

**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROJETO DE GRADUAÇÃO**



IGOR BATISTA VIEIRA

**FUSÃO SENSORIAL ENTRE CÂMERA, ENCODERS E
IMU PARA APRIMORAR A PRECISÃO DA
ESTIMATIVA DE POSIÇÃO E ORIENTAÇÃO DE UM
ROBÔ MÓVEL A RODAS**

VITÓRIA-ES

JULHO/2023

Igor Batista Vieira

**FUSÃO SENSORIAL ENTRE CÂMERA, ENCODERS E
IMU PARA APRIMORAR A PRECISÃO DA
ESTIMATIVA DE POSIÇÃO E ORIENTAÇÃO DE UM
ROBÔ MÓVEL A RODAS**

Parte manuscrita do Projeto de Graduação do aluno Igor Batista Vieira, apresentado ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheiro Eletricista.

Vitória-ES

Julho/2023

Igor Batista Vieira

FUSÃO SENSORIAL ENTRE CÂMERA, ENCODERS E IMU PARA APRIMORAR A PRECISÃO DA ESTIMATIVA DE POSIÇÃO E ORIENTAÇÃO DE UM ROBÔ MÓVEL A RODAS

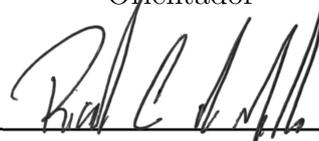
Parte manuscrita do Projeto de Graduação do aluno Igor Batista Vieira, apresentado ao Departamento de Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Engenheiro Eletricista.

Aprovado em 18 de julho de 2023.

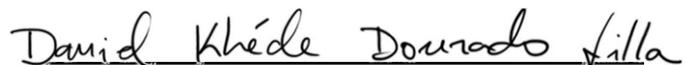
COMISSÃO EXAMINADORA:



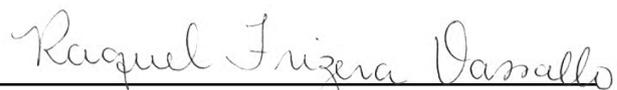
Prof. Dr. Anselmo Frizera Neto
Universidade Federal do Espírito Santo
Orientador



Prof. Dr. Ricardo Carminati de Mello
Universidade Federal do Espírito Santo
Coorientador



Prof. Dr. Daniel Khede Dourado Villa
Universidade Federal do Espírito Santo
Examinador



Profa. Dra. Raquel Frizera Vassallo
Universidade Federal do Espírito Santo
Examinadora

Vitória-ES

Julho/2023

"Por vezes sentimos que aquilo que fazemos não é senão uma gota de água no mar. Mas o mar seria menor se lhe faltasse uma gota." - Madre Teresa de Calcutá

AGRADECIMENTOS

Gostaria de agradecer a todas essas pessoas especiais que apareceram (e estão) na minha vida e tornaram possível que eu prosseguisse com os meus estudos. A todos vocês, minha eterna gratidão.

Primeiramente a Deus, fonte da minha alegria e razão do meu viver. Por ter me amparado em toda minha caminhada e sempre derramado graças na minha vida. Dos momentos mais complicados aos de pura alegria sempre se fez presente e nunca me deixou faltar nada.

Aos meus pais, Benedita e Renato, pelo apoio, amor e dedicação em toda a minha vida. Sem vocês teria sido impossível chegar onde cheguei. Aos meus irmãos Rhenzo e Beatriz, por sempre estarem ao meu lado, superando juntos os problemas e tornando os momentos mais leves. Assim como os demais familiares por sempre torcerem, apoiarem e acreditarem em mim, entendendo minha ausência em determinados momentos.

Aos meus amigos Matheus, Amanda, Arthur e Yuri, por terem me apoiado durante toda a trajetória dentro e fora da UFES, por serem os melhores amigos que eu poderia ter, se fazendo sempre presentes em toda e qualquer situação.

Ao Shalom, por vir sendo uma fonte de graças na minha vida e base da minha vida espiritual, em especial a Giselle, Yago e Gabriel por virem intercedendo nessa jornada, assim como todos os amigos que venho fazendo nessa caminhada.

Ao meu orientador Anselmo Frizera e coorientador Ricardo Caminati por todo suporte, ajuda, orientação e dedicação durante o desenvolvimento deste trabalho, ajudando além do esperado e necessário. Menção honrosa ao professor Daniel Khede por auxiliar na pesquisa.

Aos membros do Núcleo de Tecnologia Assistiva da UFES, em especial à Fabiana, por terem me oferecido auxílio técnico e teórico para que este trabalho pudesse ser realizado e cederem o espaço e estrutura para que pudesse realizar os experimentos.

À banca examinadora pela aceitação do convite e pelo tempo investido para leitura e avaliação desse trabalho.

Agradeço à Universidade Federal do Espírito Santo pela minha formação, assim como os membros do colegiado e do departamento de engenharia elétrica.

Agradeço aos projetos de pesquisa da FAPES (2021-V4J3L, 2022-D48XB, 2022-C5K3H), do CNPq (304049/2019-0, 403753/2021-0) e I2CA (FAPES 2021-07KJ2) pelo suporte financeiro fornecido ao longo deste estudo. O apoio desses projetos foi fundamental para a realização e conclusão deste trabalho.

RESUMO

Estimar uma posição e orientação mais precisa de um robô móvel é de extrema importância, uma vez que erros provenientes desta estimativa são propagados onde é utilizada, podendo causar problemas na localização do robô no desenvolvimento de estratégias. A fusão sensorial surge como uma solução para mitigar esses erros, integrando dados provenientes de múltiplos sensores. No entanto, é necessário explorar métodos eficazes para combinar esses dados, levando em consideração as incertezas, ruídos e possíveis falhas associadas a cada sensor. Neste trabalho, foi realizada a fusão sensorial entre a estimativa de posição obtida por uma câmera, *encoders* e uma IMU, com o intuito de avaliar o desempenho das diferentes configurações adotadas, assim como seu comportamento em um ambiente dinâmico, no qual são inseridos ruídos que podem impactar na precisão da estimativa realizada pela câmera. Para isso, foi integrada em uma plataforma robótica, uma câmera capaz de estimar a posição do robô. A seguir, seus dados são combinados com os dos *encoders* e da IMU e avaliado o desempenho ao executar um caminho pré-programado, utilizando um sistema de captura de movimentos para gerar uma referência. Os resultados obtidos mostram que a integração da câmera sendo realimentada pela estimativa dos *encoders*, mesmo em condições desfavoráveis, se mostrou a configuração mais robusta dentre as analisadas, encontrando um erro médio de posição até 82,52% menor em relação aos erros obtidos com o filtro de Kalman utilizando os dados da estimativa das rodas e orientação da IMU, e até 89.58% menor em relação a odometria puramente das rodas.

Palavras-chave: Fusão sensorial, câmera, *encoders*, IMU, robô móvel, Filtro de Kalman.

ABSTRACT

Estimating a more accurate position and orientation of a mobile robot is of utmost importance, as errors originating from this estimation are propagated wherever it is used, potentially causing problems in the robot's localization during strategy development. Sensor fusion emerges as a solution to mitigate these errors by integrating data from multiple sensors. However, it is necessary to explore effective methods to combine this data, taking into account uncertainties, noise, and possible failures associated with each sensor. In this work, sensor fusion was performed between the position estimate obtained from a camera, encoders, and an IMU, aiming to evaluate the performance of different configurations adopted, as well as their behavior in a dynamic environment where noise is introduced, which can impact the accuracy of the camera's estimation. To achieve this, a camera capable of estimating position was integrated into a robotic platform, combining its data with that of the encoders and IMU, and the performance was evaluated by executing a pre-programmed path, using a motion capture system to generate a reference. The results obtained show that the integration of the camera, being fed back by the encoder estimate, even under unfavorable conditions, proved to be the most robust configuration among those analyzed, with an average position error up to 82.52% smaller compared to the errors obtained with the Kalman filter using wheel estimation and IMU orientation data, and up to 89.58% smaller compared to purely wheel-based odometry.

Keywords: Sensor fusion, camera, encoders, IMU, mobile robot, Kalman Filter.

LISTA DE FIGURAS

Figura 1 – Modelo do robô diferencial do tipo Skid-Steer.	20
Figura 2 – Modelo bicicleta para o robô tipo carro.	22
Figura 3 – Velocidade e raio de giro desejado.	23
Figura 4 – Velocidade desejada do robô, das rodas e ângulo de deflexão.	23
Figura 5 – Encoder óptico em quadratura: A relação de fase observada entre os trens de pulsos dos canais A e B é usada para determinar a direção da rotação.	27
Figura 6 – Sensores não ortogonais (acelerômetros ou giroscópios) eixos (x^S, y^S, z^S) , eixos do sistema de referência do corpo (x^B, y^B, z^B)	28
Figura 7 – (a) Marinheiros experientes utilizavam as estrelas para navegar. (b) Capaz de alcançar o rastreamento em 6 graus de liberdade, a odometria visual é uma excelente opção para o uso em robôs (c) e no uso de óculos de realidade virtual.	29
Figura 8 – Exemplo de detecção de features pelo algoritmo <i>Harris Corner Features</i>	30
Figura 9 – Erro de odometria devido a um erro entre o diâmetro das rodas.	32
Figura 10 – Caminho pré programado a ser executado pelo robô.	33
Figura 11 – Odometria gerada contendo um erro de Tipo A, onde o valor com a distância entre as rodas nominal é menor do que a atualmente configurada, inicializando no sentido horário (a) e anti horário (b).	33
Figura 12 – Odometria gerada contendo um erro de Tipo B, onde o diâmetro da roda traseira esquerda ligeiramente maior do que o diâmetro da roda traseira direita, inicializando no sentido horário (a) e anti horário (b).	34
Figura 13 – Robô R550 (AKM) Plus modelo Arckeman da empresa Wheeltec.	38
Figura 14 – Sensores nativos do robô Ackerman e suas conexões.	38
Figura 15 – A câmera Intel RealSense Tracking Camera T265 possui um algoritmo de SLAM embarcado que usa a VIO para calcular sua posição e orientação no plano 3D.	39
Figura 16 – Arquitetura interna da Intel® Movidius™ Myriad™ 2 VPU.	39
Figura 17 – Diagrama de blocos da Intel® RealSense™ Tracking Camera T265.	40
Figura 18 – Visão geral do <i>setup</i> experimental utilizado.	40
Figura 19 – Marcadores posicionados no robô	41
Figura 20 – Corpos rígidos inicializados para o referencial do robô e da câmera	41
Figura 21 – Suporte 3D utilizado para fixação da câmera T265.	42
Figura 22 – Fluxo de dados no ambiente ROS	43
Figura 23 – Sistema de coordenadas utilizadas pela câmera T265.	43
Figura 24 – Referencial do robô e o referencial onde a câmera foi posicionada.	44

Figura 25 – Fluxo das transformações aplicadas na odometria visual.	45
Figura 26 – Caminho realizado pelo robô de forma teleoperada.	46
Figura 27 – Odometria pura das rodas e sua trajetória real.	47
Figura 28 – Odometria das rodas após calibração e sua referência real.	48
Figura 29 – Comportamento da odometria da câmera em relação ao seu referencial.	49
Figura 30 – Odometria das rodas e odometria obtida com a fusão da IMU pelo Filtro de Kalman.	49
Figura 31 – Odometria Visual levando em consideração a odometria das rodas.	50
Figura 32 – Caminho programado a ser seguido pelo robô.	51
Figura 33 – Transformações aplicadas na odometria visual e odometria das rodas para o ponto a	52
Figura 34 – Odometrias a serem avaliadas pelo robô nos Experimentos 1, 2, 3 e 4. (a) Odometrias independentes. (b) Odometria visual com realimentação da odometria das rodas. c) Odometria visual com realimentação do Filtro de Kalman. (d) Filtro de Kalman adicionando a odometria visual.	53
Figura 35 – Odometria das rodas, odometria Visual, e a odometria o Filtro de Kalman com IMU e odometria das rodas.	54
Figura 36 – Odometria das rodas, odometria Visual com realimentação da odometria das rodas, Filtro de Kalman com IMU e odometria das rodas.	55
Figura 37 – Odometria das rodas, Odometria Visual com realimentação do Filtro de Kalman, Filtro de Kalman com IMU e odometria das rodas.	56
Figura 38 – Odometria das rodas, odometria Visual, Filtro de Kalman com IMU e odometrias visual e das rodas.	57
Figura 39 – Queda de luz durante o experimento - Odometria das rodas, Odometria Visual, Filtro de Kalman com IMU e odometria das rodas.	59
Figura 40 – Queda de luz durante o experimento - Odometria das rodas, Odometria Visual com realimentação da odometria das rodas, Filtro de Kalman com IMU e odometria das rodas.	60
Figura 41 – Queda de luz durante o experimento -Odometria das rodas, Odometria Visual, Filtro de Kalman com IMU e Odometrias Visual e das rodas.	61
Figura 42 – Erro euclidiano médio - Luzes em condições normais.	64
Figura 43 – Erro euclidiano médio - Queda de luz no meio	65
Figura 44 – Erro euclidiano no tempo - Experimento 2.	65
Figura 45 – Erro euclidiano no tempo - Experimento 6.	66
Figura 46 – Orientações obtidas em cada configuração durante o 7 ^o experimento.	67
Figura 47 – Erro Angular Médio ($^{\circ}$)	68

LISTA DE TABELAS

Tabela 1 – Erro euclidiano médio - Luzes em condições normais.	63
Tabela 2 – Erro euclidiano médio - Queda de luz no meio.	64
Tabela 3 – Erro de orientação médio - Queda de luz no meio.	67

LISTA DE ABREVIATURAS E SIGLAS

EKF	Filtro de Kalman Estendido (<i>Extended Kalman Filter</i>)
ICR	Centro de Rotação Instantâneo (<i>Instantaneous Center of Rotation</i>)
IMU	Unidade de Medição Inercial (<i>Inertial measurement unit</i>)
LiDAR	<i>Light Detection And Ranging</i>
PF	Filtro de Partículas (<i>Particle Filter</i>)
SLAM	Localização e Mapeamento Simultâneos (<i>Simultaneous Localization and Mapping</i>)
VIO	Odometria Visual Inercial (<i>Visual Inertial Odometry</i>)
VO	Odometria Visual (<i>Visual Odometry</i>)
VRPN	<i>Virtual-Reality Peripheral Network</i>
UFES	Universidade Federal do Espírito Santo

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Motivação	14
1.2	Objetivos	15
1.2.1	Objetivo Geral	15
1.2.2	Objetivos Específicos	16
1.3	Justificativa	16
1.4	Organização do Texto	18
2	REFERENCIAL TEÓRICO	19
2.1	Conceitos gerais sobre Robótica Móvel	19
2.1.1	Modelo cinemático diferencial	19
2.1.2	Modelo cinemático Ackerman	21
2.2	Odometria	25
2.2.1	Encoders	26
2.2.2	Sensores Inerciais para Odometria	27
2.2.3	Odometria Visual e Odometria Visual Inercial	28
2.3	Erros de Odometria	30
2.3.1	Erros na Odometria das Rodas	31
2.3.2	Erros na Odometria causados por IMUs	32
2.3.3	Erros na Odometria Visual e Odometria Visual Inercial	34
2.4	Fusão Sensorial para redução do Erro de Odometria	35
3	METODOLOGIA	37
3.1	Materiais	37
3.2	Integração do robô Ackerman com a câmera T265	42
3.3	Transformações de referencial	43
3.4	Validação Experimental	45
3.4.1	Experimentos Preliminares: Teleoperação	45
3.4.2	Seguidor de Caminho e Experimentos Automatizados	50
4	RESULTADOS	54
4.1	Câmera operando em condições nominais	54
4.2	Influência de Ruído	58
4.3	Cálculo de Erro de Posição	63
4.4	Calculo de Erro de Orientação	66
5	CONCLUSÃO	69

REFERÊNCIAS	71
--------------------	-----------

1 INTRODUÇÃO

1.1 Motivação

O termo Indústria 4.0 vem se tornando cada vez mais popular no âmbito acadêmico e industrial. Koch et al. (2014) destaca a Indústria 4.0 pelo uso de processos autônomos mais eficientes, proporcionando soluções customizadas para a produção e clientes distintos. Utilizando tecnologias de automação industrial através de sensores, Koch et al. (2014) cita que o objetivo da quarta revolução industrial é obter sistemas mais inteligentes e eficientes capazes de lidar com problemas sem que haja interferência humana.

A robótica dentro da Indústria 4.0 vem trazendo inovações e aumentando a produtividade, contribuindo consideravelmente para o desenvolvimento da indústria (BAHRIN et al., 2016). Os robôs são capazes de desempenhar diversas tarefas complexas de forma mais econômica e padronizada em relação ao desempenho humano. Atualmente, indústrias já trabalham com robôs, também conhecidos como *cobots*, atuando como força de trabalho colaborativa, sendo capazes de desempenhar múltiplas tarefas em ambientes adversos (REIS, 2021).

Simhon (2011) afirma que o objetivo da robótica dentro do ambiente industrial representa o aumento da eficiência e da otimização dos processos produtivos, auxiliando a mão de obra humana, e não sua substituição. Além disso, a segurança desempenha um papel crucial nesse contexto. Os robôs têm a capacidade de assumir tarefas perigosas e repetitivas, reduzindo significativamente os riscos para os trabalhadores e aumentando a segurança no ambiente de trabalho.

Cenários que podem causar impacto direto na segurança do ser humano são aplicações especiais onde se faz fundamental o uso da robótica para que sua integridade seja mantida. Um bom exemplo disto é o robô de resgate utilizado em Tóquio para limpar áreas devastadas pelo Tsunami (AMARAL, 2011). Graças aos avanços tecnológicos é possível realizar operações de transporte de forma autônoma até o uso de robôs que permitam o monitoramento do ambiente em tempo real através de câmeras integradas (FORESIGHT, 2021).

Robôs que desempenham rotinas autônomas vem se tornando cada vez mais populares no dia a dia, como é o caso de robôs de limpeza, onde os mais avançados são capazes de mapear o imóvel à medida que limpam, gravando na memória o mapa no ciclo de limpeza¹.

¹ Website: <<https://ropo.com.br/loja/mapeamento- robo- aspirador/>>. Acessado 27 de julho de 2023.

A Amazon vem investindo no desenvolvimento de robôs de entregas, seguindo uma rota de entregas de forma autônoma, parando na porta do cliente e abrindo sua tampa permitindo que ele retire sua compra (CAMARGO, 2022).

Visando tornar um robô móvel autônomo, três problemas fundamentais devem ser solucionados: onde ele está, onde quer chegar e como chegar lá. Cada um destes problemas se resume a um campo de estudo, sendo o primeiro relacionado ao problema de localização e sensoriamento, onde o robô deve determinar a posição no ambiente através das leituras e interpretações dos sensores (LEONARD; DURRANT-WHYTE, 1991).

A localização de um robô móvel é estimada em uma posição e uma orientação, sendo esta combinação definida como pose (CORKE, 2013). Esta posição é obtida ao realizar a leitura dos sensores disponíveis, estimando a sua localização através do modelo cinemático do robô utilizado. O deslocamento realizado pelo robô em um determinado tempo é somado aos valores de deslocamentos previamente realizados. Tal processo onde é determinada a localização do robô através das informações de sensores é nomeada odometria.

Valores incorretos de estimativa de posição são propagados em todas as outras etapas onde esta é utilizada, sendo elas: cognição e controle de movimento. Devido a isto, a etapa de localização vem sendo estudada rigorosamente nas ultimas décadas (DOGRUER; KOKU; DOLEN, 2007). Sensores e atuadores desempenham um papel importante na localização em robótica móvel. Devido a imprecisão e incompletude destes sensores e efetores que a localização apresenta desafios (SIEGWART; NOURBAKHS, 2011).

Tais problemas de estimativa de posição são mitigados através da fusão sensorial. Através da fusão de diferentes fontes é possível obter uma estimativa de posição melhor do que uma obtida de forma individual, cobrindo incertezas e limitações dos sensores. Um dos métodos de fusão que leva em consideração a incerteza e confiabilidade dos sensores é o filtro de Kalman, sendo muito utilizado devido a sua eficiência e simplicidade (ASSA; JANABI-SHARIFI, 2015).

1.2 Objetivos

1.2.1 Objetivo Geral

O objetivo deste trabalho é aprimorar a precisão da estimativa de posição e orientação de um robô móvel por meio da fusão sensorial de dados provenientes de sensores proprioceptivos e

exteroceptivos, buscando uma configuração que seja suficientemente robusta para minimizar os erros quando o sistema enfrentar distúrbios.

1.2.2 Objetivos Específicos

Visando alcançar o objetivo geral, são definidos os seguintes objetivos específicos a serem realizados, sendo estes:

- Implementar um método de fusão sensorial adequado para combinar os dados de odometria visual e odometria do robô.
- Coletar e analisar dados de odometria visual e dados de odometria do robô em diferentes configurações e condições.
- Validar experimentalmente as técnicas de fusão sensorial utilizadas, comparando os resultados com um método de referência estabelecido.
- Analisar o comportamento da estimativa de localização diante de perturbações, como mudanças repentinas no ambiente ou ruídos nos sensores.

1.3 Justificativa

Estimar a localização de uma plataforma móvel no ambiente é de grande interesse no campo de robótica móvel devido a sua importância na navegação. A determinação da pose, sendo esta a combinação entre posição e orientação (CORKE, 2013), apresenta-se como um desafio quando ações são executadas em um robô e é necessária a estimativa da posição em um mapa por meio de sensores (GUTMANN; FOX, 2002).

Realizar a navegação e localização de um robô móvel em um ambiente arbitrário é um desafio devido à complexidade e particularidades de cada ambiente, assim como os métodos e sensores envolvidos no processo. Portanto, é necessária a pesquisa sobre novos sistemas e métodos com o objetivo de parametrizar sensores específicos e resolver problemas de fusão relacionados à navegação e localização de robôs (ALATISE; HANCKE, 2020).

Um dos métodos utilizados na robótica e na navegação para estimar a posição e a orientação de um veículo móvel, como um robô ou um veículo autônomo, é a odometria. Tal método consiste na estimativa da pose atual com base nas mudanças realizadas no tempo. Os

robôs utilizam dados de sensores que sejam capazes de captar o movimento realizado para estimar a sua posição atual com base na posição inicial (AQEL et al., 2016a).

A odometria através das rodas é uma das formas mais simples de estimar a localização do próprio robô (MOHAMED et al., 2019). O acúmulo de erros devido a efeitos de derrapagem e leituras incorretas dos *encoders* fazem com que esta se torne imprecisa, sendo necessário ser corrigida (CHONG; KLEEMAN, 1997). Os erros no cálculo da odometria podem ser reduzidos através da fusão sensorial, combinando dados de várias fontes a fim de reduzir as incertezas e discrepâncias (ABDULHAFIZ; KHAMIS, 2013).

Astolfi et al. (2018) realizaram a fusão sensorial entre a odometria através das rodas e uma Unidade de Medição Inercial (IMU). Ao observarem os dados obtidos, foi verificado o acúmulo de erros com o tempo. Tal problema foi amenizado ao introduzir um sensor GPS na fusão sensorial, que foi capaz de realizar pequenas correções periodicamente, obtendo uma odometria mais confiável para ser utilizada na navegação autônoma. A fusão entre a IMU e odometria através das rodas é algo que vem sendo explorado desde os anos 90 na robótica (SUKKARIEH; NEBOT; DURRANT-WHYTE, 1999).

