1.1	OBJETIVO	11
1.2	JUSTIFICATIVA	11
1.3	ESTRUTURA DA DISSERTAÇÃO	12
<b>2</b>	<b>FUTEBOL DE ROBÔS</b>	13
2.1	<i>SMALL SIZE LEAGUE</i>	13
<b>2.1.1</b>	<b>Sistema de visão da categoria SSL</b>	15
2.2	SENSORES	17
<b>2.2.1</b>	<b>Tipos de sensores</b>	17
<b>2.2.2</b>	<b>Sistema de câmeras</b>	18
<b>2.2.3</b>	<b>Encoder</b>	20
<b>2.2.4</b>	<b>Giroscópio</b>	21
<b>2.2.5</b>	<b>Acelerômetro</b>	22
<b>2.2.6</b>	<i>Inertial Measurement Unit (IMU)</i>	23
<b>2.2.7</b>	<b>Calibração dos sensores</b>	24
<b>2.2.7.1</b>	<i>Calibração do acelerômetro</i>	25
<b>2.2.7.2</b>	<i>Calibração do giroscópio</i>	26
<b>3</b>	<b>LOCALIZAÇÃO DE ROBÔS MÓVEIS</b>	28
3.1	O PROBLEMA DA LOCALIZAÇÃO	28
3.2	OS DESAFIOS DA LOCALIZAÇÃO	29
3.3	INFORMAÇÃO DISPONÍVEL	30
3.4	FUSÃO DE SENSORES	31
<b>3.4.1</b>	<b>Vantagens e Desvantagens</b>	32
<b>3.4.2</b>	<b>Classificação de técnicas</b>	35
<b>3.4.2.1</b>	<i>Classificação baseada na relação entre as fontes de dados</i>	35
<b>3.4.2.2</b>	<i>Classificação em três níveis</i>	37
<b>3.4.2.3</b>	<i>Classificação baseada na entrada e saída do sistema</i>	37
3.5	FILTRO DE KALMAN	38
<b>3.5.1</b>	<b>Introdução</b>	40
<b>3.5.2</b>	<b>Premissas</b>	41
<b>3.5.2.1</b>	<i>Sistema dinâmico linear</i>	41

3.5.2.2	<i>Características do ruído</i>	42
3.5.2.3	<i>Processo a ser estimado</i>	43
3.5.3	<b>Equações</b>	44
3.5.3.1	<i>Predição</i>	45
3.5.3.2	<i>Correção</i>	45
3.6	FILTRO DE KALMAN ESTENDIDO	48
4	<b>TRABALHOS RELACIONADOS</b>	51
5	<b>METODOLOGIA</b>	60
5.1	DOMÍNIO DE TESTES	60
5.2	COMBINAÇÃO DOS SENSORES	61
5.3	CALIBRAÇÃO DOS SENSORES	62
5.4	IMPLEMENTAÇÃO	63
5.5	<i>GROUND TRUTH</i>	64
5.6	TESTES	64
5.7	AVALIAÇÃO	67
6	<b>RESULTADOS</b>	69
6.1	AQUISIÇÃO DO GROUND TRUTH	69
6.1.1	<b>Técnica utilizada e visão geral do sistema</b>	69
6.1.2	<b>Resultados do sistema desenvolvido</b>	72
6.2	IMPLEMENTAÇÃO DOS CENÁRIOS	76
6.2.1	<b>Predição com modelo</b>	76
6.2.2	<b>Predição com IMU</b>	77
6.2.3	<b>Predição com encoders</b>	77
6.3	QUADRADO MENOR	78

## 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 **robotics\_market**, 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 (**su14010360**) e carros autônomos (**parekh2022review**)

A característica que difere os robôs móveis de robôs industriais (basicamente robôs manipuladores, como os utilizados para montagem de carros na indústria automotiva, por exemplo) 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 (**app12146951**).

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 *tracking* da posição de robôs móveis dentro de um mapa conhecido.

De acordo com **PANIGRAHI20226019**, o objetivo do *tracking* 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 **PANIGRAHI20226019**, a localização pode ser dividida em duas etapas: a predição e a percepção. Na etapa de predição o robô faz o *tracking* 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 (s16101569). 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.

## 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 **alalise2020review**, 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.

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 realizados 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.

## 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* (**OnSeeingRobots**). 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**).

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**).

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**). Além disso, os robôs da liga são omnidirecionais, o que proporciona jogos dinâmicos com jogadas imprevisíveis.

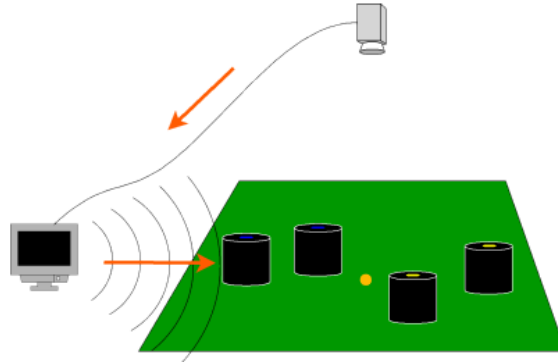
Os jogos desta categoria são destacados por conta da alta velocidade tanto dos robôs, que podem chegar até 4m/s, quanto da bola utilizada, que pode chegar até 6.5m/s, mas também pela quantidade de robôs numa partida, que acontece entre dois times que podem ter de 6 a 11 robôs num campo de 9m X 6m a 12m X 9m, dependendo da divisão da partida (**10332958**).

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  $\theta$  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 **about-ssl**

Além disso, os robôs possuem limitações de tamanho, i.e., eles devem caber num diâmetro de 180mm e possuir uma altura máxima de 150mm (**rules**). 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 (**aguiarreformulacao**).

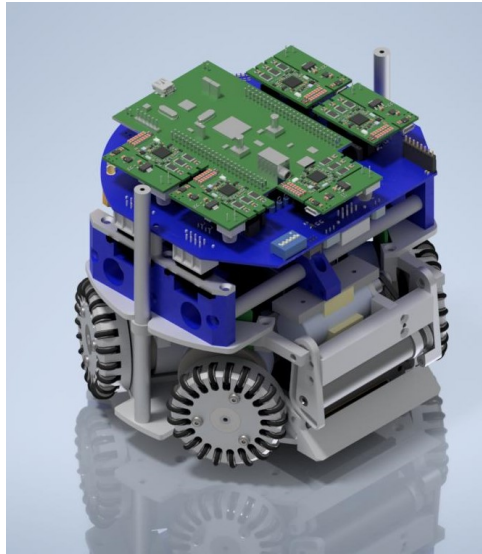
Com esse sistema de deslocamento omnidirecional, o robô torna-se um sistema holonômico, i.e., o robô possui controle sobre todos os graus de liberdade da sua movimentação, ou seja, a rotação dele não interfere na translação, o que torna a SSL uma liga muito dinâmica e imprevisível. A Figura 2 mostra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

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

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  $\sigma$ .

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



Fonte: Autor

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

### 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 **tdpZJUNlict2020**, 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.

