



INSTITUTO TECNOLÓGICO VALE



**Programa de Pós Graduação em Instrumentação, Controle e Automação de
Processos de Mineração - PROFICAM
Universidade Federal de Ouro Preto - Escola de Minas
Associação Instituto Tecnológico Vale - ITV**

Dissertação

**IMPLEMENTAÇÃO E AVALIAÇÃO DE TÉCNICAS DE ODOMETRIA
APLICADAS A UM DISPOSITIVO ROBÓTICO MÓVEL**

Eduardo Cota

**Ouro Preto
Fevereiro de 2019**

Eduardo Cota

**IMPLEMENTAÇÃO E AVALIAÇÃO DE TÉCNICAS DE ODOMETRIA
APLICADAS A UM DISPOSITIVO ROBÓTICO MÓVEL**

Dissertação apresentada ao curso de Mestrado Profissional em Instrumentação, Controle e Automação de Processos de Mineração da Universidade Federal de Ouro Preto e do Instituto Tecnológico Vale, como parte dos requisitos para obtenção do título de Mestre em Engenharia de Controle e Automação.

Linha de Pesquisa: Robótica

Orientador: Prof. D.Sc. Gustavo Medeiros Freitas

Ouro Preto, MG – Brasil
Fevereiro de 2019

C843i

Cota, Eduardo.

Implementação e Avaliação de técnicas de odometria aplicadas a um dispositivo robótico móvel [manuscrito] / Eduardo Cota. - 2019.

109f.: il.: color; grafs; tabs; mapas.

Orientador: Prof. Dr. Gustavo Medeiros Freitas.

Dissertação (Mestrado) - Universidade Federal de Ouro Preto. Escola de Minas. Departamento de Engenharia de Controle e Automação e Técnicas Fundamentais. Programa de Pós Graduação em Instrumentação, Controle e Automação de Processos de Mineração.

Área de Concentração: Engenharia de Controle e Automação de Processos Minerais.

1. Robô móvel. 2. Técnicas de odometria. 3. Modelagem cinemática. I. Medeiros Freitas, Gustavo. II. Universidade Federal de Ouro Preto. III. Título.

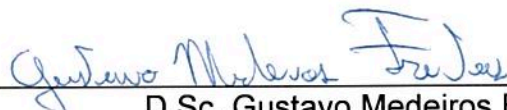
CDU: 681.5:622.2

Mestrado Profissional em Instrumentação, Controle e Automação de Processos
de Mineração - PROFICAM

Implementação e Avaliação de Técnicas de Odometria Aplicadas a um
Dispositivo Robótico Móvel

Eduardo Cota

Dissertação defendida e aprovada em 20 de fevereiro de 2019 pela banca
examinadora constituída pelos professores:



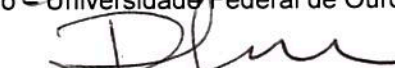
D.Sc. Gustavo Medeiros Freitas

Orientador – Universidade Federal de Minas Gerais (UFMG)



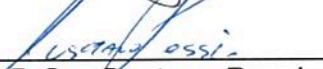
D.Sc. José Alberto Naves Cocota Júnior

Membro interno – Universidade Federal de Ouro Preto (UFOP)



D.Sc. Renato José Martins

Membro externo – Universidade Federal de Minas Gerais (UFMG)



D.Sc. Gustavo Pessin

Membro interno – Instituto Tecnológico Vale Mineração (ITV)

*Dedico esta dissertação aos meus
pais, minha irmã e meu amor,
Paula. Vocês são a melhor parte
de mim!*

Agradecimentos

Agradeço inicialmente à minha família e namorada pelo amor, apoio, suporte e paciência. À minha família de Luziânia por ter me acolhido inúmeras vezes durante o mestrado. À Lunáticos e à Damas de Ouro por terem sido minhas casas e refúgio em Ouro Preto - sem vocês não teria sido tão prazeroso! Aos grandes amigos Gabriel, Filipe e Marcos, por todos os momentos e participação ativa na construção desta dissertação. Ao Rodrigo Pastl pela inspiração, motivação e suporte quando eu pensei que não ia dar certo. Ao Thiago pelos conselhos e amizade. Ao Marcelo, Mocô, Ênio, Alexandre, Jacó, Guilherme, Luiz Eduardo, Luiz Guilherme e tantas outras amizades e parcerias construídas no PROFICAM e Laboratório de Robótica do ITV. Ao pessoal do ITV, em especial à Rubinha, Fran, Jamilly e turma da pelada. À minha tia Tina e família pela hospitalidade em BH em um momento crucial da finalização da dissertação. Por fim, aos meus grandes mentores: Hector e Gustavo, sem os quais não seria possível conquistar tudo que conquistei nestes últimos dois anos. Muito obrigado!

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001; do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq); da Fundação de Amparo à Pesquisa do Estado de Minas Gerais (FAPEMIG); e da Vale SA.

Resumo da Dissertação apresentada à Escola de Minas/UFOP e ao ITV como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

IMPLEMENTAÇÃO E AVALIAÇÃO DE TÉCNICAS DE ODOMETRIA APLICADAS A UM DISPOSITIVO ROBÓTICO MÓVEL

Eduardo Cota

Fevereiro/2019

Orientador: Gustavo Medeiros Freitas

Projeto: Dispositivo de Inspeção e Monitoramento Remoto de Cavidades - EspeleoRobô;
Equipe de Espeleologia e Tecnologia da Diretoria de Planejamento e Desenvolvimento de Ferrosos; Robótica Aplicada à Mineração.

O EspeleoRobô é uma plataforma robótica móvel projetada para realizar inspeção, monitoramento e mapeamento de cavidades naturais ao redor de áreas de mineração. O dispositivo tem sido utilizado não somente para este fim, mas também para diversas atividades em ambientes que ofereçam dificuldades ou riscos à realização por humanos, como dutos e galerias de barragens. Para melhorar a qualidade das operações com o robô, um sistema que calcule a distância total percorrida se faz necessário. Neste contexto, considerando que não há sinal GPS disponível nos locais de operação, este trabalho implementa e avalia técnicas de odometria utilizando abordagens com pacotes do ROS (*Robot Operating System*) e diferentes sensores incluindo *encoders*, câmera RGB-D, sensor LiDAR e unidade de medição inercial (IMU), para estimar o deslocamento do dispositivo robótico. Câmeras Vicon foram empregadas para capturar os movimentos definindo o *ground truth* com o intuito de avaliar os resultados obtidos. Os testes foram realizados em laboratório e o desempenho de cada técnica foi analisado visando encontrar o método mais apropriado e eficaz para a aplicação proposta.

Palavras-chave: Robô móvel, Técnicas de odometria, Modelagem cinemática.

Abstract of Dissertation presented to Escola de Minas/UFOP and ITV as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

IMPLEMENTATION AND EVALUATION OF ODOMETRY TECHNIQUES APPLIED TO A MOBILE ROBOTIC DEVICE

Eduardo Cota

February/2019

Advisor: Gustavo Medeiros Freitas

Project: Remote Cave Inspection and Monitoring Device - EspeleoRobô; *Equipe de Espeleologia e Tecnologia da Diretoria de Planejamento e Desenvolvimento de Ferrosos*; Mining Robotics.

The EspeleoRobô is a mobile robotic platform designed to perform inspection, monitoring and mapping of natural caves around mining areas. The device has been used not only for this purpose, but also for several activities that must be done in hazardous environments, such as ducts and dam's galleries. In order to improve the quality of those operations, an odometry system is mandatory. Considering that these environments are often GPS-denied, we implemented and evaluated odometry techniques using ROS (Robot Operating System) packages and different sensors, including encoders, RGB-D cameras, LiDAR and inertial measurement unity (IMU), in order to estimate the device displacement. We used Vicon cameras as motion capture system to define a ground truth and evaluate the obtained results. The tests were performed in laboratory and each technique was analyzed so that we could find the most appropriate and effective one for the proposed applications.

Keywords: Mobile robotics, Odometry techniques, Kinematic modeling.

Lista de Figuras

1.1	Cavidades naturais já classificadas e catalogadas em Carajás-PA.	3
1.2	Riscos da inspeção de cavidades.	3
1.3	(a) Primeira versão do EspeleoRobô; (b) Modelo RHex (DYNAMICS, 2017).	4
1.4	Novos mecanismos de locomoção: (a) Rodas; (b) Rodas tipo estrela; (c) Esteiras.	5
1.5	Sistema de engate rápido.	5
1.6	Modos híbridos de locomoção.	5
1.7	(a) Esquema da base de controle; (b) Interface gráfica.	6
1.8	Diagrama de códigos de locomoção (ROCHA, 2018).	6
1.9	(a) Torre de instrumentação durante teste; (b) Mapa de nuvem de pontos coloridos 3D.	7
1.10	EspeleoRobô realizando inspeções em mina abandonada localizada em área de risco, galeria de barragem e dutos.	8
1.11	<i>DARPA SubT Challenge</i>	9
2.1	Movimentos gerados por diferentes acionamentos dos motores esquerdo e direito de um modelo diferencial.	13
2.2	Plataforma robótica móvel Pioneer 3-AT (BASACA-PRECIADO <i>et al.</i> , 2014)	14
2.3	Localização dos centros instantâneos de rotação CIR (WANG <i>et al.</i> , 2018)	15
2.4	(a) Quadrotor equipado com câmeras; (b) Quadrotor equipado com laser (ACHTELIK <i>et al.</i> , 2009).	15
2.5	(a) Plataforma robótica Ariadne equipada com scanner laser 3D; (b) laser scanner 3D AIS (SURMANN <i>et al.</i> , 2003).	16
2.6	Diferença de performance entre trajetória com (vermelho) e sem (azul) o uso de IMU (HESCH <i>et al.</i> , 2014).	17
2.7	<i>Ground truth</i> (vermelho pontilhado) e trajetória 2D estimada (azul) (EN-DRES <i>et al.</i> , 2012).	17
2.8	Princípio do efeito <i>hall</i> , sem presença de campo magnético (HONEYWELL, 2017).	19

2.9	Princípio do efeito <i>hall</i> , com presença de campo magnético (HONEYWELL, 2017).	20
2.10	<i>Encoder</i> magnético com imã fixo, <i>switch</i> de efeito hall e eixo com engrenagem (PREDKO e MC COMB, 2011).	20
2.11	Esquema de funcionamento simplificado da medição de distância por meio do princípio <i>time-of-flight</i> (WALLHOFF <i>et al.</i> , 2007)	21
2.12	Posicionamento dos sensores Hokuyo UTM-30LX acoplados no protótipo BRAiVE e faixa de detecção (em verde) (BROGGI <i>et al.</i> , 2013).	22
2.13	(a) Sensor LiDAR modelo Hokuyo UTM-30LX; (b) Área de varredura (STUFF, 2018).	22
2.14	(a) Padrão emitido; (b) Imagem de profundidade; (c) <i>Skeleton</i> formado; (d) Imagem RGB (MACCORMICK, 2011).	23
2.15	Identificação de objetos com câmera RGB-D após processo de aprendizagem (LAI <i>et al.</i> , 2011).	24
2.16	Câmera RGB-D modelo Kinect da Microsoft.	24
2.17	Sensor IMU modelo MTi-G-710 da Xsens.	25
2.18	Câmeras Vicon modelos Bonita (a) e Vero (b).	26
2.19	Esferas refletivas posicionadas sobre mão humana formando objeto no <i>software</i> Tracker (BLANA, 2018).	27
2.20	Exemplo de volume de rastreamento formado com seis câmeras Vicon.	27
3.1	EspeleoRobô, sensores e suportes utilizados nos desenvolvimentos.	29
3.2	Modelo do uniciclo.	30
3.3	Modelo diferencial.	32
3.4	Modelo <i>skid steer</i> .	33
3.5	Localização dos CIR's dos flancos e do veículo (ROCHA, 2018).	34
3.6	Média de frequência ("average rate") de atualização dos valores de velocidade rotacional exercida nos atuadores.	35
3.7	Identificação (<i>matching</i>) de características correspondentes em duas imagens diferentes (RUBLEE <i>et al.</i> , 2011).	36
3.8	Modelo de gerenciamento de memória para garantir o processamento dos nós em tempo real (Labbé and Michaud, 2014).	37
3.9	Características correspondentes computadas ao longo de múltiplas imagens sobrepostas sobre a primeira imagem (FRAUNDORFER e SCARAMUZZA, 2012).	38
3.10	Características correspondentes corretas (esquerda) e equivocadas (direita) (CHUM e MATAS, 2008).	38
3.11	Minimização da distância entre superfícies para se calcular a transformada ocorrida entre movimentos (SMISTAD <i>et al.</i> , 2015).	39

4.1	Objeto utilizado para calibrar câmeras Vicon e definir volume de rastreamento.	42
4.2	Disposição das 8 câmeras no Laboratório de Robótica do ITV.	43
4.3	Espaço utilizado no Laboratório de Robótica do ITV.	43
4.4	(a) Posicionamento das esferas refletivas e (b) objeto criado no <i>software</i> Tracker para captura de movimento.	44
4.5	Esquema de ligação de equipamentos para testes.	45
4.6	Registros das simulações realizadas antes da implementação dos testes no robô real.	46
4.7	Esquema de ligação de equipamentos para testes.	47
4.8	Diferencial vs <i>skid-steer</i> (1) - Comparação de precisão dos modelos.	48
4.9	Diferencial vs <i>skid-steer</i> (2) - Comparação de precisão dos modelos.	49
4.10	Diferencial vs <i>skid-steer</i> (3) - Comparação de precisão dos modelos.	50
4.11	Diferencial vs <i>skid-steer</i> (4) - Comparação de precisão dos modelos.	51
4.12	Diferencial vs <i>skid-steer</i> (5) - Comparação de precisão dos modelos.	52
4.13	Diferencial vs <i>skid-steer</i> (6) - Comparação de precisão dos modelos.	53
4.14	Teste 1 - Trajetória computada.	55
4.15	Teste 1 - Erros computados - LiDAR vs. LiDAR+IMU.	56
4.16	Teste 1 - Distância total percorrida, erros e integral do erro absoluto (IAE).	58
4.17	Teste 2 - Trajetória computada.	59
4.18	Teste 2 - Erros computados - LiDAR vs. LiDAR+IMU.	60
4.19	Teste 2 - Distância total percorrida, erros e integral do erro absoluto (IAE).	62
4.20	Teste 3 - Trajetória computada.	63
4.21	Teste 3 - Erros computados - LiDAR vs. LiDAR+IMU.	64
4.22	Teste 3 - Distância total percorrida, erros e integral do erro absoluto (IAE).	66
4.23	Teste 4 - Trajetória computada.	67
4.24	Teste 4 - Erros computados - LiDAR vs. LiDAR+IMU.	68
4.25	Teste 4 - Distância total percorrida, erros e integral do erro absoluto (IAE).	70
4.26	Teste 5 - Trajetória computada.	71
4.27	Teste 5 - Erros computados - LiDAR vs. LiDAR+IMU.	72
4.28	Teste 5 - Distância total percorrida, erros e integral do erro absoluto (IAE).	74
4.29	Teste 6 - Trajetória computada.	76
4.30	Teste 6 - Erros computados - LiDAR vs. LiDAR+IMU.	77
4.31	Teste 6 - Distância total percorrida, erros e integral do erro absoluto (IAE).	78
4.32	Comparação da precisão do cálculo odométrico feito por cada método (acima), mesmo gráfico porém ajustado para melhor visualização (abaixo).	79
4.33	Precisão no ponto de chegada do robô após finalizado cada teste.	80

Lista de Tabelas

4.1	Diferencial vs <i>skid-steer</i> (1) - Tabela comparativa dos erros de odometria. .	48
4.2	Diferencial vs <i>skid-steer</i> (2) - Tabela comparativa dos erros de odometria. .	49
4.3	Diferencial vs <i>skid-steer</i> (3) - Tabela comparativa dos erros de odometria. .	50
4.4	Diferencial vs <i>skid-steer</i> (4) - Tabela comparativa dos erros de odometria. .	51
4.5	Diferencial vs <i>skid-steer</i> (5) - Tabela comparativa dos erros de odometria. .	52
4.6	Diferencial vs <i>skid-steer</i> (6) - Tabela comparativa dos erros de odometria. .	53
4.7	Teste 1 - Tabela de erros na trajetória.	57
4.8	Teste 1 - Tabela de erros de odometria.	58
4.9	Teste 2 - Tabela de erros na trajetória.	61
4.10	Teste 2 - Tabela de erros de odometria.	63
4.11	Teste 3 - Tabela de erros na trajetória.	65
4.12	Teste 3 - Tabela de erros de odometria.	66
4.13	Teste 4 - Tabela de erros na trajetória.	69
4.14	Teste 4 - Tabela de erros de odometria.	70
4.15	Teste 5 - Tabela de erros na trajetória.	73
4.16	Teste 5 - Tabela de erros de odometria.	74
4.17	Teste 6 - Tabela de erros na trajetória.	75
4.18	Teste 6 - Tabela de erros de odometria.	77

Sumário

1	Introdução	1
1.1	A mineração e aspectos ambientais relacionados	2
1.2	O EspeleoRobô	4
1.3	Motivação	8
1.4	Objetivos	10
1.4.1	Objetivo geral	10
1.4.2	Objetivos específicos	10
1.5	Organização do trabalho	10
2	Revisão Bibliográfica	12
2.1	Sensores e sistema de validação empregados	18
2.1.1	<i>Encoder</i>	18
2.1.2	LiDAR	20
2.1.3	Câmera RGB-D	23
2.1.4	Unidade de medição inercial (IMU)	25
2.1.5	Sistema de captura de movimento com câmeras Vicon	25
3	Estudos e implementações	28
3.1	EspeleoRobô e instrumentação embarcada	28
3.2	Técnicas de odometria	29
3.2.1	Odometria com <i>encoders</i>	30
3.2.1.1	Modelo do uniciclo	30
3.2.1.2	Modelo diferencial	31
3.2.1.3	Modelo <i>skid steer</i>	33
3.2.2	Odometria visual com câmera RGB-D	36
3.2.3	Odometria com sensor LiDAR	39
3.2.3.1	Odometria com sensor LiDAR e unidade de medição inercial	40
3.3	Implementação do cálculo de distância total percorrida	41
4	Resultados experimentais	42
4.1	Sistema de validação e estrutura de testes	42

4.2	<i>Software</i>	46
4.3	Comparação preliminar de modelos cinemáticos: diferencial vs <i>skid-steer</i> . .	47
4.4	Avaliação das técnicas de odometria	53
4.4.1	Resultados experimentais em testes sem restrição de velocidade . .	54
4.4.1.1	Teste 1	55
4.4.1.2	Teste 2	59
4.4.1.3	Teste 3	63
4.4.2	Resultados experimentais com restrição de velocidade em 50% da capacidade máxima	67
4.4.2.1	Teste 4	67
4.4.2.2	Teste 5	71
4.4.2.3	Teste 6	75
4.4.3	Análise geral	78
4.4.4	Considerações sobre implementação com <i>encoders</i>	80
4.4.5	Considerações sobre implementação com câmera RGB-D	81
4.4.6	Considerações sobre implementação com sensor LiDAR e LiDAR+IMU	81
5	Conclusões	83
5.1	Contribuições da dissertação	84
5.1.1	Integração Vicon-ROS	84
5.1.2	Estrutura de testes para odometria do EspeleoRobô	84
5.1.3	Odometria <i>online</i>	84
5.1.4	Publicações e apresentações	85
5.2	Sugestões de trabalhos futuros	86
	Referências Bibliográficas	88

Capítulo 1

Introdução

Robôs móveis são dispositivos teleoperados, semi ou completamente autônomos, desenvolvidos para atividades específicas em que a mobilidade é necessária. Geralmente, essas atividades ocorrem em ambientes que oferecem algum tipo de risco ao ser humano, ou mesmo a dificuldade ou impossibilidade de acesso pelo mesmo. Desta forma, esses dispositivos são utilizados para diversos fins, tais como operações militares Hill e Bodt (2007), exploração espacial Atherton *et al.* (2006), reconhecimento e vigilância Burke *et al.* (2004), inspeção de dutos Chablat *et al.* (2019), inspeção de turbinas Sahbel *et al.* (2019), resgate em situações de desastre Allali e Benchaïba (2019), Spenko *et al.* (2018) e outros.

O sucesso na realização dessas atividades está diretamente relacionado à capacidade dos dispositivos em identificarem e interagirem com o ambiente de operação, bem como com objetos detectados no mesmo. A evolução da eletrônica trouxe a miniaturização dos componentes, que possibilitou o desenvolvimento de sensores cada vez menores e precisos, proporcionando a coleta de informações relevantes ao funcionamento do robô. O avanço na computação, por sua vez, aumentou a capacidade de processamento dos computadores, o que permitiu o desenvolvimento de algoritmos mais sofisticados para a manipulação eficaz destes dados. A combinação destes progressos resultou em grande avanço da robótica móvel nos últimos anos.

Neste sentido, a localização dos robôs móveis tem sido objeto de estudo e é considerada fator determinante para o alcance dos objetivos especificados no escopo das operações realizadas em ambientes remotos. Os robôs autônomos necessitam se localizar com eficiência, pois estes irão definir suas próprias referências e quais trajetórias traçar em determinado instante, bem como calcular o retorno à sua base. Na teleoperação, um dos maiores desafios é proporcionar ao operador uma sensação de presença estando em um ambiente remoto, logo informações sobre a localização ou distância percorrida podem aumentar a qualidade da teleoperação.

Inserido neste contexto, o presente trabalho apresenta uma avaliação de técnicas de odometria para o EspeleoRobô, um dispositivo robótico móvel idealizado para atividades de inspeção e monitoramento de cavidades, mas que tem sido empregado em diversas

outras atividades desempenhadas em ambientes que ofereçam riscos à realização ou ao acesso de humanos, como dutos e galerias de barragens. Os desenvolvimentos visam implementar a técnica de odometria mais adequada para o dispositivo e suas aplicações. Tendo em vista a complexidade do tema, as implementações realizadas são simplificadas de modo a considerar as operações em terrenos planares.

1.1 A mineração e aspectos ambientais relacionados

O Brasil possui vasta reserva mineral, o que faz da mineração uma atividade tradicional no país. Grande parte da reserva brasileira se encontra na Serra dos Carajás, no Pará e o potencial da região tem sido estudado há décadas (BEISIEGEL *et al.*, 1973). O local desde então chama atenção não apenas pelo volume da reserva, mas também pela alta concentração de ferro no minério. Atualmente, o Projeto S11D é considerado o maior complexo da mineradora Vale SA e atinge marcas expressivas, como o maior investimento privado realizado no país na última década, com valor de US\$ 14,3 bilhões (VALE, 2018).

De acordo com Cunha *et al.* (2007), o Brasil não somente está entre os países com as maiores reservas minerais de ferro do mundo, mas sem dúvida também contém o maior potencial em termos de número de cavernas com este minério. Apenas na região de Serra dos Carajás existem centenas de cavernas com esta tipologia.

Segundo Figueiredo *et al.* (2010), para obter licenças ambientais a fim de explorar tais regiões, a Vale deve atender a Legislação de Proteção ao Patrimônio Cultural Nacional e Ambiental através do Decreto nº 99.556 de 01/10/1990; portaria IBAMA nº 887/90 de 15/06/1990 e resoluções do CONAMA. Ruchkys *et al.* (2015) acrescentam que o Decreto 6.640/2008 altera o de 1990 trazendo o conceito de grau de relevância para classificar as cavernas e definir diferentes medidas de proteção. Quando o grau de relevância for classificado como máximo, a caverna não pode sofrer nenhum tipo de dano irreversível, enquanto para os outros graus, respeitadas as devidas medidas compensatórias, poderá haver exploração mineral no local. Os critérios para se definir a relevância são dispostos no Art. 2º do novo decreto: “análise de atributos ecológicos, biológicos, geológicos, hidrologicos, paleontológicos, cênicos, histórico-culturais e socioeconômicos, avaliados sob enfoque regional e local”.

Em conformidade com o disposto na legislação, as mineradoras devem atender às exigências legais e realizar o estudo espeleológico, bem como preencher outros requisitos. Monteiro (2011) define a Espeleologia como sendo a ciência que realiza estudos das cavidades naturais quanto à sua origem e evolução, ao meio físico que elas representam, ao seu ecossistema atual ou passado, e também, aos meios e técnicas que são próprias do seu estudo. Portanto, trata-se de uma ciência multidisciplinar que visa o estudo e a exploração de cavernas. Na Figura 1.1 é possível identificar a localização de cavidades naturais já classificadas e catalogadas, de acordo com o grau de relevância, em Carajás-PA.

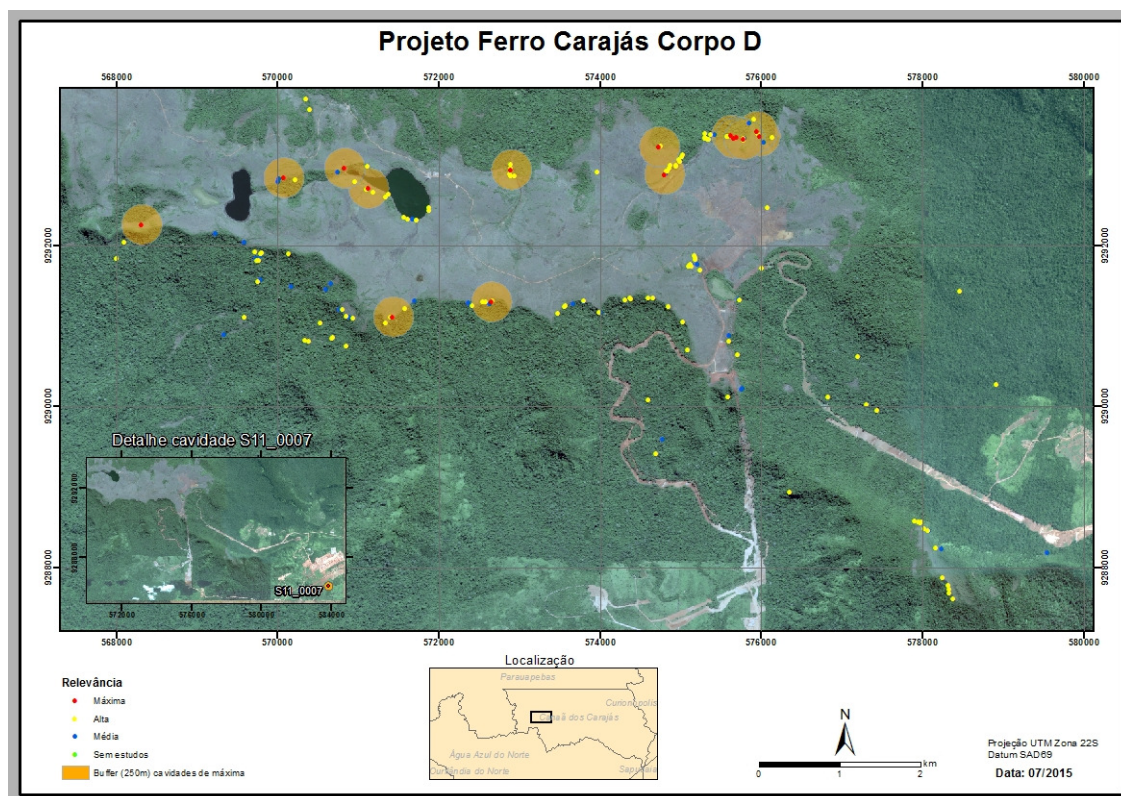


Figura 1.1: Cavidades naturais já classificadas e catalogadas em Carajás-PA.

Para realizar o estudo espeleológico a equipe de espeleologia precisa acessar o interior das cavidades para mapear, bem como coletar informações gerais das mesmas. A operação de inspeção, no entanto, pode se tornar crítica pela própria dificuldade de acesso e outros riscos inerentes à atividade como: confinamento, desabamento, fungos, gases tóxicos e ainda animais selvagens ou peçonhentos (Figura 1.2).



Figura 1.2: Riscos da inspeção de cavidades.

Considerando a relevância do tema meio ambiente nos dias atuais e visando a redução da exposição de operadores a esses riscos, foi iniciado na Vale S.A o projeto EspeleoRobô para auxiliar a equipe na realização dos estudos espeleológicos, bem como nas atividades de inspeção e monitoramento das cavernas.

Como já dito anteriormente, apesar desta proposta inicial ser voltada para cavidades naturais, o dispositivo também pode ajudar em operações de inspeção de dutos e galerias de barragens, bem como outros ambientes confinados em geral.

1.2 O EspeleoRobô

O EspeleoRobô é um projeto iniciado para reduzir a exposição de espeleólogos aos riscos encontrados em atividades de inspeção de cavidades naturais. Para cumprir os objetivos do projeto, foi adquirida uma plataforma robótica móvel (Figura 1.3a), inspirada no modelo RHex da Boston Dynamics ¹ (Figura 1.3b). A configuração inicial da plataforma era composta por seis pernas, acionadas por motores individuais coordenando sequências de movimentos pré-programados (marchas). O dispositivo, entretanto, apresentou dificuldade de mobilidade nos testes iniciais. Buscando reestruturar e estender as capacidades do robô e adequá-lo aos ambientes de operação, o Instituto Tecnológico Vale foi acionado para efetuar uma série de modificações mecânicas, de *hardware* e de *software*. As adaptações realizadas são detalhadas de forma aprofundada por Cota *et al.* (2017b). Nesta seção são destacadas as mais relevantes para o presente trabalho.



(a)



(b)

Figura 1.3: (a) Primeira versão do EspeleoRobô; (b) Modelo RHex (DYNAMICS, 2017).

Com o intuito de aumentar a capacidade de locomoção do robô, bem como sua adaptabilidade em diferentes tipos de terrenos, outros mecanismos e modos de locomoção foram propostos. Além das pernas, rodas, rodas tipo estrela e esteiras já estão sendo utilizadas no dispositivo. A criação de um sistema de engate rápido ainda possibilitou a combinação desses mecanismos formando modos híbridos de locomoção, que podem ser definidos em campo pelo próprio operador ao considerar as características do terreno. Os novos mecanismos, o sistema de engate rápido e os modos híbridos de locomoção podem ser verificados nas Figuras 1.4, 1.5 e 1.6.