A fusão sensorial desempenha um papel essencial para obter uma percepção mais precisa do ambiente. A combinação de informações de diferentes sensores permite compensar as limitações individuais de cada um, resultando em estimativas mais confiáveis e robustas. Esta abordagem encontra inspiração na percepção biológica, onde os animais combinam sinais sensoriais para criar um modelo dinâmico do mundo ao seu redor (JUNG et al., 2019).

Elmenreich (2002) descreve que integração de múltiplos sensores possibilita que o sistema consiga suprir informações em caso de falha. Além disso, áreas cobertas por um sensor podem não ser cobertas por outro, portanto, a cobertura e medição entre eles são complementares. Um exemplo disso são dados obtidos por uma IMU e uma câmera. A área coberta pela câmera é muito maior quando comparada com a IMU, que consegue medir somente sobre a rota de navegação.

Ganganath e Leung (2012) foram capazes de reduzir o erro da pose de uma plataforma móvel através do uso de um Filtro de Kalman Extendido (EKF) e de um Filtro de Partículas (PF). Os valores obtidos através dos filtros foram melhorados ao se utilizar a leitura de um *Kinect* para corrigir o erro acumulado da odometria a fim de estimar a pose da plataforma móvel.

Shen, Tick e Gans (2011) propõem um sistema capaz de utilizar a odometria visual

juntamente aos dados obtidos através da odometria das rodas e da IMU. O sistema proposto utiliza uma câmera fixada de forma rígida para estimar a velocidade do robô no tempo. A odometria obtida através das rodas é responsável por estimar as velocidades linear e angular, enquanto a IMU estima a velocidade angular do sistema.

Ouerghi et al. (2020a) comparam a câmera de tracking T265 da Intel e o algoritmo ORB-SLAM2 para gerar a odometria visual. Alapetite et al. (2020) realizam o estudo comparando o desempenho da T265, da odometria das rodas e os algoritmos ZED Mini e ORB-SLAM2, mas não realizam a fusão com a odometria das rodas. É colocado como trabalho futuro a avaliação do desempenho da mesma em um ambiente de baixa luminosidade, simulando situações extremas.

Neste trabalho, é realizada a fusão de dados provenientes da odometria das rodas e da odometria visual, com o objetivo de realizar uma análise qualitativa e quantitativa dos resultados obtidos. Além disso, é realizada a avaliação do desempenho da fusão em um ambiente de baixa luminosidade, verificando a capacidade de correção da odometria visual e o erro propagado ao longo do tempo.

1.4 Organização do Texto

No Capítulo 2, é apresentado o referencial teórico necessário para a compreensão da pesquisa realizada, assim como os conceitos envolvidos. As etapas de desenvolvimento do projeto e a metodologia adotada estão descritas no Capítulo 3. Os resultados obtidos com os experimentos realizados são apresentados no Capítulo 4. Por fim, no Capítulo 5, são discutidas as conclusões do trabalho, abordando os resultados obtidos, juntamente com as perspectivas de trabalhos futuros.

2 REFERENCIAL TEÓRICO

2.1 Conceitos gerais sobre Robótica Móvel

Corke (2013) define que robôs móveis são uma classe de robôs capazes de se mover pelo ambiente. Por mais que tenham funções distintas relacionada a suas aplicações, são muito semelhantes em termos do que fazem e como o fazem. Sua principal função é de se mover para algum lugar.

Este lugar pode ser especificado em termos de características físicas do ambiente, como mover-se em direção à luz, ou em termos de uma determinada coordenada geométrica ou referência do mapa. Em qualquer caso, o robô seguirá um determinado caminho para alcançar seu destino lidando com diferentes desafios, como os obstáculos presentes que podem bloquear seu caminho, um mapa incompleto ou até mesmo não ter mapas pré determinados.

Dentre os modelos de robôs móveis com rodas, os modelos mais estudados são os modelos diferencial e o robô do tipo carro, também conhecido como Ackerman. Tendo em vista que o equacionamento para atingir um determinado ponto de referência no espaço em um modelo Ackerman é mais complexo devido a restrições mecânicas do modelo, o controle utilizado pode ser simplificado para um modelo diferencial, desde que a posição determinada esteja dentro da região atingível pelo ângulo das rodas dianteiras.

2.1.1 Modelo cinemático diferencial

Analisando primeiramente a cinemática de um robô diferencial. Tendo como base o modelo de robô diferencial apresentado na Figura 1, para apresentar uma trajetória circular com base no centro de rotação instantâneo (*Instantaneous Center of Rotation* - ICR), a roda esquerda deve se mover com uma velocidade v_L sobre o raio R_L e a roda direita deve se mover com velocidade v_R sobre o raio R_R , a velocidade angular será dada por

$$\dot{\theta} = \frac{v_L}{R_L} = \frac{v_R}{R_R} \quad (2.1)$$

e desde que $R_R = R_L + W$, pode-se escrever a velocidade angular como

$$\dot{\theta} = \frac{v_R - v_L}{W} \tag{2.2}$$

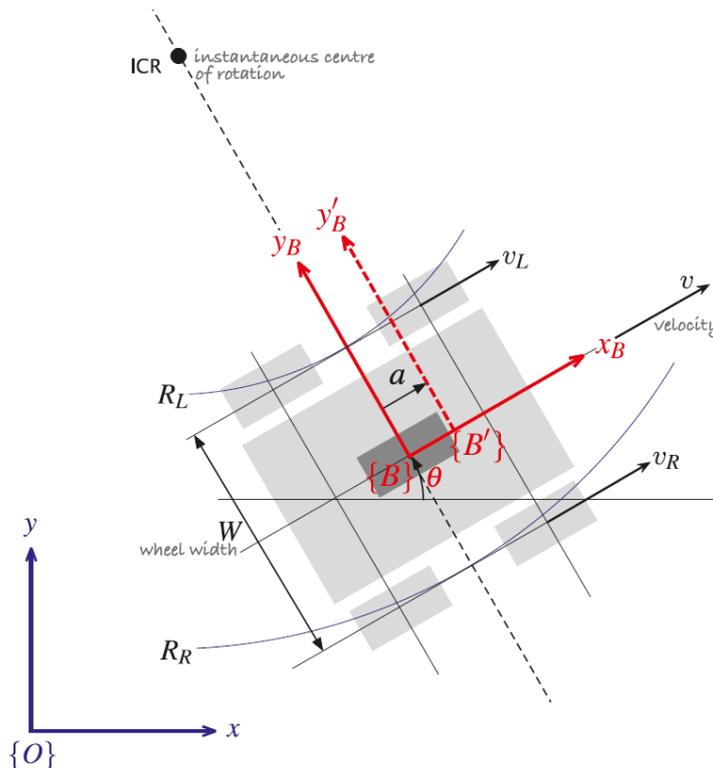
em termos de velocidade diferencial e separação das rodas W (CORKE, 2013). Desta forma, as equações de movimento finais para o robô diferencial são descritas como:

$$\dot{x} = v \cos \theta \tag{2.3}$$

$$\dot{y} = v \sin \theta \tag{2.4}$$

$$\dot{\theta} = \frac{v \Delta}{W} \tag{2.5}$$

Figura 1 – Modelo do robô diferencial do tipo Skid-Steer.



Fonte: Corke (2013).

Para mover o determinado veículo considerando o ponto de referência $\{B'\}$, ignorando a orientação, as equações de movimento para o robô diferencial podem ser escritas como

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} \cos\theta & -a\sin\theta \\ \sin\theta & a\cos\theta \end{pmatrix} \begin{pmatrix} v \\ w \end{pmatrix} \quad (2.6)$$

e desde que $a \neq 0$, esta matriz pode ser invertida, encontrando então:

$$\begin{pmatrix} v \\ w \end{pmatrix} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\frac{1}{a}\sin\theta & \frac{1}{a}\cos\theta \end{pmatrix} \begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} \quad (2.7)$$

Desta forma, é possível encontrar a velocidade linear e angular desejada para atingir uma velocidade arbitrária (\dot{x}, \dot{y}) em relação ao ponto de referência $\{B'\}$ (MARTINS; SARCINELLI-FILHO; CARELLI, 2017). Vale ressaltar que, conforme o valor do ponto de controle “a” diminuir, maiores valores serão inseridos na matriz inversa, o que irá tornar o controle mais agressivo. Quanto maior for o valor de “a”, a manobrabilidade do robô será reduzida, mas, o controle realizado será menos agressivo.

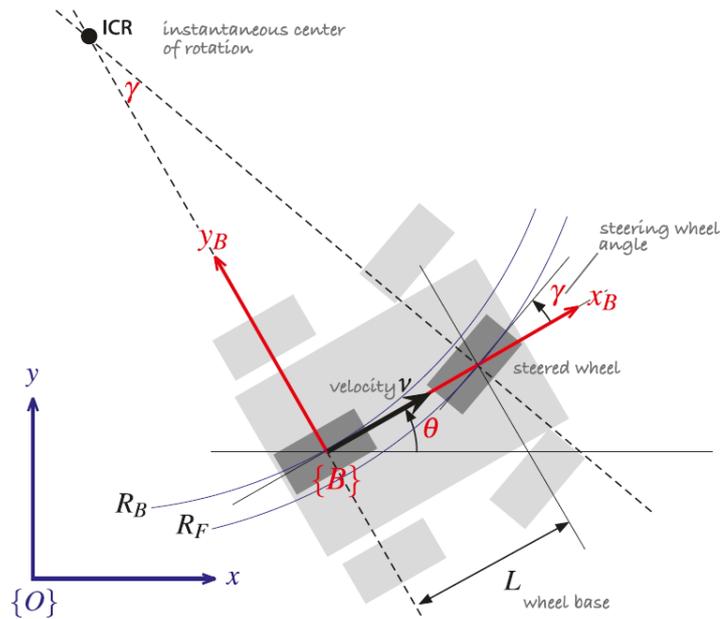
2.1.2 Modelo cinemático Ackerman

O robô de modelo cinemático Ackerman é o robô que tem como base a estrutura utilizada nos carros modernos. O Ackerman se assemelha muito ao modelo diferencial que é amplamente conhecido na robótica, com o adicional de depender de rodas de condução, sendo então necessário controlar o ângulo de virada nas rodas dianteiras de forma cooperada. Caso contrário, ocorre uma grande fricção das rodas com o solo, o que afeta seriamente o movimento de direção do robô e desgastará as rodas (ZHENG; YANG, 2020).

Comparados a outros robôs, como os omnidirecionais, possui a vantagem de poder utilizar rodas comuns, o que reduz seu custo e melhora a resistência do robô de uma forma geral. A desvantagem é que sua movimentação é limitada ao intervalo de ângulo das rodas dianteiras, não podendo fazer uma curva de raio zero, como nos modelos diferenciais e Skid-Steer (CORKE, 2013). É possível observar o estado do robô Ackerman para fazer uma curva na Figura 2. É necessário que as linhas perpendiculares às direções de movimentos das quatro rodas se cruzem em um ponto, ou seja, as quatro rodas giram em torno do mesmo ICR.

Discutindo primeiramente a cinemática inversa, a velocidade alvo das rodas é obtida a partir da velocidade desejada V_X , V_Y e w_Z dos eixos. As velocidades das rodas traseiras esquerda e direita são representadas por V_L e V_R , respectivamente. Para frente é considerado

Figura 2 – Modelo bicicleta para o robô tipo carro.



Fonte: Corke (2013).

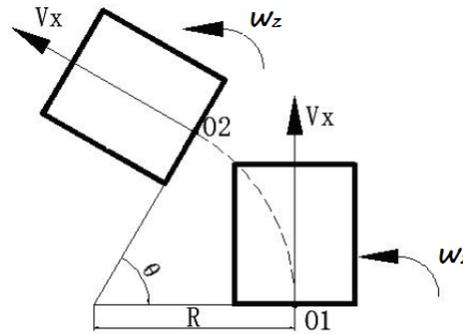
positivo e para trás é considerado negativo. O robô Ackerman requer o ângulo das duas rodas dianteiras, sendo $Angle_L$ o ângulo da roda esquerda e $Angle_R$ o ângulo da roda direita, tendo seu giro à esquerda positivo e à direita negativo. Como o robô não pode se mover horizontalmente de forma direta, a velocidade alvo no eixo Y não é discutida.

Para o robô Ackerman, a cinemática inversa é necessária para compreender a relação entre os parâmetros físicos do robô e o controle realizado no baixo nível. A relação entre V_X , w_Z e o raio de giro R devem ser determinados primeiramente. Conforme observado na Figura 3, assumindo que o robô se move com as velocidades constantes V_X e w_Z por um tempo t , o comprimento do arco O_1O_2 é a integral da velocidade V_X no tempo t : $O_1O_2 = V_X.t$. A orientação do robô é rotacionada em um ângulo θ , que é a integral da velocidade angular w_Z no dado tempo t : $\theta = w_Z.t$ Pela propriedade do círculo:

$$R = \frac{O_1O_2}{\theta} = \frac{V_X.t}{w_Z.t} = \frac{V_X}{w_Z} \quad (2.8)$$

Sabendo então as velocidades V_X , w_Z e o raio de giro $R = \frac{V_X}{w_Z}$, é possível determinar as velocidades das rodas esquerda e direita, assim como os ângulos de deflexão $Angle_L$ e $Angle_R$. Outros parâmetros importantes de serem conhecidos são a distância W entre as

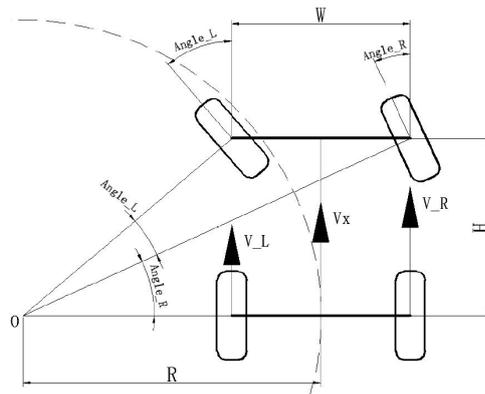
Figura 3 – Velocidade e raio de giro desejado.



Fonte: WHEELTEC (2021).

rodas e H , conforme pode ser observada na Figura 4.

Figura 4 – Velocidade desejada do robô, das rodas e ângulo de deflexão.



Fonte: WHEELTEC (2021).

Conforme determinado na Figura 4, com o robô exercendo a rotação sobre o ponto O durante todo o movimento de rotação, é possível determinar as velocidades V_L e V_R das rodas esquerda e direita:

$$V_L = V_X \frac{R - 0,5W}{R} \quad (2.9)$$

$$V_R = V_X \frac{R + 0,5W}{R} \quad (2.10)$$

Os ângulos de deflexão das rodas dianteiras $Angle_L$ e $Angle_R$ podem ser obtidos através da relação geométrica:

$$\text{Angle}_L = \tan^{-1}\left(\frac{H}{R - 0,5W}\right) \quad (2.11)$$

$$\text{Angle}_R = \tan^{-1}\left(\frac{H}{R + 0,5W}\right) \quad (2.12)$$

$$R = \frac{V_X}{w_Z} \quad (2.13)$$

Resolvendo então para a cinemática direta:

$$V_L + V_R = V_X \frac{R - 0,5W}{R} + V_X \frac{R + 0,5W}{R} \quad (2.14)$$

$$V_L + V_R = V_X \frac{R + R + 0,5W - 0,5W}{R} \quad (2.15)$$

$$V_L + V_R = 2V_X \quad (2.16)$$

$$V_X = \frac{V_L + V_R}{2} \quad (2.17)$$

Então, resolvendo para w_Z

$$V_R - V_L = V_X \frac{R - 0,5W}{R} - V_X \frac{R + 0,5W}{R} \quad (2.18)$$

$$V_R - V_L = V_X \frac{R - R + 0,5W + 0,5W}{R} \quad (2.19)$$

$$R = V_X \frac{W}{V_R - V_L} \quad (2.20)$$

Substituindo então a relação do raio:

$$\frac{V_X}{w_Z} = V_X \frac{W}{V_R - V_L} \quad (2.21)$$

$$w_Z = \frac{V_R - V_L}{W} \quad (2.22)$$

Desta forma, conhecendo as distâncias W e H do robô Ackerman, é possível determinar a relação entre as velocidades desejadas V_X e w_Z , as velocidades das rodas V_L e V_R , os ângulos de deflexão $Angle_L$ e $Angle_R$:

$$R = \frac{V_X}{w_Z} \quad (2.23)$$

$$V_L = V_X \frac{R - 0,5W}{R} \quad (2.24)$$

$$V_R = V_X \frac{R + 0,5W}{R} \quad (2.25)$$

$$Angle_L = \tan^{-1}\left(\frac{H}{R - 0,5W}\right) \quad (2.26)$$

$$Angle_R = \tan^{-1}\left(\frac{H}{R + 0,5W}\right) \quad (2.27)$$

Por fim, a partir das velocidades V_X e w_Z , é possível calcular uma nova posição do robô X_k e Y_k do robô, assim como o ângulo de rotação θ no eixo Z (CARO, 2017):

$$\begin{cases} X_{k+1} = X_k + V_X \cdot \cos(\theta) \cdot \Delta t \\ Y_{k+1} = Y_k + V_X \cdot \sin(\theta) \cdot \Delta t \\ \theta_{k+1} = \theta_k + w_Z \cdot \Delta t \end{cases} \quad (2.28)$$

2.2 Odometria

Conforme descrito por Thrun, Burgard e Fox (2005), uma maneira de armazenar dados de informações a respeito de mudanças de estados nos ambientes é através do uso de odômetros. Odômetros são sensores que medem a revolução das rodas de um robô. O

odômetro mais comumente utilizado na robótica são os *encoders*, que medem a revolução das rodas, permitindo gerar uma odometria do robô e posicionar o robô no espaço.

A maioria dos robôs comerciais disponibiliza essa estimativa integrando a posição em intervalos de tempo regulares, por exemplo, a cada décimo de segundo. Também conhecida por *Dead reckoning*, ou “Navegação por dedução”, é a estimativa da posição de um robô com base em sua velocidade estimada, direção e tempo de deslocamento em relação a uma estimativa anterior (CORKE, 2013).

Os valores integrados pelos sensores são convertidos em velocidades linear e angular, com base no modelo cinemático do robô em quem está sendo avaliado. A partir destas velocidades, a Equação 2.28 é utilizada para calcular a posição do robô no espaço, calculando a variação do deslocamento do robô a cada ciclo de tempo Δt .

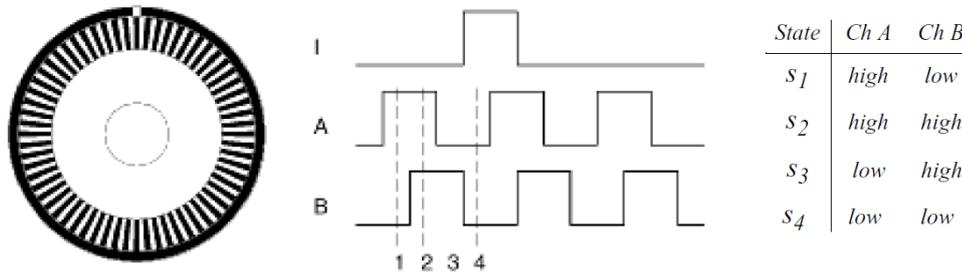
Devido aos erros de medição do sensor serem acumulados na integração, o erro de posição aumenta ao longo do tempo. Portanto, a posição precisa ser atualizada periodicamente por outros mecanismos de localização. Caso contrário, o robô não será capaz de manter uma estimativa de posição e orientação significativa a longo prazo (SIEGWART; NOURBAKHSH, 2011).

2.2.1 Encoders

Os *encoders* se tornaram o dispositivo mais popular para medir a velocidade angular e a posição dentro de um acionamento de motor, no eixo de uma roda ou mecanismo de direção. Na robótica móvel, os *encoders* são usados para controlar a posição ou velocidade das rodas e outras articulações acionadas por motor. Por serem sensores proprioceptivos, sua estimativa de posição é melhor no referencial do robô e, ao serem aplicados ao problema da localização do robô, são necessárias correções significativas (SIEGWART; NOURBAKHSH, 2011).

Siegwart e Nourbakhsh (2011) descrevem que, na robótica móvel, é comum utilizar o *encoder* em quadratura. Nesse caso, um segundo par de iluminação e detector é posicionado deslocado 90 graus em relação ao original, em termos do disco do rotor. As duas ondas quadradas resultantes podem ser observadas na Figura 5. A ordem em que a onda quadrada produz uma borda ascendente primeiro identifica a direção de rotação. Além disso, os quatro estados detectavelmente diferentes melhoram a resolução em um fator de quatro sem alterar o disco do rotor.

Figura 5 – Encoder óptico em quadratura: A relação de fase observada entre os trens de pulsos dos canais A e B é usada para determinar a direção da rotação.



Fonte: Siegwart e Nourbakhsh (2011).

2.2.2 Sensores Inerciais para Odometria

Sensores inerciais são dispositivos projetados para medir e fornecer informações sobre a orientação, velocidade angular e aceleração linear de um objeto em relação a uma referência, como o norte magnético ou um sistema de coordenadas tridimensional. Esses sensores são amplamente utilizados em diversas aplicações, como navegação, robótica, realidade virtual e sistemas de estabilização (TEDALDI; PRETTO; MENEGATTI, 2014).

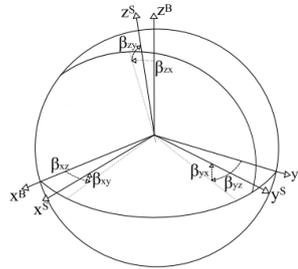
Os sensores de orientação podem ser proprioceptivos (giroscópio, acelerômetro) ou exteroceptivos (bússola). Eles são usados para determinar a orientação e inclinação do robô. Eles nos permitem, juntamente com informações de velocidade apropriadas, integrar o movimento em uma estimativa de posição. Alguns dos sensores de orientação muito populares na robótica são as IMUs (Unidades de Medição Inercial). Elas são compostas por um conjunto de conjuntos triaxiais: acelerômetros, giroscópios e frequentemente um conjunto de magnetômetros (CORKE, 2013).

Utilizando os valores dos sensores inerciais, é possível gerar a odometria do robô ao utilizar a técnica INS (Inertial Navigation System). Tal técnica consiste em integrar as leituras dos sensores ao longo do tempo, obtendo informações acerca de posição, velocidade e orientação. Isso é feito através da integração das leituras dos acelerômetros para estimar a velocidade e a posição, e da integração das leituras dos giroscópios para estimar a orientação (AQEL et al., 2016b).

Em uma IMU ideal, os conjuntos triaxiais devem compartilhar os mesmos eixos de sensibilidade ortogonais em 3D que abrangem um espaço tridimensional (Figura 6), enquanto o fator de escala deve converter a quantidade digital medida por cada sensor na quantidade física real (por exemplo, acelerações e taxas de giro) (TEDALDI; PRETTO;

MENEGATTI, 2014).

Figura 6 – Sensores não ortogonais (acelerômetros ou giroscópios) eixos (x^S, y^S, z^S) , eixos do sistema de referência do corpo (x^B, y^B, z^B) .