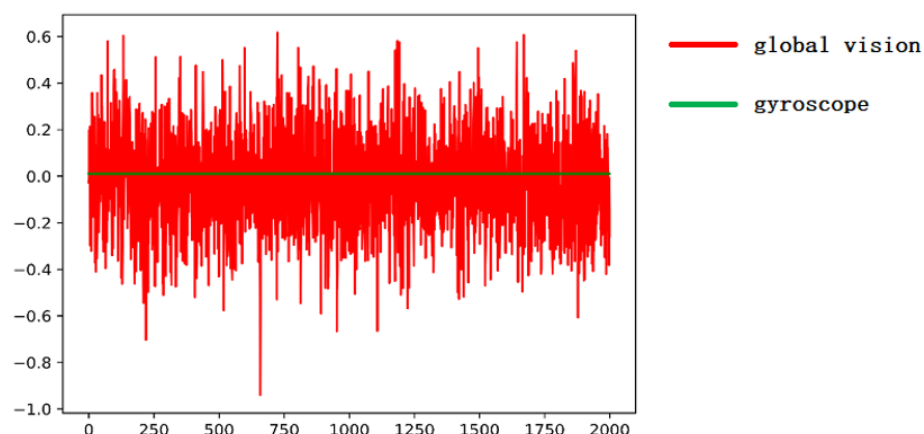


O problema de alta latência de envio das imagens compromete a realização de jogadas durante as partidas e gera dificuldades no controle de posicionamento dos robôs. Assim, nota-se uma necessidade de aplicação de outras tecnologias para resolver esse problema de posicionamento, como utilização de outros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

Segundo **tdpZJUNlict2020**, 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, consequentemente, 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.

Figura 3 – Comparação da informação recebida pelo sistema de visão global e pelo giroscópio.



Fonte: Retirado de **tdpZJUNlict2020**

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 (tdptwente2018). 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 posição embarcado que realize estimativas de posicionamento dele por conta do tempo de atualização dos dados das câmeras ser relativamente alto para a aplicação.

## 2.2 SENSORES

A utilização de sensores é parte essencial para o funcionamento correto de um robô. Segundo **de2017tipos**, 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 **de2017tipos**, 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 **sensorFusionKalmanFilter**, 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**

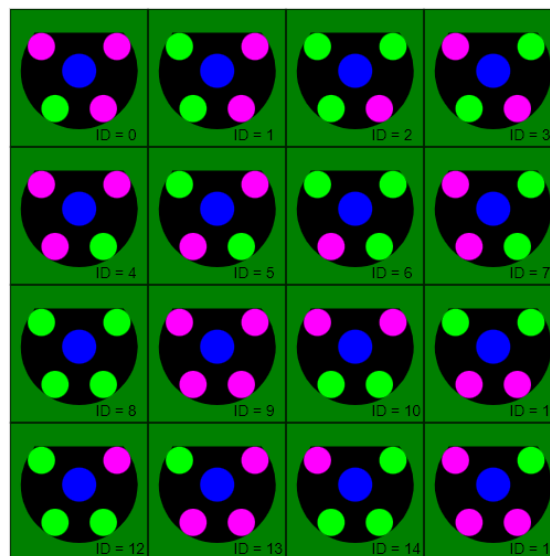
A câmera é um instrumento cujo uso em aplicações na área da robótica tem crescido bastante. Mapeamento, localização, navegação, desvio de obstáculos, reconhecimento de objetos e inspeção de qualidade são só alguns exemplos de possíveis utilizações de câmeras para realização de tarefas por parte de robôs.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

Segundo **cameras\_technexion**, 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 **melo2022embedded**.

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 (10.1007/978-3-642-11876-0\_37).

Figura 4 – Padrões de cores para identificação dos robôs.



Fonte: **rules**

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 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 **siegwart2011introduction**, 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.

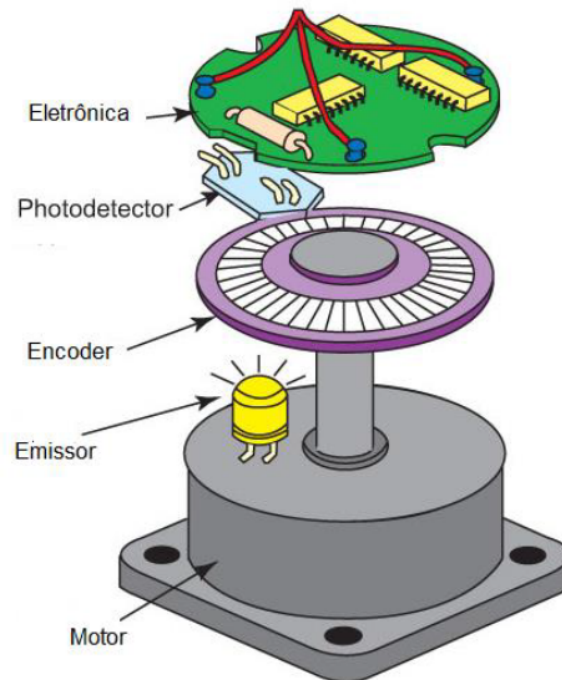
Um encoder óptico é basicamente um picador de luz mecânico que produz uma certa quantidade de pulsos na forma de um seno ou quadrado para cada revolução. No caso, o sensor consiste de uma fonte de iluminação, uma "grade" fixa que mascara a luz, um disco rotor com uma grade óptica fina que gira com o 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 (**ec45\_maxon**). Já o encoder utilizado é da empresa US Digital®(**e4t\_encoder**), no caso é utilizado um encoder do modelo E4T por roda também, sendo que eles ficam acoplados diretamente no eixo do motor.

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).

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

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



Fonte: Adaptado de **opticalencoder**

#### 2.2.4 Giroscópio

Giroscópios também são um dos principais sensores utilizados em robôs para realização de tarefas básicas como navegação. De acordo com **jeremydingman2020**, 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 **s17102284**, 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*', **siegwart2011introduction** 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 **siegwart2011introduction**, 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 (**datasheet\_gyro**). 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 I<sup>2</sup>C (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 **NISTLER2011413**, 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 (**persson1998we**) 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 **NISTLER2011413**, 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 **dadafshar2014accelerometer**, 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 **siegwart2011introduction**, 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 (**datasheet\_accel**). O componente inclui uma interface serial I<sup>2</sup>C, 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*) (**siegwart2011introduction**).

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 **sensor\_market**, 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 **calibration\_av**, 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 **lv2020targetless**, 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 **9181212**, 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 **menezes2020triaxial**, 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 **menezes2020triaxial**, 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 **hassan2020field**, 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 **menezes2020triaxial** 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 **hassan2020field** 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  $\omega$  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 **siegwart2011introduction**, 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 **thrun2001robust**, 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 **borenstein1997mobile**, 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, **siegwart2011introduction** 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 reduz 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 **siegwart2011introduction**, 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 recalculá-lo 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 (**mendes2017perceccao**), 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

**alatis2020review** 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 **castanedo2013review** e **nagla2014multisensor**, 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, **luo2011multisensor** 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 **hall1997introduction**, 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 **marton2013two**, 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

**fung2017sensor** 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.

**fung2017sensor** 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 **alalise2020review**, 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 **fung2017sensor**, 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 **castanedo2013review**, 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 **castanedo2013review**, a relação entre as fontes de dados é uma maneira de dividir as diferentes técnicas de fusão de sensores, separando nas seguintes três categorias.