¹www.bostondynamics.com/robot_rhex.html

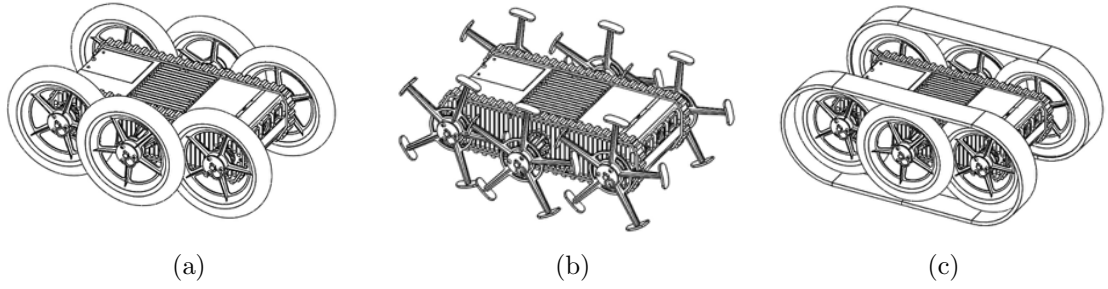


Figura 1.4: Novos mecanismos de locomoção: (a) Rodas; (b) Rodas tipo estrela; (c) Esteiras.

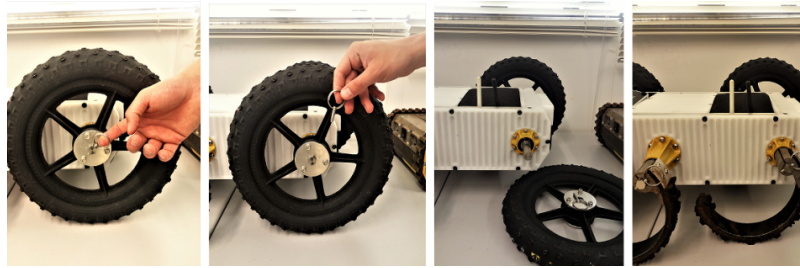


Figura 1.5: Sistema de engate rápido.

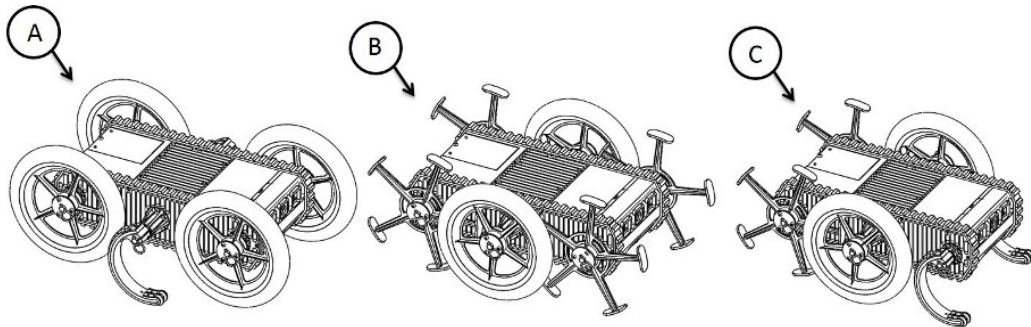


Figura 1.6: Modos híbridos de locomoção.

Uma nova base de controle também foi implementada para melhorar o envio dos comandos, alcance dos sinais e interação com o operador por meio de nova interface gráfica (Figura 1.7). O processamento central do robô passou a ser executado por um minicomputador Intel NUC, aumentando as capacidades de velocidade e memória do dispositivo robótico, que era anteriormente controlado por uma placa microcontrolada Arduino.

As novas especificações de *hardware* permitiram uma reformulação geral no *software* do robô com o *Robot Operating System*² (ROS). Este é um *middleware* utilizado para prototipagem rápida de robôs (QUIGLEY *et al.*, 2009) que permite uma comunicação mais eficiente, além de viabilizar mais facilmente a integração da plataforma robótica com sensores de diversos tipos e fabricantes. Esta ferramenta também possibilita a gravação de

²www.ros.org

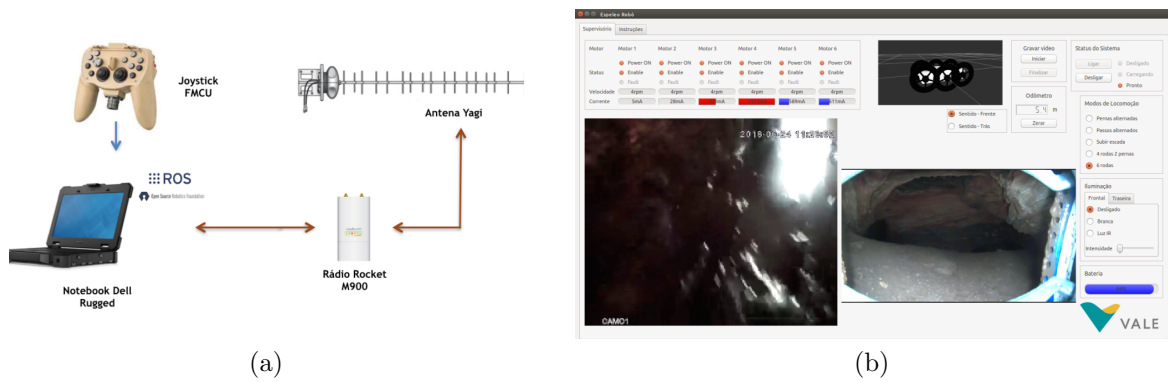


Figura 1.7: (a) Esquema da base de controle; (b) Interface gráfica.

relatórios, chamados *bags*, para reproduzir experimentos virtualmente, sem a necessidade de repetir fisicamente procedimentos onerosos.

Com ROS, a programação é desenvolvida de forma modular: cada função do dispositivo é separada em um nó. Os nós se comunicam por meio de tópicos, publicando ou subscrevendo nestes. Na Figura 1.8 é possível observar os tópicos (vermelho), nós (verde), e bibliotecas (azul) implementados para realizar a locomoção do dispositivo.

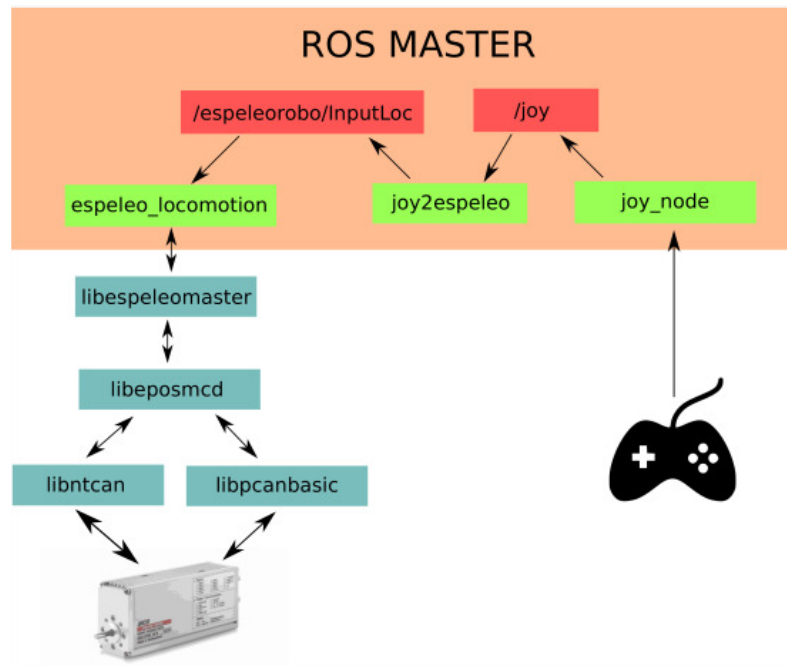


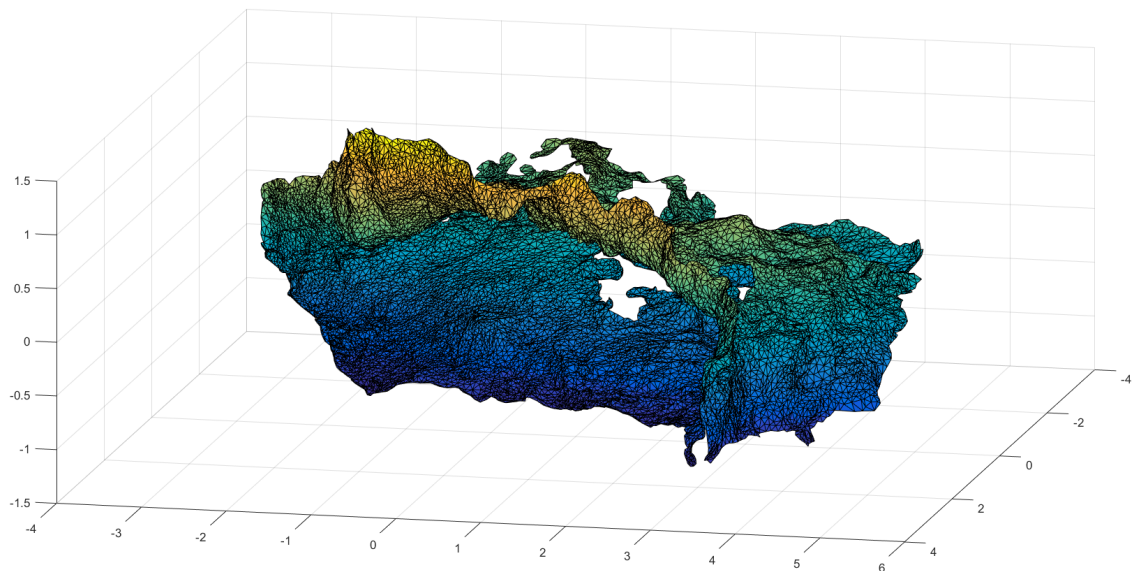
Figura 1.8: Diagrama de códigos de locomoção (ROCHA, 2018).

Primeiramente o nó “`joy_node`” interpreta os comandos recebidos na base de controle através do *joystick* e publica no tópico “`/joy`”. Em seguida, o nó “`joy2espeleo`” acessa (subscreve) os comandos publicados no tópico “`/joy`” e os traduz para serem publicados no tópico “`/espeleorobo/InputLoc`” de forma que o robô entenda o comando. Por fim, o nó “`espeleo_locomotion`” então acessa as informações de comando traduzidas e invoca as bibliotecas responsáveis pelo acionamento dos motores por meio de protocolo CAN.

Além da reestruturação implementada, também foi iniciado, em conjunto com o SENAI Cimatec da Bahia, um projeto para construção de uma torre de instrumentação para a realização do mapeamento de ambientes monitorados com nuvem de pontos coloridos 3D (Figura 1.9).



(a)



(b)

Figura 1.9: (a) Torre de instrumentação durante teste; (b) Mapa de nuvem de pontos coloridos 3D.

Após as adaptações e melhorias, o EspeleoRobô se mostrou uma solução versátil, não somente para inspeção e monitoramento de cavernas, mas também para atividades em ambientes de difícil acesso, espaços confinados e locais onde a entrada de humanos é perigosa. Nesse sentido, o robô tem sido útil em inspeções de galeria de barragens, minas desativadas, áreas de risco de desabamento, além de cavidades localizadas em áreas de mineração (Figura 1.10).

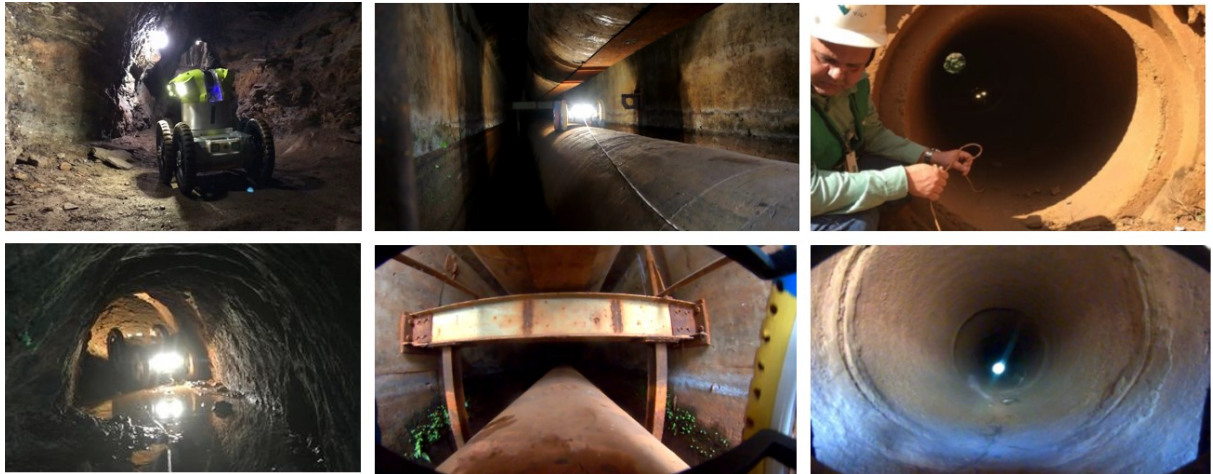


Figura 1.10: EspeleoRobô realizando inspeções em mina abandonada localizada em área de risco, galeria de barragem e dutos.

1.3 Motivação

Uma das maiores dificuldades da operação de dispositivos robóticos móveis em ambientes confinados ou restritos é medir a distância total percorrida, posição e orientação do robô no interior destes locais. Isso se deve basicamente à ausência de localização global (uso de GPS ou similares) e ao escorregamento das rodas. Em relação ao EspeleoRobô e suas aplicações, existem outros dois fatores dificultadores para a realização do cálculo de odometria: a baixa confiabilidade de magnetômetros em ambientes com concentração de ferro (como cavernas); e a utilização de diferentes mecanismos de locomoção, o que dificulta o estudo cinemático e a definição de modelos para descrever matematicamente os movimentos do robô.

A complexidade, relevância e atualidade do tema é confirmada pelo desafio *Subterranean (SubT) Challenge* (Figura 1.11), proposto pela DARPA (*Defense Advanced Research Projects Agency*)³ em 2018, que conta com circuito em caverna agendado para 2020 e evento final em 2021. O desafio propõe que competidores desenvolvam conceitos de operação que explorem a capacidade de conduzir missões subterrâneas de forma autônoma e eficiente, possibilitando o salvamento de vidas; e a composição de sistema para oferecer mobilidade com velocidade operacional relevante em ambientes subterrâneos complexos, imprevisíveis e diversos.

Tendo em vista as dificuldades descritas e relevância do assunto abordado, esta dissertação corresponde a uma primeira etapa da solução do problema de cálculo de odometria do EspeleoRobô, implementando técnicas desenvolvidas para ambientes estruturados, e validando as mesmas por meio de testes em laboratório.

³A DARPA é a Agência de Projetos de Pesquisa Avançada de Defesa do governo americano. Com origem militar, visando alcançar avanços tecnológicos, a agência tem promovido diversos desafios no ramo da robótica, como o DARPA Urban Challenge e o DARPA Gran Challenge, relacionados à navegação autônoma.

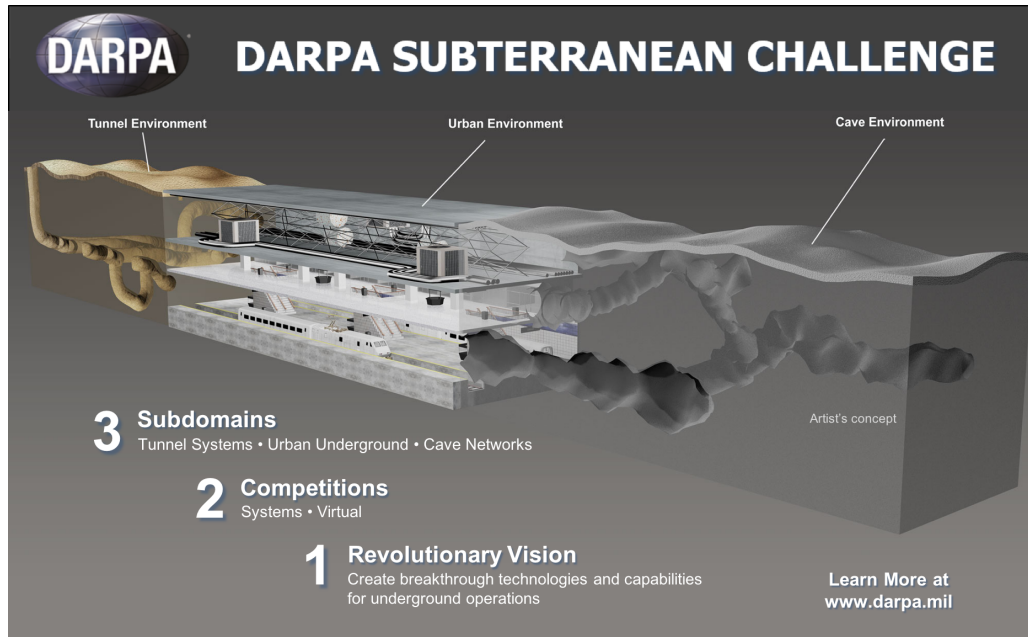


Figura 1.11: *DARPA SubT Challenge*.

Apesar das simplificações adotadas, as estratégias para estimação de odometria investigadas nesta dissertação apresentam resultados satisfatórios em laboratório, sendo provavelmente capazes de calcular com precisão adequada a distância percorrida por um dispositivo robótico durante a inspeções realizadas em ambientes estruturados, como dutos e galerias de barragens. Além disso, os estudos apresentados aqui serão necessários para a implementação da odometria 3D do EspeleoRobô, que possibilitará a solução de problemas mais complexos como a localização e mapeamento simultâneo (do inglês, *Simultaneous Localization and Mapping* - SLAM) e navegação autônoma, independentemente das características do terreno.

No caso do EspeleoRobô, uma potencial aplicação da odometria 3D é na geração de mapas tridimensionais dos ambientes inspecionados, que é diretamente influenciada por problemas de localização do robô durante a fusão de mapas locais. Para mapear todo o interior de uma caverna são escolhidos determinados pontos onde a torre deve gerar um mapa local. O mapeamento completo é feito de vários desses mapas locais que devem ser fundidos para se completar, formando um só mapa, que corresponde ao interior da cavidade como um todo. O sucesso desse procedimento está atrelado ao cálculo correto da posição e orientação de cada um dos pontos onde foi gerado um mapa local. Essa localização relativa entre os pontos deve ser conhecida para que sejam calculadas as transformadas corretas no procedimento de fusão dos mapas locais.

Não se limitando ao EspeleoRobô, os desenvolvimentos propostos podem ser aplicados em diferentes arquiteturas de robôs móveis, com diferentes mecanismos de locomoção. Foram desenvolvidas quatro implementações utilizando *encoders*, câmera RGB-D, sensor LiDAR e sensor LiDAR combinado com IMU. Para integrar os sensores com o robô e

processar os dados gerados foram utilizados pacotes do ROS. Testes realizados em laboratório compararam a performance das abordagens com um preciso *ground truth*, formado por um sistema de captura de movimento com oito câmeras Vicon⁴.

1.4 Objetivos

Os objetivos são divididos em geral e específicos. O objetivo geral apresenta a ideia central a ser desenvolvida pelo trabalho, e os específicos são os desdobramentos necessários para a consecução da meta principal.

1.4.1 Objetivo geral

Implementar e avaliar técnicas para calcular a distância percorrida por um robô móvel em ambientes 2D estruturados, como solução preliminar do problema de localização e mapeamento.

1.4.2 Objetivos específicos

- Realizar estudos de modelos cinemáticos que representam veículos com rodas;
- Escolher sensores e técnicas de odometria adequadas ao dispositivo utilizado e suas aplicações;
- Preparar metodologia de testes interligando sistema de captura de movimento (*ground truth*), sensores e robô utilizando ROS;
- Desenvolver código para processar dados e gerar informações de odometria online;
- Executar testes em laboratório e analisar os respectivos resultados.

1.5 Organização do trabalho

Este capítulo introduziu o contexto, as propostas de desenvolvimento e os objetivos desta dissertação. O aspecto ambiental no qual estão inseridas as atividades de mineração, bem como as melhorias acarretadas pelas implementações propostas, destacam a relevância do trabalho.

O Capítulo 2 apresenta uma discussão sobre trabalhos relacionados, com estudos de modelos cinemáticos e técnicas de odometria para robô móveis. Nesse capítulo também são apresentados os sensores utilizados por esta dissertação e sistema de validação empregado.

⁴<https://www.vicon.com/>

Em seguida, o Capítulo 3 descreve as técnicas de odometria implementadas com *encoders*, câmera RGB-D, LiDAR e IMU. São detalhados os funcionamentos das técnicas a partir da leitura dos sensores utilizados e a manipulação dessas informações por meio de algoritmos para obtenção dos dados odométricos.

O Capítulo 4 apresenta maiores detalhes do sistema de validação, a estrutura de testes realizados no laboratório, a metodologia dos experimentos, resultados experimentais encontrados, bem como análise geral dos mesmos.

Por fim, o Capítulo 5 traz as conclusões, além das contribuições da dissertação e sugestões de trabalhos futuros que podem complementar os resultados alcançados.

Capítulo 2

Revisão Bibliográfica

Propostas de localização de dispositivos robóticos móveis são vastamente discutidas na literatura (ACHTELIK *et al.*, 2009; MAIMONE *et al.*, 2007; PEI e KLEEMAN, 2017). Geralmente, a localização do robô móvel está associada à atividade de mapeamento de determinado local ou ambiente. Dessa forma, a solução do problema de localização e mapeamento simultâneos (SLAM) tem estimulado a produção de diversos trabalhos e técnicas na área da robótica.

A maioria dos desenvolvimentos propostos envolve o conceito de odometria para estimar a localização dos dispositivos. De forma simplificada, a odometria está presente na vida da maioria das pessoas por meio do odômetro, dispositivo que computa e indica, no painel dos veículos automotivos, a distância (popularmente dita quilometragem) percorrida pelo carro. Segundo Bruno e Oussama (2008), a palavra odometria vem do grego *hodos* que significa viagem ou jornada, e *metron* que significa medida ou medição. Ainda segundo os autores, no contexto de veículos autônomos, a odometria se refere ao uso de dados de atuadores para estimar o movimento geral do veículo. De acordo com Dudek e Jenkin (2010), o conceito básico consiste em desenvolver um modelo matemático que descreve como o movimento coordenado de juntas induz movimento no veículo como um todo, e então integrar esses movimentos específicos ao longo do tempo com o intuito de criar um modelo de posição e orientação atual do veículo.

Um dos maiores problemas do uso destes modelos matemáticos é que estes são representações limitadas e simplificadas da realidade, portanto trazem consigo erros que se propagam e aumentam ao longo do tempo. Dessa forma, muitos autores propõem combinações de diferentes abordagens de cálculos visando minimizar os erros aos quais o sistema está submetido.

O modelo mais básico para representar veículos com rodas é o modelo do uniciclo, que consiste basicamente em uma roda com velocidade linear e angular. Para representar veículos com rodas sem sistema de direção, duas arquiteturas são comumente utilizadas: diferencial e *skid-steer*. Nessas arquiteturas, a direção do dispositivo é baseada no controle das velocidades relativas dos lados esquerdo e direito. Ao contrário do que é comum nos

carros de passeio e veículos de transporte, as rodas dianteiras não realizam ângulos de esterçamento para desempenhar curvas.

O modelo diferencial é composto por duas rodas paralelas montadas em um eixo comum, controladas por motores separados. O robô diferencial navega em linha reta, para frente ou para trás, caso as velocidades aplicadas tenham mesmo sentido e módulo. As curvas são realizadas quando uma roda tem velocidade maior que a outra. E caso as rodas tenham velocidades iguais em sentidos opostos, o dispositivo gira em torno do próprio eixo sem se deslocar. Na Figura 2.1 é ilustrado o movimento gerado por um robô diferencial em relação ao acionamento das rodas direita e esquerda. Apesar da simplicidade do modelo, o mesmo possui algumas limitações e para representá-lo são assumidas algumas hipóteses, dentre elas a de que não há escorregamento das rodas.

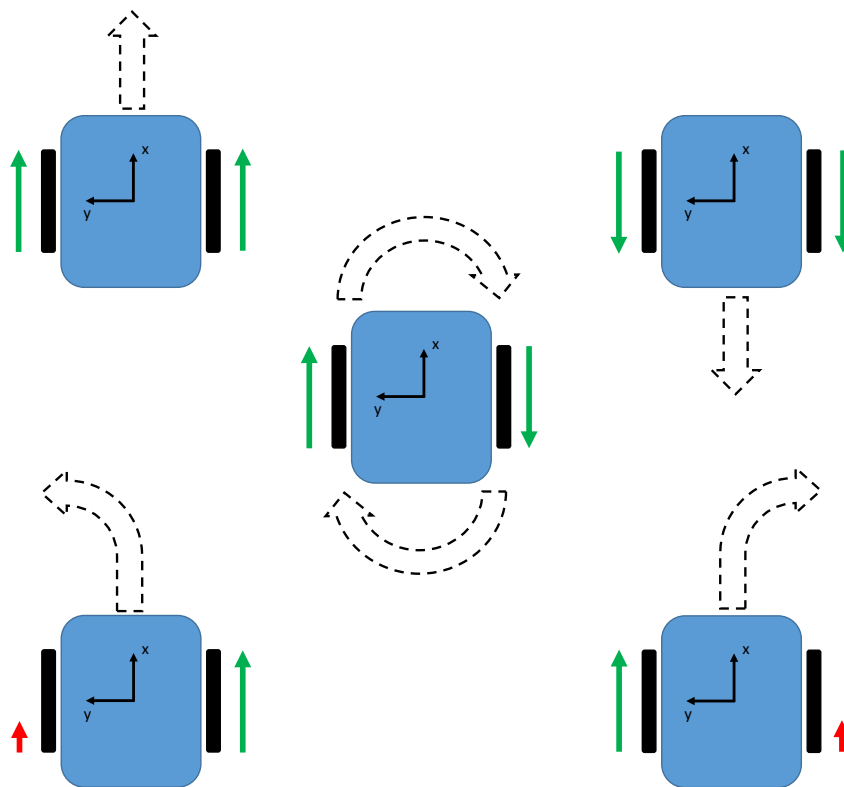


Figura 2.1: Movimentos gerados por diferentes acionamentos dos motores esquerdo e direito de um modelo diferencial.

O modelo *skid steer* geralmente utiliza dois ou mais pares de rodas ou esteiras para se locomover (Figura 2.2). O veículo *skid steer* também não realiza o esterçamento das rodas, logo a execução de curvas pelo robô requer o deslizamento das rodas (WANG *et al.*, 2015). Isso ocorre porque as esteiras ou rodas de cada lado do robô são acionadas com uma mesma velocidade, apesar de corresponderem a inúmeros centros de rotação instantâneos distintos. Segundo Pentzer *et al.* (2014), assumir que não há escorregamento em veículos *skid-steer* causa má estimativa de posição e orientação.

Geralmente, para computar o movimento das rodas são utilizados *encoders*, sensores empregados para medir o giro destas. Quando a roda escorrega, o movimento é computado, porém o robô não se desloca. Dessa forma, esse erro será computado e propagado por todo o cálculo de localização, somando-se a outros erros e causando imprecisão.



Figura 2.2: Plataforma robótica móvel Pioneer 3-AT (BASACA-PRECIADO *et al.*, 2014)

Para atenuar efeitos do escorregamento, diversos trabalhos têm sido desenvolvidos, entretanto cada dispositivo e local de operação possuem características próprias, acarretando diferentes variáveis, como a interação com diferentes solos e materiais, o que dificulta a consolidação de técnica eficientes para determinar e compensar o escorregamento das rodas. Wang *et al.* (2018) propõe a estimativa adaptativa de centros instantâneos de rotação (CIR) de acordo com o terreno para calcular a odometria de robôs com esteiras. Segundo os autores as localizações dos CIR's são capazes de refletir o escorregamento lateral e longitudinal do robô (Figura 2.3). Dessa forma, elas podem ser consideradas como um tipo de parâmetro de escorregamento. Para implementar, os autores utilizaram fusão de dados dos *encoders* com GPS-compass¹ para prover estimativa em tempo real dos CIR's e melhorar os resultados de odometria em 4 terrenos testados.

Devido à ausência do sinal GPS em ambientes fechados, à imprecisão da odometria utilizando a leitura incremental de *encoders* (proveniente do escorregamento), e ao avanço na área da visão computacional, surge o conceito de odometria visual citado por Nistér *et al.* (2004), Maimone *et al.* (2007), Steinbrücker *et al.* (2011) e outros, além da utilização de odometria baseada em sensores laser (ARRAS e VESTLI, 1998; HAHNEL *et al.*, 2003; LU e MILIOS, 1997; WULF *et al.*, 2004). Nestas odometrias alternativas são utilizados câmeras e sensores laser para aquisição de informações referenciais. As informações adquiridas, juntamente com cálculos matemáticos, permitem a determinação de distâncias

¹Variação do GPS comum patenteado por Chang *et al.* (1998)

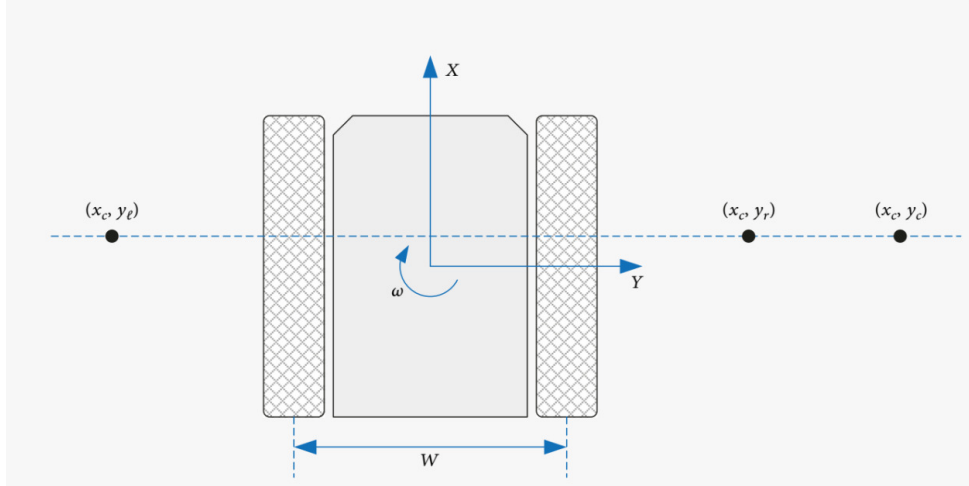


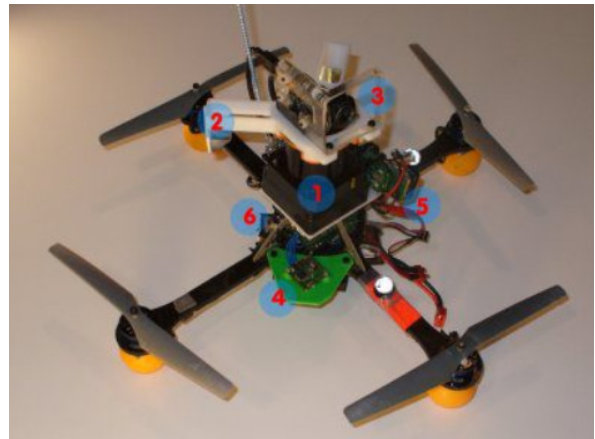
Figura 2.3: Localização dos centros instantâneos de rotação CIR (WANG *et al.*, 2018)

entre o robô móvel e objetos fixos ao seu redor, que passam a ser utilizados como referência durante a navegação. Em ambientes controlados, é possível ainda a utilização de *landmarks* artificiais (CHENAVIER e CROWLEY, 1992), que são referências adicionadas ao ambiente a fim de facilitar a resolução do problema de localização.