Fonte: Tedaldi, Pretto e Menegatti (2014).

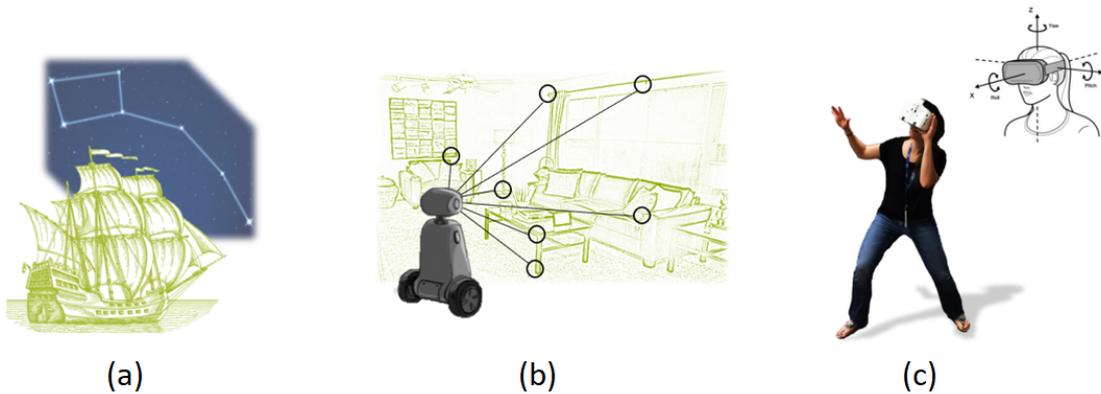
2.2.3 Odometria Visual e Odometria Visual Inercial

Odometria visual, ou *Visual Odometry* (VO), é o processo de utilizar informações de imagens consecutivas para estimar o movimento relativo do robô de uma imagem para a próxima (CORKE, 2013). A odometria visual é uma técnica de odometria mais precisa e econômica em comparação com técnicas convencionais, como a odometria das rodas e INS, com um erro de posição relativa variando de 0,1% a 2% (SCARAMUZZA; FRAUNDORFER, 2011).

O conceito central de seu funcionamento lembra tempos anteriores à bússola e ao GPS, quando os seres humanos navegavam de acordo com as estrelas, como pode ser observado na Figura 7 (a). Ao identificar constelações específicas, é possível identificar um norte (ou seja, a direção) e calcular sua própria latitude e longitude (ou seja, a localização), geralmente usando tabelas de consulta (GRUNNET-JEPSEN; FULKERSON, 2023). Por possuir um alto grau de liberdade, a odometria visual é uma boa opção para aplicações que exploram os 6 graus de liberdade, assim como a robótica (Figura 7 (b)) e realidade virtual (Figura 7 (c)).

Na visão computacional, busca-se realizar a mesma tarefa de forma mais rápida e robusta possível. Uma determinada foto eletrônica de um dado ambiente normalmente é constituída de milhões de pixels. A partir desta, é utilizado algum algoritmo de detecção de características, para condensar o conteúdo informativo sobre algumas centenas de pontos nomeados. Fazendo uma analogia com a navegação por estrelas, é como se a imagem fosse substituída por uma imagem preta, com apenas alguns pontos brancos, cada um com uma identidade (GRUNNET-JEPSEN; FULKERSON, 2023).

Figura 7 – (a) Marinheiros experientes utilizavam as estrelas para navegar. (b) Capaz de alcançar o rastreamento em 6 graus de liberdade, a odometria visual é uma excelente opção para o uso em robôs (c) e no uso de óculos de realidade virtual.



Fonte: Grunnet-Jepsen e Fulkerson (2023).

Uma característica pode ser dada como as quinas identificadas em um objeto no ambiente. Na realidade, essa é uma área extremamente ativa de pesquisa, e existem muitos algoritmos de detecção de características, como por exemplo, o estudo comparativo entre SIFT, SURF, KAZE, AKAZE, ORB, BRISK, AGAST, FAST, MSER, MSD, GFTT e Harris Corner, realizado por Tareen e Raza (2023). Uma vez detectadas, essas características são mapeadas pelo algoritmo, atribuindo a cada característica um descritor ou um vetor de características.

Este processo é repetido a cada novo quadro de imagens que é recebido e, a cada quadro consecutivo (ou quadros-chave), são comparadas e rastreadas as características entre eles. Cada característica é mapeada para o espaço 3D, e não apenas em duas dimensões. A Figura 8 é um exemplo de detecção de features em uma imagem através do algoritmo *Harris Corner Features*.

Visando cobrir limitações presentes na odometria visual, assim como a alta dependência do ambiente, surgiu a *Visual Inertial Odometry* - VIO (Odometria Visual Inercial), que consiste em integrar uma IMU ao sistema, para juntos poderem estimar de uma forma mais robusta a posição, velocidade e orientação, podendo ser facilmente utilizados em algoritmos de mapeamento e navegação já consolidados em robótica móvel (ZHANG; SCARAMUZZA, 2018).

Grunnet-Jepsen e Fulkerson (2023) descrevem que a Odometria Visual Inercial apresenta pontos que são compensados através de uma IMU. Baixas frequências de atualização por parte da odometria visual são compensadas com frequências mais altas da IMU, assim como um baixo consumo computacional e uma menor dependência do ambiente. Já a

Figura 8 – Exemplo de detecção de features pelo algoritmo *Harris Corner Features*.

Fonte: Grunnet-Jepsen e Fulkerson (2023).

odometria visual apresenta propriedades que levam em consideração o ambiente ao qual está inserida, assim como a estabilidade com o tempo e a re-localização da mesma ao detectar uma determinada característica no ambiente.

2.3 Erros de Odometria

Os sensores estão sujeitos a erros que podem afetar e, até mesmo, comprometer a leitura e confiabilidade dos dados obtidos. Tais erros podem ser causados por uma série de fatores. Alguns são inerentes ao próprio sensor, podem ser introduzidos durante o processo de medição ou fatores do ambiente, sendo alguns deles impossíveis de serem corrigidos, mesmo que por uma calibração perfeita (YOON; YU, 2017).

Siegwart e Nourbakhsh (2011) caracterizam os erros em *sistemáticos* (determinísticos) e *aleatórios* (não determinísticos).

Erros *sistemáticos* são causados por fatores ou processos que podem ser modelados. Esses erros são, portanto, determinísticos, ou seja, previsíveis. São exemplos de erros sistemáticos calibrações inadequadas de um determinado sensor, inclinações não modeladas de um piso do ambiente e câmeras descalibradas devido a colisões.

Erros *aleatórios* não podem ser modelados e nem mitigados por sensores mais precisos. Estes erros podem apenas ser descritos em termos probabilísticos, ou seja, estocasticamente. São exemplos de erros aleatórios atraso de comunicação entre controladores e atuadores,

falha dos sensores, e reflexão indevida da luz em sensores que utilizam a luminosidade, como sensores LiDAR.

2.3.1 Erros na Odometria das Rodas

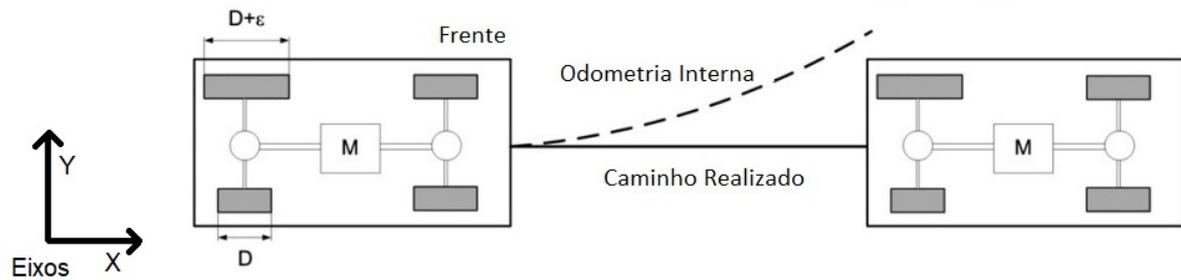
A odometria a partir das rodas pode ser atualizada com base na leitura de *encoders*. Como base no modelo cinemático, sua posição atual é integrada com base na posição anterior e na variação de movimento em cada ciclo. Ou seja, por se tratar de uma integração, o erro de posição acumula com o tempo. Desta forma, quanto maior for a rotina realizada por um dado robô, maior será o erro acumulado pela odometria das rodas. Existem várias fontes causadoras desses erros (SIEGWART; NOURBAKHS, 2011):

- Resolução limitada dos sensores (incrementos de tempo, resolução de medição, etc);
- Desalinhamento das rodas (determinístico);
- Incerteza no diâmetro das rodas e, em particular, diâmetros de rodas desiguais (determinístico);
- Variações no ponto de contato da roda (aleatório);
- Contato desigual com o piso (deslizamento, superfície não plana, etc.).

Alguns desses erros (sistemáticos) podem ser eliminados ou atenuados pela calibração adequada do sistema. De qualquer forma, ainda existem erros aleatórios que continuam no sistema, que resultam no acúmulo de erro com o tempo (BORENSTEIN et al., 1997). Do ponto de vista geométrico, os erros podem ser classificados em erro de distância, erro de curva e erro de derrapagem .

Siegwart e Nourbakhsh (2011) também descrevem que o erro de distância ocorre ao integrar o movimento das rodas para obter a distância percorrida. O erro de curva surge quando a distância percorrida durante uma curva é integrada, levando em consideração a diferença de rotação das rodas. Em relação ao erro de derrapagem, essa ocorre devido à discrepância no deslizamento entre as rodas de um robô, resultando em imprecisões na orientação. No regime permanente, os erros de curva e derrapagem superam em muito os erros de distância, visto que sua contribuição para o erro de posição não é linear. Considerando um robô onde sua posição inicial seja perfeitamente conhecida, movendo-se para frente ao longo do eixo-x. O erro na posição y introduzido terá uma componente $d\text{sen}\Delta\theta$, que pode se tornar bastante grande a partir do momento em que o erro angular aumenta (Figura 9).

Figura 9 – Erro de odometria devido a um erro entre o diâmetro das rodas.



Fonte: Lee, Chung e Yoo (2010).

Com o passar do tempo, à medida que um robô móvel se move pelo ambiente, o erro de rotação entre sua referência interna e sua posição inicial cresce rapidamente. Ou seja, à medida que o robô se distancia da origem, o erro linear resultante aumenta consideravelmente (SIEGWART; NOURBAKHS, 2011).

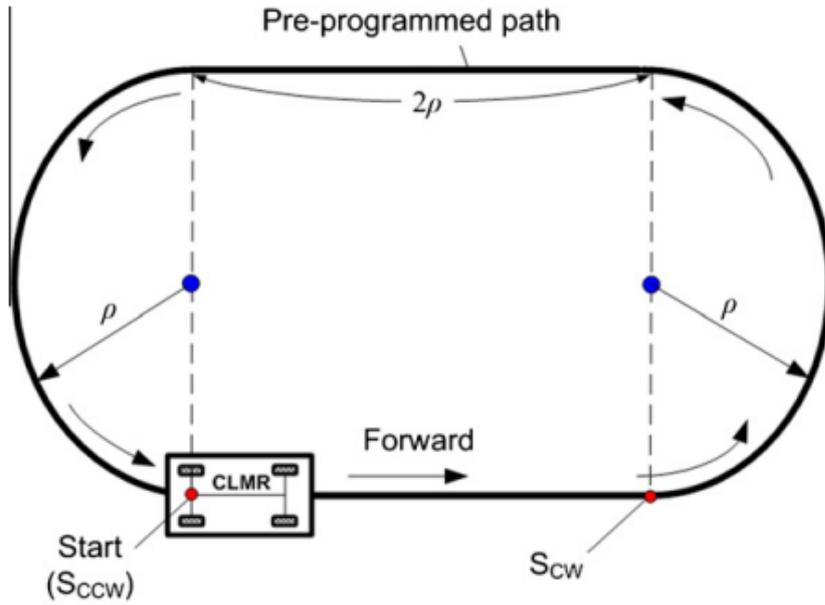
Lee, Chung e Yoo (2010) apresentam que dentre os erros determinísticos, os mais comuns são divididos entre *Tipo A* e *Tipo B*. O erro do Tipo A acontece quando ocorrem o erro de modelagem a respeito da distância entre as rodas traseiras. Já o erro do Tipo B acontece quando ocorre um erro de modelagem do diâmetro das rodas. Os erros são calculados com base em uma referência dada a partir de um caminho pré programado (Figura 10), inicializando no sentido anti-horário no ponto S_{CCW} ou inicializando no sentido horário no ponto S_{CW} .

Com o robô se movendo ao longo do caminho de referência, é possível observar nas Figura 11 e 12 como a odometria se comporta, sendo esta a linha tracejada, com o robô percorrendo o caminho no sentido horário e anti horário quanto ocorre os erros de Tipo A e Tipo B. Na Figura 11, a distância entre as rodas nominal é menor do que a atualmente configurada, apresentando um erro do Tipo A. Já na Figura 12, o diâmetro da roda traseira esquerda é ligeiramente maior do que o diâmetro da roda traseira direita, apresentando um erro do Tipo B.

2.3.2 Erros na Odometria causados por IMUs

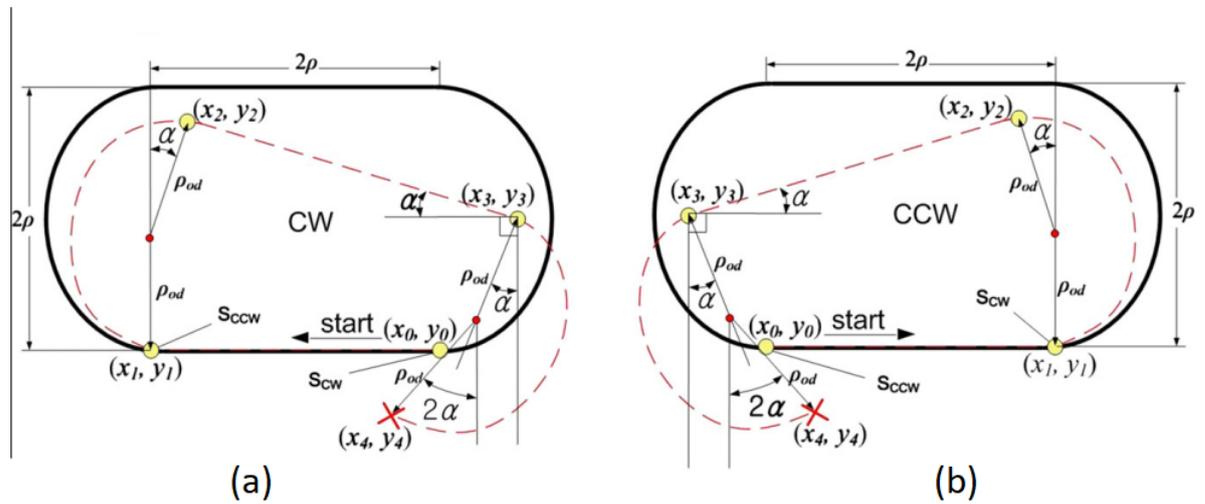
IMUs mal calibradas geram medições com erros sistemáticos não negligenciáveis (TEDALDI; PRETTO; MENEGATTI, 2014). Considerando as leituras de um dado sensor - giroscópio ou acelerômetro - o sinal medido está relacionado ao valor verdadeiro e desconhecido de x , sendo aplicado um fator de escala s , um deslocamento b e um ruído aleatório ε (Equação

Figura 10 – Caminho pré programado a ser executado pelo robô.



Fonte: Lee, Chung e Yoo (2010).

Figura 11 – Odometria gerada contendo um erro de Tipo A, onde o valor com a distância entre as rodas nominal é menor do que a atualmente configurada, inicializando no sentido horário (a) e anti horário (b).



Fonte: Lee, Chung e Yoo (2010).

2.29).

$$x = sx + b + \varepsilon \tag{2.29}$$

Embora seja possível conseguir altas taxas de quadros, os algoritmos possuem um custo computacional elevado, demandando uma maior capacidade de processamento e mais energia, além de captar menos luz ao tentar aumentar sua velocidade de captura. Mas, o problema mais grave a ser destacado é de um ambiente com poucas características detectáveis. Ao se mover em um ambiente com baixa quantidade de características, vale ressaltar em especial paredes brancas, a odometria visual fica muito prejudicada e dependente das poucas características detectadas em seu campo de visão, mostrando assim a vantagem de se utilizar câmeras com lente olho de peixe para obter um maior campo de visão (GRUNNET-JEPSEN; FULKERSON, 2023). Por outro lado, o uso de lentes olho de peixe inclui muita distorção nas imagens.

2.4 Fusão Sensorial para redução do Erro de Odometria

Diversas definições podem ser encontradas na literatura acerca de fusão sensorial. A abordada neste trabalho será a descrita por Elmenreich (2002), o qual descreve que a fusão sensorial é a combinação de dados providos por diversos sensores afim de obter algo melhor do que o uso dessas fontes de forma individual. Dentre as definições presentes na literatura o objetivo comum de realizar a fusão é obter uma maior precisão e qualidade dos dados e desempenho final do robô (HALL; LLINAS, 1997)(CASTANEDO, 2013).

As técnicas utilizadas para a fusão em robótica móvel são influenciadas pelos objetivos das aplicações que vão realizar a modelagem do ambiente para que o robô se comporte de uma forma mais fidedigna com a situação. As três formas fundamentais para combinar os dados dos sensores são tidas por: competitiva, complementar e cooperativa.

Quando realizada de forma competitiva, os sensores são configurados de forma que podem ter várias fontes de medição para um dado específico. Desta forma, dados de diferentes sensores podem ser fundidos mesmo se os dados possuírem taxas de aquisições diferentes, fundindo os dados com base em sua confiabilidade. Uma grande vantagem desta forma de fusão é uma maior robustez a falhas ao combinar dados de forma redundante (MARZULLO, 1990).

A fusão complementar garante que os sensores não dependam entre si, mas que o dado lido por eles seja complementado. Um exemplo é quando dados obtidos através de uma câmera possui seus pequenos erros acumulados corrigidos através de uma IMU. Uma desvantagem deste tipo de fusão é que em determinadas condições os sensores podem ter sua eficiência reduzida, como uma câmera ao ter sua visibilidade reduzida (BROOKS; IYENGAR, 1998).

O método cooperativo utiliza a informação de dois sensores para obter um dado que não seria possível ser obtido de forma única. Implementar a fusão cooperativa é complicado devido a sensibilidade a imprecisões nas participações individuais dos sensores (BROOKS; IYENGAR, 1998). Sendo assim, a fusão competitiva aumenta a robustez do sistema, enquanto a cooperativa e complementar estendem e aumentam o grau de liberdade dos sensores. Estes métodos estão subjetivamente ligados aos tipos de sensores. De qualquer forma, as três formas descritas não são mutuamente exclusivas, tendo aplicações que implementam aspectos de um ou mais dos três tipos.

É necessário o uso de algoritmos para traduzir a entrada de diversos sensores em valores confiáveis para serem utilizados nos subsistemas de navegação. Usualmente são implementados algoritmos para lidar com modelos lineares e não lineares. Para localizar o robô, alguns sensores padrões vêm sendo adotados e métodos de fusão desenvolvidos. Estes algoritmos utilizam uma série de equações com um custo computacional reduzido para estimar o estado de um processo. Sinha et al. (2008) categoriza os métodos utilizados pelos algoritmos em dois grupos: estimador de estados e fusão de decisões.

O estimador de estados é utilizado para determinar antecipadamente o estado de um sistema que muda continuamente suas medições. É comumente utilizado em algoritmos de fusão de dados por possibilitar a aquisição de diversos sensores e fontes, visando obter um estado final derivado das observações realizadas. Os dois métodos de estimador de estados mais comuns são o Filtro de Kalman e o Filtro de Partículas. O Filtro de Kalman (KF) é um estimador eficiente utilizado em vários campos para estimar estados indeterminados de um sistema. Uma série de aplicações foram desenvolvidas utilizando como base o Filtro de Kalman, assim como módulos da navegação e *tracking* de objetos.

A fusão de decisões é uma forma de fusão de dados que combina decisões de diferentes classificadores em uma decisão mútua sobre uma decorrida atividade. Este método de fusão reduz o nível de incerteza ao maximizar uma medição que se sobressaia (TEXIER et al., 2019). Essas técnicas frequentemente usam informações simbólicas e processam a fusão através de um objetivo, ao mesmo tempo que contabiliza incertezas e restrições. Os dois métodos de fusão de decisão mais comuns são a Abordagem Bayesiana e a Abordagem de Dempster-Shafer.

3 METODOLOGIA

Este trabalho pode ser classificado como uma pesquisa aplicada quanto à sua natureza, visto que os resultados obtidos foram destinados a solucionar um problema específico. Ao final deste trabalho, deseja-se obter uma melhor leitura de odometria para ser utilizada nos algoritmos de robótica móvel.

No que diz respeito aos seus objetivos, trata-se de uma pesquisa exploratória, já que o trabalho desenvolvido busca comparações e análises para uma discussão mais detalhada. Em relação à abordagem adotada, pode ser classificado como pesquisa quantitativa, pois busca coletar e analisar dados numéricos para identificar padrões, tendências e relações a fim de melhorar a precisão da odometria.

Com relação aos procedimentos técnicos, pode-se concluir que esta é uma pesquisa experimental, uma vez que, a partir dos dados coletados e da identificação dos padrões, o fluxo de dados e a configuração do sistema serão alterados quando julgado necessário, visando obter uma odometria mais precisa e com menor acúmulo de erros.

3.1 Materiais

Os materiais utilizados foram uma plataforma robótica móvel, uma câmera para gerar a odometria visual, dois computadores e um sistema de captura de movimento, disponibilizados pelo Núcleo de Assistência Assistiva (NTA), Laboratório de Computação de Alto Desempenho (LCAD) e Lab AIR da UFES.

A plataforma robótica móvel utilizada é a plataforma de modelo cinemática Ackerman (R550 (AKM) Plus Wheeltec, China)(Figura 13). Esta possui um motor de passo na parte dianteira para controlar a direção, enquanto as rodas traseiras são responsáveis pela tração. Este robô é fabricado pela empresa Wheeltec, que fabrica robôs com código aberto, com o objetivo de fomentar o desenvolvimento e aprendizado no ramo da robótica móvel.

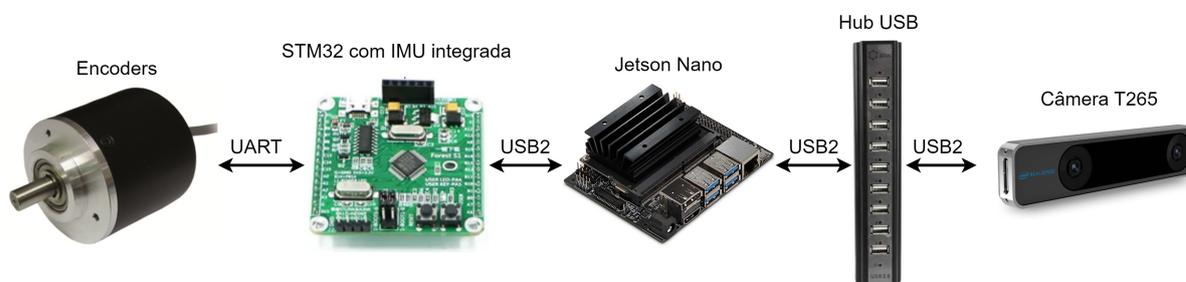
Este robô possui sensores integrados, incluindo encoders nas rodas e uma IMU, que são fundamentais para realizar a estimativa da odometria durante o deslocamento. Além disso, seu sistema operacional está embarcado em um kit de Desenvolvedor Nvidia Jetson Nano, conectado a um Hub USB, o que permite a integração de sensores externos (Figura 14).