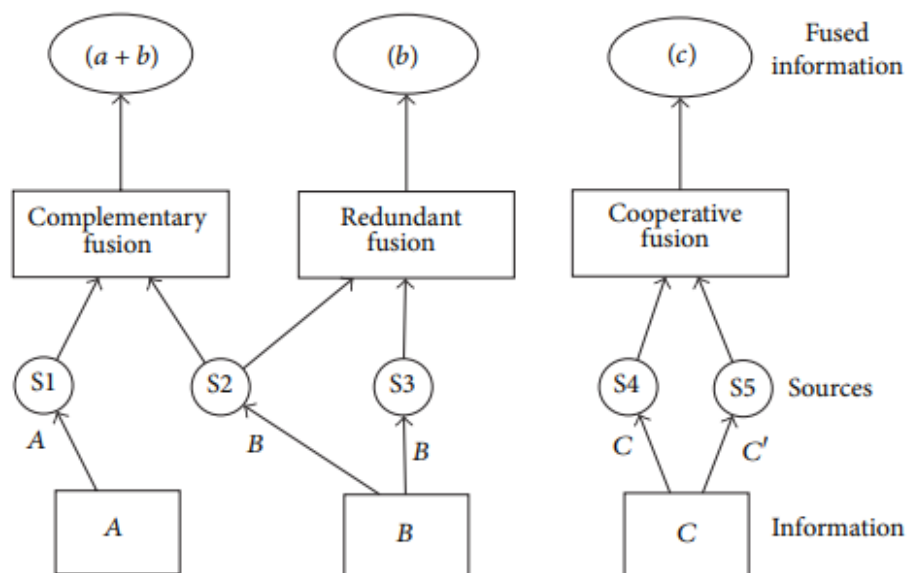
- a) **Complementar:** é o caso de quando os sensores não dependem diretamente entre si, mas podem ser combinados de uma maneira que entreguem uma visão mais completa do fenômeno sendo observado. Ou seja, a informação providenciada pelas diferentes fontes representam diferentes partes do cenário. Um 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. **visser1999organisation** 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 **brooks1998multi**, 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 **castanedo2013review**.

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



Fonte: Retirado de **castanedo2013review**

### 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 (castanedo2013review).

- 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 **dasarathy1997sensor** (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 (li2021data; vakil2021survey).

- 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.

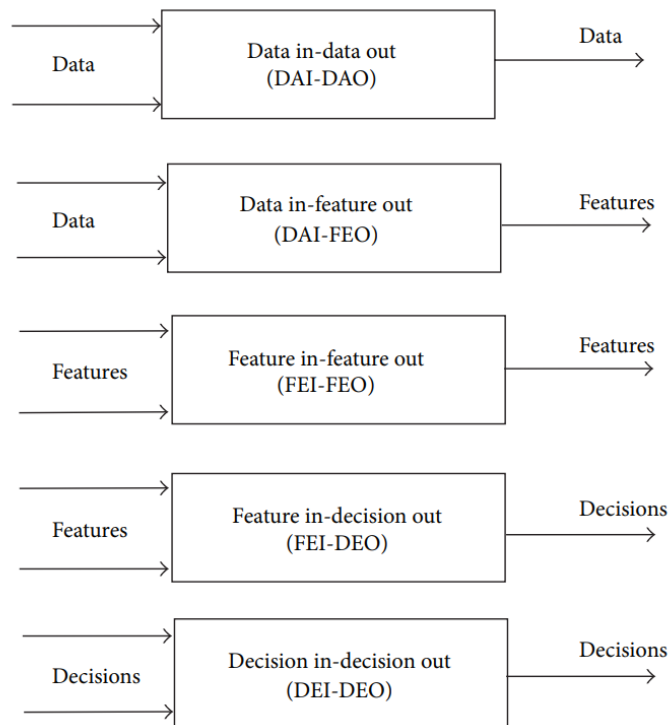
A Figura 7 representa claramente a diferença entre as cinco categorias de classificação proposta por **dasarathy1997sensor**. Já a Figura 8 relaciona e mostra as diferenças entre os modelos de classificação em três níveis e Dasarathy.

### 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.

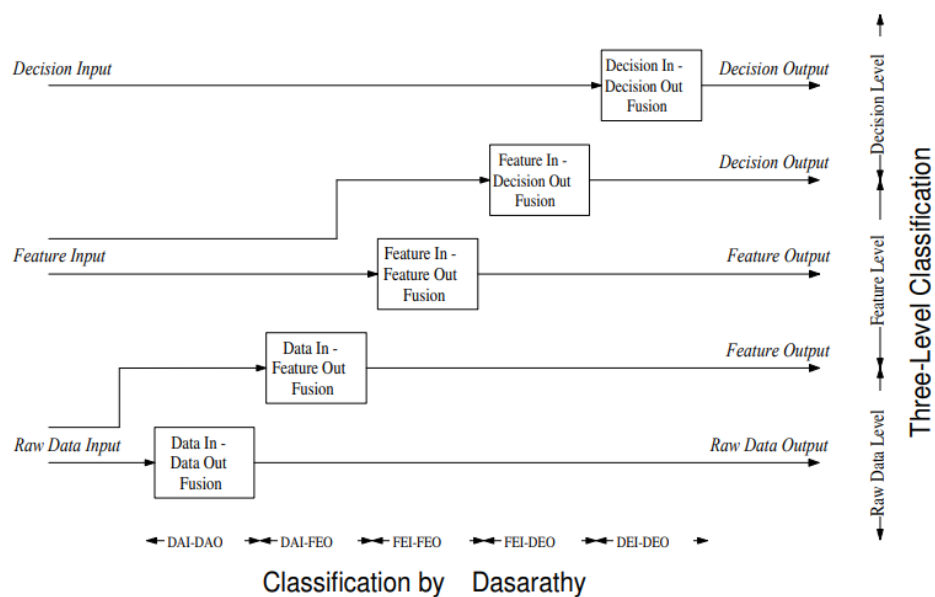
No caso desse trabalho, o filtro de Kalman é o algoritmo utilizado para realizar a fusão de diferentes sensores a fim de estimar a localização de robôs móveis, por isso a importância da descrição detalhada do seu funcionamento ao longo dessa seção.

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



Fonte: Retirado de **castanedo2013review**

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



Fonte: Retirado de **elmenreich2002introduction**



### 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 **khodarahmi2023review**, 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 (**urrea2021kalman**). Além de que 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 (**khodarahmi2023review**).

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  $\theta$  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 **siegwart2011introduction**, 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 **phdthesisNegenborn**, 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 **phdthesisNegenborn**, 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 (**urrea2021kalman**).

#### 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 (**urrea2021kalman**).

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

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 uma 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 (khodarahmi2023review).

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

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

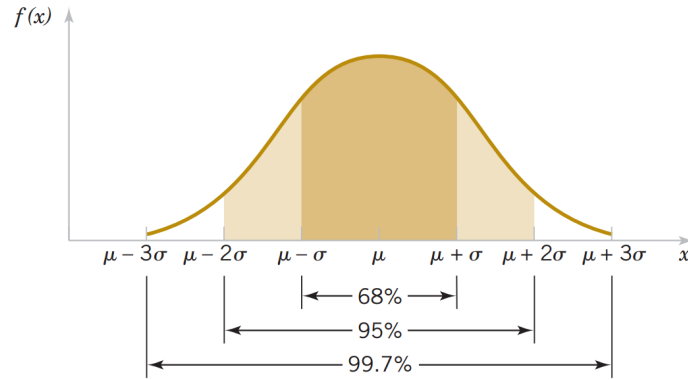
Uma característica necessária do ruído para o filtro de Kalman é a independência, que torna o cálculo envolvido na estimativa de estado mais fácil. De acordo com phdthesisNegenborn em geral é justo assumir que os ruídos no sistema e medição são independentes.