No desenvolvimento de Achtelik *et al.* (2009) é proposta uma comparação de duas abordagens para estimar posição e orientação de um helicóptero *quadroto* em ambiente fechado não estruturado e com ausência de sinal GPS. Foi implementada uma abordagem utilizando um sensor laser *rangefinder* e outra utilizando câmera *stereo*. Ao analisar os resultados é possível observar clara superioridade da técnica com laser. Para validação dos resultados, o *ground truth* foi obtido através de um sistema de captura de movimento com câmeras Vicon. Na Figura 2.4 é possível observar as duas montagens propostas com câmera e com laser, além das esferas refletivas fixadas no Quadroto, utilizadas para possibilitar o rastreo dos movimentos descritos com o sistema de captura.



(a)



(b)

Figura 2.4: (a) Quadroto equipado com câmeras; (b) Quadroto equipado com laser (ACHTELIK *et al.*, 2009).

Surmann *et al.* (2003) propõem um sistema automático de exploração e digitalização de ambientes fechados utilizando um robô móvel (Figura 2.5 (a)), um sensor laser *rangefinder* (Figura 2.5 (b)) e três módulos de *software*. O primeiro módulo utiliza um variante do algoritmo ICP (*Iterative Closest Point*) para registrar *scans* 3D em um sistema de coordenadas comum e relocalizar o robô. O segundo e terceiro módulos são responsáveis pelo desvio de obstáculos e navegação autônoma do robô. Na navegação, os *encoders* são utilizados para calcular e estimar a pose (posição e orientação) do robô. A estimativa é então usada como pose inicial para o registro de *scans*.

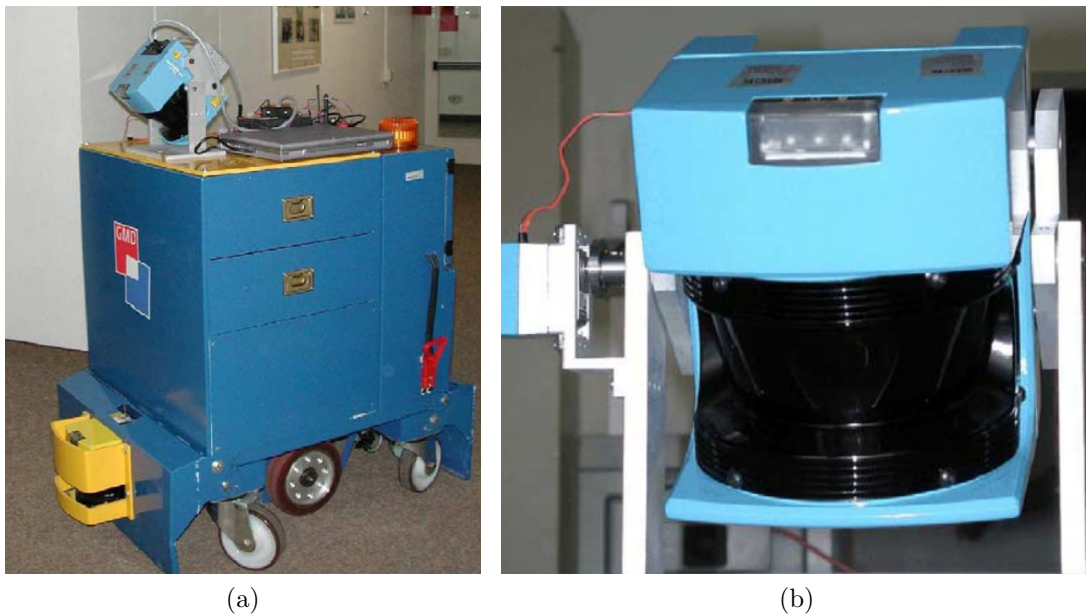


Figura 2.5: (a) Plataforma robótica Ariadne equipada com scanner laser 3D; (b) laser scanner 3D AIS (SURMANN *et al.*, 2003).

A combinação de câmera e unidade de medição inercial (IMU) para localização de dispositivos móveis é apresentada na implementação de Hesch *et al.* (2014). O sistema de odometria visual-inercial (*Visionaided Inertial Navigation System* - VINS) utiliza um filtro de Kalman estendido para fundir as medições das câmeras e IMU. Os resultados mostram a influência positiva do uso do IMU para corrigir a trajetória calculada pelo método durante movimentos curvos.

Na Figura 2.6 pode ser analisado um comparativo de resultados com IMU e sem IMU. Observa-se que a abordagem sem IMU apresenta resultado impraticáveis como o atravessamento de estruturas e paredes, enquanto a abordagem que combina câmera e IMU apresenta resultados de trajetória dentro dos limites possíveis.

Endres *et al.* (2012) apresentam um sistema para a solução do problema de SLAM utilizando câmera RGB-D, como o Microsoft Kinect. A avaliação é feita observando acurácia, robustez e tempo de processamento para três diferentes descritores de características, incluindo o SURF (*Speeded Up Robust Features*), proposto por Bay *et al.* (2008).

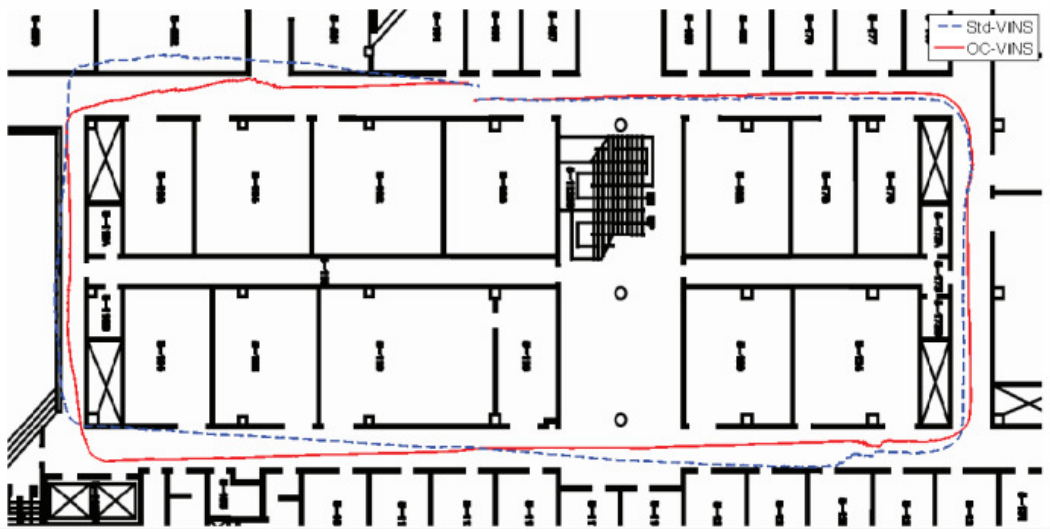


Figura 2.6: Diferença de performance entre trajetória com (vermelho) e sem (azul) o uso de IMU (HESCH *et al.*, 2014).

A abordagem dos autores do sistema pode ser dividida em quatro passos: extração de características visuais (do inglês *visual features extraction*) a partir de imagens coloridas; correspondência (do inglês *matching*) destas características com características de imagens anteriores; avaliação da profundidade da imagem na posição das características detectadas; e estimativa de transformação relativa entre os *frames* usando RANSAC (*Random Sample Consensus*), proposto por Fischler e Bolles (1987). A média de acurácia no cálculo da trajetória 2D foi de 9,7cm e 3,95°. Na Figura 2.7 é possível verificar a trajetória calculada em comparação com o *ground truth* em *dataset* (conjunto de dados) gravados. Os autores ainda criaram um pacote no ROS, disponibilizando os códigos e resultados completos para reprodução e verificação científica.

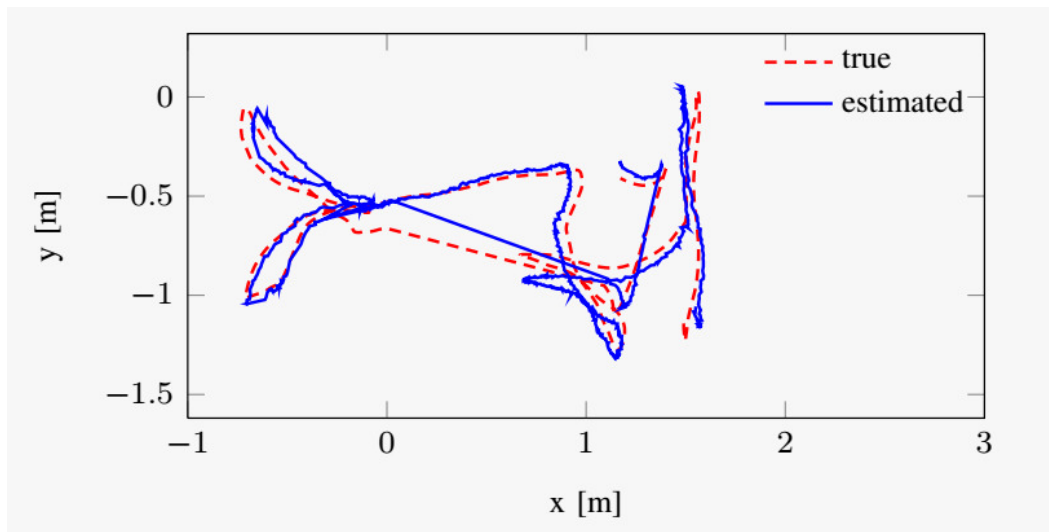


Figura 2.7: *Ground truth* (vermelho pontilhado) e trajetória 2D estimada (azul) (ENDRES *et al.*, 2012).

Quando não há definição de dados de referência para comparação de resultados, se faz necessário um método confiável para geração do *ground truth*. Neste sentido, a utilização de sistemas de câmeras Vicon para captura de movimentos e validação de cálculos de odometria tem sido registrada na literatura. O trabalho de Komsuoglu *et al.* (2014) chama atenção pelo fato de utilizar o sistema para definir o comportamento dinâmico de um robô hexápode (com seis pernas), configuração original do EspeleoRobô.

A seção a seguir apresenta os sensores utilizados nas técnicas de odometria, bem como câmeras e sistema de captura de movimentos empregados para validar os resultados gerados pelo presente trabalho, de forma a facilitar o entendimento das implementações realizadas.

2.1 Sensores e sistema de validação empregados

Cada abordagem proposta por este trabalho utiliza diferentes sensores para calcular o deslocamento do robô. Nesta seção são apresentados os tipos dos sensores, seu funcionamento e mais especificamente o modelo utilizado pelas abordagens propostas.

2.1.1 *Encoder*

Encoders são dispositivos eletromecânicos responsáveis por medir movimentos rotacionais gerando um impulso elétrico proporcional ao movimento, que representa a posição do objeto em questão. A partir desta característica o sensor pode auxiliar na medição de rotações, ângulos, distâncias, e velocidades, etc. As aplicações do dispositivo na indústria podem variar de um simples controle de velocidade de uma esteira, ao controle de posição de ferramenta de um braço robótico.

Geralmente, são encontrados dois tipos de *encoder*: absoluto e incremental. O absoluto indica a posição real do objeto, enquanto o incremental indica a posição do objeto em relação ao ponto no qual o dispositivo foi energizado. Outra classificação comum é feita em relação à construção ou arquitetura do sensor, sendo ópticos, mecânicos ou magnéticos.

O funcionamento do *encoder* óptico é basicamente composto por um disco acoplado ao eixo no qual haverá a rotação que se deseja medir, um emissor de luz e um receptor. O disco, posicionado entre o emissor e receptor, possui orifícios ou seções translúcidas, através dos quais é permitida a passagem de luz. Durante a rotação, o disco se movimenta e a luz chega até o receptor apenas quando passa por essas regiões do disco, e cada vez que o receptor detecta a luz, ocorre um pulso. Dessa forma, é possível contar quantos furos foram medidos (pulsos) e consequentemente a rotação exercida no eixo. Dependendo da precisão requerida no processo, a resolução da medida varia conforme a quantidade de furos (geralmente entre 16 e 512). Quanto maior o número de furos, maior a resolução e a sensibilidade da medição.

O *encoder* mecânico, por sua vez, utiliza contatos mecânicos para computar o movimento rotacional. Com funcionamento similar à um potenciômetro, este tipo de *encoder* varia a resistência em sua saída de acordo com o movimento exercido.

O funcionamento do *encoder* magnético baseia-se no princípio físico conhecido como efeito *hall*. Segundo Rezende Neto *et al.* (2010), o efeito Hall é gerado pela diferença de potencial através de um condutor elétrico. Essa tensão é transversa à corrente no condutor, sendo perpendicular ao campo magnético. Dentro de um mesmo material, por exemplo um fio metálico que esteja conduzindo corrente e sendo afetado por um campo magnético, surgiria uma diferença de tensão entre dois locais, pertencente a um plano perpendicular à corrente e ao campo magnético incidente ao fio, dessa mesma barra. O Efeito Hall foi descoberto por Edwin H. Hall no final do século XIX.

Segundo Honeywell (2017), na Figura 2.8 é ilustrado o princípio básico do efeito hall. Nessa imagem é mostrada uma fina camada de um material semiconductor (elemento hall) através do qual a corrente passa. As saídas das conexões são perpendiculares à direção da corrente. Quando o campo magnético é presente, a distribuição da corrente é uniforme e não há, portanto, diferença de potencial na saída do circuito.

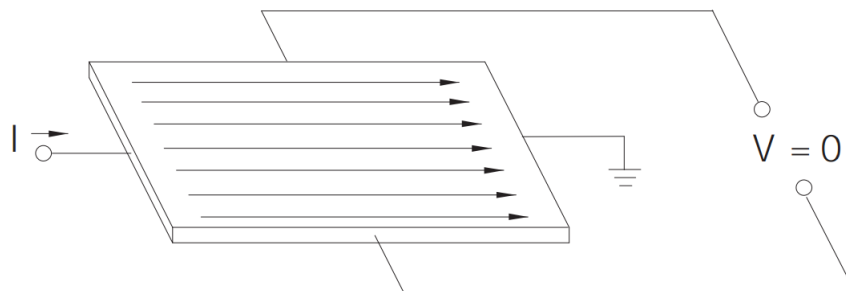


Figura 2.8: Princípio do efeito *hall*, sem presença de campo magnético (HONEYWELL, 2017).

Já quando há campo magnético perpendicular, como mostrado na figura 2.9, a força de Lorentz² é exercida na corrente. Esta força causa distúrbios na distribuição da corrente, resultando em uma diferença de potencial na saída. Esta tensão é a tensão de hall (V_H). A interação do campo magnético (B) com a corrente (I) é mostrada em forma matemática na equação 2.1.

$$V_H \propto I \times B \quad (2.1)$$

Por meio do princípio supracitado, o *encoder* magnético pode utilizar ímãs e semicondutores sensíveis a campos magnéticos (*switch* de efeito *hall*) para medir o movimento rotacional. Predko e Mc Comb (2011) apresentam duas variantes de montagem: uma

²Força exercida sob uma partícula carregada eletricamente devido à existência de um campo eletromagnético

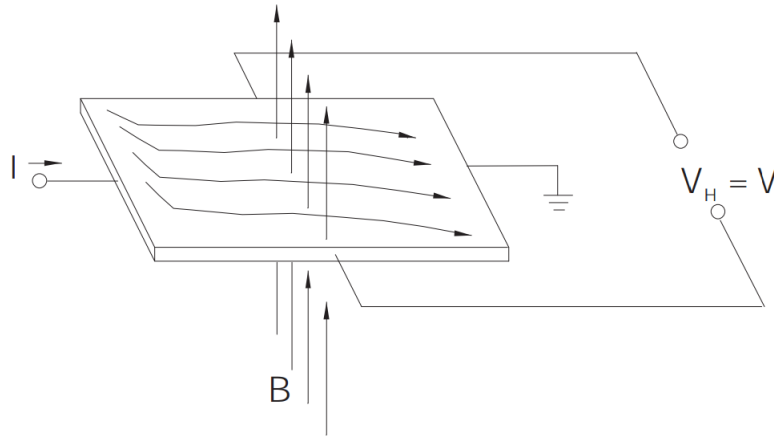


Figura 2.9: Princípio do efeito *hall*, com presença de campo magnético (HONEYWELL, 2017).

com um ou mais ímãs acoplados no eixo rotacional que são detectados quando passam em frente ao *switch*; e outra com um ímã fixo por trás do *switch* que detecta dentes de uma engrenagem de material ferroso quando passam em frente ao sensor, como é mostrado na Figura 2.10.

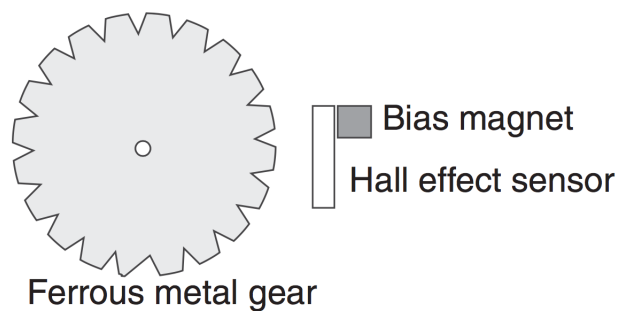


Figura 2.10: *Encoder* magnético com ímã fixo, *switch* de efeito hall e eixo com engrenagem (PREDKO e MC COMB, 2011).

Nesse trabalho, o EspeleoRobô conta com *encoders* digitais magnético-resistivos acoplados a cada um dos seus seis motores. Esse tipo de *encoder* mescla características do *encoder* mecânico com o magnético. Segundo HONEYWELL (2011), esse tipo de construção traz benefícios em relação à construção puramente magnética, pois apresenta maior sensibilidade e o dobro de pulsos por revolução, que é devido à independência da variação da resistência em relação à polaridade do campo magnético.

2.1.2 LiDAR

O dispositivo LiDAR, do inglês *Light Detection And Ranging*, é um tipo de sensor *laser* que utiliza feixes de luz refletidos para medir distâncias. O funcionamento baseia-se no

princípio chamado *time-of-flight*, amplamente difundido e aplicado na construção desses dispositivos, como apresentado no estudo realizado por Blais (2004). Esse princípio, basicamente, utiliza o tempo de voo de um feixe de luz para determinar a distância entre um objeto e o sensor. Para isso, o dispositivo conta com um emissor para emitir o feixe, e um receptor que detecta o retorno do feixe após colisão com o objeto em questão. No momento da emissão, um relógio é acionado, e quando detectado o retorno, a contagem de tempo é pausada, indicando o tempo de voo (Figura 2.11). De forma simplificada, a distância (d) pode ser descrita em função da velocidade da luz (c) e da variação de tempo entre emissão e detecção de retorno (Δt) (Equação 2.2).

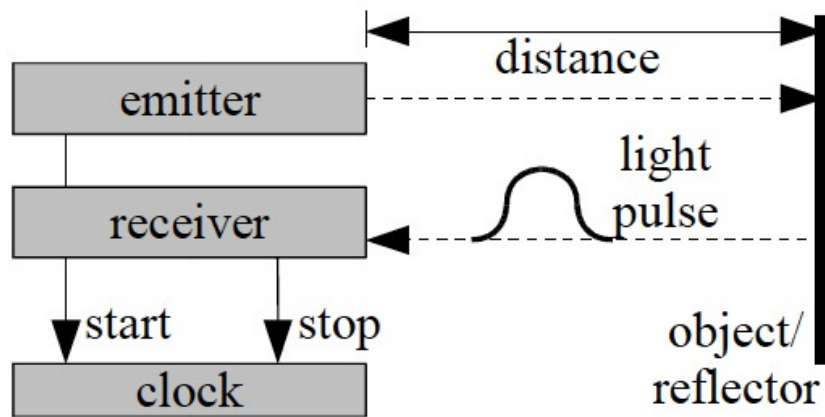


Figura 2.11: Esquema de funcionamento simplificado da medição de distância por meio do princípio *time-of-flight* (WALLHOFF *et al.*, 2007)

$$d = \frac{c\Delta t}{2} \quad (2.2)$$

As funcionalidades apresentadas pelo sensor podem ser aplicadas na agricultura, geologia, hidrologia, arqueologia, construção civil e modelagem de superfícies e objetos em geral.

Na robótica, esse sensor tem sido largamente utilizado para a solução do problema de SLAM, tanto para a parte do mapeamento quanto da localização, auxiliando de forma significativa na construção de robô móveis e carros autônomos. Na Figura 2.12 pode-se observar três sensores LiDAR Hokuyo modelo UTM-30LX no protótipo BRAiVE, citado por Broggi *et al.* (2013) como base para o desenvolvimento de quatro carros autônomos, com os quais realizaram um teste extensivo de desempenho relativa à autonomia de navegação em uma viagem da Itália à China.

Nas implementações realizadas pelo presente trabalho, o modelo de sensor LiDAR utilizado também foi o Hokuyo UTM-30LX (Figura 2.13a). Com um ângulo de abrangência de 270°, resolução de 0.25° e alcance máximo de 30m, o sensor precisa de apenas 25ms para realizar uma varredura, retornando então 1080 medições de distâncias (Figura 2.13b).



Figura 2.12: Posicionamento dos sensores Hokuyo UTM-30LX acoplados no protótipo BRAiVE e faixa de detecção (em verde) (BROGGI *et al.*, 2013).

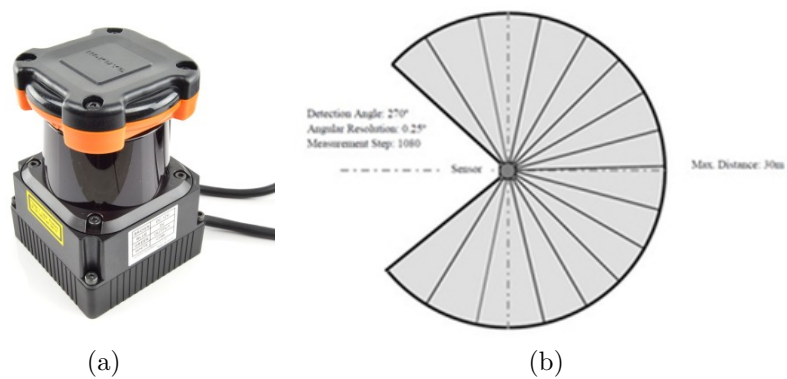


Figura 2.13: (a) Sensor LiDAR modelo Hokuyo UTM-30LX; (b) Área de varredura (STUFF, 2018).

2.1.3 Câmera RGB-D

A representação de cores no meio digital pode ser dada de várias maneiras. Uma forma bastante utilizada é a RGB (*Red, Green, Blue*) que associa componentes de cor vermelha, verde e azul para representar inúmeras cores. Uma câmera RGB-D fornece dados de cor RGB e profundidade que podem também ser utilizados para mapeamento e localização de robôs móveis.

Nos desenvolvimentos realizados foi utilizado o primeiro modelo da Kinect da Microsoft, que tem como objetivo primário a interação de usuários com o videogame XBOX através de movimentos, mas também tem sido largamente utilizado em aplicações acadêmicas e profissionais no ramo da robótica. Segundo MacCormick (2011), este sensor usa luz estruturada e aprendizagem de máquina (do inglês, *machine learning*) para identificar objetos e movimentos (Figura 2.14). O princípio geral da técnica de luz estruturada consiste em projetar um padrão conhecido em um cenário e inferir a profundidade por meio da deformação sofrida pelo padrão Zhang *et al.* (2002). As técnicas de aprendizagem de máquina são utilizadas para a identificação de partes do corpo humano. Segundo Shotton *et al.* (2011), mais de 1 milhão de exemplos de treinamento foram realizados para o reconhecimento de gestos humanos. Da mesma forma, essas técnicas podem ser utilizadas para a identificação de objetos em geral, como mostrado na Figura 2.15.

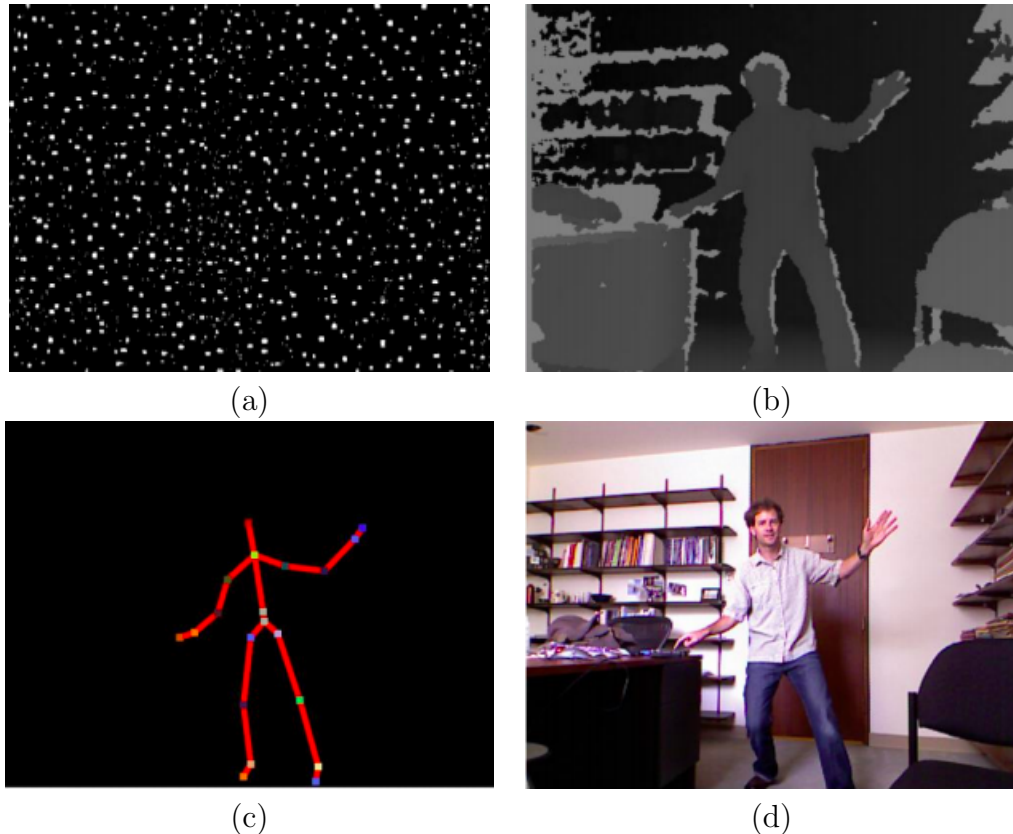


Figura 2.14: (a) Padrão emitido; (b) Imagem de profundidade; (c) *Skeleton* formado; (d) Imagem RGB (MACCORMICK, 2011).

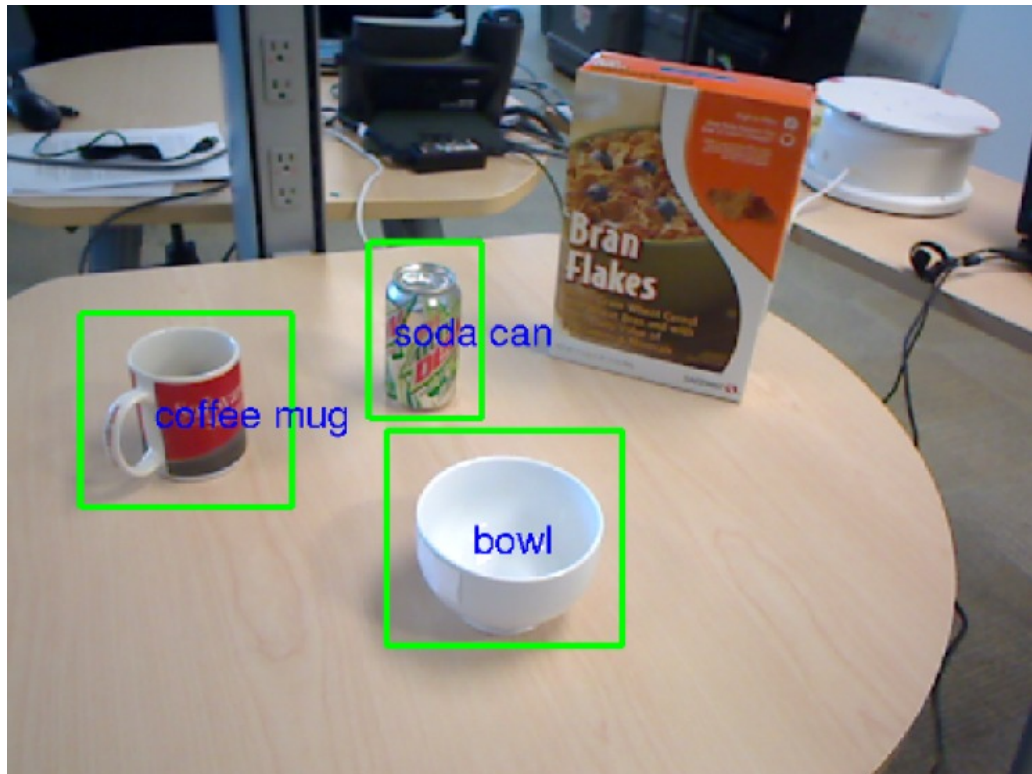


Figura 2.15: Identificação de objetos com câmera RGB-D após processo de aprendizagem (LAI *et al.*, 2011).