Figura 13 – Robô R550 (AKM) Plus modelo Arkeman da empresa Wheeltec.



Fonte: WHEELTEC (2021).

Figura 14 – Sensores nativos do robô Ackerman e suas conexões.



Fonte: Produção do próprio autor.

A escolha deste robô foi baseada em sua robustez, versatilidade e desempenho, além de um fator crucial: seu código-fonte aberto. Isso permite o acesso e a manipulação das variáveis e sensores internos, proporcionando um melhor entendimento do processamento em níveis mais baixos e mais altos. Essa característica torna o robô adequado para o objetivo deste estudo, que é otimizar a configuração dos sensores visando obter uma odometria mais precisa.

A câmera T265 (Intel® RealSense™ Tracking Camera T265, Estados Unidos) de *tracking* da Intel (Figura 15) foi selecionada devido às suas características funcionais, como baixo consumo de energia e capacidade de realizar o algoritmo de SLAM (*Simultaneous Localization and Mapping*) internamente com uma IMU. Além disso, essa câmera permite a integração da odometria proveniente dos encoders das rodas em suas leituras, o que torna o sistema mais robusto e confiável. Além disso, a câmera é de fácil integração e oferece suporte a uma variedade de plataformas de desenvolvimento, ampliando suas possibilidades de aplicação.

Ela usa um algoritmo de código fechado da Intel que funciona com base na VIO para realizar um rastreamento nos 6 graus de liberdade. Esta é composta por um conjunto

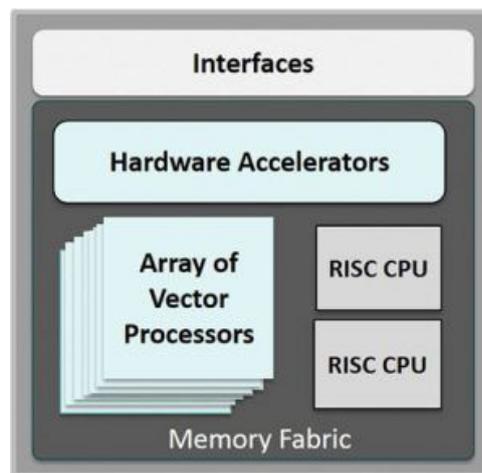
Figura 15 – A câmera Intel RealSense Tracking Camera T265 possui um algoritmo de SLAM embarcado que usa a VIO para calcular sua posição e orientação no plano 3D.



Fonte: Grunnet-Jepsen e Fulkerson (2023).

de câmeras olho de peixe, incorporada por um giroscópio BMI055 de 200 Hz da Bosh e um acelerômetro de 62.5 Hz sincronizados por hardware, além de uma Unidade de Processamento Visual (VPU) Intel® Movidius™ Myriad™ 2 (Figura 16).

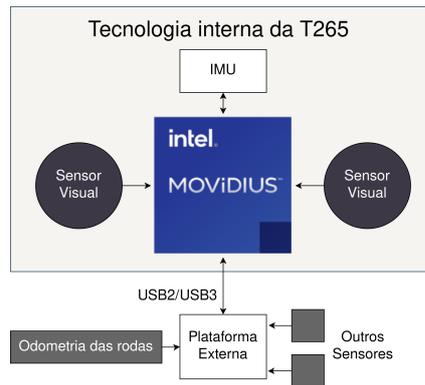
Figura 16 – Arquitetura interna da Intel® Movidius™ Myriad™ 2 VPU.



Fonte: Intel (2022).

A VPU executa o algoritmo de SLAM de forma embarcada, fundindo os dados dos sensores para obter o rastreamento no espaço 3D, com um baixo consumo de potência (1.5 W), forçando informações a uma taxa de 200 Hz. É possível integrar a leitura da odometria a partir das rodas em tal câmera para aumentar a robustez do sistema (INTEL, 2022). Para isso, é necessário realizar a transformação de referencial da odometria das rodas para o referencial da câmera. O diagrama de blocos do fluxo de dados do robô para a câmera pode ser observado na Figura 17.

Figura 17 – Diagrama de blocos da Intel® RealSense™ Tracking Camera T265.

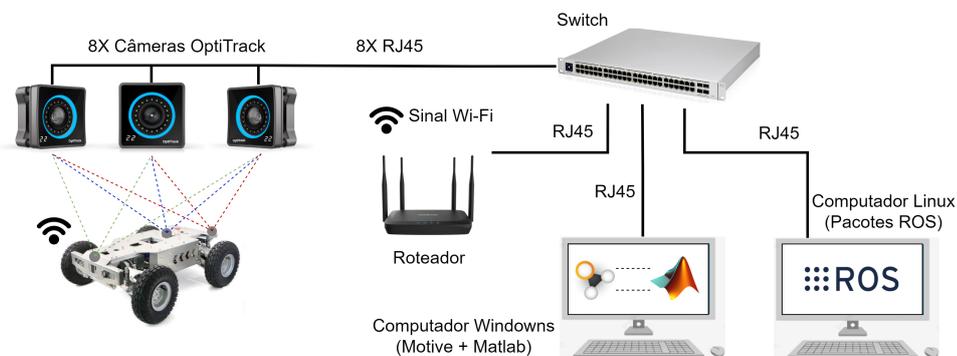


Fonte: Adaptado de Intel (2022).

Uma característica importante a ser ressaltada a respeito do SLAM realizado a partir da VIO, é a relocalização. A relocalização é a capacidade do algoritmo de usar as características que já viu antes para reconhecer quando retornou a um local com característica previamente detectada. Ou seja, se o robô realiza uma determinada rotina e retorna a um ponto com uma característica previamente detectada em seu banco de dados, realiza a relocalização da sua posição atual em meio às referências estáticas previamente determinadas, geralmente diminuindo o erro em relação à origem (INTEL, 2022). Tal funcionalidade é de fundamental importância para entender o comportamento da T265.

Visando gerar uma referência absoluta para o trajeto realizado pelo robô nos experimentos, foi utilizado um sistema de captura de movimentos, chamado de OptiTrack. O sistema utilizado é composto pelo software *Motive* (OptiTrack, Estados Unidos) em conjunto com oito câmeras do modelo *Prime^x 22* (OptiTrack, Estados Unidos).

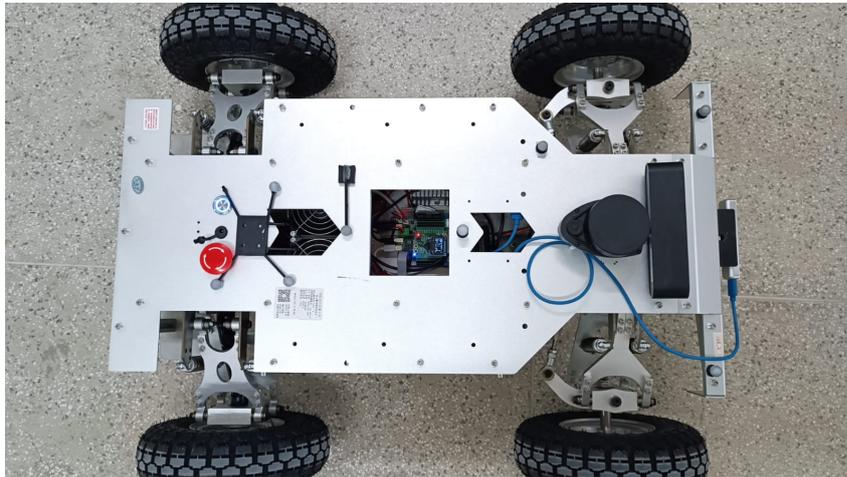
Uma visão geral do *setup* experimental utilizado pode ser observado na Figura 18.

Figura 18 – Visão geral do *setup* experimental utilizado.

Fonte: Produção do próprio autor.

São posicionados no robô marcadores reflexivos (Figura 19), que são projetados para refletir a luz infravermelha emitida pelas câmeras do sistema do OptiTrack. Os marcadores são utilizados pelo *software Motive* para criar um corpo rígido e, a partir do dado corpo, estimar a posição e orientação do robô.

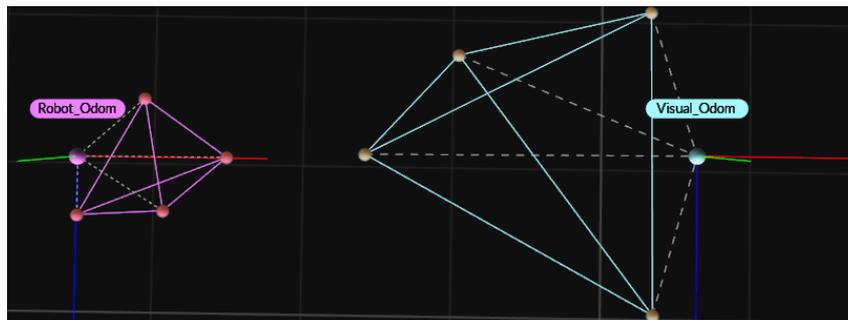
Figura 19 – Marcadores posicionados no robô



Fonte: Produção do próprio autor.

É importante ressaltar que, ao criar um corpo dentro do *software Motive*, seu ponto de referência será inicializado em um ponto geométrico que é a média espacial dos marcadores utilizados. Ou seja, é fundamental reposicionar a referência do corpo para o ponto de interesse a ser analisado. Na Figura 20 é possível observar os corpos rígidos inicializados dentro do *software Motive*, tendo seus referenciais já realocados para os pontos de interesse.

Figura 20 – Corpos rígidos inicializados para o referencial do robô e da câmera



Fonte: Produção do próprio autor.

Por meio de um nó do ROS é estabelecida uma comunicação através do protocolo VRPN (Virtual-Reality Peripheral Network) entre os computadores Linux e Windows utilizando o pacote *vrpn_client_ros* do ROS para ter acesso aos valores aos corpos rígidos criados pelo Motive. A partir de tais valores, é criada uma mensagem de odometria completa para

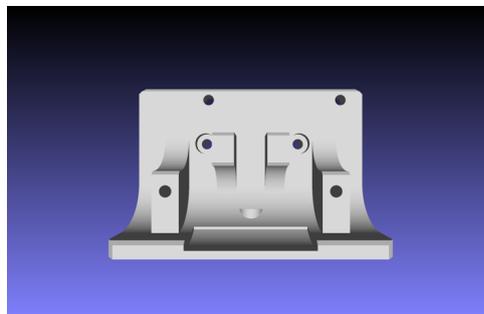
cada corpo identificado, incluindo posição, velocidades relativas e orientação através do uso de quatérnions e publicada em um tópico do ROS.

A comunicação entre o robô e os computadores é realizada via conexão Wi-Fi. O robô é definido como a central de processamentos do ROS, visto que nele são processados os dados com maiores frequência, como por exemplo, a câmera T265, que publica a mensagem de VIO com uma frequência de 200 Hz.

3.2 Integração do robô Ackerman com a câmera T265

Primeiramente, a câmera foi posicionada na parte posterior do robô, de forma que não comprometesse o uso de sensores nativos fornecidos pelo fabricante, assim como um sensor LiDAR e uma câmera de profundidade. Foi fabricada uma peça em uma impressora 3D que possui encaixe para a câmera T265 e das câmeras da série D400 da Intel (Figura 21), o que viabiliza seu reuso em conjunto com outras câmeras em futuras aplicações (SCHMIDT, 2023).

Figura 21 – Suporte 3D utilizado para fixação da câmera T265.



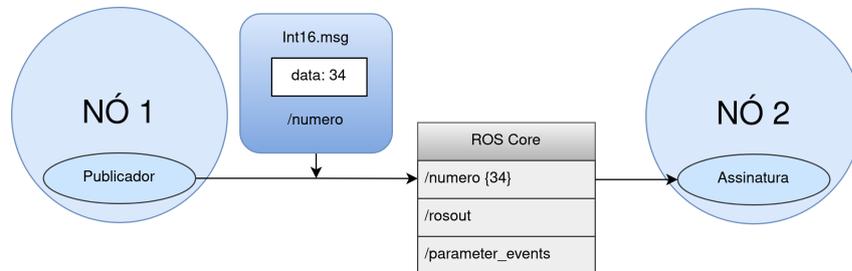
Fonte: Schmidt (2023).

Para realizar a integração da câmera, foi utilizado o *middleware* ROS, como um intermediário entre os diferentes componentes do sistema. O ROS (*Robot Operating System*) é um kit de desenvolvimento de software de código aberto para aplicações de robótica. O ROS fornece um conjunto abrangente de bibliotecas, ferramentas e recursos que ajudam na comunicação entre os diferentes componentes do sistema, como sensores, atuadores, algoritmos e processos (ROS, 2021).

Os algoritmos no ROS são inicializados através de um arquivo *launch* que contém os parâmetros necessários para a inicialização. O *software* é inicializado como um nó no sistema do ROS, se comunicando com outros nós através de tópicos que são registrados no sistema através de mensagens, onde são atualizados por publicações (*publishers*) e

requisitados através de uma assinatura (*subscriber*) (ROS, 2023). Conforme ilustrado na Figura 22, é registrado no ROS (ROS Core) através de uma mensagem um tópico /numero contendo o número inteiro 34.

Figura 22 – Fluxo de dados no ambiente ROS



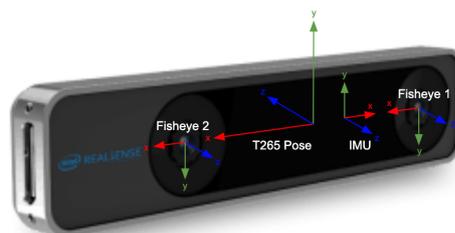
Fonte: Produção do próprio autor.

O robô possui instalado o Ubuntu 18.04 e o ROS Melodic. Foi necessário instalar um kit de desenvolvimento (SKD), da Intel, que fosse compatível com a versão do Ubuntu e do ROS instalados no sistema. Em seguida, foi feita a instalação do pacote RealSense específico para o ROS. Dentro do pacote, foram criados os arquivos *launch* para inicializar a câmera utilizando o pacote ROS. A transformação de referencial da câmera para o referencial do robô é realizada através de um arquivo do formato JSON, que é utilizado como parâmetro no *launch* que inicializa a câmera.

3.3 Transformações de referencial

Foi necessário realizar a transformação de referencial para realizar a integração da odometria das rodas com o algoritmo de *tracking* da câmera. Para isso, é necessário observar que os referenciais adotados pela câmera são diferentes dos referenciais padrões adotados na robótica (Figura 23).

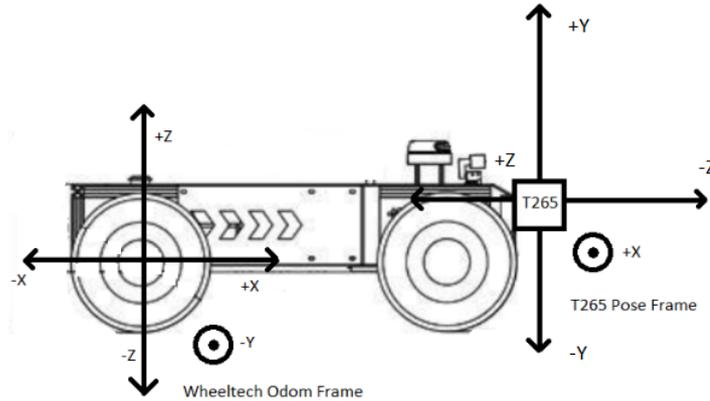
Figura 23 – Sistema de coordenadas utilizadas pela câmera T265.



Fonte: Grunnet-Jepsen e Fulkerson (2023)

Na Figura 24, é possível observar a relação entre a referência base do robô, centralizada entre as rodas, que é o ponto onde a odometria do robô é calculada a partir da leitura dos *encoders*, e o local onde foi posicionada a câmera, assim como seu referencial no espaço tridimensional.

Figura 24 – Referencial do robô e o referencial onde a câmera foi posicionada.



Fonte: Produção do próprio autor.

Conforme orientações do próprio fabricante, é necessário deslocar as coordenadas câmera para o referencial onde está posicionada a odometria das rodas. Isto é representado pela a matriz de transformação H_1 (Equação 3.1) que converte coordenadas da câmera para o referencial da odometria das rodas. Tal matriz inclui, além da translação, uma inclinação de 5° no eixo X da câmera devido a seu posicionamento.

$$H_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\left(\frac{1}{36}\pi\right) & -\text{sen}\left(\frac{1}{36}\pi\right) & -0,1038 \\ 0 & \text{sen}\left(\frac{1}{36}\pi\right) & \cos\left(\frac{1}{36}\pi\right) & 0,68 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.1)$$

Além disso, visando utilizar a leitura da câmera em um Filtro de Kalman, é necessário transformar a leitura da câmera para o referencial do robô para que o filtro tenha como entrada leituras de pontos correspondentes no ambiente. Os valores retornados pela VIO feita pela câmera estão deslocados em 0,68 metros no eixo X do referencial do robô e são inicializados em zero. Portanto, foi criada uma odometria correspondente com esse deslocamento para possibilitar a devida transformação de referencial.

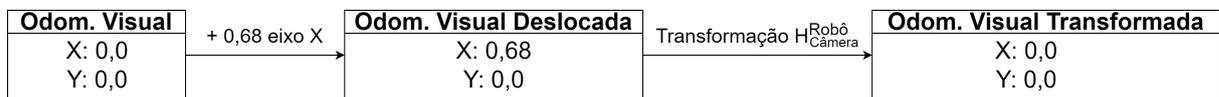
A partir da VIO que inclui essa transformação, é aplicada uma transformação de referencial no plano Z, deslocando-a para o referencial do robô. Sendo $H_{C\grave{a}m\grave{e}r\grave{a}}^{Rob\grave{o}}$ a matriz de transfor-

mação das leituras feitas no referencial da câmera para o referencial do robô visto que a única translação necessária a ser realizada é a referente ao posicionamento da câmera.

$$H_{C\grave{a}mera}^{Rob\hat{o}} = \begin{bmatrix} 1 & 0 & 0 & -0,68 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

Esta transformação é aplicada ao valor de odometria visual deslocada no espaço, conforme apresentado no diagrama apresentado na Figura 25.

Figura 25 – Fluxo das transformações aplicadas na odometria visual.



Fonte: Produção do próprio autor.

3.4 Validação Experimental

Foram realizados experimentos de forma teleoperada para realizar a validação da integração do sistema, assim como a identificação de possíveis padrões que ajudem a entender o comportamento dos resultados. Além disso, foram definidos experimentos automatizados através de um seguidor de caminho para que ocorra a repetibilidade da rotina a ser realizada pelo robô, possibilitando comparar o erro acumulado entre elas.

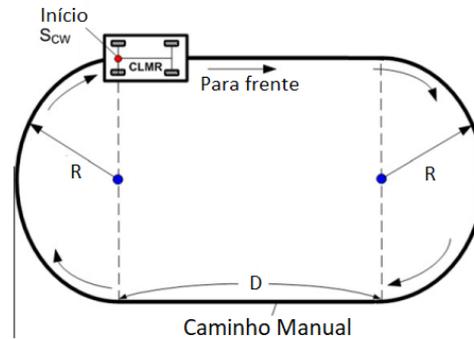
3.4.1 Experimentos Preliminares: Teleoperação

Com o objetivo de validar o funcionamento do sistema e compreender o comportamento da câmera, foram realizados experimentos iniciais nos quais o robô executou uma rotina de forma teleoperada. Esses experimentos permitiram verificar o desempenho tanto da odometria das rodas quanto da odometria da câmera, fornecendo informações essenciais sobre o funcionamento e a precisão do sistema.

Os primeiros experimentos foram realizados com base no caminho exemplificado na Figura 26. Em cada experimento, o robô realizou trechos retos arbitrários de comprimento D

e curvas com o raio mínimo permitido pelo robô $R = 1,318$ m, segundo o manual do fabricante.

Figura 26 – Caminho realizado pelo robô de forma teleoperada.



Fonte: Produção do próprio autor.

O comportamento da odometria das rodas pode ser observado na Figura 27. É possível concluir que o comportamento apresentado é semelhante ao apresentado na Figura 11, que é um erro de odometria do Tipo A, onde a distância entre as rodas traseiras está mal calibrada no baixo nível. O que é chamado de baixo nível neste trabalho são os códigos embarcados na STM32 que são responsáveis por realizar a aquisição dos dados dos *encoders* e realizar o controle de velocidade das rodas.

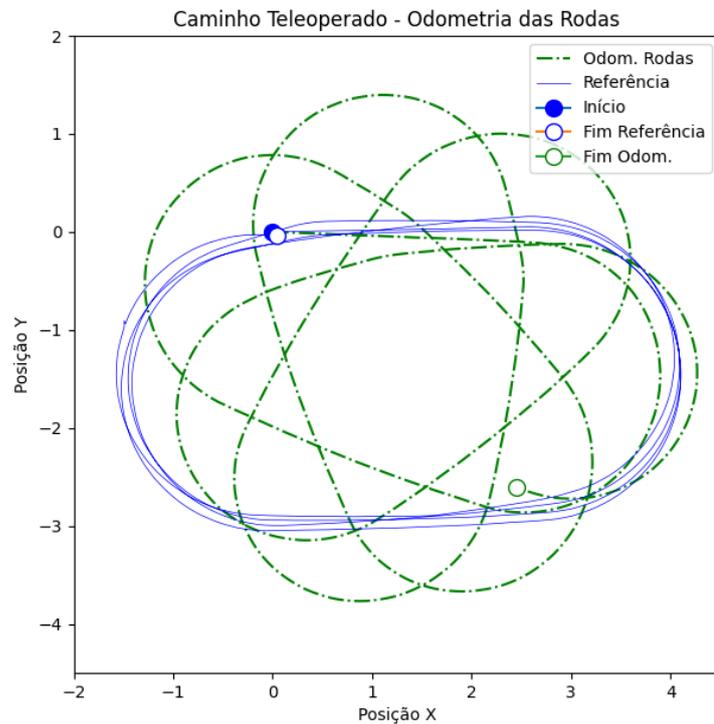
Para validar essa afirmação, realizou-se a medição do raio efetivamente percorrido pelo robô nessa trajetória. Foi enviado ao robô um comando de velocidade para ele descrevesse o raio de rotação mínimo, medindo o valor realizado através do sistema de captura de movimentos. Obtendo-se o valor de $R = 1,3792$ m.

Sendo assim, se o robô sempre percorresse o mesmo trajeto de forma fixa, seria possível corrigir esse erro de forma indireta através da Equação 2.22. Isso poderia ser feito multiplicando o valor da Velocidade Angular V_Z por uma constante que levasse em consideração a relação entre o raio do centro de rotação efetivamente percorrido pelo robô e o raio configurado no baixo nível (Equação 3.3).

$$V_Z \cdot \frac{1,3792}{1,318} = 1,046433 \cdot V_Z \quad (3.3)$$

A calibração foi realizada multiplicando na odometria final a velocidade angular do robô pela constante encontrada através da relação dos raios. Realizando tal calibração, foi possível obter valores de odometria que se aproximam da referência estabelecida pelo robô, conforme é possível observar na Figura 28.