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 vezes 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  $\mu$  dos valores, já a dispersão (ou largura) do gráfico é representada pelo desvio padrão  $\sigma$  (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 **normal\_distribution**

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  $\mu$  e covariância  $\Sigma$ .

### 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 **welch1995introduction**, 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.

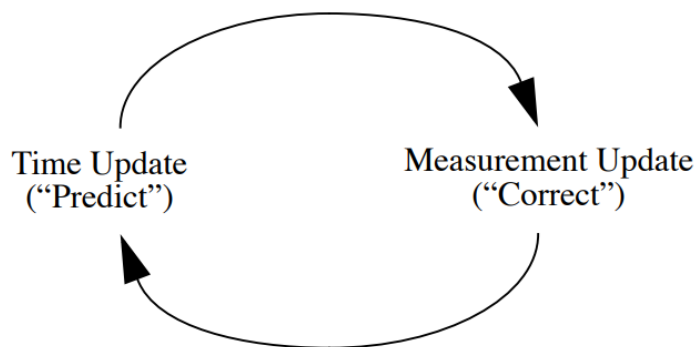
A matriz  $A$  da Equação (10) possui tamanho  $n \times n$  e relaciona o estado no passo de tempo anterior  $k - 1$  com o estado no passo de tempo atual  $k$  na ausência de uma função ou ruído de processo. Já a matriz  $B$  possui tamanho  $n \times l$  e relaciona a entrada de controle  $u \in \mathbb{R}^l$  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 **khodarahmi2023review**, 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 **welch1995introduction**

### 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 **thrun2002probabilistic**,  $bel(x_k) = N(\hat{x}_k^-, P_k^-)$ , 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 **thrun2002probabilistic**, 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 previsão 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 previsão 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 previsão 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 previsão 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).

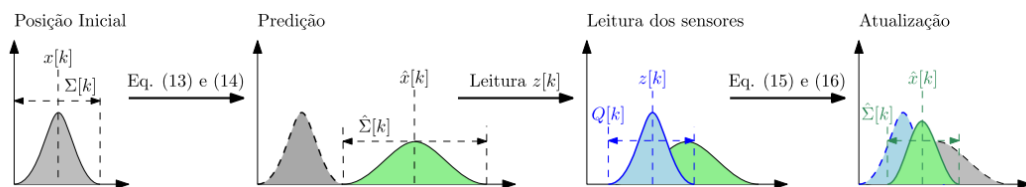
A fim de comparar a incerteza da estimativa do estado anterior no espaço de estados com a incerteza da inovação no espaço de medição, o KF converte a incerteza no espaço de medição para o espaço de estados por meio da matriz  $H^T$ .

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          $K_k = P_k^- H^T (HP_k^- H^T + R_k)^{-1}$
- 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 **costaleonardo2023**



### 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 (**thrun2002probabilistic**). 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 (**khodarahmi2023review**). 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 (**thrun2002probabilistic**).

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).

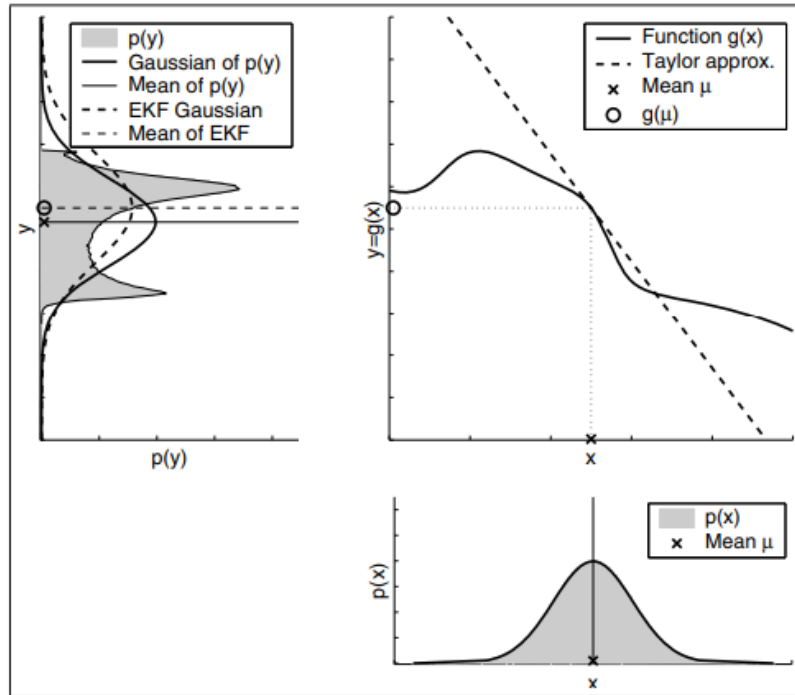
O EKF segue a mesma ideia do filtro de Kalman linear, isto é, com a separação nas etapas de predição, que projeta o sistema a frente para obter uma estimativa no próximo período de tempo, e correção, que incorpora uma nova medição na estimativa da predição a fim de obter uma melhor estimativa. A diferença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

De acordo com **thrun2002probabilistic**, 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 **thrun2002probabilistic**

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

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 **eman2020mobile**, 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  $\theta$  a orientação do robô,  $\mu$  a velocidade linear e  $\omega$  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  $0.1s$ ,  $\mu$  sendo  $5m/s$  e  $\omega$  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 **korotaj2021kalman** 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 **ismail2022soccer** 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.

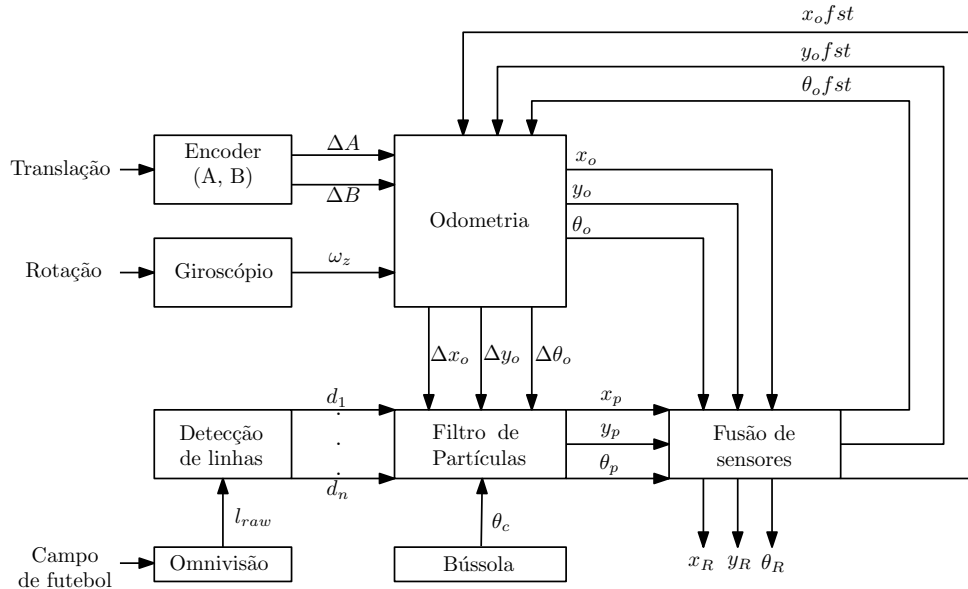
A maior parte dos robôs dessa categoria utilizam câmeras omnivisão, permitindo que os robôs tenham visão 360°. A partir dessas imagens é utilizado filtro de partículas (FP) para estimar a posição e orientação do robô em campo, embora o filtro necessite de um processamento computacional pesado.