O sensor é dotado de emissor e receptor de luz infravermelha que são responsáveis pela emissão do padrão, que é então utilizado para a construção de um mapa de profundidade. A disposição do conjunto de sensores integrados no dispositivo pode ser observada na Figura 2.16.

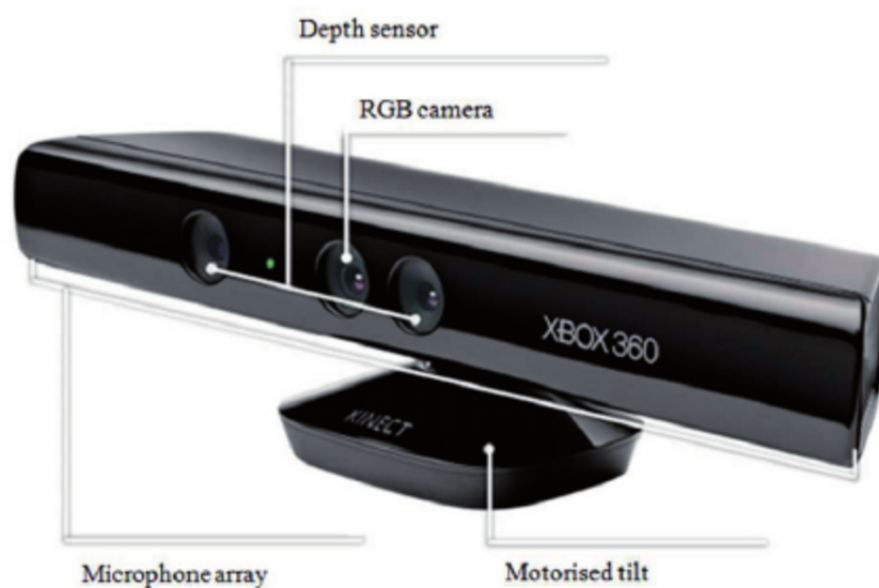


Figura 2.16: Câmera RGB-D modelo Kinect da Microsoft.

2.1.4 Unidade de medição inercial (IMU)

Unidades de medição inercial, também conhecidas como IMU (*Inertial Measurement Unit*), são dispositivos que utilizam acelerômetros e giroscópios concomitantemente para medir acelerações e inclinações em aplicações dinâmicas. Modelos mais completos podem utilizar também magnetômetros, barômetros e antena GPS. Geralmente, são empregados algoritmos para a realização de fusão de sensores, combinando as leituras feitas e filtrando-as para uma melhor qualidade de medição.

Nesse trabalho foi utilizado o modelo MTi-G-710 da Xsens (Figura 2.17) que conta com 3 acelerômetros, 3 giroscópios, 3 magnetômetros, 1 barômetro e 1 antena GPS tipo GNSS (*Global Navigation Satellite System*). Esses sensores, respectivamente, são capazes de medir a aceleração dos movimentos computados, os ângulos de giro desempenhados, o referencial em relação ao polo magnético da Terra, a pressão relativa à altitude e a posição em relação ao Sistema de Posicionamento Global (GPS).

O próprio fabricante oferece vários tipos de filtro para fusão das medições dos sensores, utilizando métodos estatísticos para definir qual medição deve influenciar mais, de acordo com determinado ambiente ou aplicação. Os filtros melhoram os resultados de medição fornecidos pelo dispositivo, atenuando ruídos e aumentando a confiabilidade dos dados gerados.



Figura 2.17: Sensor IMU modelo MTi-G-710 da Xsens.

2.1.5 Sistema de captura de movimento com câmeras Vicon

A captura de movimento é objeto de estudo na área da robótica, visão computacional e computação gráfica, visando verificar movimentos realizados por robôs e outros equipamentos, durante operações que requeiram precisão. Além disso, a medição precisa de movimentos tem contribuído também na ciência, biologia e medicina, com estudos de locomoção de animais e humanos, e até mesmo em entretenimento, tornando possíveis efeitos especiais cada vez mais realísticos nos cinemas.

Com intuito de validar os resultados de odometria alcançados com as técnicas implementadas, um sistema de captura de movimentos confiável para gerar o *ground truth* se fez necessário.

Nos desenvolvimentos realizados por esta dissertação foi utilizado um sistema composto por oito câmeras Vicon, duas modelo Vero e seis modelo Bonita. Essas câmeras utilizam taxas de até 330 FPS (*frames per second*), identificam reflexos luminosos em esferas refletivas, denominadas *markers*. Por meio de triangulação, elas geram dados de posição e orientação de objetos formados por essas esferas, com acurácia de 1mm. Na Figura 2.18 é possível verificar os modelos utilizados e na Figura 2.19 pode-se observar um exemplo de utilização das esferas refletivas para identificar uma mão humana, separando-a em quatro diferentes estruturas representadas com cores diferentes (objetos).



(a)



(b)

Figura 2.18: Câmeras Vicon modelos Bonita (a) e Vero (b).

Geralmente, as câmeras são dispostas em um ambiente, formando um volume de rastreamento, dentro do qual é possível detectar as esferas (Figura 2.20). A integração é feita através de um *switch*, que liga as câmeras a um computador, responsável por processar os dados lidos por meio de um *software* próprio, chamado Tracker (Versão 3.3).

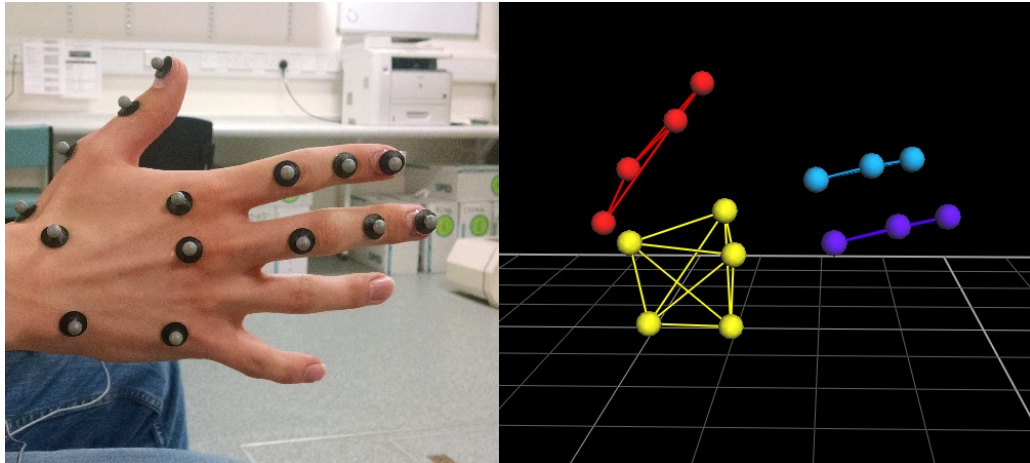


Figura 2.19: Esferas refletivas posicionadas sobre mão humana formando objeto no *software* Tracker (BLANA, 2018).

Nesse *software*, é possível selecionar um conjunto de esferas refletivas, defini-las como um objeto a ser rastreado e configurar um sistema de coordenadas próprio com respeito ao objeto. Dessa forma, é possível rastrear o movimento do objeto definido, obtendo valores de posição e orientação durante os testes realizados.

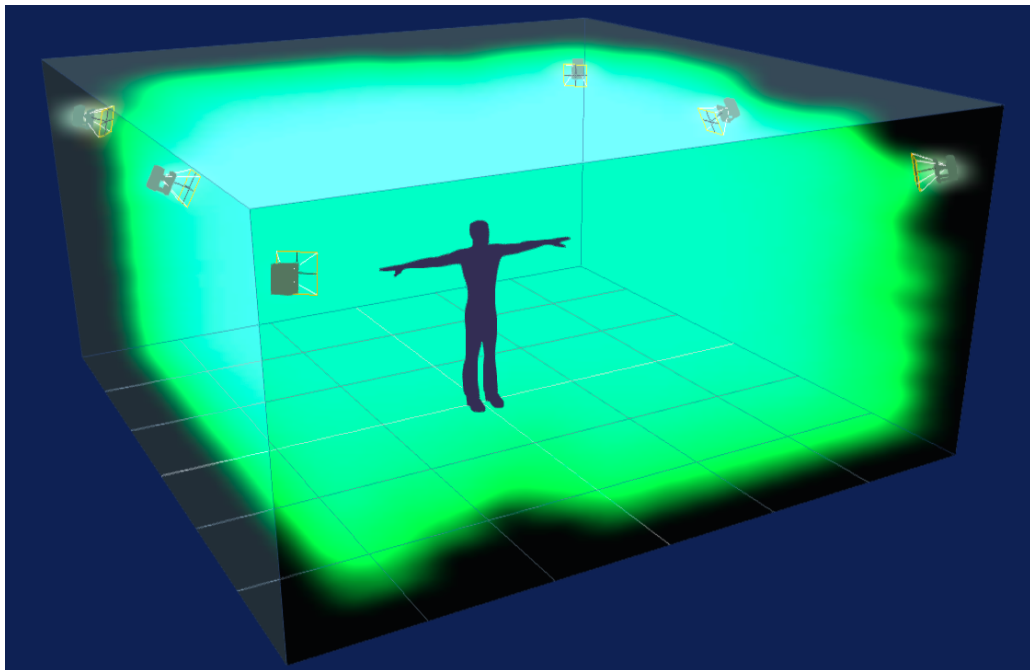


Figura 2.20: Exemplo de volume de rastreamento formado com seis câmeras Vicon.

Neste capítulo foram apresentados trabalhos relacionados, bem como os sensores utilizados nesta dissertação. No capítulo a seguir serão descritas as técnicas implementadas utilizando esses sensores para realizar o cálculo de odometria do EspeleoRobô.

Capítulo 3

Estudos e implementações

Este capítulo descreve o dispositivo robótico e sensores utilizados nos desenvolvimentos, os modelos cinemáticos adotados para testes com *encoders* e os algoritmos e métodos usados para o cálculo de odometria.

3.1 EspeleoRobô e instrumentação embarcada

Os desenvolvimentos foram realizados com o EspeleoRobô configurado com modo de locomoção de seis rodas. A locomoção é realizada por meio do acionamento individual de seis motores Maxon MCD EPOS de 60W, que possuem acoplados os *encoders* digitais magneto-resistivos descritos na Seção 2.1. Com esses sensores é possível medir a velocidade rotacional real exercida em cada atuador. Os valores medidos servem de entrada para o cálculo realizado pela implementação descrita na Subseção 3.2.1.

Na Figura 3.1 é mostrado o EspeleoRobô equipado com os dispositivos necessários ao desenvolvimentos propostos. Uma impressora 3D e o material ABS foram utilizados para imprimir um suporte central e um suporte em formato de trapézio para compor, junto dos sensores, uma estrutura de instrumentação.

O suporte central foi fixado no centro do dispositivo e sobre ele foram posicionados um sensor LiDAR (Hokuyo UTM-30LX) e uma câmera RGB-D (Primeiro modelo do Microsoft Kinect). Foi necessário elevar a altura desses sensores para que as rodas do robô não interferissem nas medições realizadas. A câmera RGB-D (Kinect) foi posicionada sobre o LiDAR (Hokuyo) para que as medições de ambos sensores fossem alinhadas com o centro do robô. Esse alinhamento foi realizado para que fosse possível iniciar os testes de um mesmo ponto em comum, sem a necessidade de cálculos para compensação. Essa medida, porém, só foi possível e viável visto que os sensores são facilmente sobrepostos.

Sobre o Kinect foi fixado o suporte trapezoidal para posicionar as esferas refletivas, necessárias para que o sistema de captura de movimento com câmeras Vicon seja capaz de rastrear o robô e computar seu deslocamento durante os testes. Maiores detalhes sobre esse sistema serão apresentados na Seção 4.1.

Por fim, a unidade de medição inercial (IMU) da Xsens modelo MTi-G-710 foi fixada à direita do suporte central, com a mesma orientação dos sensores citados e do robô. Esse dispositivo não precisou ser posicionado sob o centro do robô, pois foi apenas utilizado para gerar dados de orientação.

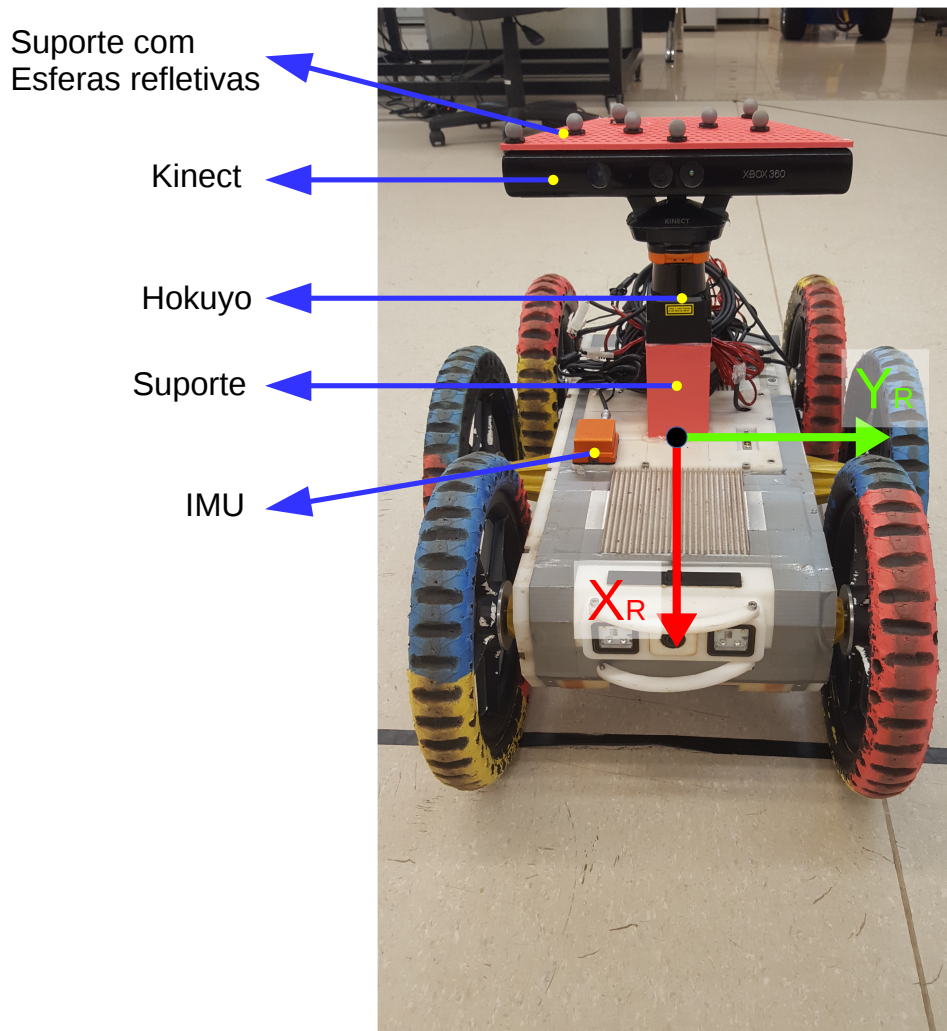


Figura 3.1: EspeleoRobô, sensores e suportes utilizados nos desenvolvimentos.

3.2 Técnicas de odometria

Nesta seção são apresentadas as técnicas de odometria implementadas. É importante destacar que no restante do trabalho serão mencionadas as abordagens referindo-se apenas ao nome do sensor, porém as mesmas não se limitam a cada sensor, mas são compostas, cada uma, de algoritmos que utilizam as leituras de sensores para gerar dados de odometria. Logo, como exemplo, a “odometria com LiDAR” ou “odometria com sensor *laser*” ou ainda “odometria com Hokuyo” é uma técnica ou abordagem que utiliza as medições desse sensor para realizar os cálculos odométricos.

3.2.1 Odometria com *encoders*

Na implementação de odometria com *encoders* foram utilizadas as informações de giro das rodas do meio do robô, lidas por estes sensores como entradas em modelos cinemáticos para se calcular o deslocamento do robô.

Visando a simplificação dos cálculos, uma primeira abordagem foi realizada considerando o dispositivo como tendo um sistema de direção diferencial. Após a primeira abordagem com modelo diferencial, foi estudada uma abordagem utilizando o modelo cinemático do *skid-steer*, com o qual o dispositivo utilizado nesta pesquisa pode ser melhor representado.

Para introduzir os modelos utilizados, será explicado a seguir o modelo cinemático do unicycle, comumente adotado para simplificar modelagens e comandos de robôs móveis terrestres.

3.2.1.1 Modelo do unicycle

O modelo do unicycle, ilustrado pela Figura 3.2, é geralmente utilizado como simplificação para representar a dinâmica de um robô móvel com rodas. O modelo é basicamente uma roda que possui uma velocidade linear v e uma velocidade angular ω . Para representar o modelo são assumidas as seguintes simplificações:

- o robô se move em uma superfície plana;
- não há deslizamento;
- não há deformação nas rodas;
- os eixos de orientação são perpendiculares ao solo.

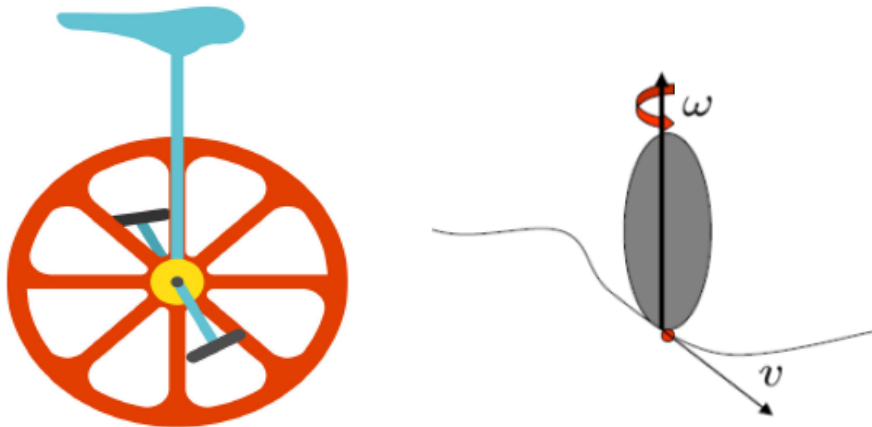


Figura 3.2: Modelo do unicycle.

Considerando o modelo planar (plano x - y), tem-se que a pose do robô é dada por:

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \quad (3.1)$$

onde x e y são as componentes da posição nos eixos ortogonais (x , y) e θ é o ângulo de orientação em torno do eixo z .

A Equação 3.2 apresenta a relação da posição (x , y) e orientação (θ) do unicycle em função da sua velocidade linear v e angular ω :

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v \cos(\theta) \\ v \sin(\theta) \\ \omega \end{bmatrix} \quad (3.2)$$

Em geral, a cinemática de um robô móvel pode ser expressada na forma de modelo cinemático, onde $\dot{q}$ são as variáveis de estados, $\dot{\theta}$ são as variáveis de atuação e $J(\theta)$ é a matriz Jacobiana:

$$\dot{q} = J(\theta)\dot{\theta} \quad (3.3)$$

Relacionando as Equações 3.2 e 3.3, tem-se que:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (3.4)$$

3.2.1.2 Modelo diferencial

O modelo diferencial consiste basicamente em duas rodas paralelas de raio r separadas por uma distância L (Figura 3.3). O acionamento das rodas é feito de forma separada, o que possibilita a realização de curvas e giro em torno do próprio eixo se: uma roda for acelerada mais que outra; ou as acelerações aplicadas forem iguais e com sentido inverso; respectivamente.

No modelo diferencial tem-se a seguinte relação:

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{L} & -\frac{1}{L} \end{bmatrix} \begin{bmatrix} r\omega_d \\ r\omega_e \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{L} & -\frac{1}{L} \end{bmatrix} \begin{bmatrix} v_d \\ v_e \end{bmatrix} \quad (3.5)$$

onde r é o raio da roda, L é a distância entre as rodas (comprimento do eixo), ω_d e ω_e são as velocidades angulares na rodas direita e esquerda, respectivamente. As velocidades lineares das rodas direita e esquerda são representadas pelas variáveis v_d e v_e , respectivamente.

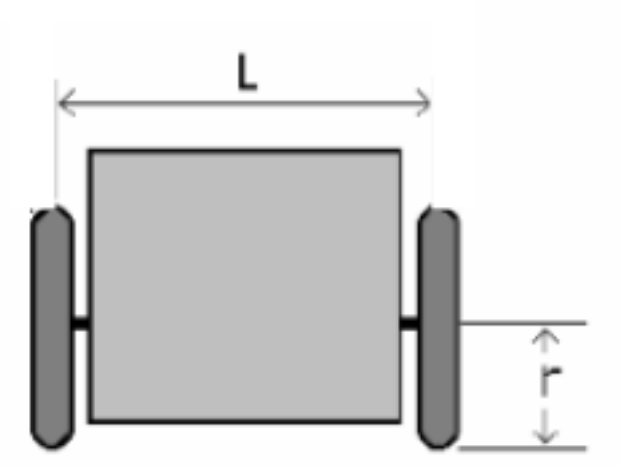


Figura 3.3: Modelo diferencial.

Logo, observando as Equações 3.4 e 3.5 temos que as velocidades lineares $\dot{x}$ e $\dot{y}$, e a velocidade angular do robô $\dot{\theta}$ podem ser expressadas da seguinte forma:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ \frac{1}{L} & -\frac{1}{L} \end{bmatrix} \begin{bmatrix} v_d \\ v_e \end{bmatrix} \quad (3.6)$$

Realizando as devidas multiplicações, tem-se que:

$$\dot{x} = \frac{(v_d + v_e)}{2} \cos(\theta) \quad (3.7)$$

$$\dot{y} = \frac{(v_d + v_e)}{2} \sin(\theta) \quad (3.8)$$

$$\dot{\theta} = \frac{(v_d - v_e)}{L} \quad (3.9)$$

Adquirindo os valores de ω_d e ω_e através de *encoders* acoplados aos motores e utilizando a integração numérica de 1^a pelo método de Euler (SICILIANO *et al.*, 2010) nas equações 3.7, 3.8 e 3.9, é possível obter os valores de posição e orientação ao longo do tempo (Equações 3.10, 3.11 e 3.12).

$$\Delta x = \frac{(v_d + v_e)}{2} \cos(\theta) \Delta t \quad (3.10)$$

$$\Delta y = \frac{(v_d + v_e)}{2} \sin(\theta) \Delta t \quad (3.11)$$

$$\Delta \theta = \frac{(v_d - v_e)}{L} \Delta t \quad (3.12)$$

3.2.1.3 Modelo *skid steer*

O modelo *skid steer* é utilizado para representar veículos com pares de rodas ou esteiras (Figura 3.4). De forma semelhante ao modelo diferencial, as curvas e rotações em torno do próprio eixo são executadas variando-se as acelerações aplicadas do lado direito e esquerdo.

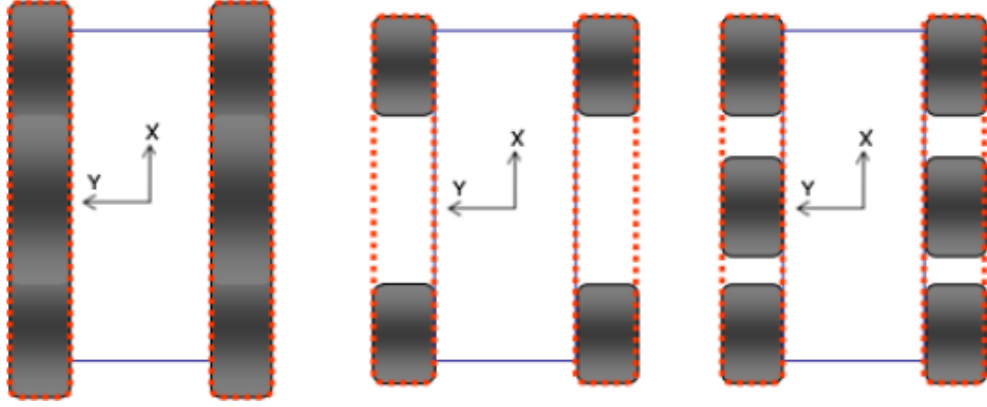


Figura 3.4: Modelo *skid steer*.

Observando os desenvolvimentos de (MANDOW *et al.*, 2007) e (MARTÍNEZ *et al.*, 2005), a cinemática de um robô *skid steer* pode ser descrita por:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = R_z(\theta) A \begin{bmatrix} v_e \\ v_d \end{bmatrix} \quad (3.13)$$

onde $R_z(\theta) \in$ grupo especial ortogonal $SO(3)$ é a matriz de rotação elementar no eixo z (Equação 3.14), $A \in \mathbb{R}^{3 \times 2}$ é a matriz de modelagem cinemática do dispositivo *skid-steer* (Equação 3.15) e v_e e v_d representam as velocidades lineares nos flancos esquerdo e direito, respectivamente.

$$R_z(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.14)$$

$$A = \frac{\alpha}{2y_{CIR}} \begin{bmatrix} -y_{CIR} & -y_{CIR} \\ 0 & 0 \\ -1 & 1 \end{bmatrix} \quad (3.15)$$

Uma maneira prática de se encontrar a matriz A é apresentada por Mandow *et al.* (2007). Estes propõem experimentos que utilizam os centros instantâneos de rotação (CIR, Figura 3.5) para encontrar fatores de correção devido à mecânica do robô e ao tipo de terreno afim de calcular a cinemática do dispositivo. Por meio da utilização destes fatores, os erros causados por escorregamentos devem ser menores.

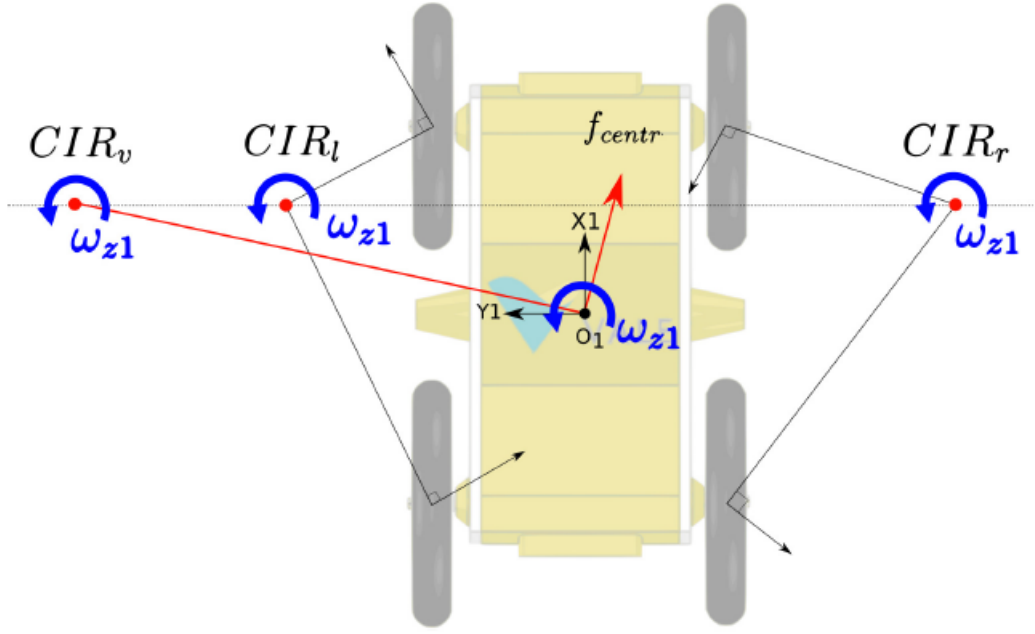


Figura 3.5: Localização dos CIR's dos flancos e do veículo (ROCHA, 2018).

Na abordagem dos autores são encontrados dois valores empíricos (α e y_{CIR}) para ajustar o modelo cinemático ao dispositivo real utilizado. Experimentos foram desenvolvidos em laboratório para a obtenção dos valores do fator de correção α e o valor de y_{CIR} que são calculados pelas Equações 3.16 e 3.17:

$$\alpha = \frac{2d}{\int v_d(t)dt + \int v_e(t)dt} \quad (3.16)$$

$$y_{CIR} = \frac{\int v_d(t)dt - \int v_e(t)dt}{2\theta^*} \quad (3.17)$$

onde d é uma distância percorrida pelo robô em movimento de linha reta; θ^* é o ângulo obtido após uma sequência de giros em torno do próprio eixo; e $v_d(t)dt$ e $v_e(t)$ são as velocidades, do lado direito e esquerdo, respectivamente, exercidas durante os movimentos descritos.

Dessa forma, dois tipos de experimentos foram conduzidos para determinar os parâmetros. Para encontrar α foram conduzidos testes de locomoção em linha reta, aplicando velocidade de módulos e sentidos iguais. Enquanto para y_{CIR} foram executados testes de rotação pura (em torno do próprio eixo), aplicando velocidades de módulos iguais, porém com sentidos diferentes.

Cada experimento foi realizado cinco vezes, obtendo-se os valores finais procurados por meio de médias simples. Os valores de d e θ^* foram obtidos utilizando o sistema de captura de movimento com câmera Vicon (Seção 4.1), enquanto os valores de velocidades desenvolvidas foram coletados com a leitura dos *encoders*.

Ao final dos experimentos foram obtidos os valores de 0,9838 para α e 0,3045 para y_{CIR} , para as condições do laboratório. Os valores mostram-se coerentes, se comparados ao valores encontrados por Rocha (2018) em testes simulados: 0,965 para α e 0,400 para y_{CIR} . Após determinar os valores para os parâmetros supracitados, torna-se possível o cálculo dos valores da matriz de modelo cinemático A (Equação 3.15).