Figura 27 – Odometria pura das rodas e sua trajetória real.



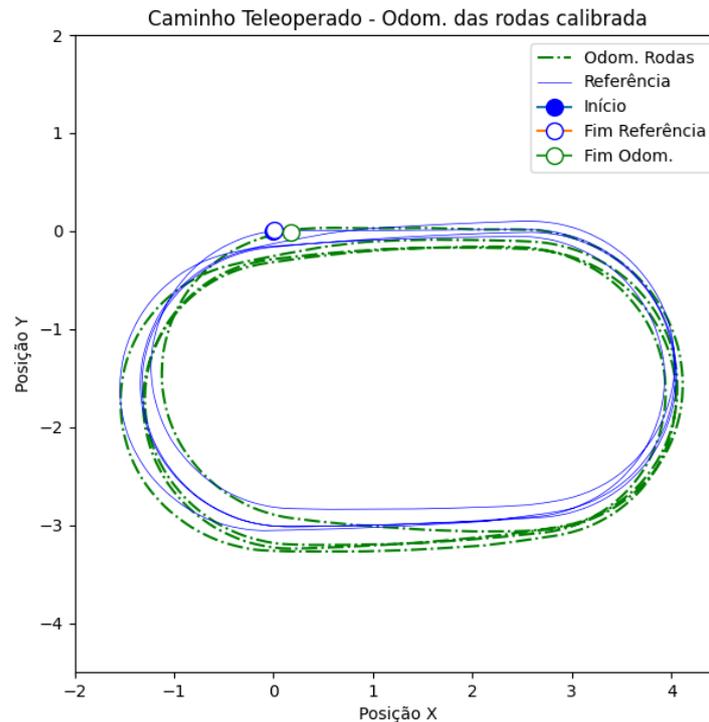
Fonte: Produção do próprio autor.

Visando a validação dos algoritmos de correção e filtragem e sua capacidade de lidar com os erros sistemáticos, e tendo noção da magnitude do erro e sua propagação quando integrado na câmera, a odometria das rodas utilizada nos experimentos será a odometria não calibrada.

O comportamento apresentado pela VIO acoplada a T265 está apresentada na Figura 29. O primeiro ponto importante a ser discutido é que, por estar posicionada em um referencial diferente, a câmera não descreve uma trajetória igual a que está centrada no referencial da odometria, sendo bem visíveis os pontos onde ocorreu o ajuste manual durante a teleoperação para que a referência da odometria das rodas seguisse o trajeto desejado.

É possível observar que, neste experimento em específico, a câmera calculou erroneamente a odometria inicial, calculando somente a partir do momento em que o robô começa a realizar a curva. Tal comportamento pode ser associado ao ambiente onde a parede é predominantemente branca, para o qual o algoritmo de VIO possui dificuldade de detectar características iniciais. Outro ponto importante a ser discutido é sobre o comportamento observado pelo formato do gráfico final da trajetória descrita pela própria câmera. O formato final é muito semelhante ao formato da referência, mas, deslocado no espaço. Dito

Figura 28 – Odometria das rodas após calibração e sua referência real.



Fonte: Produção do próprio autor.

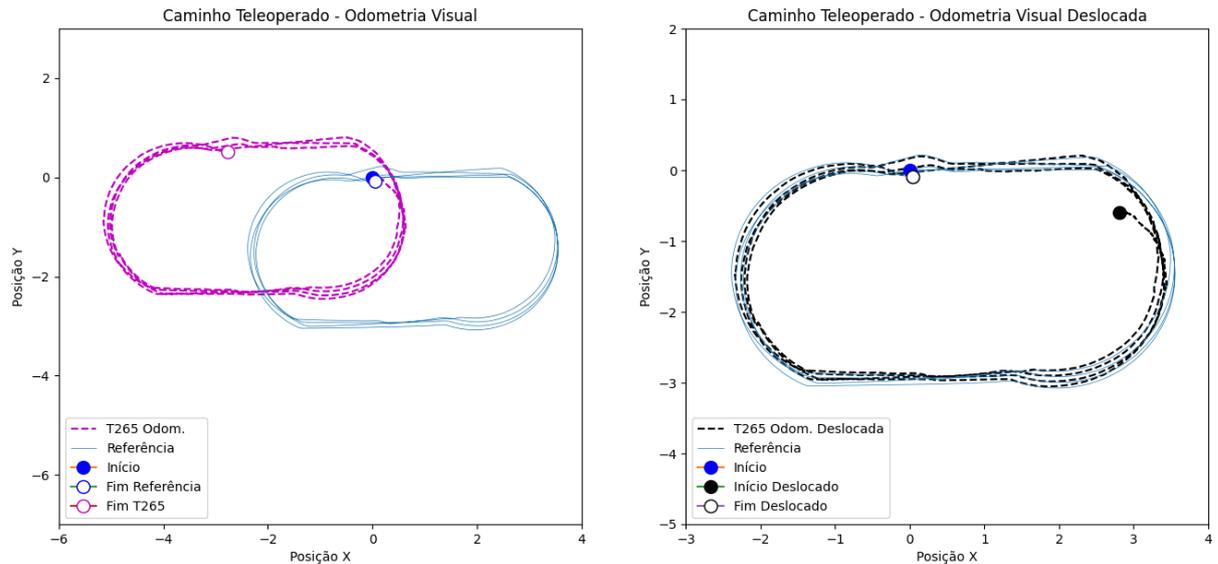
isso, pode-se concluir que a câmera é capaz de realizar a relocalização de uma forma mais robusta mas, caso ocorra problemas relacionados ao ambiente ou nos seus pontos iniciais, sua leitura terá uma alta precisão, mas baixa acurácia, tendo leituras válidas, porém, não confiáveis.

Há disponível na comunidade do ROS pacotes para realizar a fusão sensorial com o objetivo de melhorar odometria através de múltiplos sensores, sendo algum deles o *robot_pose_ekf* e o *robot_localization* (MOORE; STOUCH, 2014). O utilizado neste trabalho é o *robot_pose_ekf*, que realiza a fusão dos sensores para estimar a pose de um robô móvel. Este pacote utiliza os dados provenientes de sensores inerciais e sensores de odometria para calcular uma estimativa mais precisa do robô.

Este pacote utiliza o filtro de Kalman Estendido (*Extended Kalman Filter* - EKF) para realizar a fusão dos dados dos sensores. O EKF é um método amplamente utilizado em filtragem de sensores e é capaz de lidar com não-linearidades presentes no sistema, tornando-o adequado para estimar a pose do robô em ambientes complexos (ROS, 2022).

Para utilizar este pacote, é necessário que as mensagens contenham as matrizes de covariância de forma correta de cada sensor para que seja levado em consideração o peso

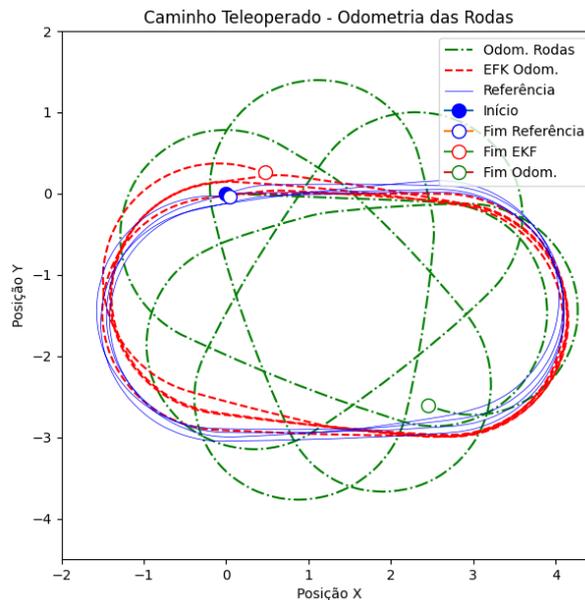
Figura 29 – Comportamento da odometria da câmera em relação ao seu referencial.



Fonte: Produção do próprio autor.

adequado das medições, assim como redução da propagação de erros. Na Figura 30 é possível observar o resultado ao realizar a fusão da odometria das rodas com os dados da IMU.

Figura 30 – Odometria das rodas e odometria obtida com a fusão da IMU pelo Filtro de Kalman.

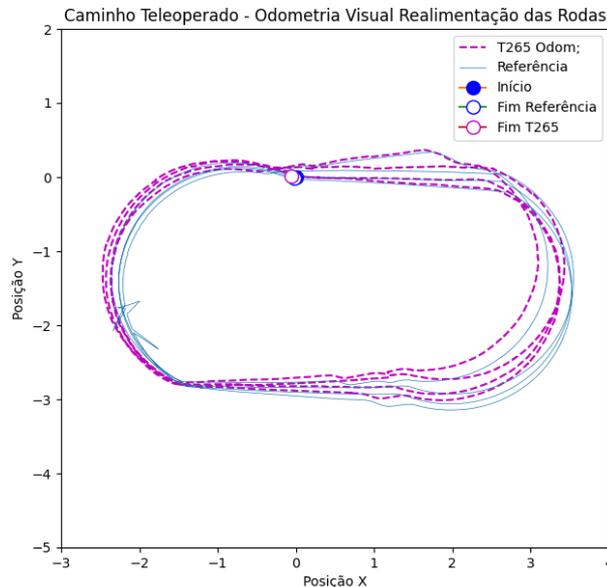


Fonte: Produção do próprio autor.

A transformação descrita na Equação 3.1 é utilizada para realizar a transformação na câmera para que esta leve em consideração a leitura da odometria das rodas ao realizar a

VIO. O comportamento da câmera levando em consideração a odometria das rodas pode ser observado na Figura 31.

Figura 31 – Odometria Visual levando em consideração a odometria das rodas.



Fonte: Produção do próprio autor.

É possível observar que, comparado ao experimento onde a câmera realizou a VIO independente da odometria das rodas, a trajetória se aproxima mais da referência durante todo o experimento, havendo mais precisão e acurácia do que quando a câmera opera de forma independente.

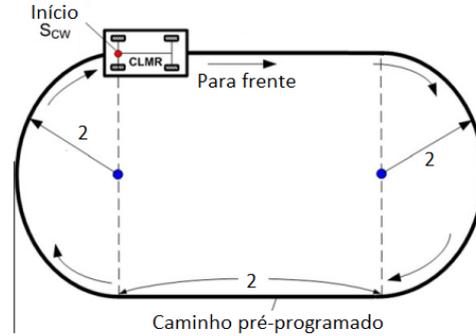
3.4.2 Seguidor de Caminho e Experimentos Automatizados

Realizar todos os testes de forma padronizada é o ponto chave para a análise de erros, já que experimentos feitos de forma arbitrária ou manual iriam resultar em erros inconsistentes e não padronizados. Nesse sentido, é essencial assegurar a repetibilidade dos experimentos. Para isso, foi programado um seguidor de caminho para aumentar a precisão e exatidão dos testes, reduzindo os erros sistemáticos e aleatórios, obtendo resultados mais confiáveis e consistentes.

O caminho pré programado a ser seguido (Figura 32), levando em consideração as restrições do robô, como o raio mínimo de rotação, foi parametrizado no Matlab. A lei de controle utilizada é a lei apresentada pela Equação 2.7, partindo do princípio que não há pontos em seu caminho que não sejam restritos pelo seu modelo cinemático, podendo então ser simplificado ao modelo de robô diferencial. O *script* do Matlab calcula qual o ponto mais

próximo do caminho parametrizado ao ponto a definido, incrementando à medida que o robô progride.

Figura 32 – Caminho programado a ser seguido pelo robô.



Fonte: Produção do próprio autor.

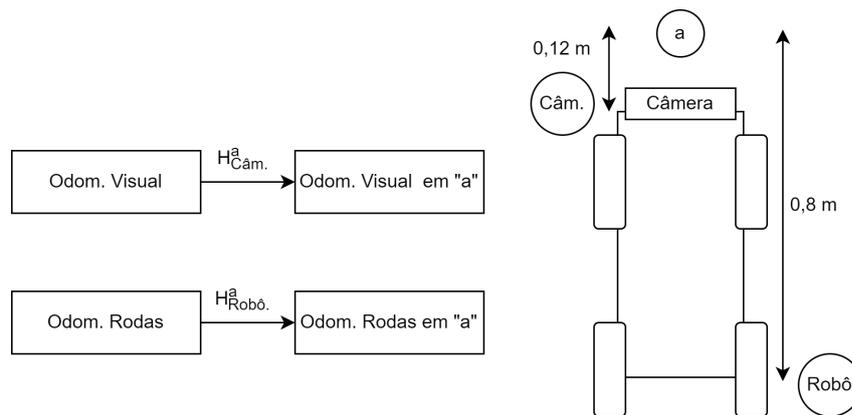
É necessário definir um ponto a , conforme descrito na Equação 2.7. Quanto maior o valor do ponto a , maior será a estabilidade do sistema, tornando o controle menos agressivo e, conseqüentemente, reduzindo os desvios em relação ao caminho determinado. O ponto a definido foi igual $a = 0.8 m$, fora do corpo do robô e à frente da câmera.

Sendo o ponto a a referência dos experimentos, é necessário que a referência do robô e da odometria visual sofram uma transformação para que seja calculado o erro associado. A matriz de transformação da odometria do referencial do robô para o ponto a é descrita pela Equação 3.4, e a matriz de transformação da odometria do referencial da câmera para o referencial do ponto a é descrita pela Equação 3.5.

$$H_{Robô}^a = \begin{bmatrix} 1 & 0 & 0 & 0,8 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

$$H_{Câm}^a = \begin{bmatrix} 1 & 0 & 0 & 0,12 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

O esquemático apresentando as transformações utilizadas em relação aos referenciais adotados está apresentado na Figura 33.

Figura 33 – Transformações aplicadas na odometria visual e odometria das rodas para o ponto a .

Fonte: Produção do próprio autor.

O sistema utiliza três métodos para estimar a odometria: odometria das rodas, odometria visual e odometria gerada pelo Filtro de Kalman através da fusão da odometria das rodas com a IMU do robô. Inicialmente, foram realizados quatro experimentos. Cada um dos experimentos tem como finalidade alterar um dos métodos com o objetivo de melhorar a estimativa de posição. Exceto pelo primeiro experimento, que avalia cada odometria de forma isolada.

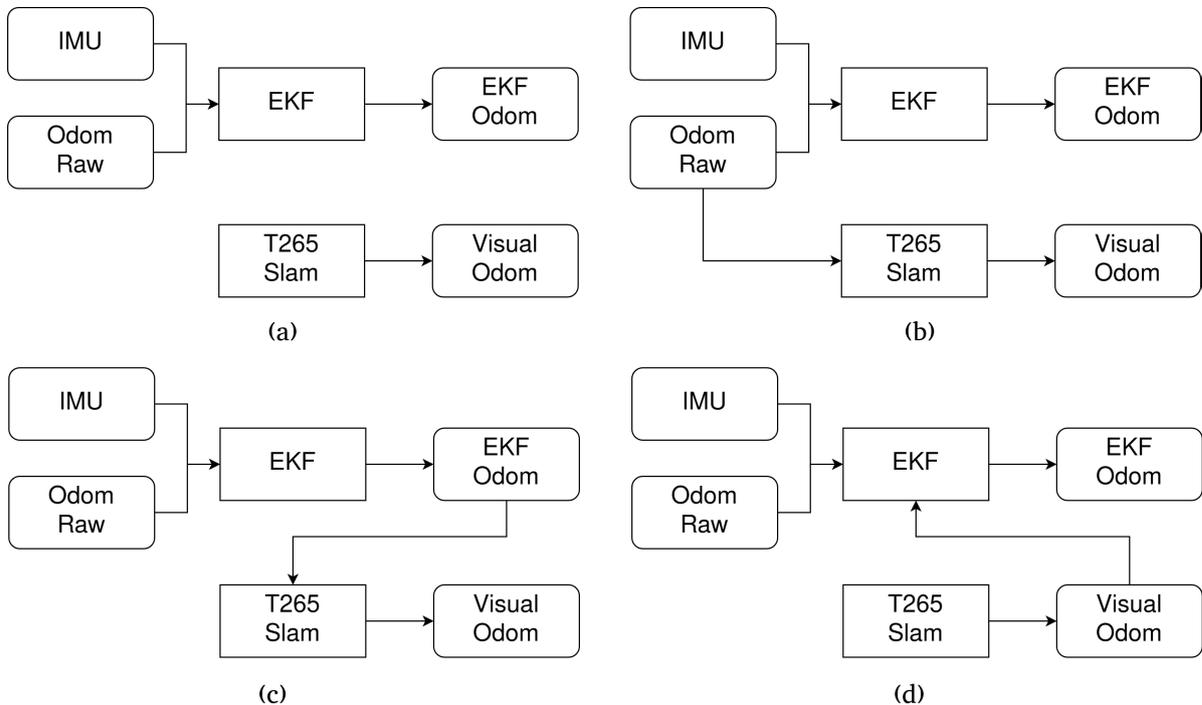
Sendo os seguintes experimentos realizados, tendo como foco as seguintes configurações:

1. Odometrias isoladas (Figura 34 (a));
2. Odometria Visual com realimentação da odometria das rodas (Figura 34 (b));
3. Odometria Visual com realimentação do Filtro de Kalman (Figura 34 (c));
4. Filtro de Kalman com IMU e Odometrias Visual e das rodas (Figura 34 (d)).

Os Experimentos 1, 2, 3 e 4 consistem em inicializar o robô em um ponto próximo ao caminho pré-programado (Figura 32), fazendo com que este se posicione no caminho e o execute. Em cada experimento é alterado algum ponto onde visando melhorar a odometria visual ou utiliza-la para tentar obter uma odometria melhor através da fusão com outras fontes, conforme apresentado na Figura 34. Com o objetivo de avaliar o erro em regime permanente, os experimentos foram realizados durante o período de 10 minutos.

Visando avaliar o comportamento e robustez da odometria visual quando submetida a um distúrbio, foram realizados experimentos simulando uma queda de luz. Para isso, o robô foi posicionado em um ponto próximo ao caminho, fazendo com que este se posicione e o

Figura 34 – Odometrias a serem avaliadas pelo robô nos Experimentos 1, 2, 3 e 4. (a) Odometrias independentes. (b) Odometria visual com realimentação da odometria das rodas. (c) Odometria visual com realimentação do Filtro de Kalman. (d) Filtro de Kalman adicionando a odometria visual.



Fonte: Produção do próprio autor.

siga durante o intervalo de tempo de 5 minutos. A partir deste momento, a luz da sala é apagada, fazendo com que o robô tenha somente fontes externas à sala onde estava sendo realizado os experimentos e luz provenientes do monitor dos computadores.

Os Experimentos 5 ao 8 tem por finalidade avaliar as mesmas configurações de odometria definidas pela Figura 34, diferindo pelo fato destas estarem sendo influenciadas pelo ruído de uma odometria visual mal realizada:

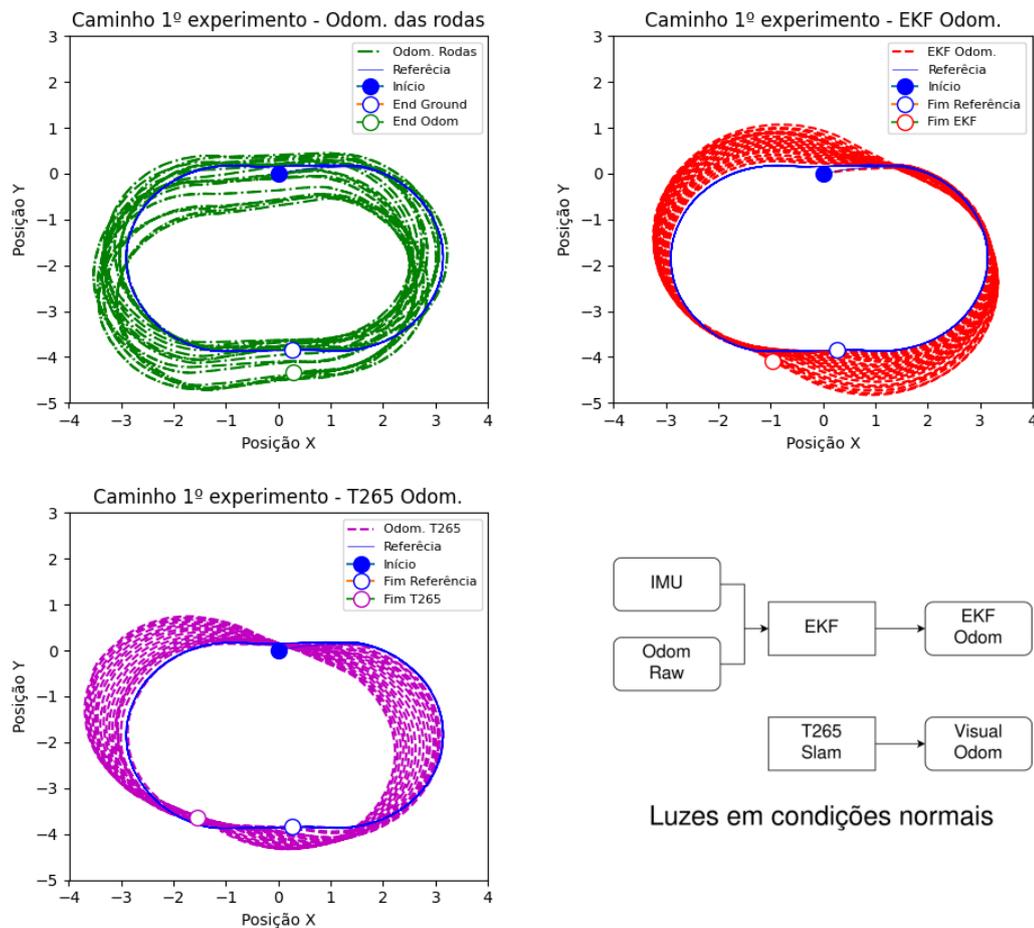
5. Odometrias isoladas - Queda de luz aos 5 min;
6. Odometria Visual com realimentação da odometria das rodas - Queda de luz aos 5 min;
7. Odometria Visual com realimentação do Filtro de Kalman - Queda de luz aos 5 min;
8. Filtro de Kalman com IMU e Odometrias Visual e das rodas - Queda de luz aos 5 min.

4 RESULTADOS

4.1 Câmera operando em condições nominais

Os resultados da primeira configuração, mostram a odometria das rodas, odometria Visual, e a odometria e do Filtro de Kalman com IMU e odometria das rodas (Figura 35). Realizando uma análise qualitativa visual dos caminhos realizados, observa-se que neste experimento específico a odometria das rodas manteve o formato base do caminho de referência, mas se deslocando no ambiente com o tempo. Já o Filtro de Kalman fundindo o valor obtido da odometria das rodas com a IMU manteve a odometria mas próxima ao valor de referência, mas rotacionando em um pequeno valor a cada volta realizada. O comportamento da câmera T265 é similar ao apresentado pelo Filtro de Kalman.

Figura 35 – Odometria das rodas, odometria Visual, e a odometria o Filtro de Kalman com IMU e odometria das rodas.