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 **ismail2022soccer**.

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 **ismail2022soccer**

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  $\theta$  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 **10.1007/978-3-642-54734-8\_23** 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.

$$\begin{aligned} Z_{k+1} &= A_k Z_k + G_k n_k \\ Y_k &= C_k Z_k + w_k \end{aligned} \tag{24}$$

$$\begin{aligned} A_k &= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ \delta t_k & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & \delta t_k & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\ G_k &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{25}$$

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 **aguiar2017kalman** 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 **10333060** 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 (**sae2016board**).

Apesar dos resultados não serem satisfatórios para os autores por não cumprirem o determinado pela norma SAE J2945 (**sae2016board**), o modo que a etapa de predição é descrita no trabalho é uma ideia diferente que pode ser utilizada neste trabalho.

Já em **9233826** 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

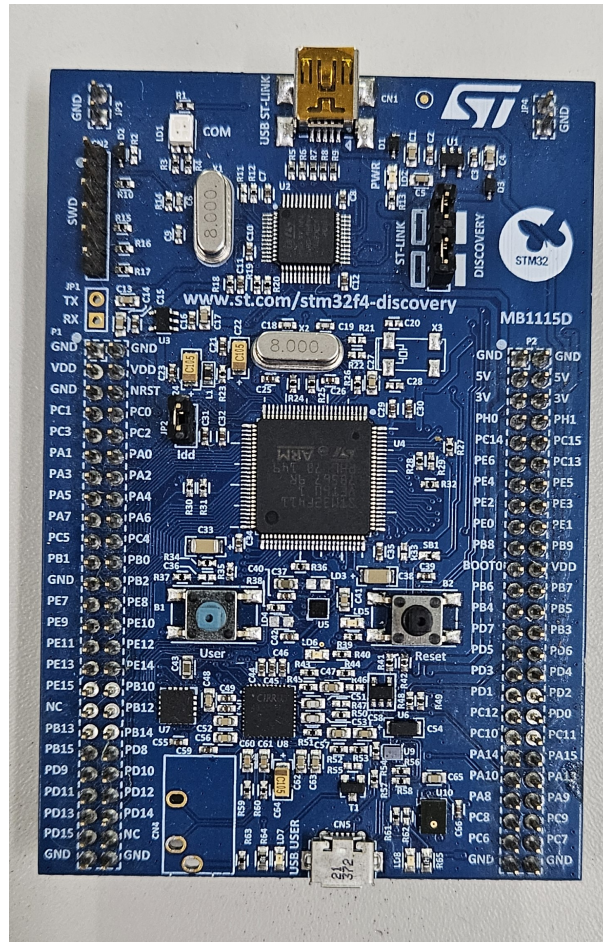
Neste trabalho será utilizado o domínio de teste da categoria SSL de futebol de robôs da RoboCup (**RoboCup**), descrito na Seção 2.1. Nele, as combinações de sensores serão testadas e comparadas. Para a realização dos testes propostos, será utilizado o campo do laboratório da equipe RoboFEI, que possui 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, 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 serão 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 CALIBRAÇÃO DOS SENSORES

Como descrito na Seção 2.2.7, há a necessidade da calibração dos sensores giroscópio e acelerômetro presentes neste estudo. Toda a calibração será realizada com a placa eletrônica posicionada no robô, visto que posteriormente a ideia é realizar uma calibração individual dos sensores de cada um dos robôs.

Assim, a primeira calibração a ser descrita é a do acelerômetro, que utiliza do método de **menezes2020triaxial**. No caso, serão tomadas amostras da aceleração nos 3 eixos em 9 posições diferentes para realização do método. Basicamente, a ideia é pegar amostras nas posições em que há os valores máximo e mínimo de aceleração em cada um dos eixos, e mais 3 posições em que o robô permanece em equilíbrio em cima de uma roda por vez. Em cada uma das posições serão lidas 4000 amostras em cada um dos eixos e, assim, será feita a média delas para cada um deles.

Já para o giroscópio, a ideia é realizar a calibração em uma só posição, diferente do giroscópio. Para isso serão tomadas 30000 amostras de velocidade angular nos 3 eixos e, assim, será feita a média dessas amostras.

Para validar a calibração de ambos os sensores, serão tomadas 10000 amostras nos 3 eixos antes e depois da calibração e a comparação será feita utilizando um gráfico do tipo *box plot*. A ideia é que, para o giroscópio, em todos os eixos a média dos dados fique bem próxima de zero, tal como para o acelerômetro nos eixos X e Y, enquanto para o Z espera-se que fique em torno de  $9.81m/s^2$ .

#### 5.4 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.

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) Logitech BRIO 4K UHD (**logitech\_brio**)
- b) Stingray F046C (**stingray\_f046c**) + Lente Tamron 12VM412ASIR



## 5.5 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 (**rplidar\_a1**) e RPLIDAR S1 (**rplidar\_s1**), 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	Dense Boost	925

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 15.

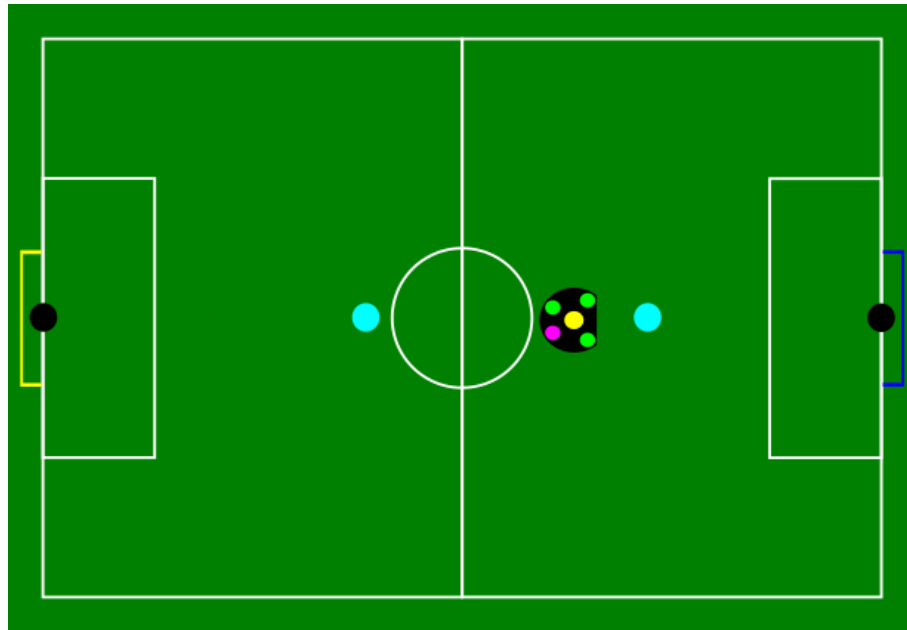
## 5.6 TESTES

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 quatro testes que buscam verificar pontos fortes e fracos de cada combinação.

O primeiro teste a ser realizado visa extrair a resposta da posição do sistema dado uma entrada de grau a fim de analisar a qualidade das combinações de sensores propostas em relação à orientação do robô. No caso, o robô estará com sua orientação em 0° e receberá um sinal para

Figura 15 – Representação do posicionamento dos lasers em campo.