Considerando a matriz de rotação $R_z(\theta)$ (Equação 3.14), a matriz A calculada e as velocidades lineares das rodas esquerda e direita (v_e e v_d), é possível somar as pequenas variações em x , y e θ para cada instante Δt (Equação 3.18) aplicando novamente a integração numérica de 1ª ordem pelo método de Euler:

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{bmatrix} = R_z(\theta) A \begin{bmatrix} v_e \\ v_d \end{bmatrix} \Delta t \quad (3.18)$$

A discretização é necessária para que seja possibilitada a implementação do cálculo de odometria no programa de processamento de dados. Entretanto, para que a odometria funcione de forma a minimizar erros, deve-se adotar o menor intervalo de tempo possível entre cálculos. Para se definir esse intervalo no caso do EspeleoRobô, deve-se ter em mente que os valores de velocidade rotacional das rodas são disponibilizados pelos *encoders* a uma frequência de 5Hz, o que significa que, idealmente, esses valores são atualizados a cada 0,2s. Apesar da definição da frequência, é possível observar na Figura 3.6 que a média de frequência varia em torno de 5Hz, apresentando também valores abaixo do definido.

```
cota ~ $ rostopic hz /espeleorobo/telemetry
subscribed to [/espeleorobo/telemetry]
average rate: 5.122
  min: 0.148s max: 0.256s std dev: 0.03966s window: 5
average rate: 4.899
  min: 0.029s max: 0.488s std dev: 0.11728s window: 10
average rate: 5.001
  min: 0.029s max: 0.488s std dev: 0.10703s window: 15
average rate: 4.952
  min: 0.029s max: 0.488s std dev: 0.10950s window: 20
average rate: 4.901
```

Figura 3.6: Média de frequência (“average rate”) de atualização dos valores de velocidade rotacional exercida nos atuadores.

Dessa forma, foi implementado o cálculo com intervalo de 0,2s, porém alguns testes de odometria *online* apresentaram problemas com essa frequência, como travamento. Considerando o bom funcionamento do programa que gera a odometria *online*, foi estabelecido então um intervalo de 0,25s para o cálculo odométrico. Tendo em vista que o robô se move relativamente a baixas velocidades tanto nos testes de laboratório, quanto em futuras operações, esse intervalo foi considerado aceitável. De toda forma, para garantir que o intervalo ligeiramente maior não prejudicasse o desempenho do método em relação aos outros, cálculos *offline* com o intervalo limítrofe de 0,2s também foram realizados; entretanto, não apresentaram melhoria significativa.

Na Seção 4.3 os modelos cinemáticos diferencial e *skid-steer* são comparados entre si, em relação à capacidade de representar matematicamente os movimentos reais descritos pelo EspeleoRobô. Como esperado e demonstrado pelos resultados expostos na referida seção, o *skid-steer* é o modelo que melhor representa a plataforma robótica utilizada. Na Seção 4.4 esse modelo é avaliado em conjunto com as outras abordagens propostas pelo presente trabalho.

3.2.2 Odometria visual com câmera RGB-D

Na avaliação da odometria visual, foi utilizado um sensor RGB-D (Kinect) de luz estruturada, o qual fornece dados de profundidade e imagens RGB simultaneamente, em conjunto com o algoritmo de SLAM proposto por Labbé e Michaud (2014). Os autores propõem um método para diminuir o problema do “robô sequestrado” ou do mapeamento multi-sessão (onde um mapa maior é gerado desde diferentes fontes ou sessões) usando fechamentos de *loop* globais com restrição de memória. Essa restrição de memória permite realizar a fusão de diferentes porções do mapa ou grandes conjunto de dados de forma computacionalmente eficiente.

O método usa técnicas de SLAM baseadas em grafos para realizar o fechamento de *loop*: a estrutura do mapa é dividida em nós e arcos, e em cada nó são armazenadas as poses de odometria do mapa, e informações visuais como imagens, palavras visuais, etc. Os arcos armazenam as transformações entre um par de nós.

Para gerar as palavras visuais (*bag-of-words*), são extraídas características marcantes (*features*) das imagens RGB geradas com *SURF*. Essas características são intensidade de cor, formas, quinas, bordas, que formam aspectos únicos, possibilitando a identificação de uma mesma referência em imagens geradas em momentos diferentes (Figura 3.7).

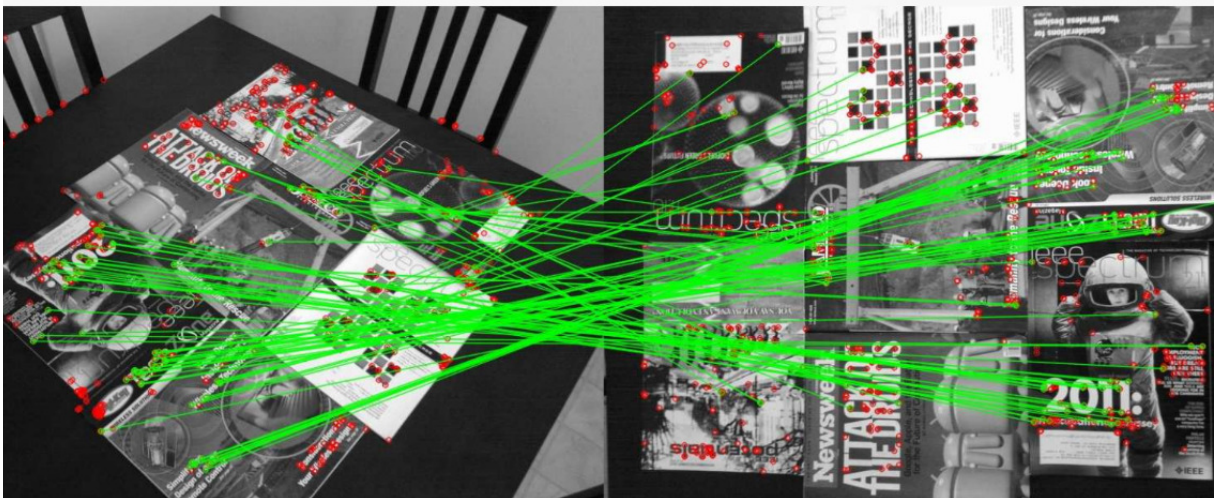


Figura 3.7: Identificação (*matching*) de características correspondentes em duas imagens diferentes (RUBLEE *et al.*, 2011).

Cada palavra visual é computada para gerar um dicionário visual incremental. Quando a hipótese de fechamento entre dois nós atinge um limite predefinido, então o fechamento de *loop* é detectado.

As restrições de memória são realizadas mediante o uso da abordagem RTAB-Map para gestão de memória (LABBE e MICHAUD, 2013) onde são usados diferentes tipos de memória de: longo prazo, curto prazo, e memória atual (Figura 3.8). O método mantém uma quantidade razoável de nós na memória atual, e esses nós são substituídos de acordo com heurísticas com a finalidade de possibilitar os cálculos de fechamento de *loop* em tempo real. O grafo é finalmente otimizado para reduzir os erros de posicionamento com o TORO (*Tree-based netwORk Optimizer*) (GRISSETTI *et al.*, 2007), usando as poses dos nós e os links como restrições.

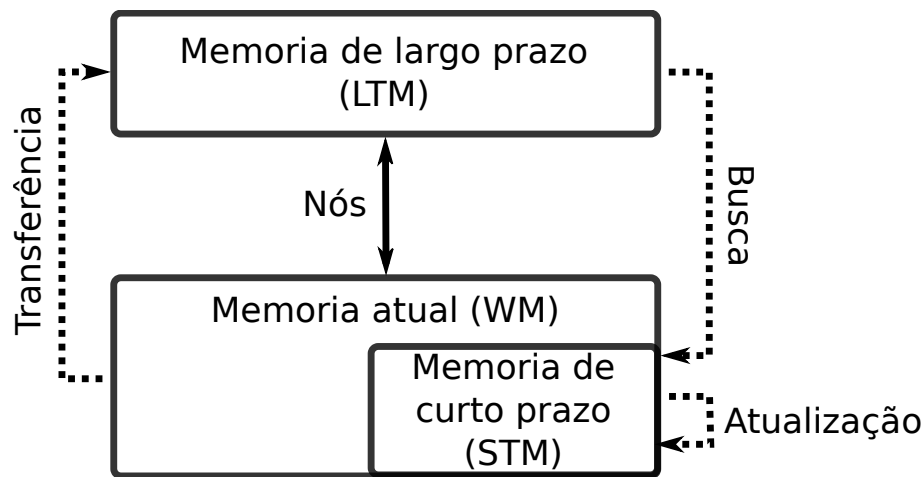


Figura 3.8: Modelo de gerenciamento de memória para garantir o processamento dos nós em tempo real (Labbé and Michaud, 2014).

Especificamente, a odometria usada neste trabalho é calculada usando a técnica F2M (*Frame-2-Map*) e não é necessário o fechamento de *loop*. Essa técnica usa uma versão reduzida e temporal do mapa gerado, onde a última imagem (*frame*) estimada do mapa é comparada com os *features* visuais e de profundidade de uma nova imagem. Finalmente é usado o método RANSAC para calcular a transformação entre as imagens. Esse método utiliza as características marcantes identificadas com SURF correspondentes nas duas imagens, calcula as transformadas entre cada uma dessas características e então usa otimização para determinar a transformação resultante entre as imagens como um todo.

Na Figura 3.9 são mostradas inúmeras correspondências de características de múltiplos *frames* sobrepostas na primeira imagem. O uso da estatística é necessário para eliminar correspondências equivocadas (*outliers*) de características marcantes que se parecem ou são idênticas, mas não são correspondentes. Na Figura 3.10 é possível observar as transformadas geradas por correspondências corretas (*inliers*) e equivocadas (*outliers*), nas imagens à esquerda e à direita, respectivamente.



Figura 3.9: Características correspondentes computadas ao longo de múltiplas imagens sobrepostas sobre a primeira imagem (FRAUNDORFER e SCARAMUZZA, 2012).

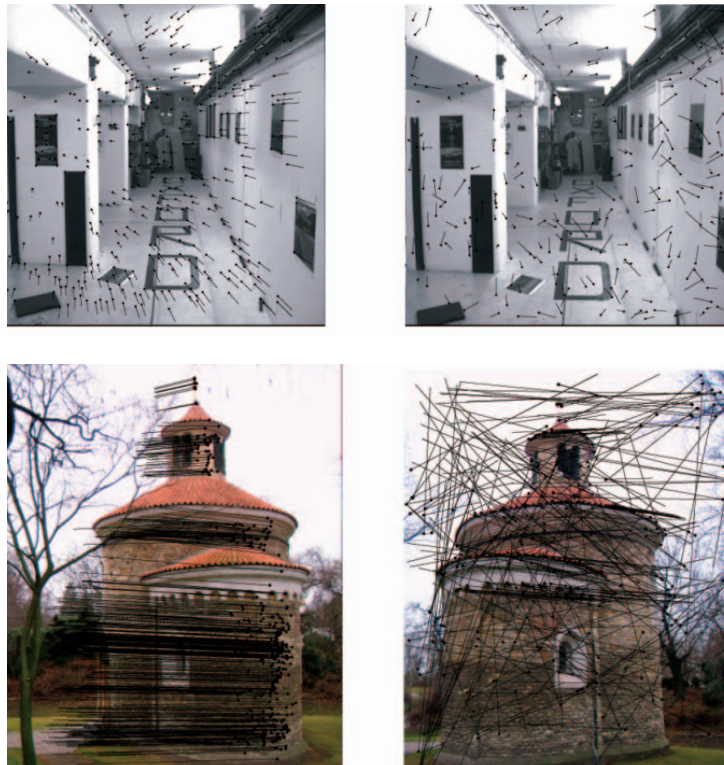


Figura 3.10: Características correspondentes corretas (esquerda) e equivocadas (direita) (CHUM e MATAS, 2008).

3.2.3 Odometria com sensor LiDAR

Como apresentado na Subseção 2.1.2, o dispositivo empregado na abordagem com sensor *laser* é do tipo LiDAR e utiliza feixes de luz refletidos para medir distâncias em uma área de varredura de até 270° e 30m de alcance. Ao manipular corretamente essas medições, é possível obter informações sobre o ambiente ou objetos localizados nele.

Em cada varredura é possível medir uma distância a cada 0,25°, logo uma varredura completa é composta por 1080 distâncias entre o sensor e pontos no ambiente. Esses pontos, quando interligados formam uma superfície e, dessa forma, cada varredura gera uma superfície que representa a forma do ambiente no momento da leitura. Para calcular então a odometria entre duas leituras, deve-se minimizar a distância entre essas superfícies, calculando a transformada entre elas (Figura 3.11).

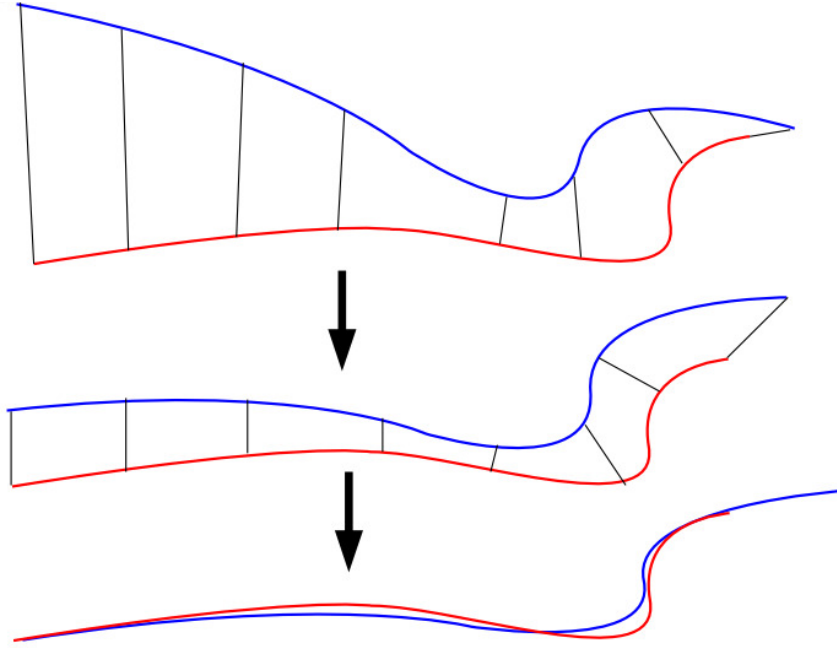


Figura 3.11: Minimização da distância entre superfícies para se calcular a transformada ocorrida entre movimentos (SMISTAD *et al.*, 2015).

Para implementar o cálculo dessas transformadas entre superfícies gerando em dados odométricos, foi avaliada a odometria com laser usando uma variante do algoritmo ICP (*Iterative Closest/Correspondent Point*) chamada PLICP, apresentada por Censi (2008). O ICP é um algoritmo que dados um conjunto de pontos $P = \{p_1, \dots, p_n\}$ e uma superfície S^{ref} estima as rotações e translações ($q = (t, \theta)$) que minimizam o erro quadrático entre esses dois conjuntos:

$$\min_{q_{k+1}} \sum_i \|p_i \oplus q_{k+1} - \Pi\{S^{ref}, p_i \oplus q_k\}\|^2 \quad (3.19)$$

onde $\oplus$ denota o operador de rotação e translação, e $\Pi\{S^{ref}, \cdot\}$ denota a projeção euclidiana em S^{ref} .

O intuito dessa técnica se baseia na teoria de que se as correspondências corretas são conhecidas entre dois conjuntos de pontos (gerados pelas distâncias medidas), então é possível calcular as transformações relativas entre eles, e assim conhecer o deslocamento (odometria).

Censi (2008) desenvolveu uma variante do algoritmo que usa uma métrica *point-to-line* ao invés da *point-to-point*, a qual é demonstrada ser mais eficiente. Segundo Li *et al.* (2016), o PLICP melhora a convergência de linear para quadrática, se comparado ao ICP clássico, precisando assim de menos iterações para convergir:

$$\min_{q_{k+1}} \sum_i (n_i^t[p_i \oplus q_{k+1} - \Pi\{S^{ref}, p_i \oplus q_k\}])^2 \quad (3.20)$$

3.2.3.1 Odometria com sensor LiDAR e unidade de medição inercial

Segundo Censi (2008), em relação a grandes deslocamentos rotacionais (a partir de aproximadamente 45°) o PLICP é menos robusto que o ICP clássico e quando há esse tipo de rotação, o método falha 25% da vezes. Apesar do autor considerar que esse movimento é incomum na robótica *indoor* (em ambientes fechados), o EspeleoRobô, quando acionando para realizar curva fechada em velocidade máxima, acaba girando em torno do próprio eixo e pode se aproximar ou ultrapassar o ângulo descrito.

Dryanovski *et al.* (2014) sugerem então que o tempo do processo de registro de escaneamentos pode ser melhorado, se provida uma estimativa do movimento rotacional do robô entre os escaneamentos. Dessa forma, em uma segunda abordagem utilizando sensor laser, o algoritmo PLICP foi parametrizado para obter a componente de rotação em z (θ) por meio da componente de orientação *yaw* gerada por um sensor IMU, fazendo com que a estimativa de rotação seja calculada mediante a diferença absoluta entre o *yaw* dos *frames* anterior e atual.

É necessário salientar que não é aplicada técnica específica de fusão de dados para adicionar a leitura da unidade de medição inercial, visto que o objetivo da utilização da IMU é aumentar robustez do método e diminuir o tempo do processo de registro. O ângulo de rotação θ deixa de ser calculado por meio das leituras do Hokuyo e passa a ser obtido exclusivamente pela componente *yaw* da IMU. Caso contrário, se fosse mantido o cálculo com os escaneamentos do sensor *laser* para posterior fusão com dados do sensor adicionado, o tempo de registro iria aumentar, ao invés de diminuir.

O modelo de IMU utilizado, previamente descrito na Subseção 2.1.4, é equipado com 3 acelerômetros, 3 giroscópios, 3 magnetômetros, barômetro, além de antena GPS, e possui filtros próprios implementados de fábrica para combinar a leitura desses sensores e gerar dados. Considerando que as implementações foram realizadas em laboratório e que as operações em campo serão feitas em locais fechados e cavernas, o filtro selecionado desconsidera informações dos magnetômetros, sinal GPS e barômetro.

A existência de equipamentos eletrônicos no laboratório e a alta concentração de ferro nas cavernas afetam a medição dos magnetômetros, que utilizam como referência o polo magnético da Terra. Além disso, em ambos os ambientes, o sinal GPS não funciona corretamente. Por fim, como demonstrado na Equação 3.2, a componente de posição em z não é considerada neste trabalho, logo a altitude medida com barômetro não tem relevância.

3.3 Implementação do cálculo de distância total percorrida

Na prática, todas as técnicas apresentadas nesse capítulo, bem como o *ground truth*, geram, a cada iteração, valores de posição e orientação durante os experimentos. Logo, para implementar o cálculo de distância total percorrida pelo robô é necessário calcular a distância entre pontos para cada iteração e somar todas essas distâncias.

$$d_{total} = \sum_{m=1}^n \sqrt{(x_m - x_{(m-1)})^2 + (y_m - y_{(m-1)})^2} \quad (3.21)$$

O desempenho das técnicas apresentadas neste capítulo pode ser verificado na Seção 4.4.

Capítulo 4

Resultados experimentais

Neste capítulo são apresentados o sistema de validação e estrutura de testes utilizados, além dos experimentos realizados e resultados atingidos.

4.1 Sistema de validação e estrutura de testes

Para formar o sistema de validação foram dispostas oito câmeras Vicon no Laboratório de Robótica do Instituto Tecnológico Vale. Após instaladas as câmeras, foi realizado o procedimento de calibração padrão utilizando o objeto chamado *active wand calibration*, “varinha ativa de calibração”, em tradução literal (Figura 4.1). Este processo otimiza o volume de rastreamento, bem como define a origem e os eixos do sistema de coordenadas do mesmo. Além deste passo, ajustes como foco e zoom foram realizados para identificar as esferas refletivas empregadas com maior precisão. A disposição das câmeras e o sistema de coordenadas definido, bem com o espaço do laboratório utilizado, podem ser observados nas Figuras 4.2 e 4.3, respectivamente.



Figura 4.1: Objeto utilizado para calibrar câmeras Vicon e definir volume de rastreamento.

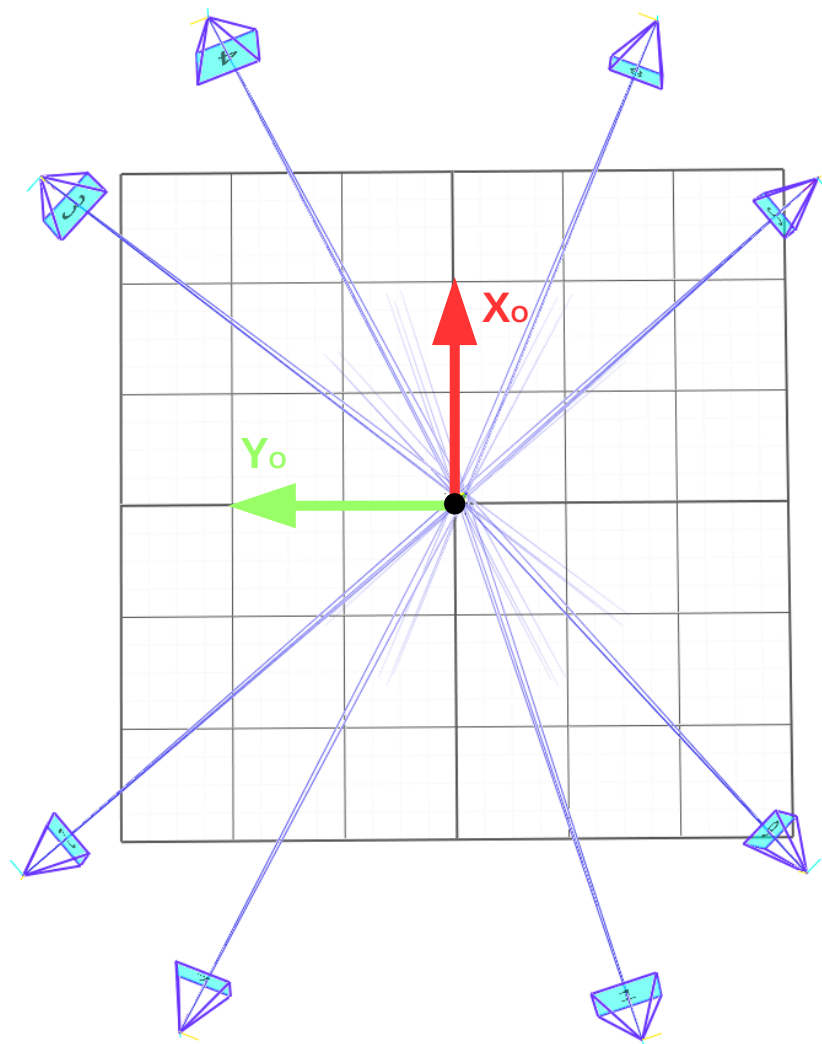


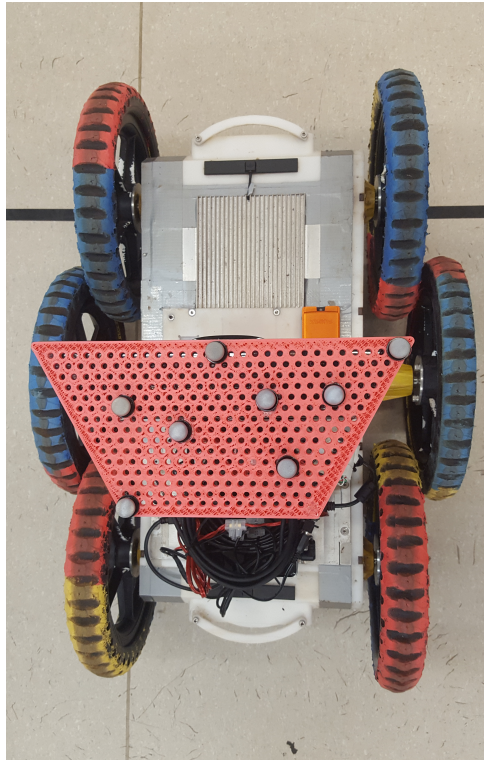
Figura 4.2: Disposição das 8 câmeras no Laboratório de Robótica do ITV.



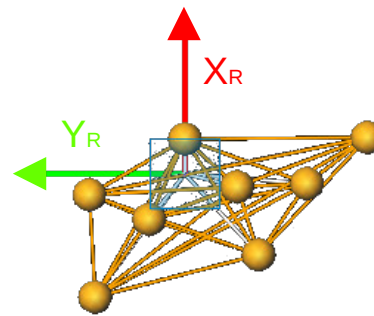
Figura 4.3: Espaço utilizado no Laboratório de Robótica do ITV.

A identificação do EspeleoRobô foi feita com as esferas refletivas mostradas na Figura 4.4a. Na figura 4.4b é possível observar o objeto EspeleoRobô criado no *software* proprietário, além dos eixos x_R e y_R do sistema de coordenadas do robô. Foram utilizadas oito marcas refletivas dispostas de forma aleatória para que não fosse formado um desenho

simétrico, o que atrapalharia na identificação da orientação feita pelas câmeras. A simetria pode causar picos nos ângulos computados, gerados pela interpretação equivocada de que o objeto, durante um intervalo curto de tempo, girou em torno do próprio eixo.



(a)



(b)

Figura 4.4: (a) Posicionamento das esferas refletivas e (b) objeto criado no *software* Tracker para captura de movimento.

Na Figura 4.5 é possível observar o esquema de ligação dos dispositivos para realização de testes. As câmeras Vicon são interligadas por um *switch* que por sua vez está conectado à estação Vicon, um computador com sistema operacional Windows 7. As câmeras capturam os movimentos descritos pelo objeto formado pelo conjunto de esferas refletivas. Os dados de leitura das câmeras são processados na estação, e disponibilizados através da API (*Application Programming Interface*) Vicon DataStream SDK¹ para uma rede *wireless* IEEE 802.11 2.4GHz.

A base de controle, controlada por um operador, envia os comandos ao robô através um rádio Ubiquiti modelo Rocket M900, ligado a uma antena omnidirecional Yagi AMY-9M16. O mini-computador embarcado no EspeleoRobô recebe e processa os comandos da base de controle, acarretando no movimento do robô. O mesmo computador acessa na rede as informações processadas das câmeras e recebe informações de todos os sensores.

¹<https://www.vicon.com/products/software/datastream-sdk>.

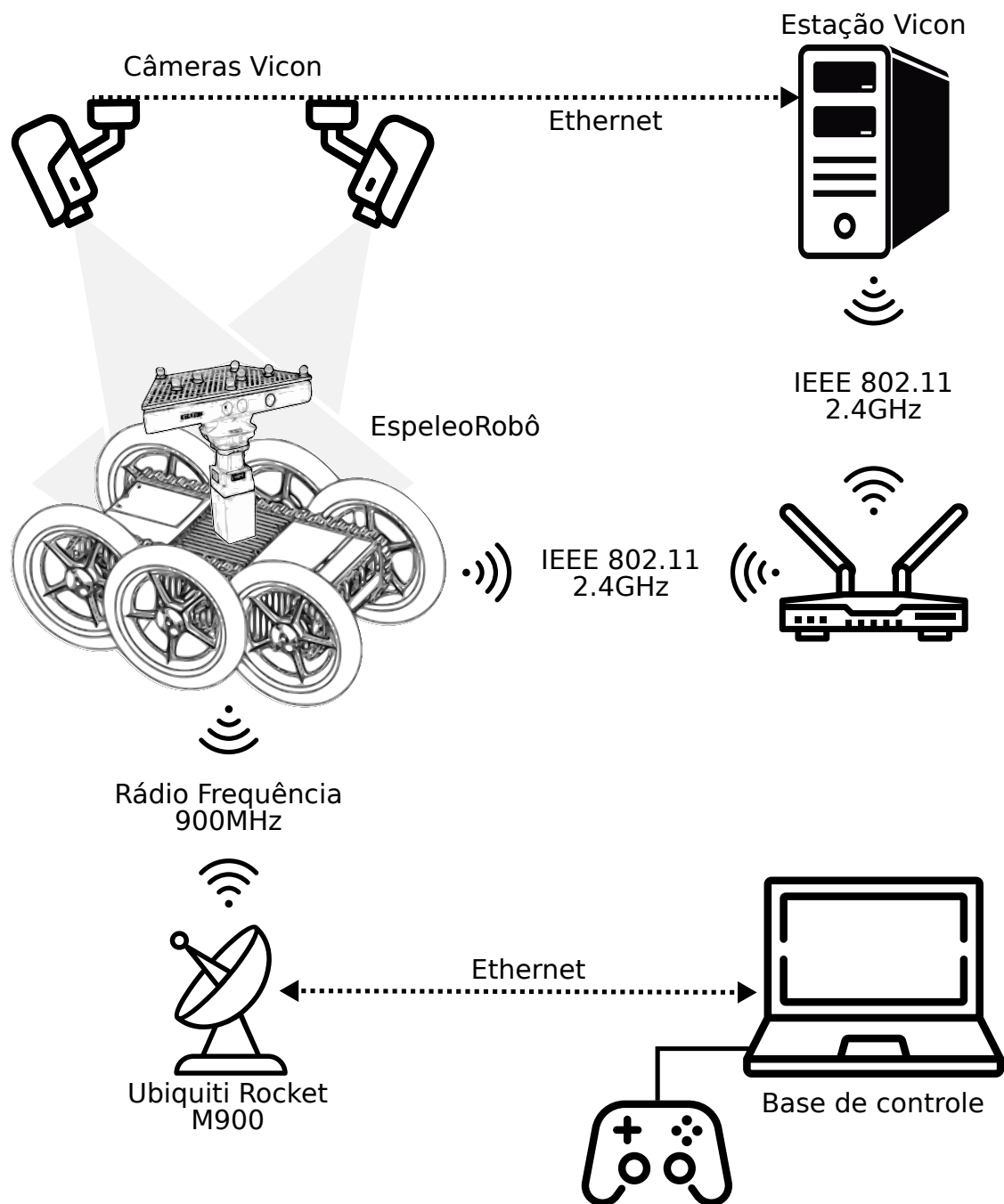


Figura 4.5: Esquema de ligação de equipamentos para testes.

4.2 Software

O primeiro passo para a implementação dos códigos necessários foi a simulação da navegação do EspeleoRobô, integração dos sensores e teste dos pacotes por meio do simulador Gazebo² (Figura 4.6). A simulação é uma ferramenta importante pois permite o esboço e a correção de erros iniciais da implementação sem a necessidade da realização de testes físicos, acelerando assim o desenvolvimento como um todo.