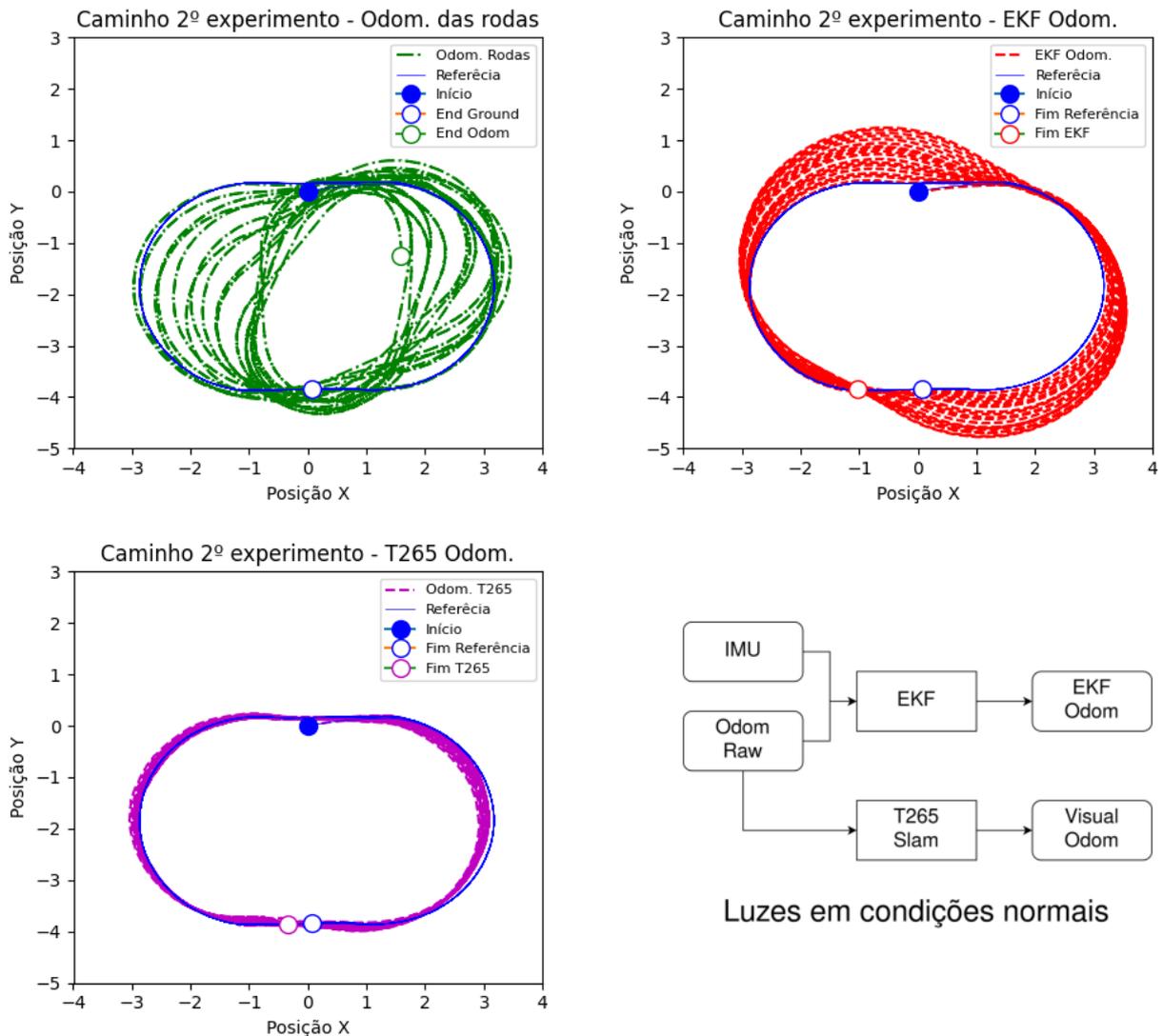
Luzes em condições normais

Fonte: Produção do próprio autor.

A configuração com a odometria das rodas, odometria visual com realimentação da odometria das rodas, Filtro de Kalman com IMU e odometria das rodas está disposta na Figura 36. Neste experimento a odometria das rodas além de sofrer uma pequena rotação, se deslocou com o tempo. Por mais que a odometria das rodas tenha um comportamento onde se distanciou mais de sua referência, o Filtro de Kalman obteve um resultado muito próximo ao observado no primeiro experimento.

Em relação à odometria visual, ao utilizar a odometria das rodas, foi capaz de se manter próxima da referência durante todo o caminho. Ainda que a odometria das rodas utilizada seja imperfeita, foi o suficiente para uma boa correção da odometria visual realizada pelo algoritmo interno da câmera.

Figura 36 – Odometria das rodas, odometria Visual com realimentação da odometria das rodas, Filtro de Kalman com IMU e odometria das rodas.

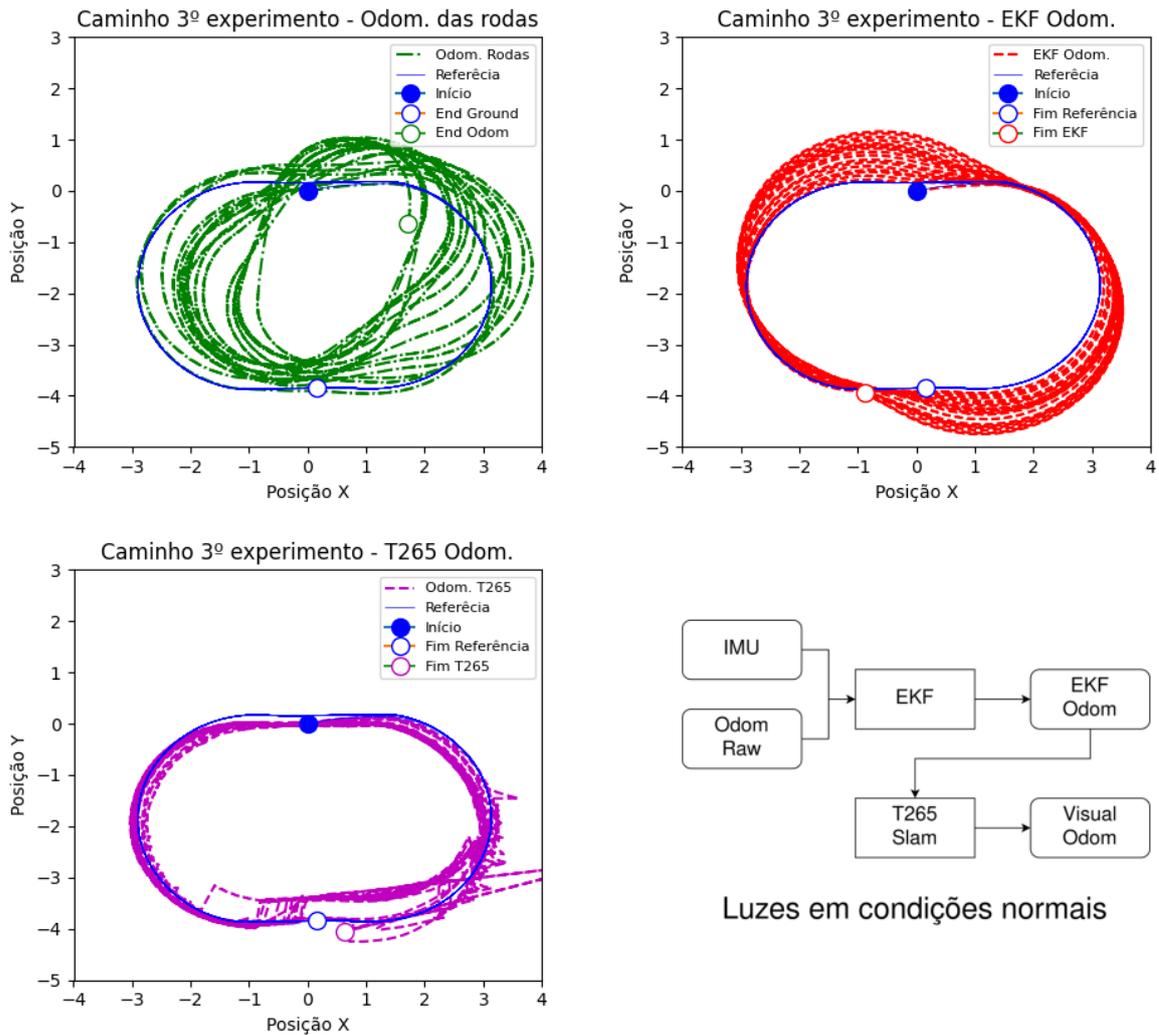


Fonte: Produção do próprio autor.

O resultado dos experimentos visando realizar uma melhora na odometria visual ao utilizar o valor de odometria corrigido do Filtro de Kalman na odometria visual está apresentado na Figura 37. A partir de um determinado momento do experimento os valores obtidos na odometria visual passaram a demonstrar uma grande instabilidade.

As prováveis causas disso são leituras atrasadas inseridas pelo filtro, já que as leituras de entrada são sempre atrasadas em relação ao valor atual calculado da câmera. Outro fator está relacionado à matriz de covariância, onde a odometria do Filtro de Kalman, por retornar um valor de maior confiabilidade, altera também a matriz de covariância, fazendo com valores de leituras antigas tenham uma maior confiabilidade do que valores atuais. Por apresentar instabilidade com o tempo, tal configuração é descartada, não sendo realizado o experimento para verificar o comportamento da influência do ruído na odometria visual quando realimentada com os valores do Filtro de Kalman.

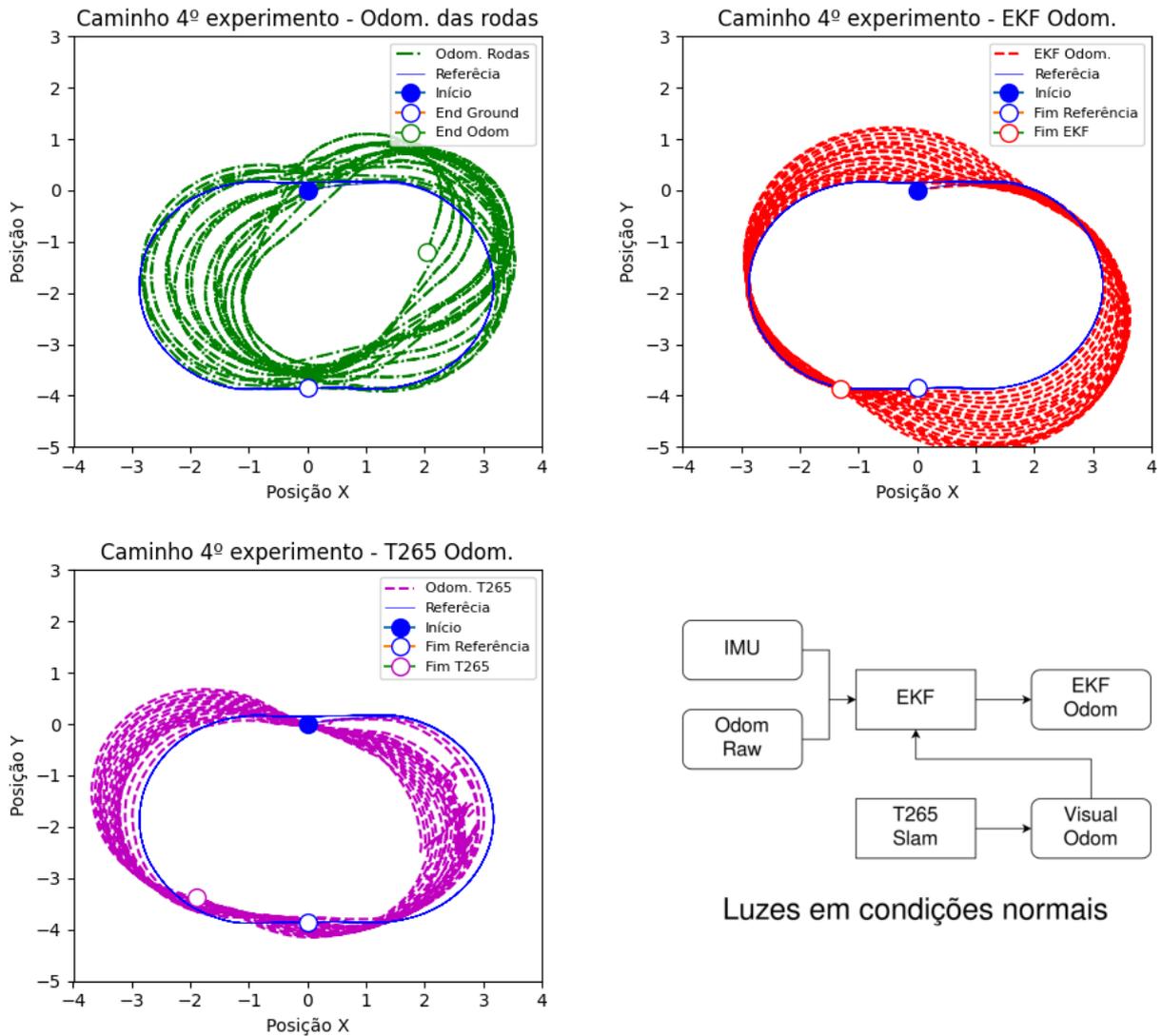
Figura 37 – Odometria das rodas, Odometria Visual com realimentação do Filtro de Kalman, Filtro de Kalman com IMU e odometria das rodas.



Fonte: Produção do próprio autor.

O resultado obtido ao fundir a odometria visual com a odometria das rodas e IMU está apresentado na Figura 38. É possível observar que a odometria do Filtro de Kalman não apresentou uma grande diferença quando comparado aos experimentos anteriores, quando a odometria da câmera não estava sendo utilizada no filtro.

Figura 38 – Odometria das rodas, odometria Visual, Filtro de Kalman com IMU e odometrias visual e das rodas.



Fonte: Produção do próprio autor.

Para entender o ocorrido, é necessário verificar as matrizes de covariância das odometrias. É possível observar através da Equação 4.1 a matriz de covariância da odometria das rodas, enquanto a equação da matriz de covariância da odometria T265 está representada pela Equação 4.2. A covariância correspondente às posições X,Y,Z da odometria das rodas é menor do que as covariâncias correspondentes da T265, fazendo com que os valores da odometria das rodas tenham mais prioridade do que os valores de posição da câmera no Filtro do Kalman.

Já em relação aos valores correspondentes às rotações sobre os eixos X,Y,Z da câmera, a matriz de covariância da T265 tem um valor muito menor quando comparado à odometria das rodas. Isto é devido à IMU interna, que realiza essa leitura de orientação de uma forma mais precisa. Como o filtro já possui uma IMU, uma nova adição não terá muito impacto em relação à orientação, mas é útil para a redução de ruídos entre elas.

$$Cov_{Odom} = \begin{bmatrix} 0,001 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0,001 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1000000 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1000000 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1000000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1000 \end{bmatrix} \quad (4.1)$$

$$Cov_{T265} = \begin{bmatrix} 0,01 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0,01 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0,01 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0,0001 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0,0001 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0,0001 \end{bmatrix} \quad (4.2)$$

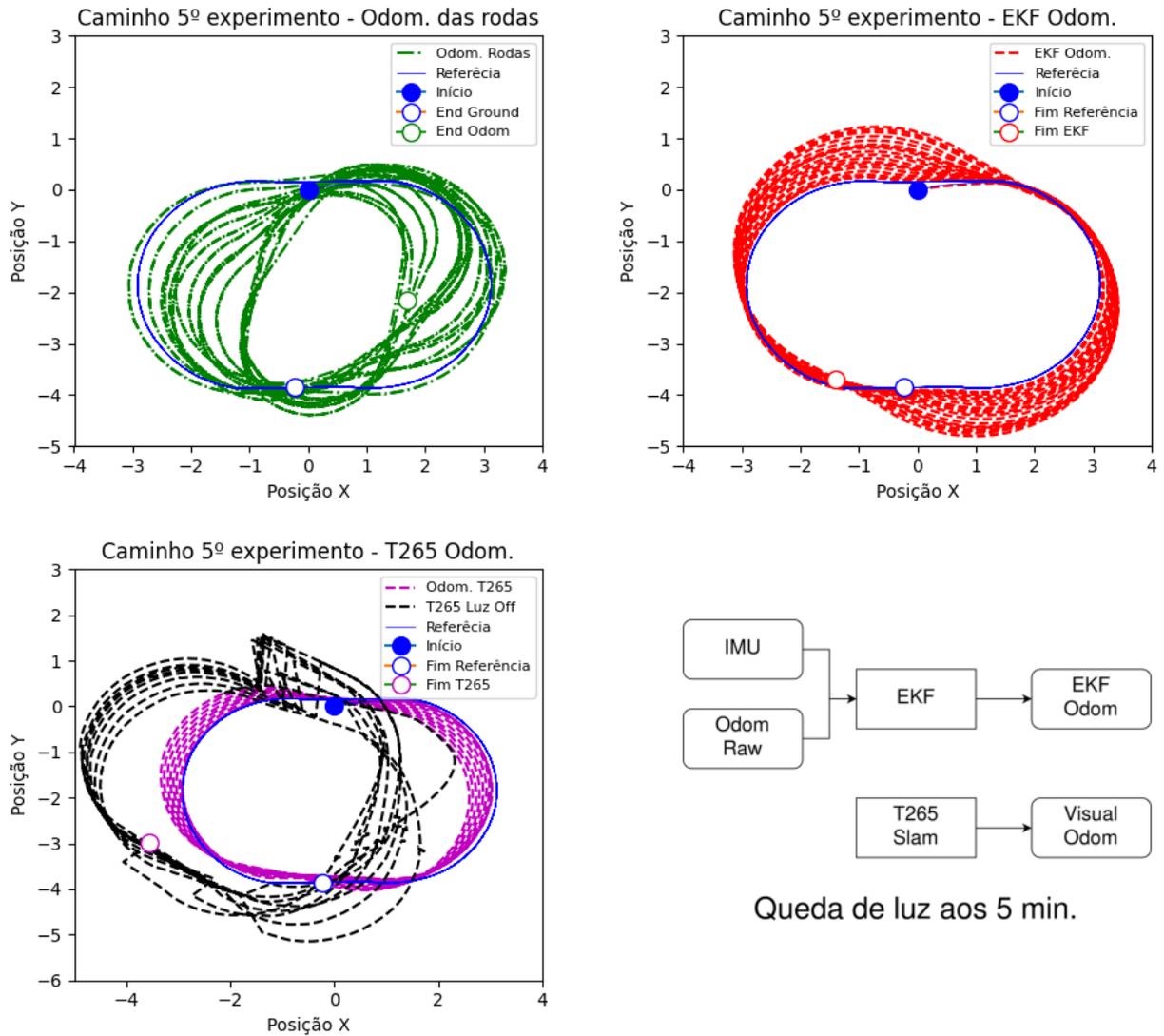
4.2 Influência de Ruído

Descartando a configuração do terceiro experimento, que apresentou instabilidade ao realimentar a câmera com a odometria do Filtro de Kalman, foram refeitos os experimentos com as configurações realizadas nos experimentos 1, 2 e 4, conforme apresentado pela Figura 34. As configurações e cenários foram replicados da primeira parte, porém, foi introduzido um erro aleatório no sistema, que é a queda de luz durante o caminho realizado.

O resultado para a quinta configuração (odometria das rodas, Odometria Visual, Filtro de Kalman com IMU e odometria das rodas) com a queda de luz durante metade do experimento pode ser observado na Figura 39. É possível observar que a odometria gerada pela câmera ficou completamente fora do caminho a partir do momento em que houve a queda de luz, se reposicionando em um novo referencial, havendo deslocamentos abruptos com a ordem de grandeza de metros. Como a odometria das rodas e o filtro utilizam sensores completamente independentes da luz, não foram afetados por esse erro.

A sexta configuração, que apresentou um bom resultado por parte da odometria visual, apresentou uma boa resposta ao distúrbio na qual foi submetida, conforme pode ser

Figura 39 – Queda de luz durante o experimento - Odometria das rodas, Odometria Visual, Filtro de Kalman com IMU e odometria das rodas.

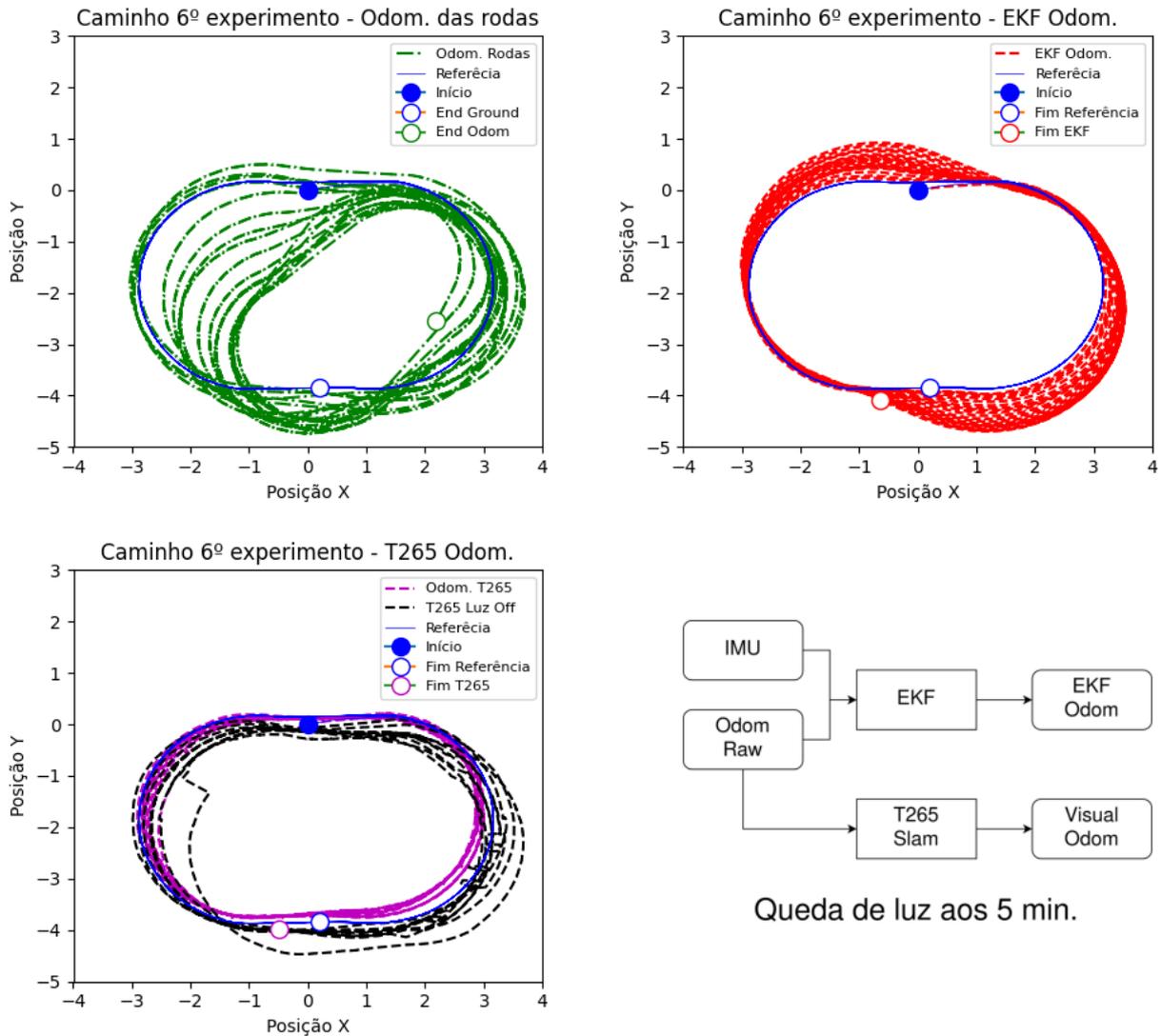


Fonte: Produção do próprio autor.

observado na Figura 40. Por mais que a odometria visual tenha ficado claramente comprometida com a queda de luz, não apresentou um formato muito diferente da referência do experimento, ocorrendo correções para se posicionar mais próximo da trajetória. Esses resultados sugerem que a combinação dos dados provenientes da odometria das rodas e da odometria visual podem compensar as limitações individuais de cada modalidade, aumentando assim a confiabilidade e a capacidade de lidar com situações adversas.