Fonte: Autor

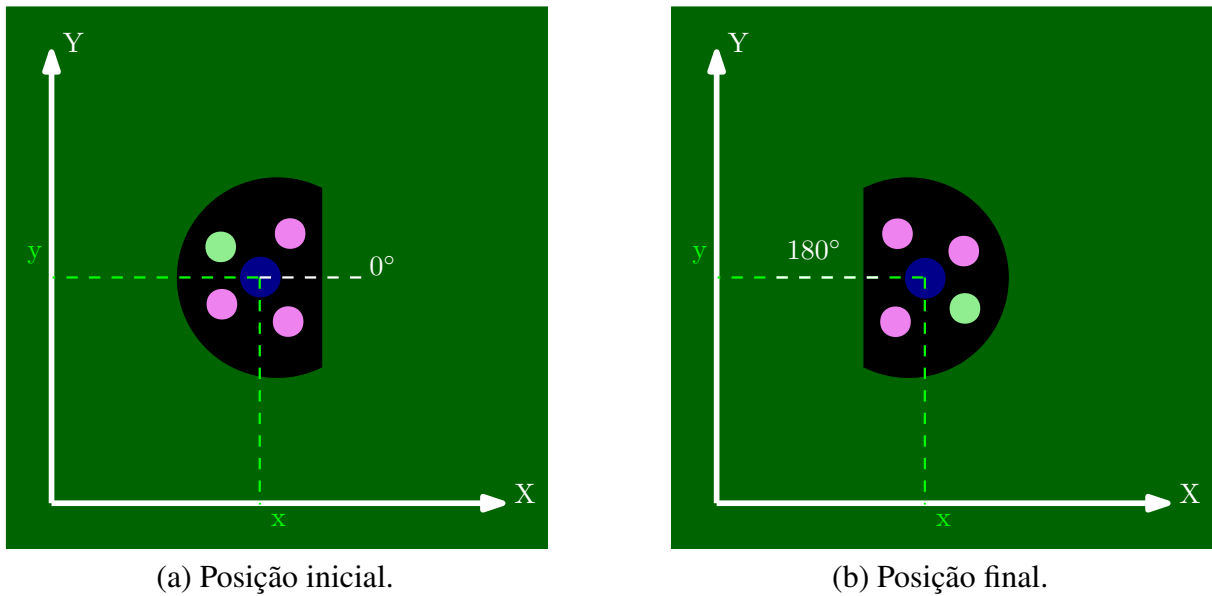
que ele se posicione nas mesmas coordenadas  $x$  e  $y$ , mas com uma orientação de  $180^\circ$ . Este teste servirá para validar as combinações dos sensores no que tange a orientação do robô, ou seja, para isolar a análise no que diz respeito à orientação  $\theta$  e não à posição  $(x, y)$ . A ilustração do teste pode ser observada na Figura 16.

O segundo cenário de teste a ser considerado é um quadrado com lado de tamanho  $1.5m$  em que o robô deverá seguir seus lados com uma orientação fixa de  $0^\circ$ . O cenário quadrado é bastante difundido para validação de estimativa de posição (rostami2018localizationgonccalves2008realsantini1997). Este teste servirá para validar as combinações de sensores no que diz respeito à coordenadas  $x$  e  $y$  do robô, ou seja, a fim de isolar a análise no que diz respeito à posição  $(x, y)$  e não à orientação  $\theta$ . A ilustração do trajeto a ser realizado pode ser verificada na Figura 17.

O terceiro teste para analisar as combinações de sensores escolhidas será um círculo de raio  $1m$  em que o robô deverá seguir sempre apontado para o centro do círculo. O cenário circular também é bastante difundido para validação de estimativa de posição (suliman2009mobilemarton2013tworigatos2010extendedeman2020mobile). Este teste servirá para validar as combinações tanto para a questão da orientação  $\theta$  quanto para a posição  $(x, y)$  do robô. A ilustração do trajeto a ser realizado pode ser verificada na Figura 18.

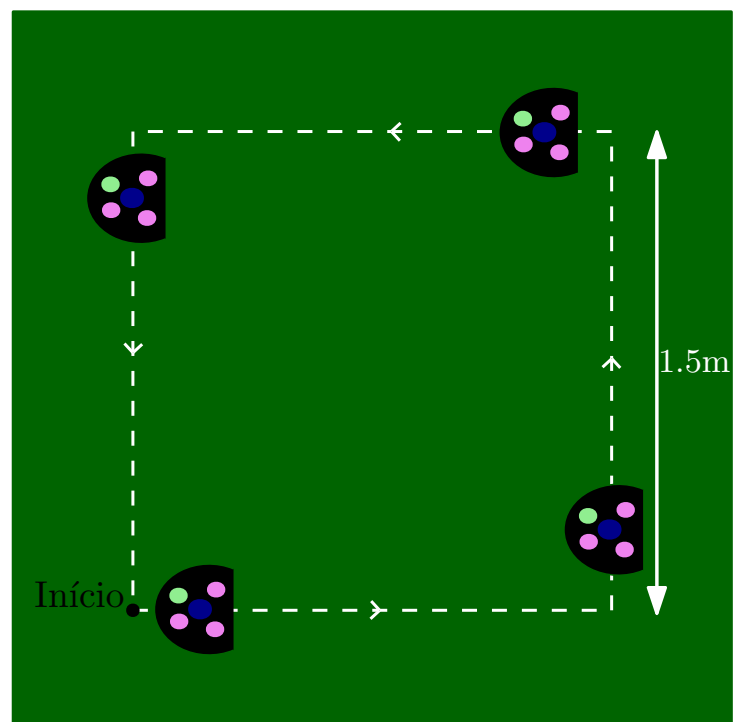
O quarto teste será um teste que está mais inserido no ambiente do futebol de robôs. No caso, o robô terá que sair de um ponto pré-determinado dentro do campo e, a partir da localização

Figura 16 – Ilustração do primeiro teste.



Fonte: o Autor

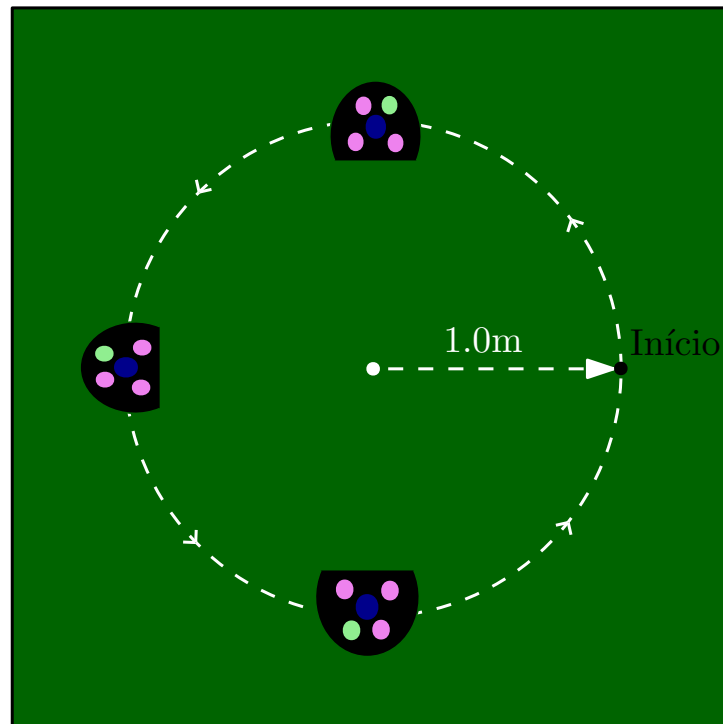
Figura 17 – Ilustração do segundo teste.