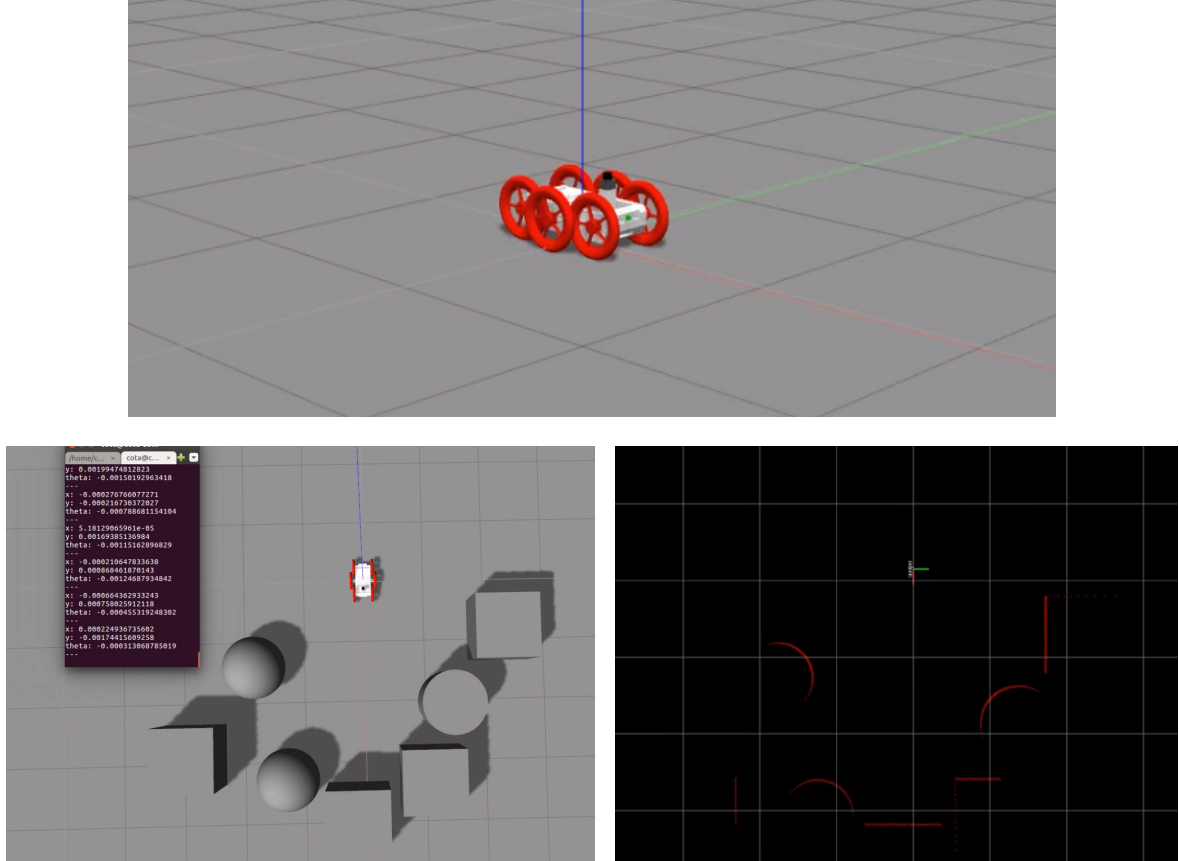


Figura 4.6: Registros das simulações realizadas antes da implementação dos testes no robô real.

Na abordagem com *encoders* foi desenvolvido um código que implementa o modelo cinemático *skid-steer*. Para a implementação de odometria visual foi utilizado o algoritmo proposto em Labbé e Michaud (2014), *rtabmap-ros*³ o qual foi disponibilizado pelos autores. Nos experimentos com LiDAR e LiDAR combinado com IMU foi usada uma implementação do algoritmo *PLICP* Censi (2008) chamada *laser_scan_matcher*⁴, também disponibilizada pelo autor. Por fim, para computar o *ground truth*, gerado pelo sistema de captura de movimento, foi utilizado o pacote *vicon*⁵.

²<http://gazebosim.org/>

³http://wiki.ros.org/rtabmap_ros

⁴http://wiki.ros.org/laser_scan_matcher

⁵<http://wiki.ros.org/vicon>

Um nó do ROS foi implementado para acessar os dados odométricos gerados pelos pacotes que processam os dados coletados pelos sensores. O nó publica a odometria online para o robô e grava *bags* contendo as informações de sensores e funcionamento do robô durante experimento. A partir dessas informações é possível gerar gráficos e tabelas visando analisar, posteriormente, o desempenho de cada método (Figura 4.7).

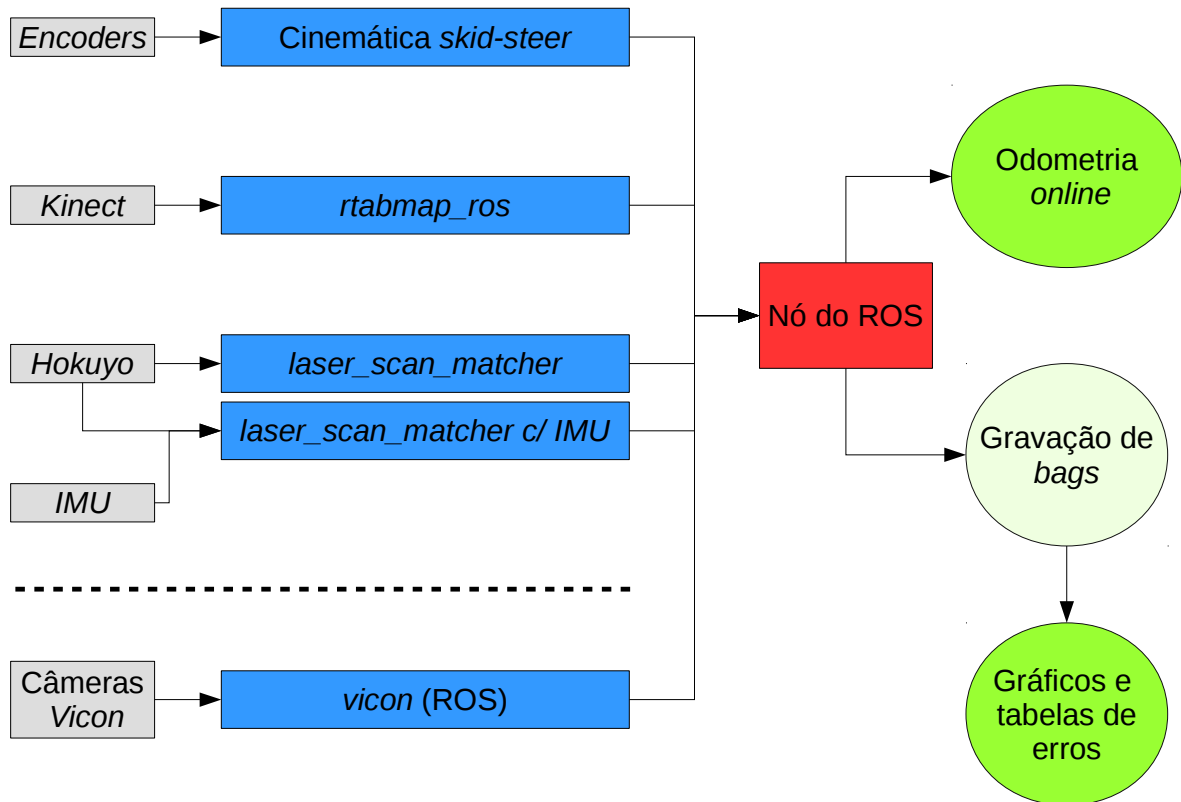


Figura 4.7: Esquema de ligação de equipamentos para testes.

4.3 Comparação preliminar de modelos cinemáticos: diferencial vs *skid-steer*

Visando maior objetividade na comparação dos resultados das implementações propostas, antes de iniciar os experimentos com todas as técnicas de odometria, foi realizada uma série de testes para definir o modelo cinemático de melhor desempenho na abordagem com *encoders*.

Nas Figuras 4.8, 4.9, 4.10, 4.11, 4.12 e 4.13 é possível observar o comportamento dos modelos em relação ao cálculo de trajetória em curvas suaves e bruscas. Cada gráfico de trajetória está acompanhado de tabela de erros (Tabelas 4.1, 4.2, 4.3, 4.4, 4.5 e 4.6) para uma análise mais aprofundada apresentando as métricas: erros máximo e médio, desvio padrão (σ) e integral dos erros absolutos (IAE), no cálculo das componentes x , y e θ . Em todas as métricas são utilizados os módulos dos erros em relação ao *ground truth*.

Analisando os dados expostos é possível afirmar que o modelo *skid-steer* comporta-se melhor quando comparado ao modelo diferencial para representar os movimentos descritos pelo EspeleoRobô utilizando a leitura dos *encoders*. Os erros médios gerados pelo modelo diferencial chegam a valores até 5x maiores que os calculados com o *skid-steer*. Dessa forma, apenas a abordagem com modelo *skid-steer* será comparada com as outras implementações de odometria.

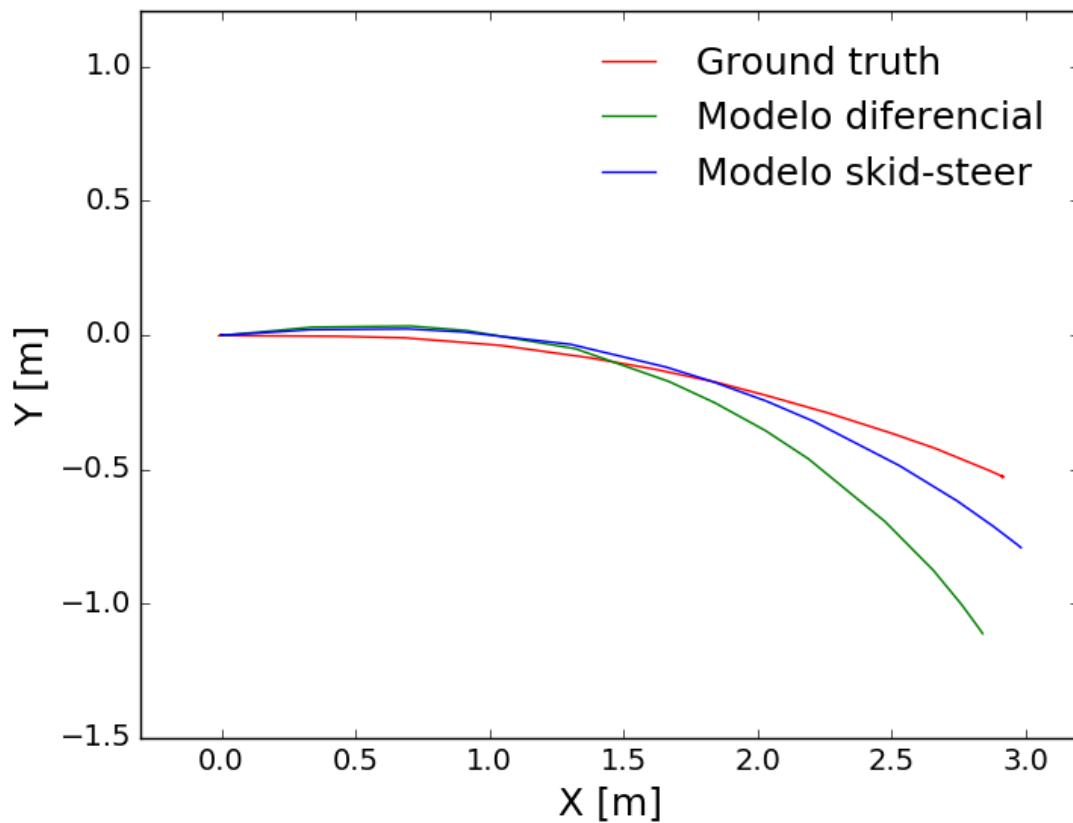


Figura 4.8: Diferencial vs *skid-steer* (1) - Comparação de precisão dos modelos.

Erro	x (m)		y (m)		θ (°)	
	Diferencial	Skid-steer	Diferencial	Skid-steer	Diferencial	Skid-steer
Máx.	0,295	0,281	0,591	0,292	61,9	34,2
Méd.	0,070	0,068	0,215	0,098	24,4	14,0
σ	0,077	0,082	0,267	0,120	27,5	14,5
IAE	4,357	4,246	13,301	6,075	1511,3	869,3

Tabela 4.1: Diferencial vs *skid-steer* (1) - Tabela comparativa dos erros de odometria.

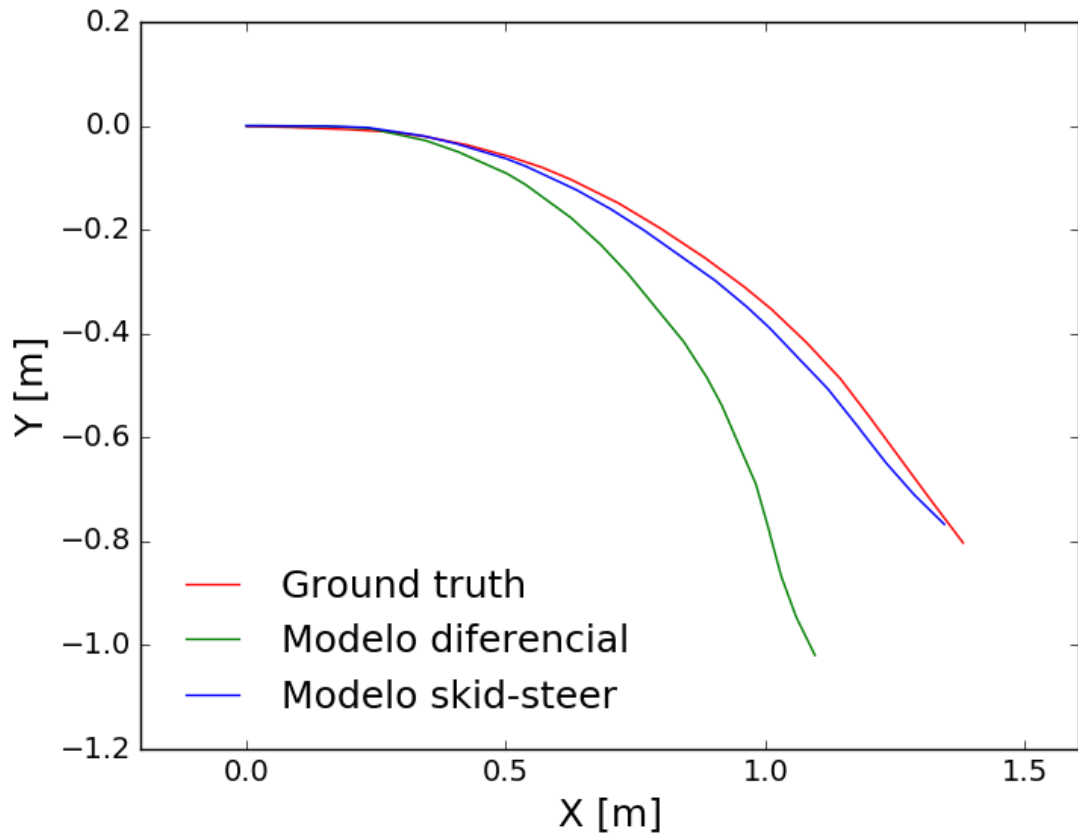


Figura 4.9: Diferencial vs *skid-steer* (2) - Comparação de precisão dos modelos.

Erro	x (m)		y (m)		θ (°)	
	Diferencial	Skid-steer	Diferencial	Skid-steer	Diferencial	Skid-steer
Máx.	0,286	0,106	0,294	0,101	107,2	28,2
Méd.	0,086	0,045	0,130	0,037	54,4	15,7
σ	0,083	0,028	0,105	0,035	39,6	9,5
IAE	1,890	0,981	2,859	0,811	1196,6	346,2

Tabela 4.2: Diferencial vs *skid-steer* (2) - Tabela comparativa dos erros de odometria.

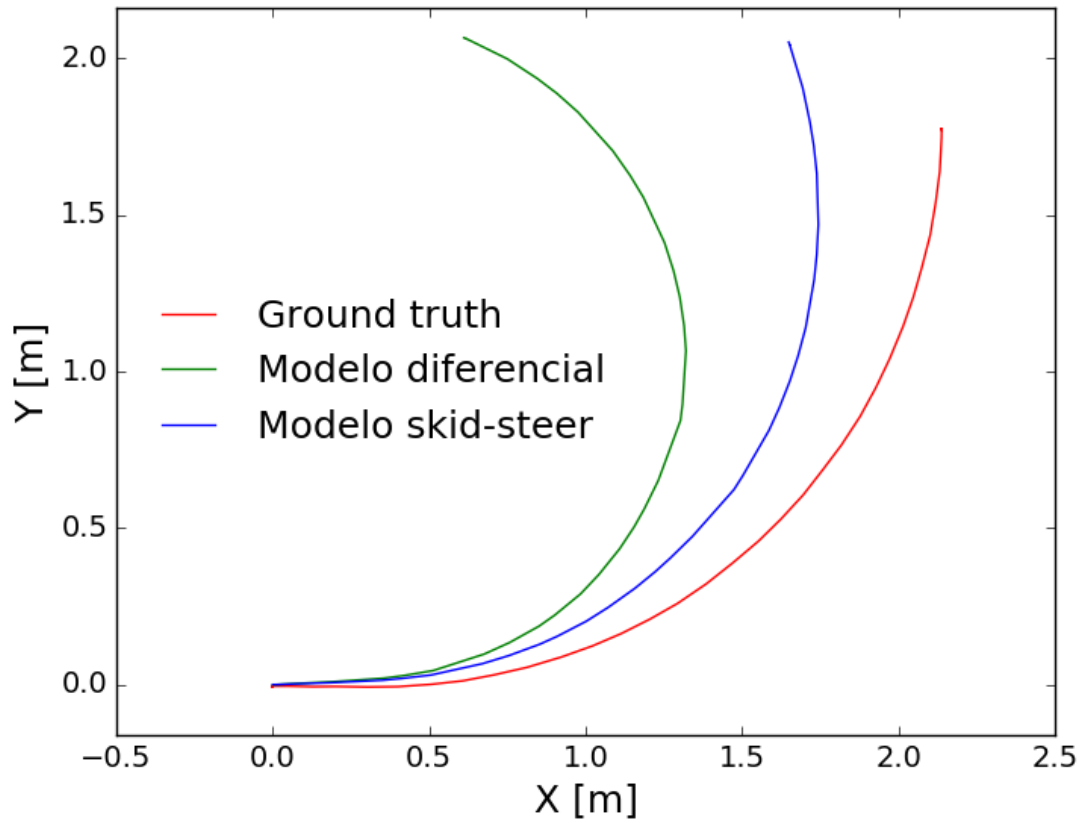


Figura 4.10: Diferencial vs *skid-steer* (3) - Comparação de precisão dos modelos.

Erro	x (m)		y (m)		θ (°)	
	Diferencial	Skid-steer	Diferencial	Skid-steer	Diferencial	Skid-steer
Máx.	1,527	0,489	0,518	0,304	202,8	71,7
Méd.	0,578	0,190	0,190	0,131	90,3	32,6
σ	0,668	0,209	0,173	0,120	87,3	29,8
IAE	43,925	14,466	14,429	9,924	6859,5	2476,9

Tabela 4.3: Diferencial vs *skid-steer* (3) - Tabela comparativa dos erros de odometria.

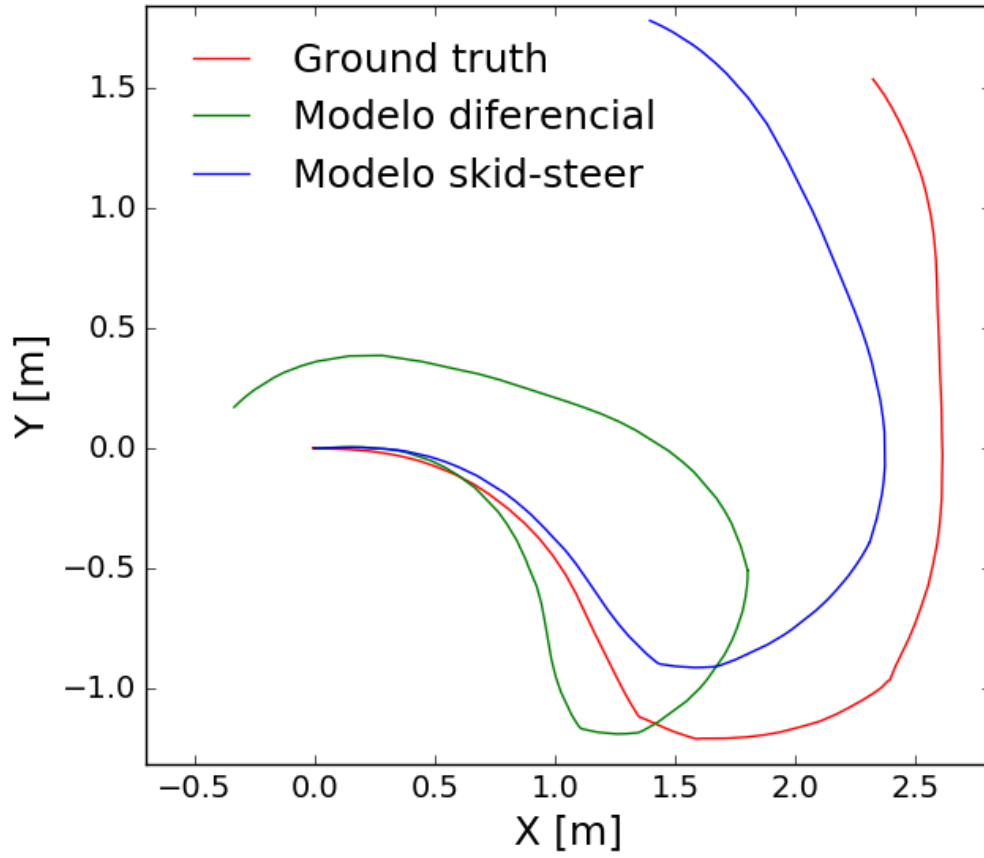


Figura 4.11: Diferencial vs *skid-steer* (4) - Comparação de precisão dos modelos.

Erro	x (m)		y (m)		θ (°)	
	Diferencial	Skid-steer	Diferencial	Skid-steer	Diferencial	Skid-steer
Máx.	259,7	85,1	107,6	65,3	262,5	91,1
Méd.	48,0	12,7	14,5	21,1	71,2	25,0
σ	74,8	22,0	22,7	23,8	82,2	29,1
IAE	5610,7	1485,8	1699,9	2468,5	8333,8	2927,1

Tabela 4.4: Diferencial vs *skid-steer* (4) - Tabela comparativa dos erros de odometria.

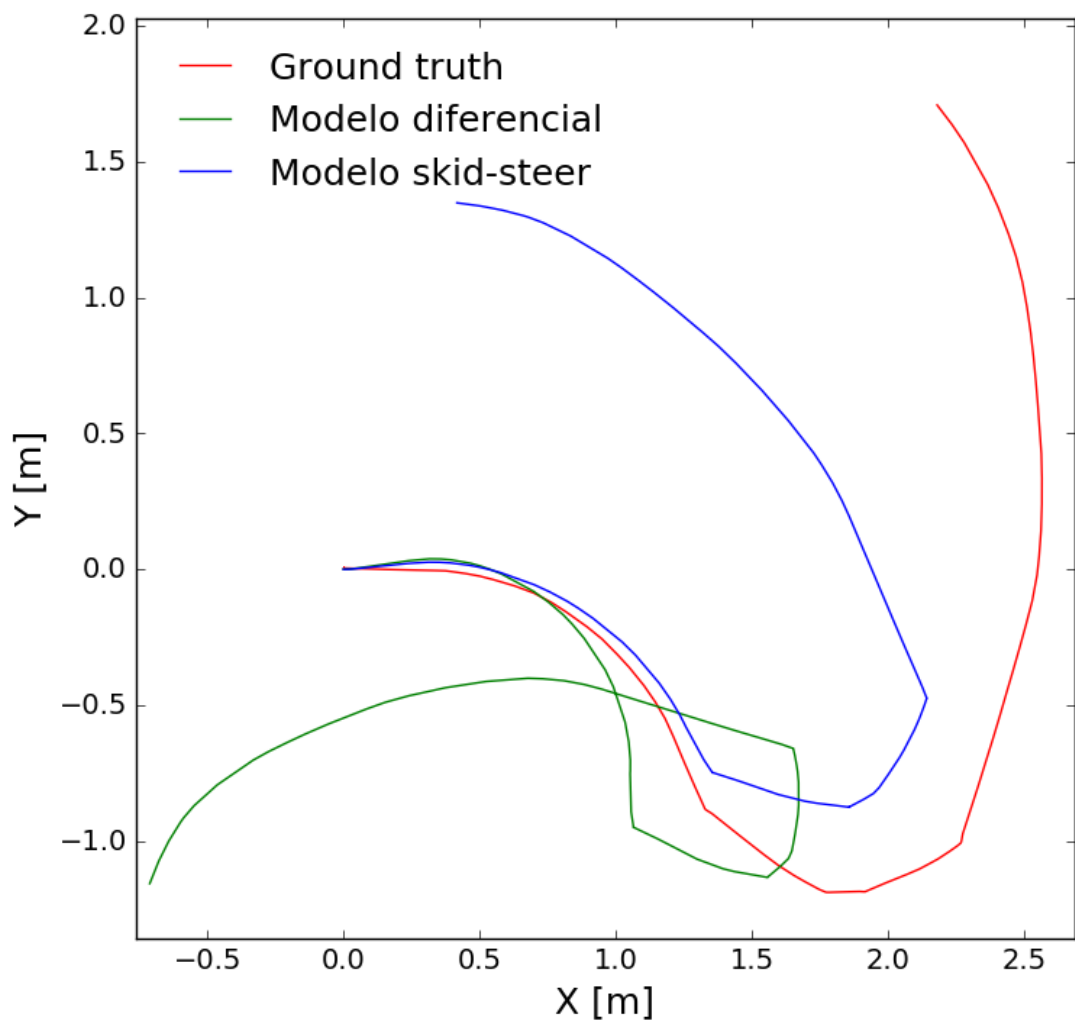


Figura 4.12: Diferencial vs *skid-steer* (5) - Comparação de precisão dos modelos.

Erro	x (m)		y (m)		θ (°)	
	Diferencial	Skid-steer	Diferencial	Skid-steer	Diferencial	Skid-steer
Máx.	2,954	1,731	2,707	0,536	294,3	122,8
Méd.	0,732	0,338	0,356	0,187	99,8	46,0
σ	0,983	0,524	0,642	0,195	92,4	41,4
IAE	74,668	34,439	36,336	19,103	10180,1	4695,7

Tabela 4.5: Diferencial vs *skid-steer* (5) - Tabela comparativa dos erros de odometria.

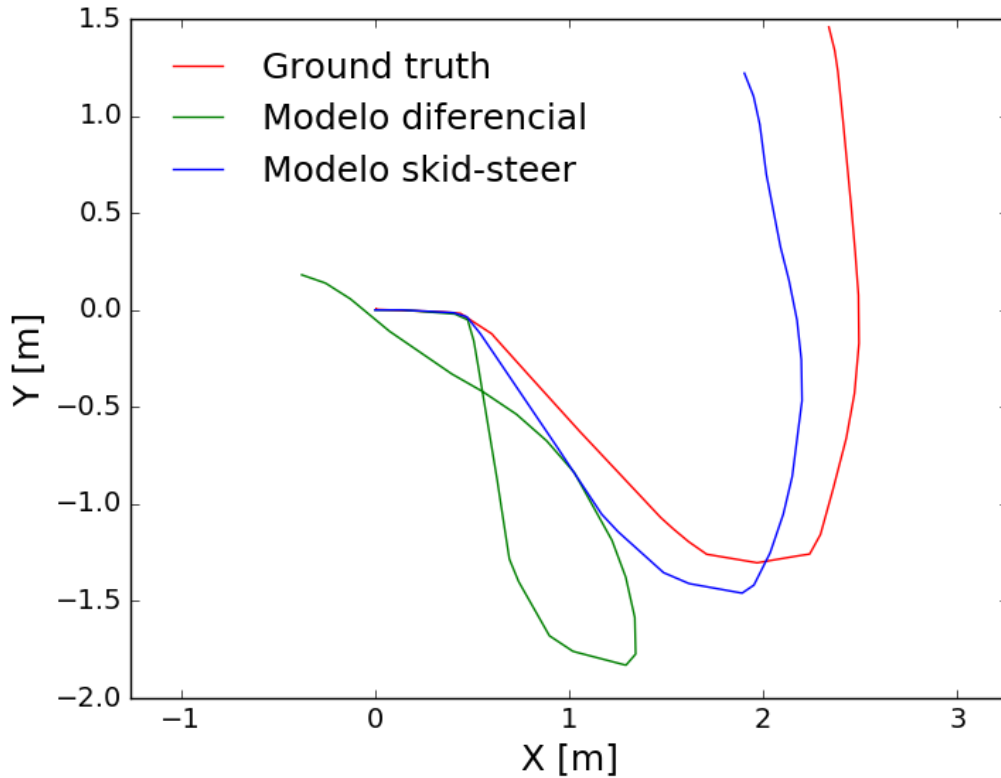


Figura 4.13: Diferencial vs *skid-steer* (6) - Comparação de precisão dos modelos.

Erro	x (m)		y (m)		θ (°)	
	Diferencial	Skid-steer	Diferencial	Skid-steer	Diferencial	Skid-steer
Máx.	2,716	0,435	1,277	0,406	227,3	74,9
Méd.	0,605	0,127	0,291	0,076	64,7	21,9
σ	0,844	0,149	0,355	0,096	77,8	24,0
IAE	32,669	6,849	15,701	4,128	3493,4	1182,4

Tabela 4.6: Diferencial vs *skid-steer* (6) - Tabela comparativa dos erros de odometria.