O sétimo experimento realizado é correspondente à oitava configuração (Figura 41) já que a sétima configuração foi invalidada. Esta configuração consiste na odometria das rodas, odometria visual, Filtro de Kalman com IMU e odometrias visual e das rodas, é interessante verificar a matriz de covariância associada à odometria visual, já que

Figura 40 – Queda de luz durante o experimento - Odometria das rodas, Odometria Visual com realimentação da odometria das rodas, Filtro de Kalman com IMU e odometria das rodas.



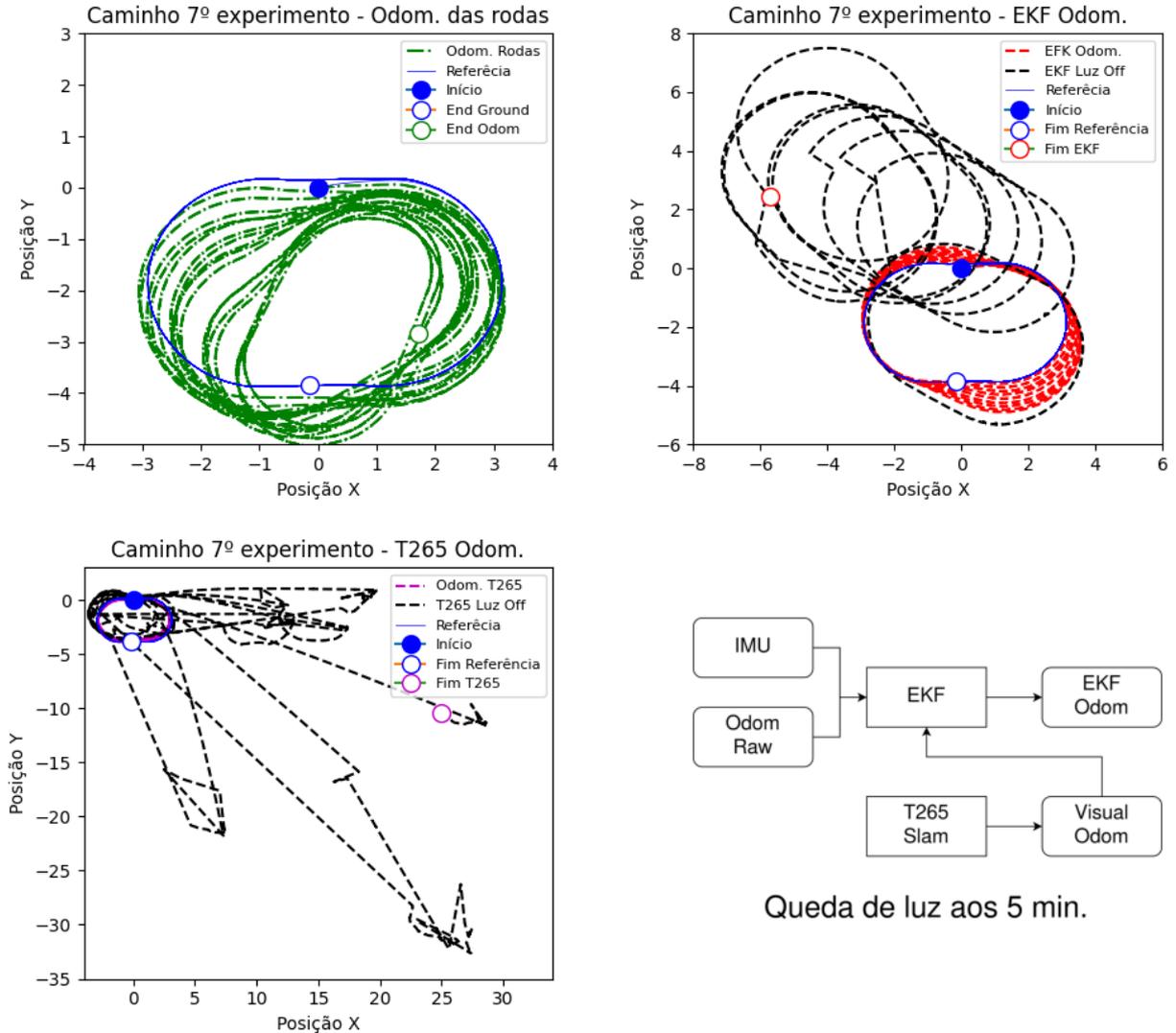
Fonte: Produção do próprio autor.

esta influencia diretamente nos resultados do Filtro de Kalman. Durante sua rotina em operações nominais, descreve o comportamento apresentado na Equação 4.2.

Com a queda de luz, sua matriz de covariância altera entre os valores de condições nominais e os valores para baixa luminosidade descritos na matriz da Equação 4.3, chegando até a apresentar em momentos críticos a matriz de covariância descrita por 4.4.

Um observação interessante é que, mesmo que possua internamente uma IMU que atua juntamente à odometria visual, a confiabilidade do sistema é tão baixa que até mesmo os termos da matriz de covariância associada ao sensor são muito altos, fazendo com que tanto os valores associados à posição quando aos de orientação tenham um peso muito

Figura 41 – Queda de luz durante o experimento -Odometria das rodas, Odometria Visual, Filtro de Kalman com IMU e Odometrias Visual e das rodas.



Fonte: Produção do próprio autor.

baixo ao ser filtrado.

Neste determinado experimento a odometria visual apresentou valores altos de odometria a partir do momento em que houve a queda de luz, se afastando muito dos pontos de referência. A matriz de covariância oscila entre os valores de operação nominal (Equação 4.2), de baixa luminosidade (Equação 4.3) e valores críticos de funcionamento (Equação 4.4).

$$Cov_{T265 \text{ Luz Baixa}} = \begin{bmatrix} 0,1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0,1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0,1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0,001 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0,001 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0,001 \end{bmatrix} \quad (4.3)$$

$$Cov_{T265 \text{ Critica}} = \begin{bmatrix} 10 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0,1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0,1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0,1 \end{bmatrix} \quad (4.4)$$

A partir do momento em que as leituras da câmera deixam de ser confiáveis a matriz de covariância é alterada e tem um peso muito baixo no filtro, passando a ter como principais valores as odometrias da roda e IMU. No momento em que o robô consegue detectar alguma característica do ambiente, seja por proximidade ou luzes externas do ambiente que auxiliem em sua tarefa, a matriz de covariância volta a assumir o valor nominal. Neste processo, a odometria visual acumula um determinado erro em sua posição, se afastando da referência. Com isso, o filtro irá convergir para um novo valor no plano, fazendo com que todas as leituras sejam deslocadas em um novo referencial.

É possível observar pelo gráfico que ocorre o deslocamento de posição nos valores de odometria do Filtro de Kalman em certos momentos durante o experimento devido a convergência do filtro. A partir do momento em que houve acúmulo de erro na posição e a matriz de covariância volta ao seu valor nominal, o Filtro de Kalman converge em um ponto qualquer. A cada vez que a câmera se perde, a odometria do filtro se desloca no momento em que a matriz de covariância da câmera volta ao seu valor nominal, repetindo o dado processo durante o experimento.

Através da análise qualitativa inicial é possível observar que a configuração que apresentou o melhor comportamento, coincidindo na maioria dos pontos com a referência é a configuração onde a câmera T265 é realimentada com a leitura da odometria das rodas.

4.3 Cálculo de Erro de Posição

Para calcular o erro da odometria, será utilizada a distância euclidiana, chamando então o valor calculado de erro euclidiano. O principal fator é que esta métrica leva em consideração todos os pontos da odometria, permitindo avaliar o desempenho geral da odometria, comparando as posições estimadas e de referência. Outro fator importante para o uso da distância euclidiana é a sensibilidade a grandes erros, dado que estimativas impressionantes resultam em um valor de erro mais alto, podendo identificar de forma fácil os casos em que a odometria está significativamente incorreta (OUERGHI et al., 2020b).

O erro euclidiano médio ε é dado pela Equação 4.5, onde Δx é a distância do ponto x da odometria até o ponto do x referencial e Δy é a distância do ponto y da odometria até o ponto do y referencial. O valor n é o número de amostras.

$$\varepsilon = \frac{1}{n} \sum \sqrt{\Delta x^2 + \Delta y^2} \quad (4.5)$$

A Tabela 1 contém o erro euclidiano médio das três primeiras configurações válidas testadas, sendo elas: todos os sistemas isolados, a odometria visual com a realimentação dos *encoders* das rodas e o Filtro de Kalman utilizando a odometria visual, correspondentes ao experimentos 1, 2 e 4, respectivamente, podendo ser analisado graficamente através da Figura 42.

Tabela 1 – Erro euclidiano médio - Luzes em condições normais.

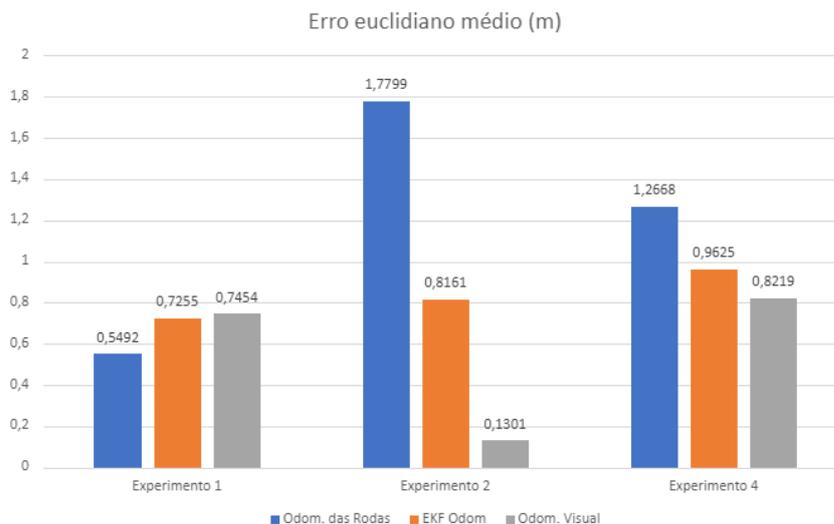
Experimento	Erro Euclidiano Médio (m)		
	Odom. das Rodas	EKF Odom.	Odom. Visual
1	0,5432	0,7255	0,7454
2	1,7799	0,8161	0,1301
4	1,2668	0,9625	0,8219

Fonte: Produção própria do autor.

A Tabela 2 contém o erro euclidiano médio da três últimas configurações testadas, sendo elas: todos os sistemas isolados, a odometria visual com a realimentação dos *encoders* das rodas e o Filtro de Kalman utilizando a odometria visual, correspondentes ao experimentos 5, 6 e 7, respectivamente (Figura 43).

Como é possível observar nas Tabelas 1 e 2, a melhor configuração obtida foi a da câmera com realimentação das rodas dos *encoders*, tendo o menor erro tanto em condições normais de operação quanto na presença de um distúrbio aleatório aplicado ao sistema e desempenho bem superior quando comparado às outras configurações. A odometria visual apresentou

Figura 42 – Erro euclidiano médio - Luzes em condições normais.



Fonte: Produção do próprio autor.

Tabela 2 – Erro euclidiano médio - Queda de luz no meio.

Experimento	Erro Euclidiano Médio (m)		
	Odom. das Rodas	EKF Odom.	Odom. Visual
5	1,7345	0,7839	1,3096
6	1,2492	0,6329	0,2907
7	1,3949	2,6809	3,5740

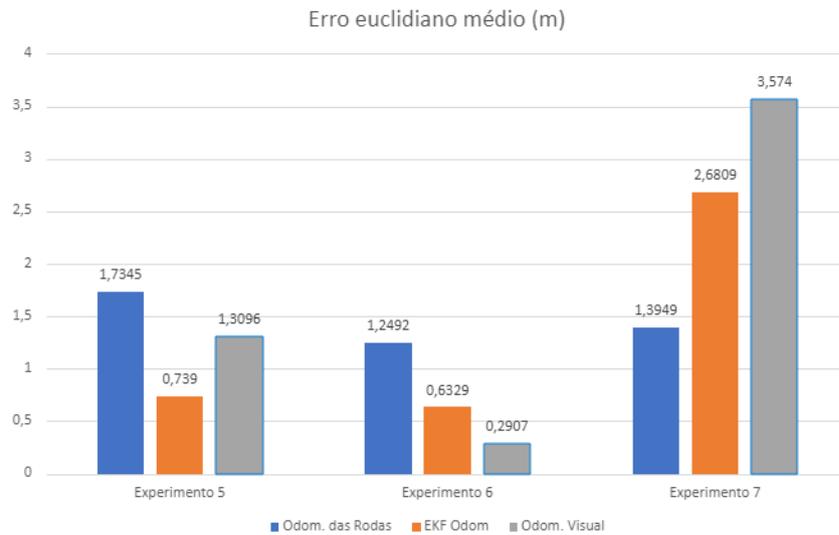
Fonte: Produção própria do autor.

um resultado semelhante ao valor obtido pelo Filtro de Kalman da odometria das rodas com a IMU, mas apresentou comportamento tão ruim quanto à odometria das rodas quando submetida a um distúrbio.

Na Figura 44 é possível observar o comportamento do erro euclidiano no tempo para as odometrias do segundo experimento, que contém a configuração que apresentou o melhor resultado. É possível observar o acúmulo progressivo do erro, conforme esperado, atingindo uma distância de mais de 4 metros em relação à referência em determinados pontos durante o período do experimento. Tal erro é atenuado pelo Filtro de Kalman, sendo reduzido a quase metade. Em relação à odometria visual, pode ser notado que com o passar do tempo a amplitude do erro aumenta, mas sofre correções de forma periódica devido à propriedade de correção da odometria através da correção de características.

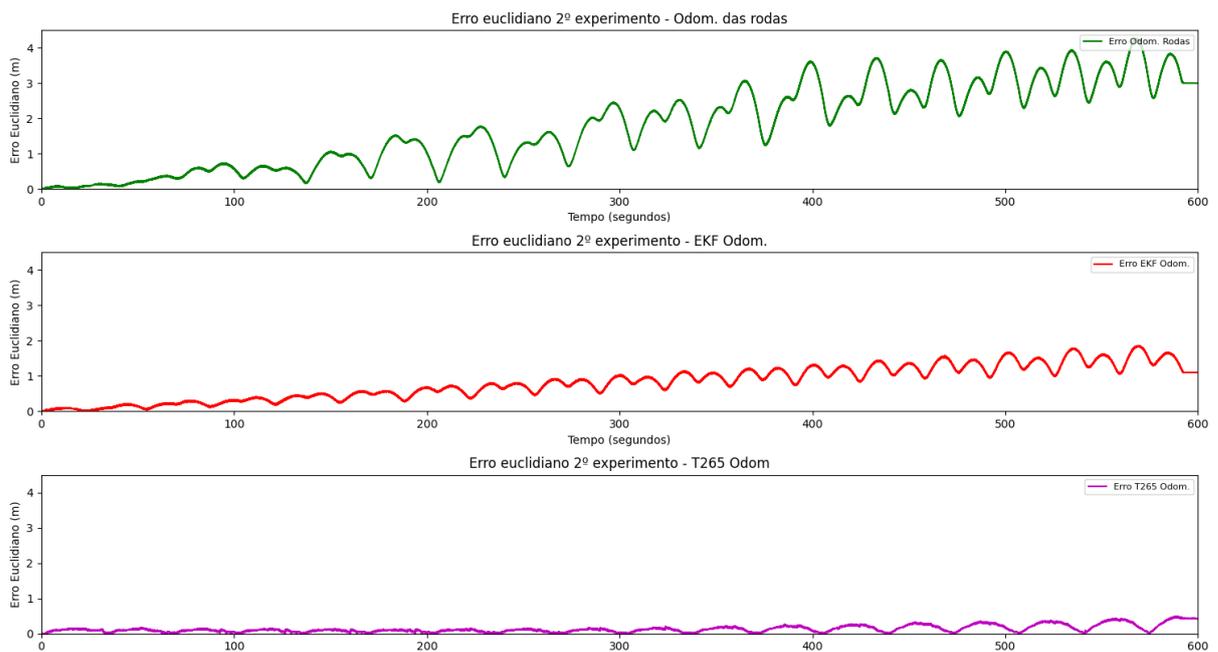
A variação do erro no tempo obtido no Experimento 6, que apresenta o comportamento do erro da odometria visual quando realimentada pela odometria das rodas quando ocorre o distúrbio (queda de luz), está apresentada na Figura 45. A linha vertical preta marca o tempo onde houve a queda de luz. Depois da queda de luz, ocorre momentos em que

Figura 43 – Erro euclidiano médio - Queda de luz no meio



Fonte: Produção do próprio autor.

Figura 44 – Erro euclidiano no tempo - Experimento 2.



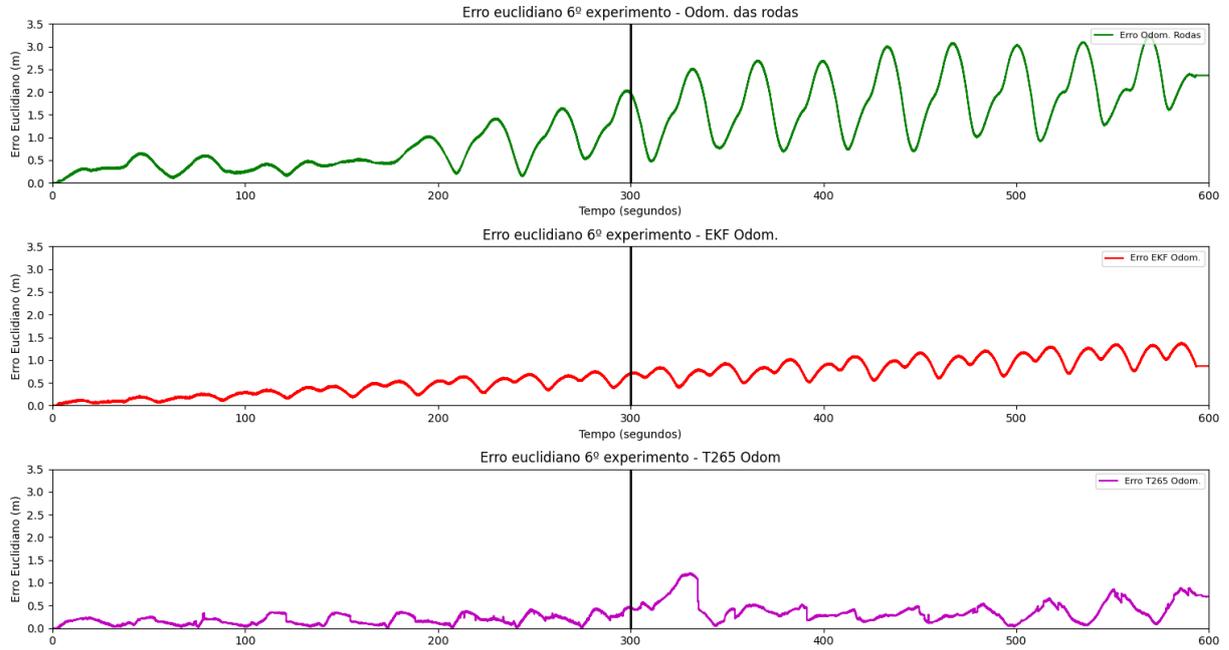
Fonte: Produção do próprio autor.

houve uma variação abrupta do erro em determinados pontos da odometria visual sendo possível observar de forma mais evidente a correção da odometria visual.

A partir do momento em que a odometria visual é comprometida, a odometria das rodas serve de referência. Mas esta possui um erro euclidiano maior do que os obtidos nos erros da odometria visual em condições nominais. Desta forma, a partir do momento em que a

odometria visual é capaz de detectar características equivalentes no ambiente, realiza a correção ao erro associado a sua própria leitura, e não à odometria das rodas.

Figura 45 – Erro euclidiano no tempo - Experimento 6.



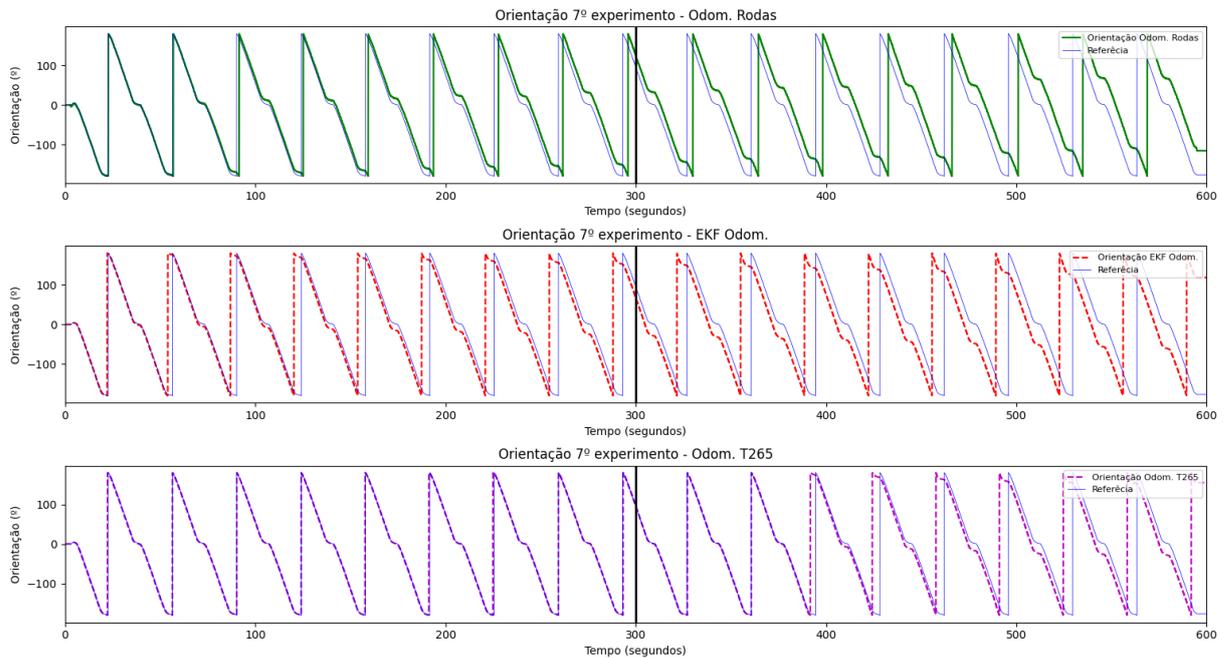
Fonte: Produção do próprio autor.