Fonte: o Autor

calculada de forma embarcada, deverá alcançar a bolinha que estará em uma outra posição em campo e fazer o gol.

Figura 18 – Ilustração do terceiro teste.



Fonte: o Autor

Para realização desse teste, algumas mudanças devem ser feitas na estratégia de controle de posição da equipe, visto que utiliza os dados puramente do sistema de visão para indicar qual posição cada robô deve ir e quais velocidades cada roda deve aplicar. Então, como o propósito desse projeto não é realizar o controle de posição de forma embarcada, a ideia é que ao invés de a estratégia utilizar os dados da câmera faça a utilização da posição calculada pelo filtro de Kalman e transmitida via rádio-frequência.

## 5.7 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  $\theta$  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  $\theta$ . No caso, será analisada a diferença do 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 será realizado 10 vezes. A partir deles, serão extraídos a média e o desvio padrão, o valor máximo e o valor mínimo de cada uma das métricas para uma análise concisa.

O quarto teste, que é o teste que irá avaliar as diferentes combinações em uma situação de jogo da SSL, também avaliará as porcentagens em que o robô conseguiu alcançar a bolinha e que o robô conseguiu marcar o gol.

Além disso, é importante realizar análises específicas do algoritmo utilizado no projeto, no caso o filtro de Kalman para realizar a fusão de sensores. Uma das métricas importantes de análise é a evolução da covariância dos estados ao longo dos trajetos. Outras duas métricas são o KG, que será relacionado ao peso que é dado ao modelo do sistema e à entrada de novas medições, e a inovação das medidas ao longo do trajeto, que mostra a diferença entre a medição e a estimativa de medição. Também será feita a comparação da posição que cada sensor calcula e a estimativa realizada pelo KF, mostrando a ponderação que o filtro calcula.

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  $\theta$  e que realiza o percurso no menor tempo possível.

## 6 RESULTADOS

Nesta seção serão apresentados os resultados obtidos neste projeto através dos experimentos descritos na Seção 5.6. 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 5.5, foi necessário desenvolver um sistema utilizando 2 LiDARs para obter um ground truth 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 19 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.

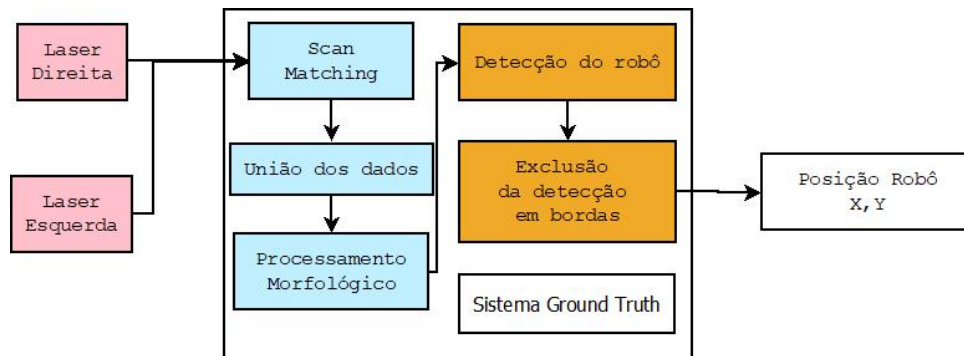


Figura 19 – Arquitetura do software.

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 (NIETO200739). 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  $\theta$  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 (lotufo2023morphological). 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.

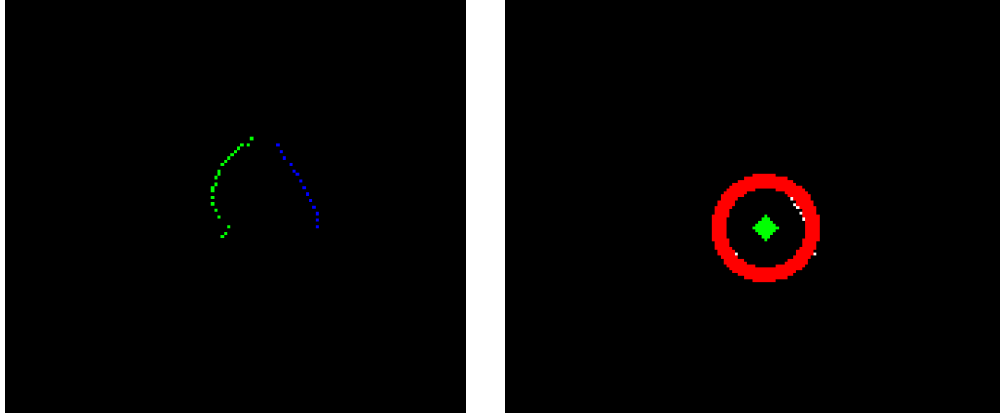
A próxima etapa do software é a detecção do robô. Esse estágio performa a detecção do padrão circular, ou semi-circular, na imagem de pontos dos lasers. O método utilizado para

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 (**mukhopadhyay2015survey**).

A HT para círculos utiliza da 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 (**hassanein2015survey**).

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

Na Figura 20 é possível observar dois momentos da arquitetura. A Figura 20a 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 20b mostra a etapa de detecção, em que a HT é aplicada na imagem de pontos dos lasers para detecção do padrão circular.



(a) Imagem crua dos pontos dos lasers.

(b) Círculo detectado pelo sistema.

Figura 20 – Duas etapas da arquitetura desenvolvida.

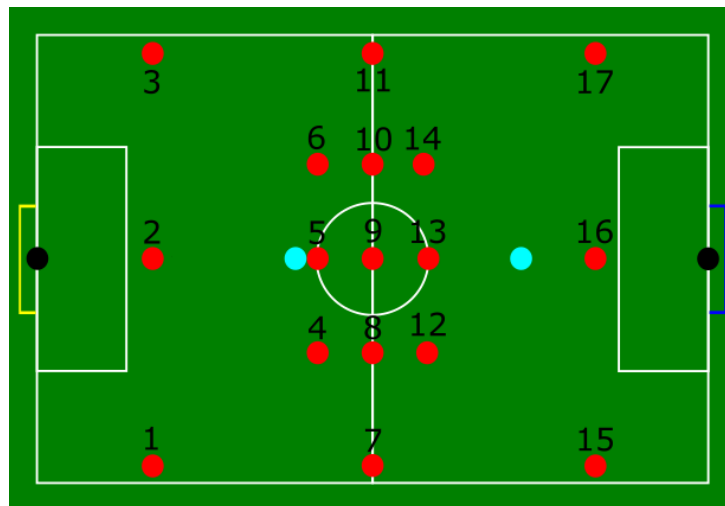
A última etapa de pós-processamento é realizada para detectar e excluir falsos positivos nas bordas do campo, já que a maior parte dos pontos detectados pelo laser são os pontos da borda do campo. Basicamente, as coordenadas de fronteira do campo são excluídos e a detecção é focada dentro do campo.



### 6.1.2 Resultados do sistema desenvolvido

Para realizar a comparação, foram decididas 17 posições arbitrárias em campo, que estão demonstrados na Figura 21. 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 21 – 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 uma diferença de aproximadamente 0.1cm entre ambos. Já para o eixo Y é possível observar uma diferença considerável entre eles, perto de 0.43cm. Essa diferença pode

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

ser explicada pela distorção ocorrida nas bordas da imagem da câmera, o que aumenta o erros 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 glsxtshortssl 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.