4.4 Avaliação das técnicas de odometria

Foram conduzidos experimentos num ambiente controlado (laboratório) visando avaliar as abordagens de odometria propostas. Os testes foram realizados com o robô se locomovendo numa trajetória de circuito fechado, com formato semelhante ao número oito ou símbolo de infinito (∞). Essa configuração é comumente utilizada em avaliações de odometria. A execução da trajetória foi comandada por um operador por meio da base de controle do EspeleoRobô. O dispositivo pode chegar a aproximadamente 1m/s e foi operado em dois modos de velocidade: (i) sem restrição de velocidade e (ii) com restrição de 50% da velocidade máxima. Os testes foram repetidos seis vezes, sendo três para cada modo de velocidade.

No início de cada teste o robô foi posicionado manualmente de modo que seu sistema de coordenadas (Figura 4.4b) fosse coincidente com o sistema de coordenadas das câmeras Vicon (Figura 4.2). Dessa forma, ao iniciar-se cada teste, o robô foi devidamente orientado e a distância do mesmo em relação à origem do sistema de coordenadas utilizado para gerar o *ground truth* seria zero. Mais especificamente, todo teste começou no ponto zero: posição $x = 0$, $y = 0$; e orientação $\theta = 0$, com respeito ao sistema de coordenadas do sistema de captura de movimento com câmeras Vicon. Após o posicionamento, todos os algoritmos eram acionados (também começando do ponto zero) para que, durante os movimentos, a posição e orientação fossem computados corretamente.

A partir da locomoção do robô é possível extrair o deslocamento em x , y , e a rotação θ em torno do eixo z ao longo do tempo. Dessa forma, tem-se a posição e orientação do robô no plano de forma temporal, as quais integradas formam a trajetória descrita.

Para melhor entendimento dos comentários sobre as trajetórias computadas é necessário salientar que todos os testes começam do ponto (0,0) do plano representado nos gráficos e o robô é movimentado primeiramente no sentido crescente do eixo x e decrescente de y . Iniciado o movimento neste sentido, o robô realiza a trajetória em oito e finaliza a mesma próximo do ponto de partida. Nas tabelas apresentadas, devido à alta precisão do *ground truth*, os valores de distância são mostrados em metros com três casas decimais.

Os resultados experimentais serão apresentados a seguir nas subseções 4.4.1 e 4.4.2, divididos por modos de velocidade sem restrição de velocidade e velocidade limitada a 50% da capacidade máxima do robô, respectivamente.

Em cada subseção serão apresentados três testes e para cada teste serão mostrados: um gráfico de posição, com o qual é possível observar a trajetória descrita pelo dispositivo; um gráfico de erros de posição e orientação evidenciando a diferença entre as abordagens LiDAR e LiDAR+IMU; uma tabela de erros relacionados a posição e orientação do robô durante o cálculo de trajetória; uma figura com gráficos relacionadas ao cálculo da distância total percorrida; e uma tabela de erros, por meio da qual pode-se verificar o desempenho dos métodos na realização desse cálculo, bem como a acurácia dos mesmos em relação ao ponto de chegada (posição final do robô ao fim do teste).

4.4.1 Resultados experimentais em testes sem restrição de velocidade

Nessa subseção, são apresentados os resultados experimentais obtidos com testes realizados sem restrição de velocidade durante a operação do robô. Isso significa que o robô operou recebendo comandos para acionar o motores em até 100% da sua capacidade, podendo chegar a aproximadamente 1m/s. Os gráficos, tabelas e análises apresentados a seguir estão separados por teste.

4.4.1.1 Teste 1

Esse teste teve duração de 36,4s entre o momento de acionamento dos algoritmo e o encerramento dos mesmos após realizada a trajetória. A distância total percorrida computada pelo sistema de captura de movimento utilizada como *ground truth* foi de 17,001m.

Em um primeiro olhar na Figura 4.14 é possível facilmente observar que o método que utiliza a leitura do giro das rodas com *encoders* se perde logo na primeira curva mais fechada e erra o restante da trajetória. Em seguida observa-se que a abordagem que utiliza as imagens geradas pela câmera RGB-D interrompe os cálculos nesta mesma curva e computa apenas pequena parte do percurso traçado.

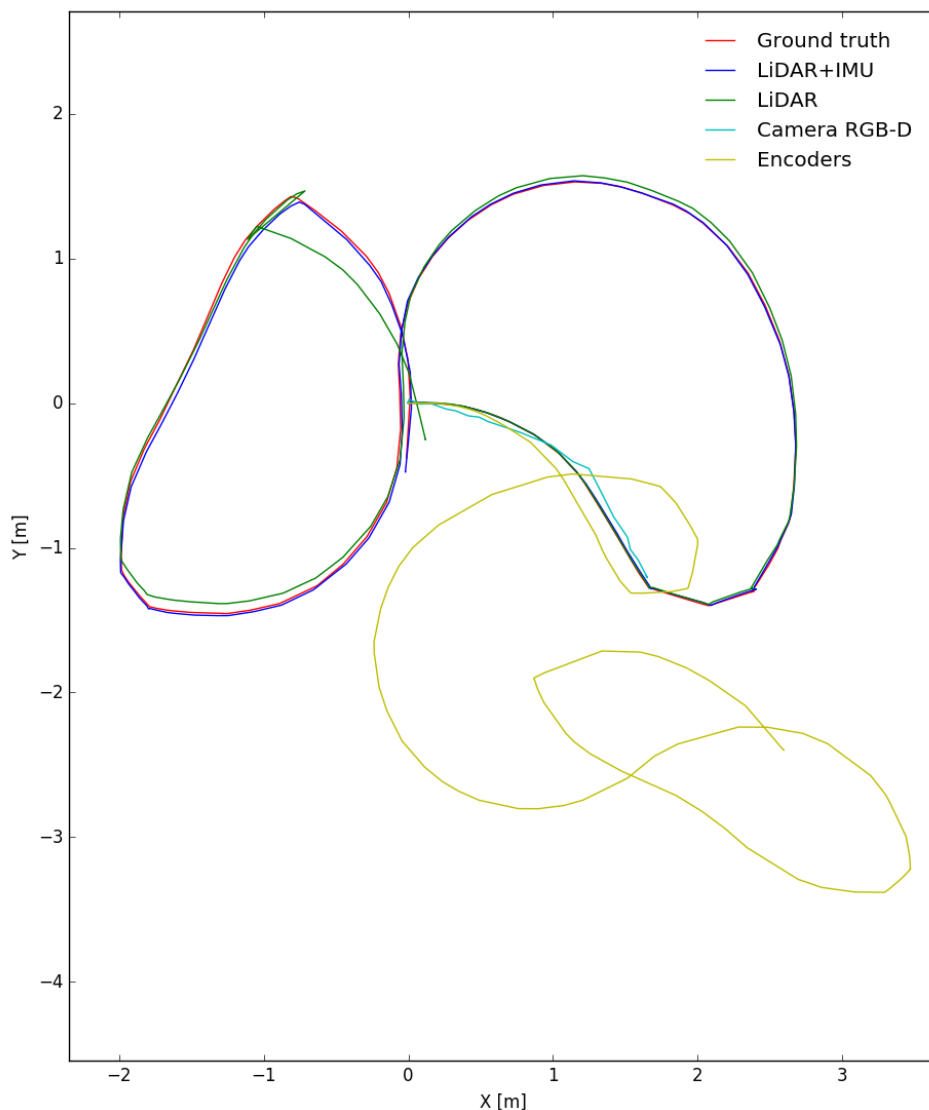


Figura 4.14: Teste 1 - Trajetória computada.

Durante toda a trajetória, as abordagens que utilizam dados do LiDAR e da combinação de LiDAR com IMU apresentam os melhores resultados. Entretanto o uso da unidade de medição faz diferença quando observada atentamente a penúltima curva, a

qual é melhor computada pela curva azul, e claramente na última curva. O movimento brusco fez com que a abordagem errasse no cálculo do deslocamento, o que prejudicou o cálculo da trajetória. Na Figura 4.15 é evidenciada a diferença de performance entre os dois melhores métodos. É possível verificar um pico próximo dos 25s no gráfico de orientação (θ), isso ocorre porque quando os cálculos ultrapassam os limites da orientação 0° e 360° o sistema computa que por um instante muito curto de tempo, o robô deu uma volta completa. Algumas vezes, o próprio sistema de captura provoca esses picos, como ocorre se as esferas refletivas não forem posicionadas de forma assimétrica. Esse problema, entretanto, não afeta o cálculo das componentes (x , y) da posição, como pode ser observado no gráfico.

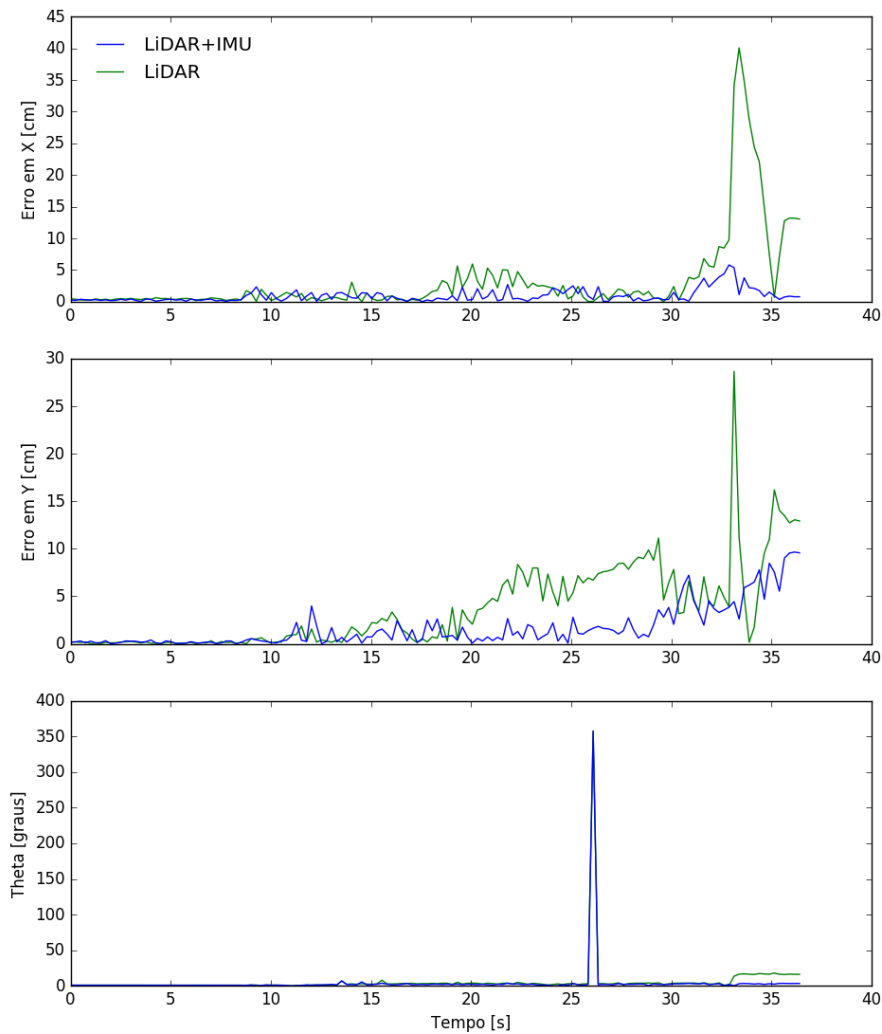


Figura 4.15: Teste 1 - Erros computados - LiDAR vs. LiDAR+IMU.

A Tabela 4.7 possibilita análise mais detalhada dos erros por meio do erro máximo e médio, desvio padrão e integral do erro absoluto (IAE). A implementação que combina o sensor LiDAR Hokuyo com a unidade de medição inercial apresenta melhor desempenho ao calcular a posição em x e y com erro médio menor que 1cm em x e menor que 2cm em y . No que diz respeito à orientação, este método também proporciona os menores valores

de erro médio, desvio padrão e IAE, apesar da câmera RGB-D apresentar menor erro máximo. É necessário destacar, no entanto, que os cálculos com câmera RGB-D foram interrompidos, logo o método não foi devidamente testado no momento do erro máximo obtido pelas outras abordagens (após 25s como é possível observar na Figura 4.15).

x (m)				
	L+IMU	LiDAR	RGB-D	Encoder
Erro máximo	0,058	0,401	3,629	5,274
Erro médio	0,009	0,032	1,160	1,480
Desvio padrão (σ)	0,010	0,065	1,246	1,703
Integral do erro absoluto (IAE)	1,353	4,733	169,346	216,061

y (m)				
	L+IMU	LiDAR	RGB-D	Encoder
Erro máximo	0,097	0,287	2,724	3,987
Erro médio	0,017	0,037	0,752	1,297
Desvio padrão(σ)	0,022	0,044	0,957	1,357
Integral do erro absoluto (IAE)	2,463	5,366	109,778	189,299

θ (°)				
	L+IMU	LiDAR	RGB-D	Encoder
Erro máximo	357,8	357,3	230,8	459,2
Erro médio	4,2	6,0	69,1	147,9
Desvio padrão(σ)	29,5	29,6	72,6	136,6
Integral do erro absoluto (IAE)	615,7	877,6	10094,3	21590,3

Tabela 4.7: Teste 1 - Tabela de erros na trajetória.

Na Figura 4.16 são comparadas as performances dos métodos implementados em relação ao cálculo da distância total percorrida (DTP). No primeiro gráfico mostrado na figura, observa-se que o movimento é iniciado antes dos 10s de teste e antes mesmo de completar 15s, a abordagem interrompe os cálculos. As outras abordagens seguem próximas do *ground truth* e a eficácia do respectivo cálculo pode ser melhor verificada no segundo gráfico mostrado, o qual apresenta o erro relativo ao *ground truth* e também no terceiro gráfico, que mostra a integral do erro absoluto, somando todos os erros computados por cada método. É válido frisar que os limites do eixo y dos últimos gráficos nessa figura foram ajustados para melhor mostrar as abordagens que completaram os cálculos. Os erros do método que utiliza câmera RGB-D superam os limites do gráfico e podem ser verificados na tabela 4.8.

Com os dados presentes nesta mesma tabela pode-se analisar o desempenho das implementações nos cálculos odométricos. A abordagem que melhor se aproxima do *ground truth* é a LiDAR+IMU com o valor de 17,148m. O erro em relação à DTP computada pelo

sistema de câmeras é de apenas 14,7cm, o que representa apenas 0,86% de erro. Isso quer dizer que para cada metro percorrido pelo robô, o método em questão computa 0,86cm a mais, ou seja, o erro de odometria durante o teste foi, em média, menor que 1cm para cada metro de deslocamento. Quando observado o erro relativo ao ponto de chegada do teste, a mesma implementação apresenta melhor desempenho entre as quatro com erro menor que 10cm.

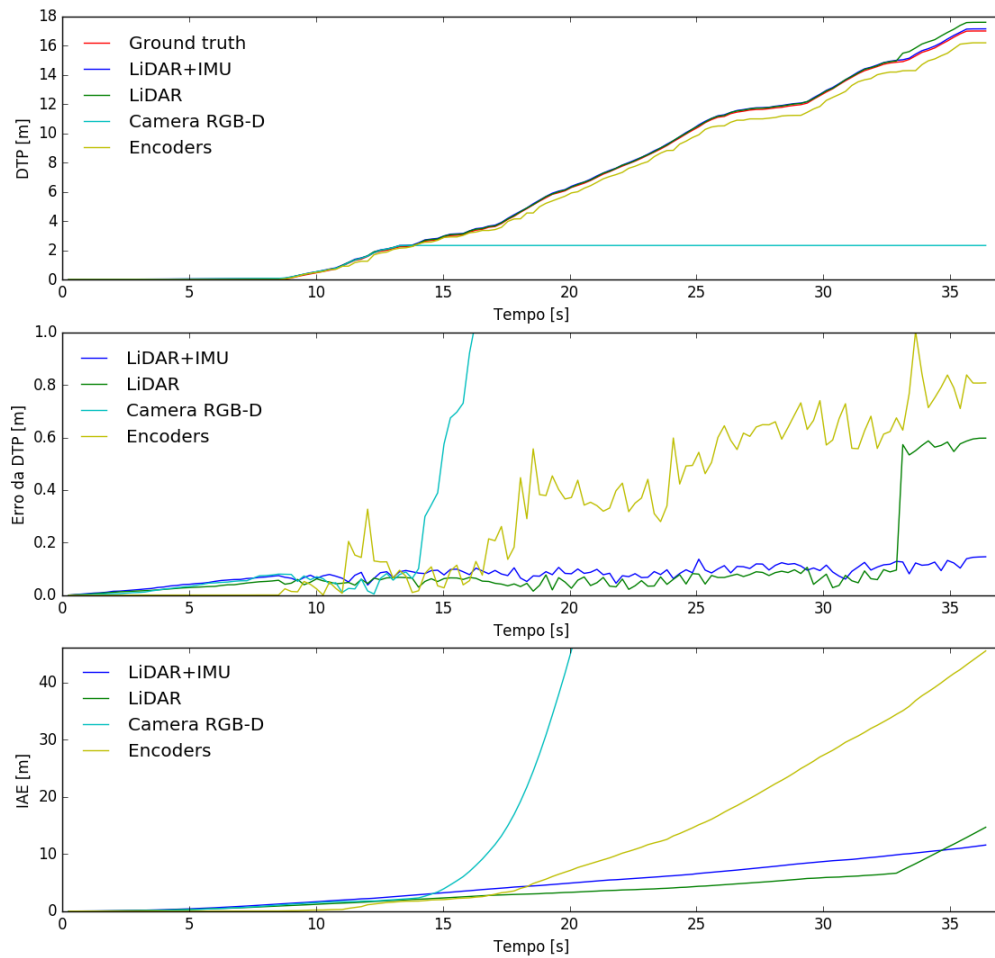


Figura 4.16: Teste 1 - Distância total percorrida, erros e integral do erro absoluto (IAE).

<i>Ground truth:</i> 17,001m	L+IMU	LiDAR	RGB-D	Encoder
Distância total percorrida (DTP) (m)	17,148	17,598	2,332	16,193
Erro de distância total percorrida (m)	0,147	0,597	14,669	0,807
% de erro em relação ao <i>ground truth</i>	0,86%	3,51%	86,28%	4,75%
Integral do erro absoluto (m)	11,561	14,667	675,643	45,494
Erro no ponto de chegada (m)	0,096	0,184	1,847	3,299
% de erro em relação ao <i>ground truth</i>	0,56%	1,08%	10,86%	19,40%

Tabela 4.8: Teste 1 - Tabela de erros de odometria.

4.4.1.2 Teste 2

Este teste teve duração de 25,1s entre o momento de acionamento dos algoritmo e o encerramento dos mesmos após realizada a trajetória. A distância total percorrida computada pelo sistema de captura de movimento utilizada como *ground truth* foi de 15,973m.

Analisando a Figura 4.17, é possível afirmar que o método com *encoders* erra, mais uma vez, na primeira curva mais fechada. Na prática, esta curva é um giro em torno do próprio eixo. Isso ocorre quando o teleoperador envia comandos para o robô curvar, quando este está em alta velocidade. Ainda observando a trajetória pode-se constatar que esta abordagem computa o giro maior que o que foi de fato realizado. Dessa forma, a curva traçada pelo desenvolvimento com *encoder* é ainda mais fechada do que a que o robô realmente descreveu. Observa-se também que a abordagem que utiliza as imagens geradas pela câmera RGB-D interrompe os cálculos e novamente computa apenas pequena parte da trajetória.

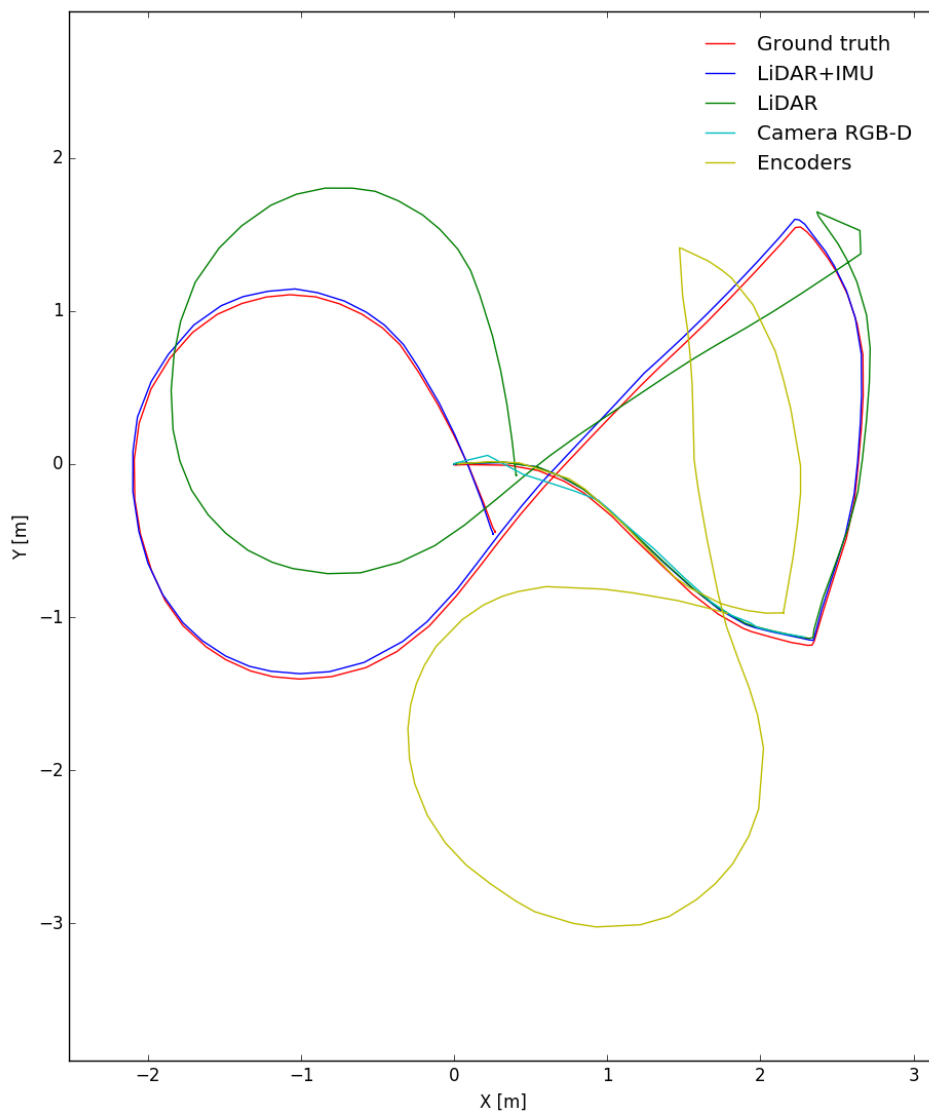


Figura 4.17: Teste 2 - Trajetória computada.

A abordagem que utiliza apenas LiDAR teve dificuldades de acompanhar o movimento realizado na curva observada no 1º quadrante. O erro causado nesta curva prejudicou o restante da trajetória calculada. Destaca-se, portanto, a diferença que o uso da unidade de medição fez, pois a abordagem que combina este sensor com o LiDAR consegue acompanhar o movimento. Na Figura 4.18 é evidenciada a diferença de desempenho causada pelo uso da IMU.

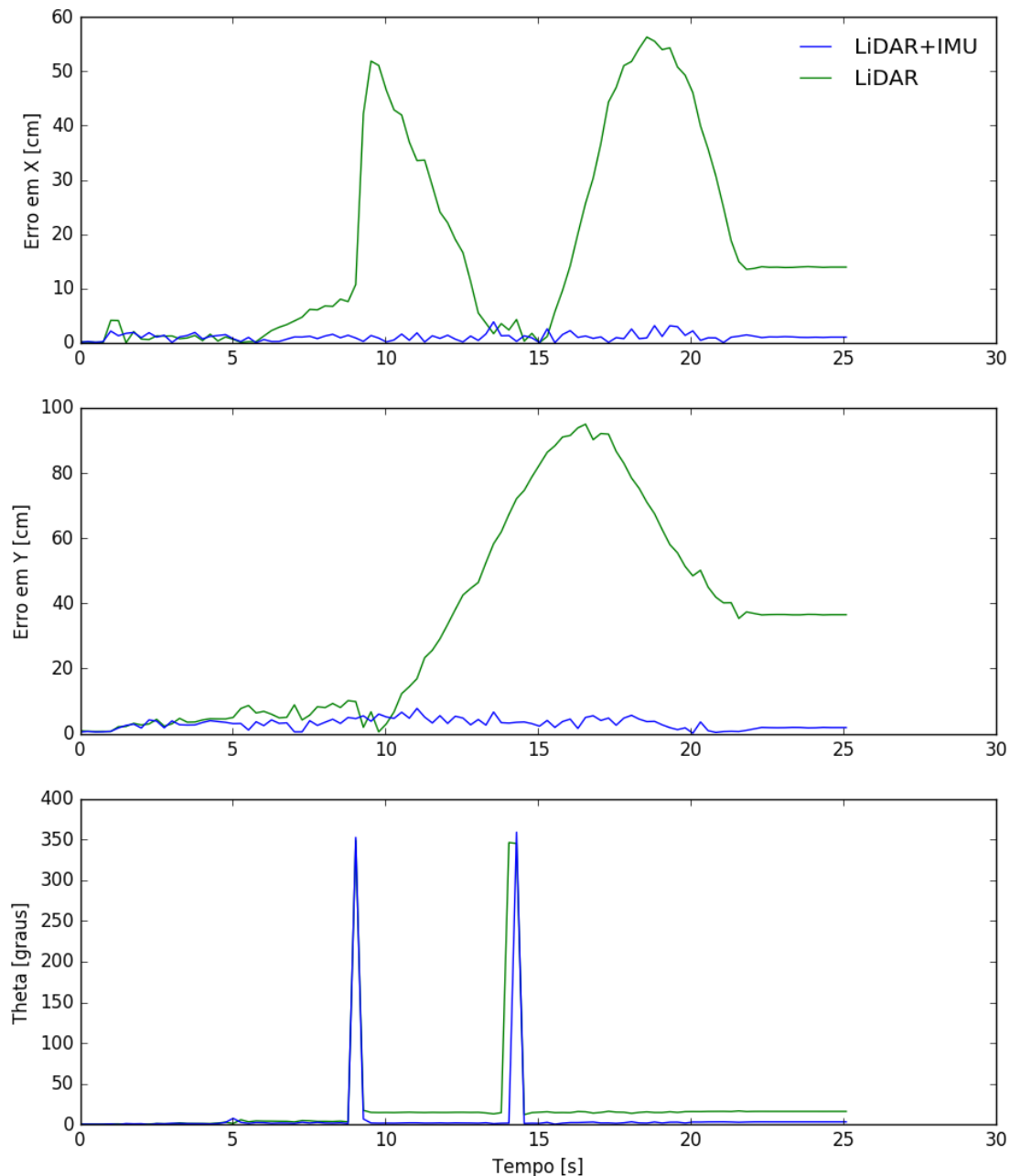


Figura 4.18: Teste 2 - Erros computados - LiDAR vs. LiDAR+IMU.

Novamente ao analisar a tabela de erros (4.9) verifica-se que a implementação que combina os sensores LiDAR e IMU apresenta os menores valores de erros máximo e médio, desvio padrão e integral do erro absoluto. Os erros médios computados no eixos x e y são de 1,1cm e 3cm, respectivamente. Em se tratando da orientação, este método também

proporciona os menores valores de erro médio, desvio padrão e IAE, apesar de novamente a câmera RGB-D apresentar menor erro máximo. É válido lembrar que os cálculos com câmera RGB-D foram interrompidos, apenas pequena parte da trajetória foi computada, excluindo-se as curvas principais. Portanto, não é justa a comparação do valor obtido pela câmera RGB-D com as outras abordagens que completaram o percurso.

x (m)				
	L+IMU	LiDAR	RGB-D	Encoder
Erro máximo	0,039	0,563	4,373	2,916
Erro médio	0,011	0,172	1,636	1,053
Desvio padrão (σ)	0,007	0,181	1,472	0,845
Integral do erro absoluto (IAE)	1,100	17,377	165,211	106,399

y (m)				
	L+IMU	LiDAR	RGB-D	Encoder
Erro máximo	0,078	0,950	2,678	2,787
Erro médio	0,030	0,336	0,975	0,837
Desvio padrão(σ)	0,016	0,304	0,861	0,848
Integral do erro absoluto (IAE)	3,068	33,970	98,428	84,552

θ (°)				
	L+IMU	LiDAR	RGB-D	Encoder
Erro máximo	359,0	348,9	175,3	390,9
Erro médio	9,1	20,5	76,5	126,9
Desvio padrão(σ)	49,5	57,7	52,9	90,7
Integral do erro absoluto (IAE)	921,0	2070,9	7725,0	12813,8

Tabela 4.9: Teste 2 - Tabela de erros na trajetória.

Na Figura 4.19 são comparados os desempenhos dos métodos implementados em relação ao cálculo da distância total percorrida (DTP). No primeiro gráfico mostrado na figura, observa-se que a abordagem com câmera RGB-D interrompe os cálculos antes de serem completados 5s de teste. No segundo gráfico é possível observar como o erro sobe bruscamente no momento da interrupção, excedendo os limites do gráfico, que foram propositalmente ajustados para evidenciar a comparação dos métodos que finalizaram a trajetória. Nota-se que o erro com utilização de *encoders* cresce relativamente constante e que a implementação com LiDAR apresentava mesma magnitude de erro da implementação que combina IMU, até o momento da curva mais brusca, pouco antes dos 10s de teste. Com isso, o método com apenas LiDAR acumula erros maiores durante mais da metade do teste, o que eleva sua integral do erro absoluto em comparação à IAE do LiDAR+IMU.