4.4 Cálculo de Erro de Orientação

Em todos os experimentos realizados, seja em condições normais ou de queda de luz, a orientação obtida foi equivalente às respectivas configurações. Isto é devido à odometria das rodas ser puramente dependente do modelo matemático e das leituras das rodas, não sendo afetada pelo distúrbio, e a odometria do Filtro de Kalman e visual terem uma IMU em sua configuração, o que torna o sistema mais robusto às variações angulares.

O comportamento geral das orientações pode ser observado pela Figura 46, que apresenta as orientações obtidas na configuração onde foram obtidos os piores resultados quanto às posições estimadas, que foi o Experimento 7, com o Filtro de Kalman levando em consideração a odometria visual na queda de luz, que está representada a partir da linha preta vertical. É possível notar através de uma análise qualitativa que a orientação tanto da odometria das rodas quanto a da odometria do Filtro de Kalman são as que apresentam uma maior divergência com o tempo, enquanto a orientação da odometria visual é que se mantém mais estável durante o tempo.

Figura 46 – Orientações obtidas em cada configuração durante o 7º experimento.



Fonte: Produção do próprio autor.

O erro de orientação no tempo será calculado através do erro médio de orientação, calculado a partir da diferença média entre as orientações estimadas e as reais. O erro angular médio é calculado através da Equação 4.6, onde $\Delta\theta$ é a diferença entre as orientações de referência e a atual.

$$Erro\ angular\ medio = \frac{1}{n} \sum |\Delta\theta| \tag{4.6}$$

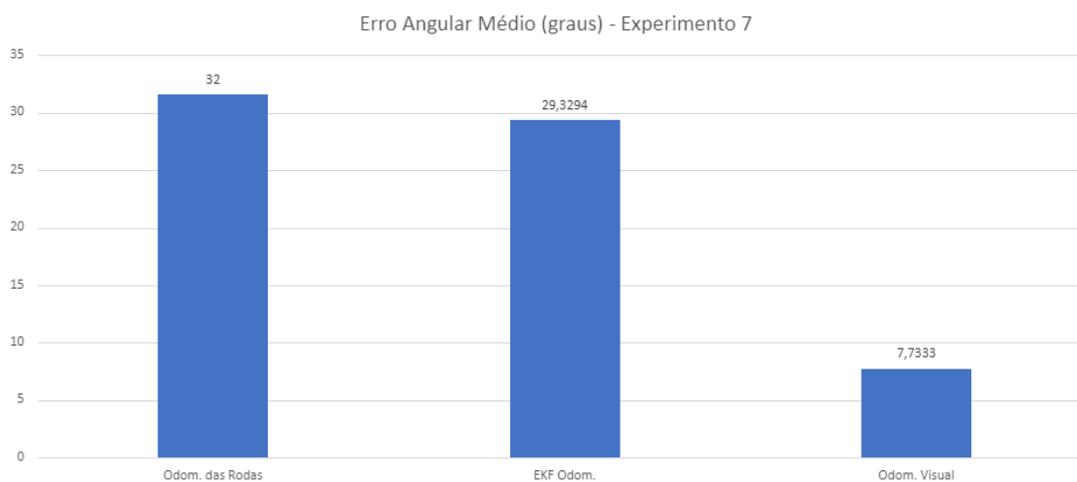
A Tabela 3 contém os erros calculados o Experimento 7, que foi similar ao obtido nas outras configurações, também apresentados graficamente na Figura 47. O erro de orientação obtido com a odometria das rodas foi o que apresentou o maior erro, conforme o esperado, já que não possui nenhum sensor associado a leituras de velocidades angulares. Já o Filtro de Kalman não é capaz de mitigar o erro associado à orientação, uma vez que a covariância associada à velocidade angular em Z é muito alta, conforme pode ser observado na Equação 4.1.

Tabela 3 – Erro de orientação médio - Queda de luz no meio.

Experimento	Erro Angular Médio (°)		
	Odom. das Rodas	EKF Odom.	Visual Odom.
7	31,5588	29,32942	7,7333

Fonte: Produção própria do autor.

As orientações obtidas através da odometria visual foram as que apresentaram o menor erro de orientação, provavelmente devido a uma maior prioridade aos valores obtidos pela IMU, mas não sendo zero devido a erros intrínsecos do próprio sensor. O erro máximo de orientação obtido na odometria das rodas é de 69.99° , enquanto o das rodas atinge erros de até 64.78° . O valor máximo de erro de orientação obtido na odometria visual foi de 30.61° .

Figura 47 – Erro Angular Médio ($^\circ$)

Fonte: Produção do próprio autor.

5 CONCLUSÕES E TRABALHOS FUTUROS

O objetivo geral deste trabalho consiste em aprimorar a estimativa de localização de um robô móvel ao realizar a fusão sensorial através da odometria das rodas e de uma fonte de odometria visual. O objetivo é buscar obter uma configuração robusta o suficiente para obter menores valores de erro em cenários onde a câmera possui sua visibilidade debilitada, comprometendo seu funcionamento.

Um dos fatores de motivação do trabalho é complementar a literatura fazendo um tipo de análise que não é o foco de outros trabalhos que usam a mesma câmera. Isso acontece porque as pesquisas geralmente realizadas a partir desta câmera têm como finalidade comparar a eficiência desta câmera com outros métodos de odometria visual, e não a uma forma de maximizar sua eficiência a partir de uma outra fonte de odometria. Além disso, também se busca compreender o comportamento da câmera em relação a um distúrbio, simulando uma possível situação real a ser enfrentada pelo robô

Como resultado, utilizando uma fonte de referência absoluta, foi possível obter e avaliar os padrões e comportamento das diferentes fontes de odometria, assim como o comportamento das configurações relacionadas à odometria visual. Essas configurações são usadas com base na odometria de um robô móvel, que incluem a odometria a partir da leitura dos *encoders* das rodas e a obtida a partir do filtro de Kalman através do pacote *robot_pose_ekf*, muito utilizado na robótica móvel

A partir do erro euclidiano médio, é possível concluir que a câmera T265 da Intel, combinada com os dados obtidos a partir da leitura da odometria das rodas permitiu uma estimativa mais precisa da posição do robô. Além disso, esta configuração se mostrou robusta mesmo em condições adversas, com a presença de ruídos, demonstrando sua capacidade de lidar com situações mais desafiadoras.

Este trabalho contribui para o avanço na área de robótica ao explorar uma fonte mais confiável de odometria para um robô móvel. Essa odometria é utilizada como base em grande parte dos algoritmos relacionados à robótica, como os de mapeamento e navegação autônoma, fornecendo uma base sólida de resultados para futuras pesquisas e aplicações práticas nesse campo.

Como trabalho futuro, é interessante avaliar o comportamento da odometria visual no cenário de navegação autônoma, verificando o impacto desta odometria nos algoritmos já

consolidados em robótica móvel, principalmente visando avaliar o impacto das correções bruscas realizadas pela odometria visual nestes algoritmos. Além disso, seria interessante avaliar seu comportamento em um caminho mais complexo, assim como alternância de caminhos, mudando as características do ambiente a serem utilizadas na odometria visual.

Alguns pontos importantes são a serem observados a respeito do funcionamento da câmera e da metodologia adotada no experimento. O primeiro ponto é estudar o comportamento da odometria visual em um ambiente dinâmico, uma vez que os experimentos foram realizados em um ambiente estático. Outro ponto relevante seria voltar com a luz ao fim do experimento para verificar se ocorre a correção do erro com base no SLAM realizado internamente pela câmera antes da queda de luz.

Como a odometria utilizada foi a odometria descalibrada, é de interesse realizar os experimentos com a odometria calibrada para verificar a importância da precisão da odometria das rodas ao integrá-la com a odometria visual. Além disso, devido à possibilidade de haver uma quantidade reduzida de características no ambiente, o que pode tornar o resultado do experimento mais específico, é recomendado conduzir o experimento em um ambiente diferente, que apresente mais variações e complexidade para avaliar a robustez do método em diferentes cenários.

Tendo em vista que a Intel possui algoritmos para realizar a integração da câmera T265 com câmeras de profundidade da série D400, vale avaliar o impacto da fusão desses dois sensores ao realizar o mapeamento 3D do ambiente, já que isto tem impacto direto no mapa gerado. Por fim, verificar se a configuração encontrada resolve, ou pelo menos mitiga, o problema do “robô sequestrado”, quando o robô é levantado ou deslocado do chão, impossibilitando a correta estimativa de sua posição por meio da odometria das rodas, assim como o deslizamento de roda.

REFERÊNCIAS

- ABDULHAFIZ, W. A.; KHAMIS, A. Handling data uncertainty and inconsistency using multisensor data fusion. *Adv. in Artif. Intell.*, Hindawi Limited, London, GBR, v. 2013, jan 2013. ISSN 1687-7470. Disponível em: <<https://doi.org/10.1155/2013/241260>>. Citado na página 17.
- ALAPETITE, A.; WANG, Z.; HANSEN, J. P.; ZAJĄCZKOWSKI, M.; PATALAN, M. Comparison of three off-the-shelf visual odometry systems. *Robotics*, v. 9, n. 3, 2020. ISSN 2218-6581. Disponível em: <<https://www.mdpi.com/2218-6581/9/3/56>>. Citado na página 18.
- ALATISE, M. B.; HANCKE, G. P. A review on challenges of autonomous mobile robot and sensor fusion methods. *IEEE Access*, v. 8, p. 39830–39846, 2020. Citado na página 16.
- AMARAL, B. do. Robô de resgate pode limpar áreas devastadas por Tsunami no Japão. 2011. <<https://www.techtudo.com.br/noticias/2011/04/robo-de-resgate-pode-limpar-areas-devastadas-por-tsunami-no-japao.ghtml>>. Citado na página 14.
- AQEL, M. O. A.; MARHABAN, M. H.; SARIPAN, M. I.; ISMAIL, N. B. Review of visual odometry: types, approaches, challenges, and applications. *SpringerPlus*, v. 5, n. 1, p. 1897, 2016. ISSN 2193-1801. Disponível em: <<https://doi.org/10.1186/s40064-016-3573-7>>. Citado na página 17.
- AQEL, M. O. A.; MARHABAN, M. H.; SARIPAN, M. I. et al. Review of visual odometry: types, approaches, challenges, and applications. *SpringerPlus*, Springer, v. 5, n. 1, p. 1897, 2016. Citado na página 27.
- ASSA, A.; JANABI-SHARIFI, F. A kalman filter-based framework for enhanced sensor fusion. *IEEE Sensors Journal*, v. 15, n. 6, p. 3281–3292, 2015. Citado na página 15.
- ASTOLFI, P.; GABRIELLI, A.; BASCETTA, L.; MATTEUCCI, M. Vineyard autonomous navigation in the echord++ grape experiment**this work has been conducted under the “ground robot for vineyard monitoring and protection (grape)” experiment funded by the european commission under the echord++ project (fp7-601116). <http://echord.eu/grape/>. *IFAC-PapersOnLine*, v. 51, n. 11, p. 704–709, 2018. ISSN 2405-8963. 16th IFAC Symposium on Information Control Problems in Manufacturing INCOM 2018. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2405896318315271>>. Citado na página 17.
- BAHRIN, M. A. K.; OTHMAN, M. F.; AZLI, N. H. N.; TALIB, M. F. Industry 4.0: A review on industrial automation and robotic. *Jurnal teknologi*, v. 78, n. 6-13, 2016. Citado na página 14.
- BORENSTEIN, J.; EVERETT, H. R.; FENG, L.; WEHE, D. Mobile robot positioning: Sensors and techniques. *Journal of robotic systems*, Wiley Online Library, v. 14, n. 4, p. 231–249, 1997. Citado na página 31.

BROOKS, R. R.; IYENGAR, S. S. Multi-Sensor Fusion: Fundamentals and Applications with Software. USA: Prentice-Hall, Inc., 1998. ISBN 0139016538. Citado 2 vezes nas páginas 35 e 36.

CAMARGO, J. C. A. Amazon encerra testes de robôs de entrega Scout. 2022. <<https://olhardigital.com.br/2022/10/07/pro/amazon-encerra-testes-de-robos-de-entrega-scout/>>. Citado na página 15.

CARO, G. A. D. Kinematics Equations Odometry, Dead Reckoning. 2017. <<https://web2.qatar.cmu.edu/~gdicaro/16311-Fall17/slides/16311-8-Kinematics-DeadReckoning.pdf>>. Citado na página 25.

CASTANEDO, F. A review of data fusion techniques. The Scientific World Journal, v. 2013, 2013. Citado na página 35.

CHONG, K. S.; KLEEMAN, L. Accurate odometry and error modelling for a mobile robot. In: Proceedings of International Conference on Robotics and Automation. [S.l.: s.n.], 1997. v. 4, p. 2783–2788 vol.4. Citado na página 17.

CORKE, P. Robotics, Vision and Control: Fundamental Algorithms in MATLAB. 1st. ed. [S.l.]: Springer Publishing Company, Incorporated, 2013. ISBN 3642201431. Citado 10 vezes nas páginas 15, 16, 19, 20, 21, 22, 26, 27, 28 e 34.

DOGRUER, C. U.; KOKU, B.; DOLEN, M. A novel soft-computing technique to segment satellite images for mobile robot localization and navigation. In: 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems. [S.l.: s.n.], 2007. p. 2077–2082. Citado na página 15.

ELMENREICH, W. Deep convolutional neural networks for image classification: A comprehensive review. Vienna University of Technology, Austria, v. 502, p. 1–28, 2002. Citado 2 vezes nas páginas 17 e 35.

ESFAHANI, M. A.; WANG, H.; WU, K.; YUAN, S. Aboldeepio: A novel deep inertial odometry network for autonomous vehicles. IEEE Transactions on Intelligent Transportation Systems, v. 21, n. 5, p. 1941–1950, 2020. Citado na página 34.

FORESIGHT. The Future of Manufacturing: a new era of opportunity and challenge for the UK. 2021. <https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/255923/13-810-future-manufacturing-summary-report.pdf>. Citado na página 14.

GANGANATH, N.; LEUNG, H. Mobile robot localization using odometry and kinect sensor. In: 2012 IEEE International Conference on Emerging Signal Processing Applications. [S.l.: s.n.], 2012. p. 91–94. Citado na página 17.

GRUNNET-JEPSEN, H.; FULKERSON. Introduction to Intel® RealSense™ Visual SLAM and the T265 Tracking Camera. 2023. <<https://dev.intelrealsense.com/docs/intel-realsensetm-visual-slam-and-the-t265-tracking-camera>>. Citado 6 vezes nas páginas 28, 29, 30, 35, 39 e 43.

GUTMANN, J.-S.; FOX, D. An experimental comparison of localization methods continued. In: IEEE/RSJ International Conference on Intelligent Robots and Systems. [S.l.: s.n.], 2002. v. 1, p. 454–459 vol.1. Citado na página 16.

HALL, D.; LLINAS, J. An introduction to multisensor data fusion. Proceedings of the IEEE, v. 85, n. 1, p. 6–23, 1997. Citado na página 35.

INTEL, C. Robust Visual-Inertial Tracking from a Camera that Knows Where it's Going. 2022. <<https://www.intelrealsense.com/visual-inertial-tracking-case-study/>>. Citado 2 vezes nas páginas 39 e 40.

JUNG, Y. H.; PARK, B.; KIM, J. U.; KIM, T.-i. Bioinspired electronics for artificial sensory systems. Advanced Materials, Wiley Online Library, v. 31, n. 34, p. 1803637, 2019. Citado na página 17.

KOCH, V.; KUGE, S.; GEISSBAUER, D. R.; SCHRAUF, S. Industry 4.0: Opportunities and challenges of the industrial internet. In: . [S.l.: s.n.], 2014. Citado na página 14.

LEE, K.; CHUNG, W.; YOO, K. Kinematic parameter calibration of a car-like mobile robot to improve odometry accuracy. Mechatronics, v. 20, n. 5, p. 582–595, 2010. ISSN 0957-4158. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0957415810001042>>. Citado 3 vezes nas páginas 32, 33 e 34.

LEONARD, J.; DURRANT-WHYTE, H. Mobile robot localization by tracking geometric beacons. IEEE Transactions on Robotics and Automation, v. 7, n. 3, p. 376–382, 1991. Citado na página 15.

MARTINS, F. N.; SARCIANELLI-FILHO, M.; CARELLI, R. A velocity-based dynamic model and its properties for differential drive mobile robots. Journal of Intelligent & Robotic Systems, v. 85, n. 2, p. 277–292, 2017. ISSN 1573-0409. Disponível em: <<https://doi.org/10.1007/s10846-016-0381-9>>. Citado na página 21.

MARZULLO, K. Tolerating failures of continuous-valued sensors. ACM Trans. Comput. Syst., Association for Computing Machinery, New York, NY, USA, v. 8, n. 4, p. 284–304, nov 1990. ISSN 0734-2071. Disponível em: <<https://doi.org/10.1145/128733.128735>>. Citado na página 35.

MOHAMED, S. A. S.; HAGHBAYAN, M.-H.; WESTERLUND, T.; HEIKKONEN, J.; TENHUNEN, H.; PLOSILA, J. A survey on odometry for autonomous navigation systems. IEEE Access, v. 7, p. 97466–97486, 2019. Citado na página 17.

MOORE, T.; STOUCH, D. A generalized extended kalman filter implementation for the robot operating system. In: Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13). [S.l.]: Springer, 2014. Citado na página 48.

OUERGHI, S.; RAGOT, N.; BOUTTEAU, R.; SAVATIER, X. Comparative Study of a commercial tracking camera and ORB-SLAM2 for person localization. In: 15th International Conference on Computer Vision Theory and Applications. Valletta, France: SCITEPRESS - Science and Technology Publications, 2020. p. 357–364. Disponível em: <<https://hal.science/hal-02567816>>. Citado na página 18.

OUERGHI, S.; RAGOT, N.; BOUTTEAU, R.; SAVATIER, X. Comparative study of a commercial tracking camera and orb-slam2 for person localization. In: 15th International Conference on Computer Vision Theory and Applications. Valletta, France: [s.n.], 2020. p. 357–364. Disponível em: <<https://hal.archives-ouvertes.fr/hal-02567816>>. Citado na página 63.

- REIS, S. V. D. Por que os robôs colaborativos estão revolucionando a indústria? 2021. <<https://inforchannel.com.br/2021/03/17/por-que-os-robos-colaborativos-estao-revolucionando-a-industria/>>. Citado na página 14.
- ROS, D. Why ROS? 2021. Disponível em: <<https://www.ros.org/blog/why-ros/>>. Citado na página 42.
- ROS, D. Understanding ROS2 Nodes. 2023. Disponível em: <<https://docs.ros.org/en/humble/Tutorials/Beginner-CLI-Tools/Understanding-ROS2-Nodes/Understanding-ROS2-Nodes.html>>. Citado na página 43.
- ROS, O. robot_pose_ekf - Package Summary. 2022. [Http://wiki.ros.org/robot_pose_ekf](http://wiki.ros.org/robot_pose_ekf). Citado na página 48.
- SCARAMUZZA, D.; FRAUNDORFER, F. Visual odometry [tutorial]. IEEE Robotics Automation Magazine, v. 18, n. 4, p. 80–92, 2011. Citado na página 28.
- SCHMIDT, P. Intel® RealSense™ Tracking Camera T265 and Intel® RealSense™ Depth Camera D435 - Tracking and Depth. 2023. <<https://www.intelrealsense.com/visual-inertial-tracking-case-study/>>. Citado na página 42.
- SHEN, J.; TICK, D.; GANS, N. Localization through fusion of discrete and continuous epipolar geometry with wheel and imu odometry. In: Proceedings of the 2011 American Control Conference. [S.l.: s.n.], 2011. p. 1292–1298. Citado na página 17.
- SIEGWART, R.; NOURBAKHSI, I. R. Autonomous Mobile Robots. [S.l.]: MIT Press, 2011. Citado 6 vezes nas páginas 15, 26, 27, 30, 31 e 32.
- SIMHON, M. Robótica Industrial. USA: Moussa Simhon, 2011. ISBN 8591069331. Citado na página 14.
- SINHA, A.; CHEN, H.; DANU, D. G.; KIRUBARAJAN, T.; FAROOQ, M. Estimation and decision fusion: A survey. Neurocomput., Elsevier Science Publishers B. V., NLD, v. 71, n. 13–15, p. 2650–2656, aug 2008. ISSN 0925-2312. Disponível em: <<https://doi.org/10.1016/j.neucom.2007.06.016>>. Citado na página 36.
- SUKKARIEH, S.; NEBOT, E.; DURRANT-WHYTE, H. A high integrity imu/gps navigation loop for autonomous land vehicle applications. IEEE Transactions on Robotics and Automation, v. 15, n. 3, p. 572–578, 1999. Citado na página 17.
- TAREEN, S. A. K.; RAZA, R. H. Potential of sift, surf, kaze, akaze, orb, brisk, agast, and 7 more algorithms for matching extremely variant image pairs. In: 2023 4th International Conference on Computing, Mathematics and Engineering Technologies (iCoMET). [S.l.: s.n.], 2023. p. 1–6. Citado na página 29.
- TEDALDI, D.; PRETTO, A.; MENEGATTI, E. A robust and easy to implement method for imu calibration without external equipments. In: 2014 IEEE International Conference on Robotics and Automation (ICRA). [S.l.: s.n.], 2014. p. 3042–3049. Citado 3 vezes nas páginas 27, 28 e 32.
- TEXIER, G.; ALLODJI, R.; DIOP, L.; MEYNARD, J.-B.; PELLEGRIN, L.; CHAUDET, H. Using decision fusion methods to improve outbreak detection in disease surveillance. BMC Medical Informatics and Decision Making, v. 19, 03 2019. Citado na página 36.

- THRUN, S.; BURGARD, W.; FOX, D. Probabilistic Robotics. [S.l.]: MIT Press, 2005. Citado na página 25.
- WANG, K.; MA, S.; CHEN, J.; REN, F.; LU, J. Approaches, challenges, and applications for deep visual odometry: Toward complicated and emerging areas. IEEE Transactions on Cognitive and Developmental Systems, v. 14, n. 1, p. 35–49, 2022. Citado na página 34.
- WHEELTEC. ROS robot technical documentation. [S.l.], 2021. V3.5. Citado 2 vezes nas páginas 23 e 38.
- YOON, S.; YU, Y. A quantitative comparison of statistical and deterministic methods on virtual in-situ calibration in building systems. Building and Environment, v. 115, p. 54–66, 2017. ISSN 0360-1323. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0360132317300136>>. Citado na página 30.
- ZHANG, Z.; SCARAMUZZA, D. A tutorial on quantitative trajectory evaluation for visual(-inertial) odometry. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). [S.l.: s.n.], 2018. p. 7244–7251. Citado na página 29.
- ZHENG, H.; YANG, S. Research on race car steering geometry considering tire side slip angle. Proceedings of the Institution of Mechanical Engineers, Part P: Journal of Sports Engineering and Technology, v. 234, n. 1, p. 72–87, 2020. Disponível em: <<https://doi.org/10.1177/1754337119872417>>. Citado na página 21.