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

Tabela 8 – Resultados dos quadrantes para o eixo Y.

	<b>Erro Y - Laser</b>	<b>Erro Y - Vis.</b>	<b>Des. Padrão Y - Laser</b>	<b>Des. Padrão Y - Vis.</b>
<b>Quad. 1</b>	0,6641	0,9951	1,0893	0,0261
<b>Quad. 2</b>	0,5844	1,0754	1,0956	0,0454
<b>Quad. 3</b>	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 IMPLEMENTAÇÃO DOS CENÁRIOS

### 6.2.1 Predição com modelo

De acordo com a Tabela 1, os três primeiros cenários consistem da fusão do modelo do sistema de um robô omnidirecional na fase de predição do EKF, alterando o sensor utilizado na fase de correção (visão, encoder e IMU, respectivamente).

Conforme representado na Equação 1, o modelo do sistema tem como entrada as velocidades lineares, nos eixos X e Y do robô, e angular ( $v_{x-t}$ ,  $v_{y-t}$  e  $v_{\theta-t}$ , respectivamente), representada pelo vetor  $\mathbf{u}_k$ , conforme Equação 33. A Equação 34 descreve a atualização dos estados na fase de predição do sistema de localização, em que os estados do sistema são a posição linear X e Y em relação ao campo ( $x$  e  $y$ ), a posição angular ( $\theta$ ), e as velocidades locais nos eixos X e Y do robô ( $v_x$  e  $v_y$ ). 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.

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

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

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

Dessa forma, a Jacobiana dos estados  $G_k$  é apresentada na Equação

$$G_k = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad (36)$$

### 6.2.2 Predição com IMU

A partir da Tabela 1, os cenários 4 e 5 utilizam das leituras da IMU posicionada na placa eletrônica principal do robô para realizar a predição dos estados.

Nesses cenários, a entrada do sistema consiste das leituras realizadas pela IMU, tanto as acelerações em relação aos eixos X e Y do robô ( $a_x$  e  $a_y$ ), quanto a velocidade angular ( $v_w$ ), conforme a Equação 37. Já a Equação 38 traz a atualização dos estados definidos com a utilização da IMU. No caso, as velocidades globais seguem as definidas na Equação 35, já as acelerações globais são calculadas conforme a Equação 39.

$$\mathbf{u}_k = \begin{bmatrix} a_x \\ a_y \\ v_w \end{bmatrix} \quad (37)$$

$$\begin{cases} x[k] = x[k-1] + V_{x-G} \times dt + a_{x-G} \times \frac{dt^2}{2} \\ y[k] = y[k-1] + V_{y-G} \times dt + a_{y-G} \times \frac{dt^2}{2} \\ \theta[k] = \theta[k-1] + v_w[k] \times dt \\ v_x[k] = v_x[k-1] + a_x[k] \times dt \\ v_y[k] = v_y[k-1] + a_y[k] \times dt \end{cases} \quad (38)$$

$$\begin{cases} a_{x-G} = \cos(\theta[k]) \times a_x[k] - \sin(\theta[k]) \times a_y[k] \\ a_{y-G} = \sin(\theta[k]) \times a_x[k] + \cos(\theta[k]) \times a_y[k] \end{cases} \quad (39)$$

### 6.2.3 Predição com encoders

Já de acordo com a Tabela 1, os cenários 6 e 7 utilizam das leituras capturadas pelos encoders acoplados às rodas do robô para realizar a predição do estado do robô.

A entrada do sistema desses cenários são as velocidades locais calculadas a partir das leituras dos encoders ( $v_{x-e}$ ,  $v_{y-e}$  e  $v_{w-e}$ ), conforme mostrado na Equação 40. A Equação 41 mostra como as velocidades locais são utilizadas na atualização dos estados. As equações são semelhantes das que foram apresentadas na Seção 6.2.1, mudando somente a fonte das velocidades locais.

$$\mathbf{u}_k = \begin{bmatrix} v_{x-e} \\ v_{y-e} \\ v_{w-e} \end{bmatrix} \quad (40)$$

$$\begin{cases} x[k] = x[k-1] + (\cos(\theta[k-1]) \times v_x[k-1] - \sin(\theta[k-1]) \times v_y[k-1]) \times dt \\ y[k] = y[k-1] + (\sin(\theta[k-1]) \times v_x[k-1] + \cos(\theta[k-1]) \times v_y[k-1]) \times dt \\ \theta[k] = \theta[k-1] + v_{\theta-e}[k] \times dt \\ v_x[k] = v_{x-e}[k-1] \\ v_y[k] = v_{y-e}[k-1] \end{cases} \quad (41)$$

### 6.3 QUADRADO MENOR

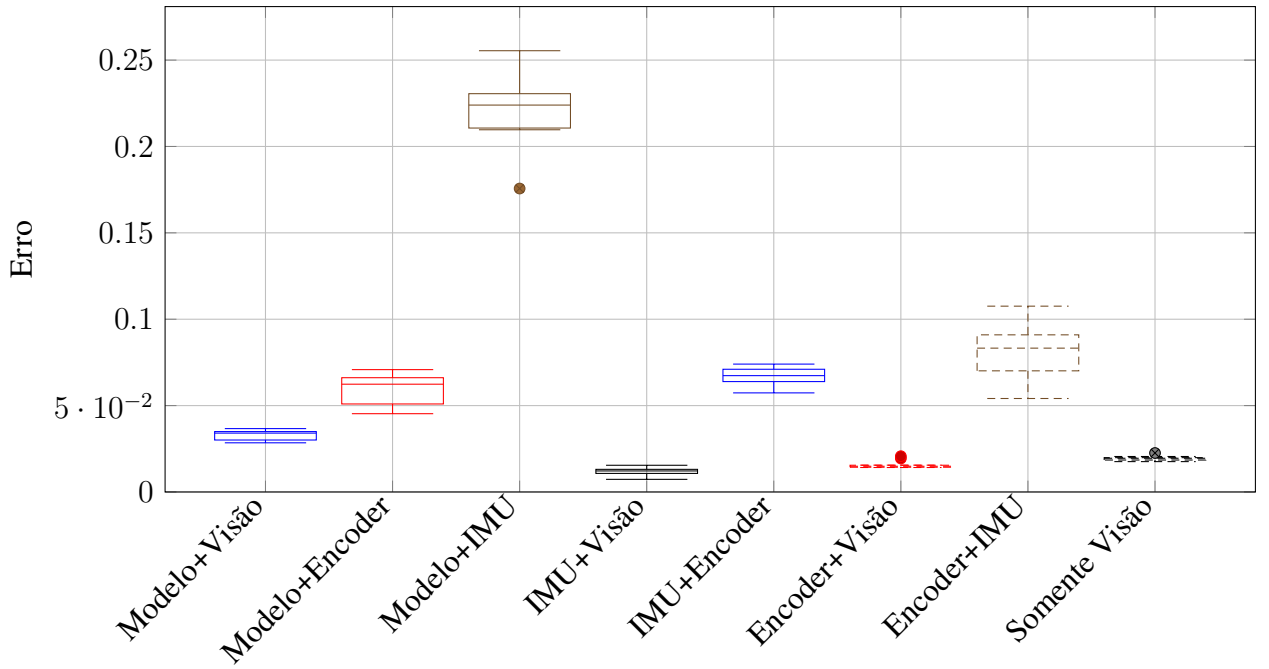


Figura 22 – Boxplot dos erros por combinação de sensores