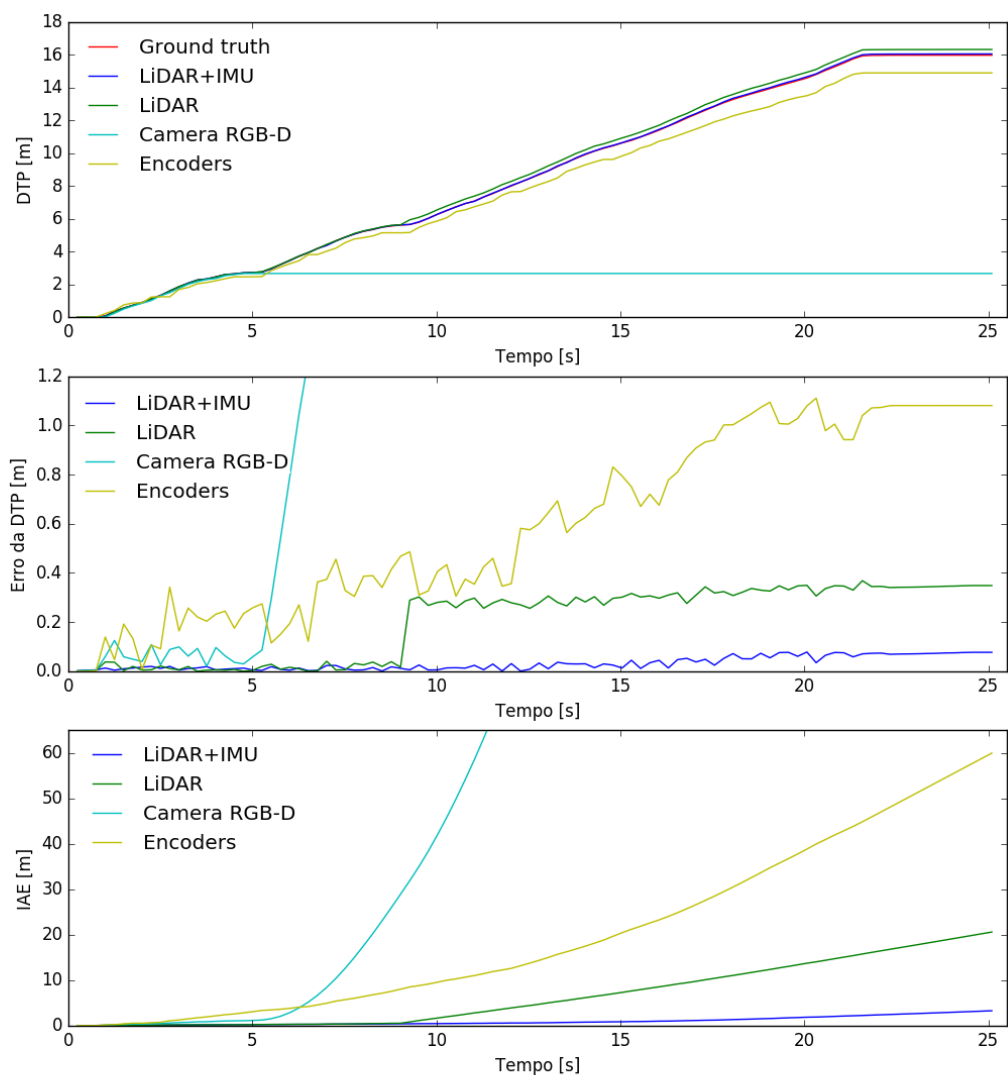


Figura 4.19: Teste 2 - Distância total percorrida, erros e integral do erro absoluto (IAE).

Com os dados apresentados na Tabela 4.10 é possível analisar o desempenho das implementações nos cálculos odométricos. A abordagem que mais se aproxima do *ground truth* é a LiDAR+IMU com o valor de 16,050m. O erro em relação à DTP computada pelo sistema de câmeras é de 7,7cm, o que representa apenas 0,48% de erro. Isso quer dizer que para cada metro percorrido pelo robô, o método em questão computa 0,48cm a mais, ou seja, o erro de odometria durante o teste foi, em média, menor que 0,5cm para cada metro navegado. Quando observado o erro relativo ao ponto de chegada do teste, a mesma implementação apresenta melhor desempenho entre as quatro com erro de 2,1cm ou 0,13% da distância total de fato percorrida (*ground truth*).

<i>Ground truth: 15.973m</i>	L+IMU	LiDAR	RGB-D	Encoder
Distância total percorrida (DTP) (m)	16,050	16,321	2,662	14,892
Erro de distância total percorrida (m)	0,077	0,349	13,311	1,081
% de erro em relação ao <i>ground truth</i>	0,48%	2,18%	83,33%	6,77%
Integral do erro absoluto (m)	3,258	20,594	626,388	59,990
Erro no ponto de chegada (m)	0,021	0,391	2,133	1,608
% de erro em relação ao <i>ground truth</i>	0,13%	2,45%	13,35%	10,06%

Tabela 4.10: Teste 2 - Tabela de erros de odometria.

4.4.1.3 Teste 3

Este teste teve duração de 34,9s entre o momento de acionamento dos algoritmos e o encerramento dos mesmos após realizada a trajetória. A distância total percorrida computada pelo sistema de captura de movimento utilizada como *ground truth* foi de 17,279m.

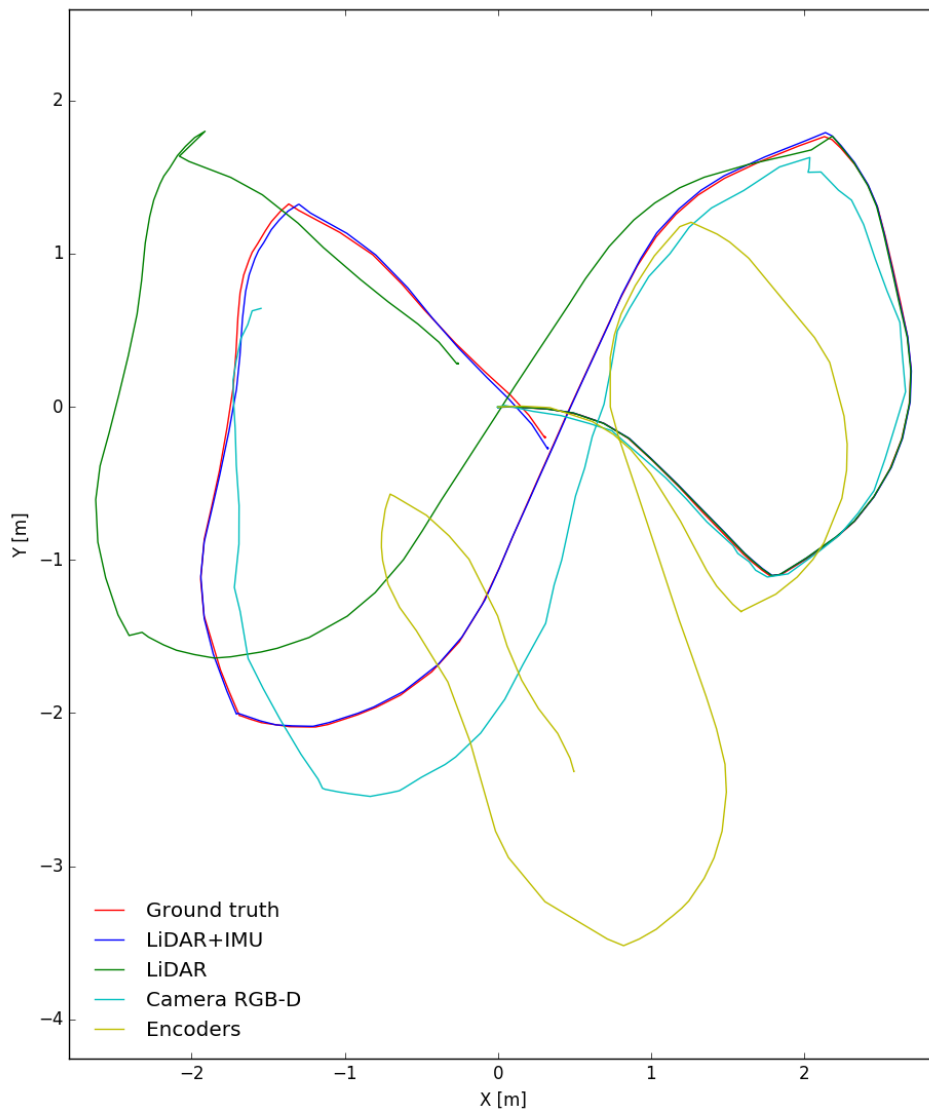


Figura 4.20: Teste 3 - Trajetória computada.

Analisando a Figura 4.20, outra vez é visível que o método com *encoders* calcula curvas mais fechadas do que as registradas pelo sistema de câmeras. Desde a primeira curva, o ângulo de curvatura observado é menor se comparada à curva do *ground truth*. Esse problema se torna ainda mais evidente quando observada a última curva da trajetória calculada com a leitura dos *encoders*.

Dessa vez, a implementação que usa imagens geradas pela câmera RGB-D conseguiu calcular mais da metade da trajetória realizada. Entretanto, não sobrevive novamente a uma curva brusca e interrompe os cálculos.

A abordagem que utiliza apenas LiDAR teve dificuldades em acompanhar o movimento realizado na curva observada no 1º quadrante. O erro causado nesta curva prejudicou o cálculo da trajetória a partir desse momento. Essa limitação se mostra mais uma vez na última curva, a trajetória calculada parece passar do ponto onde se inicia a curva, mas se recupera e corrige o movimento no sentido que a curva foi de fato realizada.

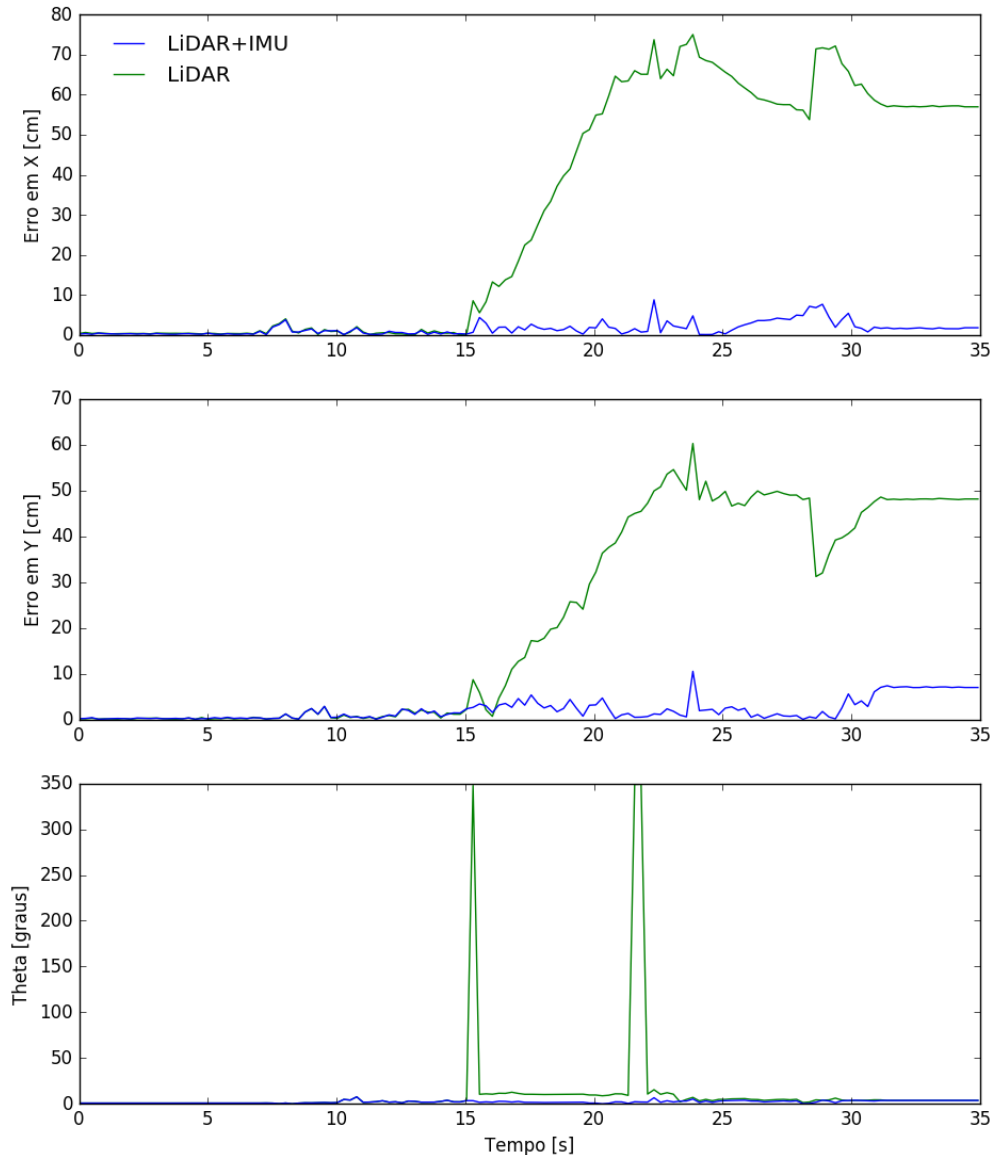


Figura 4.21: Teste 3 - Erros computados - LiDAR vs. LiDAR+IMU.

A relevância do uso da unidade de medição se reafirma dessa forma, pois, mais uma vez, a abordagem que combina este sensor com o LiDAR não foi afeta pelas curvas fechadas. Na Figura 4.21 são mostrados os erros entre as duas abordagens: com e sem IMU. Na Tabela (4.9) o desempenho da implementação LiDAR+IMU tem os menores valores de erro em todas as métricas calculas. Os erros médios computados no eixos x e y são de 1,6cm e 2,1cm, respectivamente.

x (m)				
	L+IMU	LiDAR	RGB-D	Encoder
Erro máximo	0,088	0,750	1,856	2,541
Erro médio	0,015	0,305	0,407	0,653
Desvio padrão (σ)	0,016	0,295	0,604	0,761
Integral do erro absoluto (IAE)	2,168	42,652	57,029	91,442

y (m)				
	L+IMU	LiDAR	RGB-D	Encoder
Erro máximo	0,105	0,603	0,847	2,242
Erro médio	0,021	0,222	0,295	0,937
Desvio padrão(σ)	0,023	0,221	0,311	0,859
Integral do erro absoluto (IAE)	2,930	31,101	41,276	131,158

θ (°)				
	L+IMU	LiDAR	RGB-D	Encoder
Erro máximo	7,7	349,2	353,4	396,0
Erro médio	2,2	11,7	32,2	120,7
Desvio padrão(σ)	1,4	50,2	62,8	95,1
Integral do erro absoluto (IAE)	302,5	1642,3	4514,2	16899,0

Tabela 4.11: Teste 3 - Tabela de erros na trajetória.

Na Figura 4.22 são comparados os desempenhos dos métodos implementados em relação ao cálculo da distância total percorrida (DTP). No primeiro gráfico mostrado na figura, observa-se que a abordagem com câmera RGB-D interrompe os cálculos próximo do fim do teste. No segundo gráfico é possível observar como esta abordagem apresenta desempenho semelhante à abordagem com LiDAR. No momento em os cálculos com câmera RGB-D são interrompidos, o erro aumenta e ultrapassa os limites do gráfico representado. No mesmo instante que este erro se eleva, também é registrado um pico no erro do LiDAR, que aumenta consideravelmente o erro da distância total percorrida. As integrais do erro absoluto (IAE) podem ser verificadas no terceiro gráfico da figura.

Com os dados da Tabela 4.12 pode-se analisar o desempenho das implementações nos cálculos odométricos. A abordagem que mais se aproxima do *ground truth* é a LiDAR+IMU com o valor de 17,423m. O erro em relação à DTP computada pelo sistema de

câmeras é de 14,4cm, o que representa apenas 0,83% de erro. Isso quer dizer que para cada metro percorrido pelo robô, o método em questão computa 0,48cm a mais, ou seja, o erro de odometria durante o teste foi, em média, menor que 1cm para cada metro navegado. Quando observado o erro relativo ao ponto de chegada do teste, a mesma implementação apresenta melhor desempenho entre as quatro com erro de 7,2cm ou 0,42% da distância total de fato percorrida (*ground truth*).

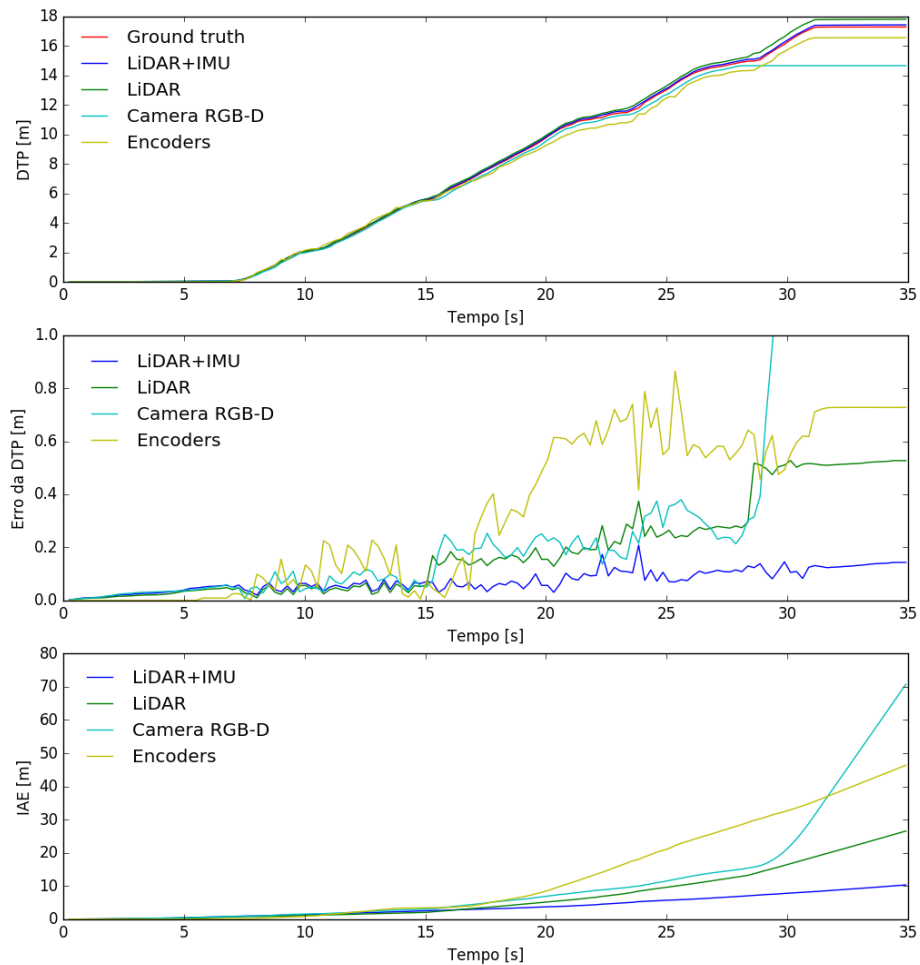


Figura 4.22: Teste 3 - Distância total percorrida, erros e integral do erro absoluto (IAE).

<i>Ground truth:</i> 17,279m	L+IMU	LiDAR	RGB-D	Encoder
Distância total percorrida (DTP) (m)	17,423	17,806	14,659	16,552
Erro de distância total percorrida (m)	0,144	0,527	2,621	0,727
% de erro em relação ao <i>ground truth</i>	0,83%	3,05%	15,17%	4,21%
Integral do erro absoluto (m)	10,319	26,559	70,741	46,375
Erro no ponto de chegada (m)	0,072	0,746	2,039	2,191
% de erro em relação ao <i>ground truth</i>	0,42%	4,32%	11,80%	12,68%

Tabela 4.12: Teste 3 - Tabela de erros de odometria.

4.4.2 Resultados experimentais com restrição de velocidade em 50% da capacidade máxima

Nessa subseção, são apresentados os resultados experimentais em testes nos quais há restrição de velocidade durante a teleoperação. Os comandos enviados são limitados proporcionalmente de forma que os motores sejam exigidos em no máximo 50% da sua capacidade. Os gráficos, tabelas e análises apresentados a seguir estão separados por teste.

4.4.2.1 Teste 4

Esse teste teve duração de 61,2s entre o momento de acionamento dos algoritmo e o encerramento dos mesmos após realizada a trajetória. A distância total percorrida computada pelo sistema de captura de movimento utilizada como *ground truth* foi de 17,279m. Na Figura 4.23 pode-se observar curvas um pouco menos fechadas, se comparadas com as dos testes sem restrição de velocidade.

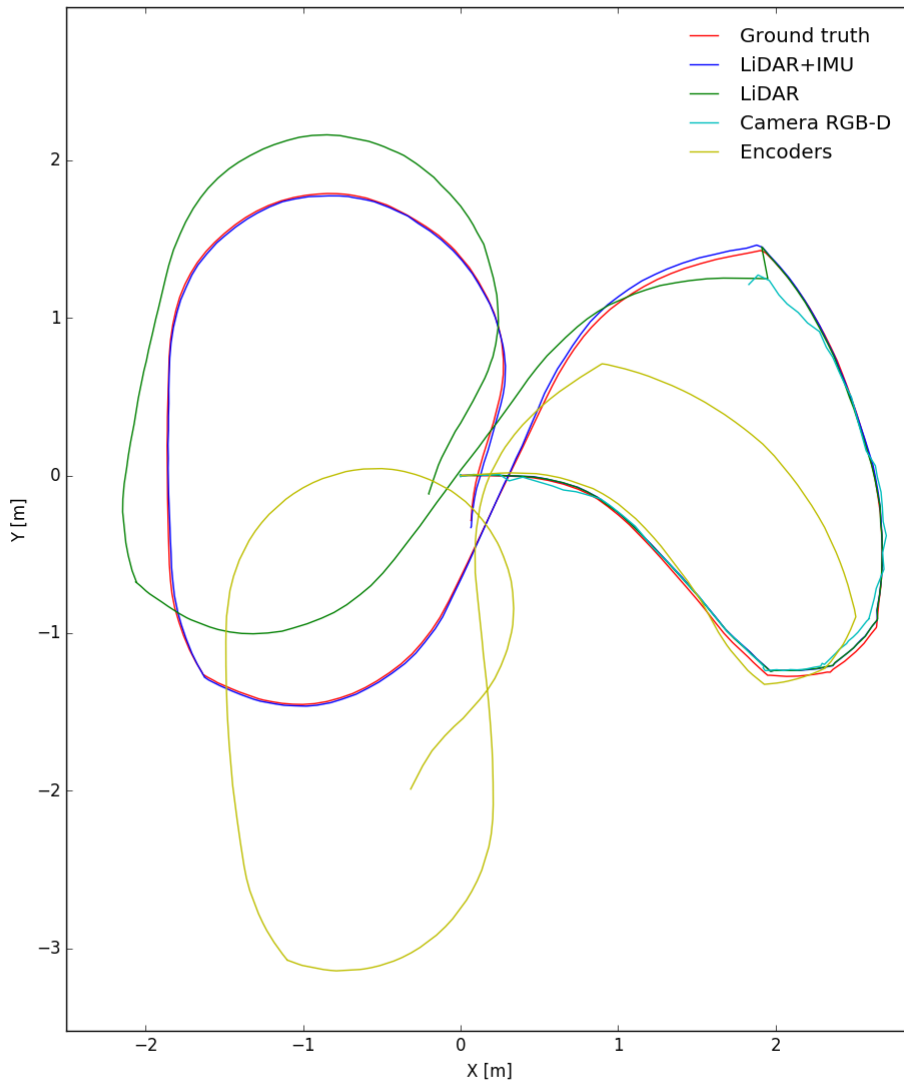


Figura 4.23: Teste 4 - Trajetória computada.

Apesar das curvas mais suaves, dois problemas persistiram: a implementação com *encoders* calculou curvas mais fechadas do que o movimento real do robô e a implementação com câmera RGB-D interrompeu os cálculos durante a curva que se encontra no 1º quadrante do plano. Esta mesma curva prejudicou o desempenho da abordagem com apenas LiDAR, o que causou maior acúmulo de erros no restante do teste. A implementação que combina LiDAR com IMU superou a curva e concluiu o teste sem maiores problemas. Os erros desta abordagem em comparação ao erros da abordagem que utiliza apenas LiDAR podem ser verificados na Figura 4.24, evidenciando assim a influencia do uso da unidade de medição inercial nos resultados obtidos. Também nesta figura é possível ver o momento em que a curva causa o erro do desenvolvimento com LiDAR: logo após os 25 segundos de teste os erros em x e y elevam-se consideravelmente.

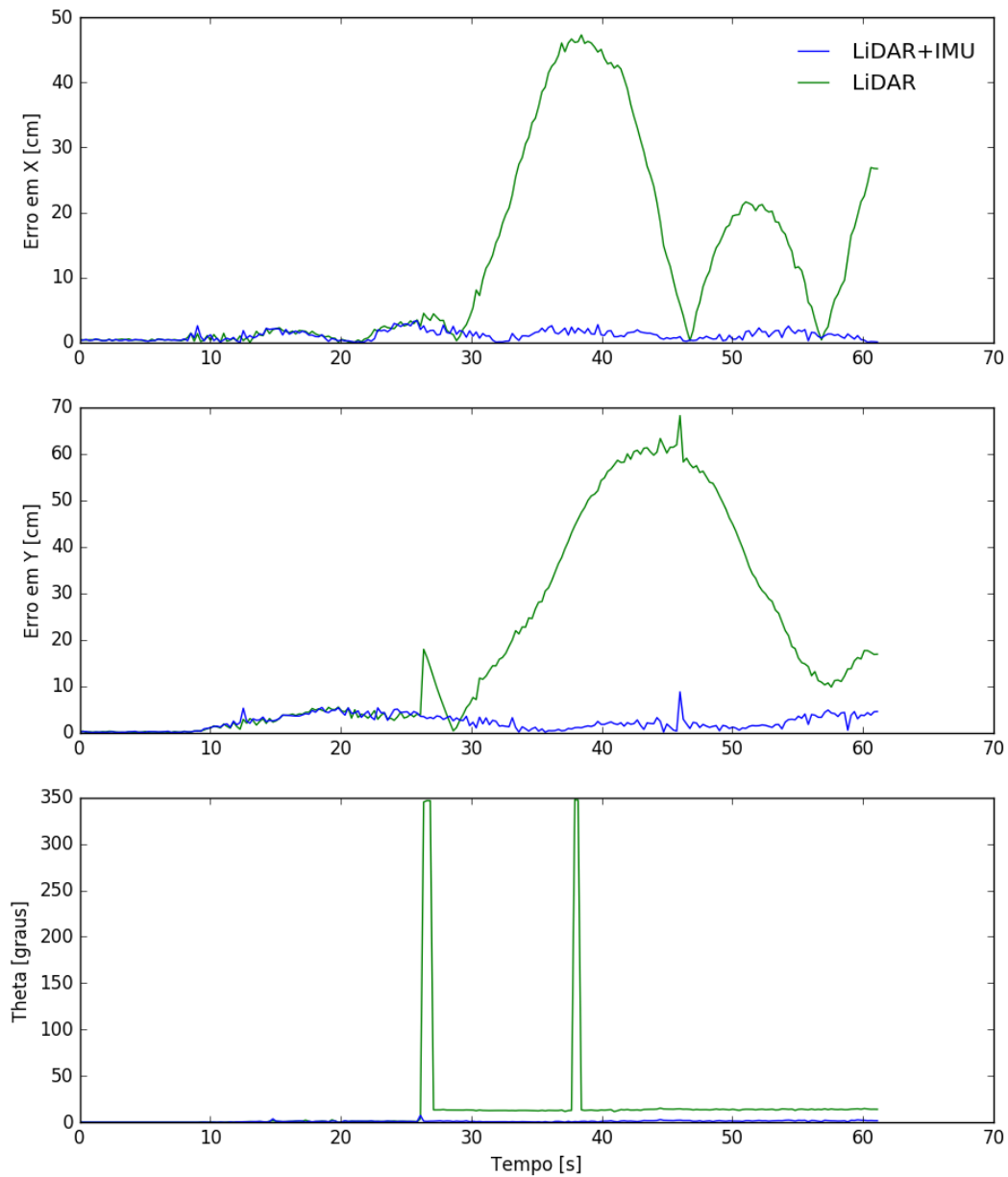


Figura 4.24: Teste 4 - Erros computados - LiDAR vs. LiDAR+IMU.

Na Tabela 4.13 são apresentados os erros de posição e orientação durante o movimento. A implementação que combina o sensor LiDAR com a unidade de medição inercial apresenta melhor desempenho em todos os quesitos apresentados. É interessante notar que o uso da IMU reduz em mais de 10x o erro médio em x (1,1cm contra 12,0cm) e em quase 10x em y (2,1cm contra 19,3cm). Em relação ao desvio padrão a diferença é ainda maior: aproximadamente 20x em x e 13x em y .

x (m)				
	L+IMU	LiDAR	RGB-D	Encoder
Erro máximo	0,033	0,473	3,692	1,063
Erro médio	0,011	0,120	1,333	0,319
Desvio padrão(σ)	0,007	0,146	1,368	0,278
Integral do erro absoluto (IAE)	2,639	29,310	325,274	77,879
y (m)				
	L+IMU	LiDAR	RGB-D	Encoder
Erro máximo	0,088	0,683	2,664	1,978
Erro médio	0,021	0,193	0,636	0,937
Desvio padrão(σ)	0,016	0,212	0,854	0,788
Integral do erro absoluto (IAE)	5,240	47,158	155,078	228,563
θ (°)				
	L+IMU	LiDAR	RGB-D	Encoder
Erro máximo	7,6	347,6	315,9	382,3
Erro médio	1,1	14,9	98,0	125,4
Desvio padrão(σ)	0,8	48,5	112,6	96,8
Integral do erro absoluto (IAE)	279,5	3640,2	23906,3	30596,6

Tabela 4.13: Teste 4 - Tabela de erros na trajetória.

No primeiro gráfico mostrado na Figura 4.25, observa-se que o método com câmera RGB-D interrompe os cálculos computando aproximadamente um terço da distância total percorrida. O segundo gráfico mostra o erro no cálculo da odometria. Novamente é possível perceber a influência da curva logo após os 25s de teste. Uma curiosidade é que mesmo errando mais que o LiDAR+IMU ao longo do teste, a abordagem com *encoders* termina o experimento com o menor erro no cálculo da DTP. O acúmulo de erros no decorrer do experimento, porém, é maior, se comparado ao LiDAR+IMU.

Como já destacado, a Tabela 4.14 confirma que a implementação com *encoders* apresenta o melhor resultado para este teste com apenas 0,08% de erro em relação ao *ground truth*. Neste caso, faz-se muito importante a análise do ponto de chegada, visto que apesar dos *encoders* oferecerem melhor cálculo de odometria, o ponto de chegada apresenta 10,11% de erro, ou seja, em um percurso de 17,279m, a implementação calculou um ponto

final 1,746m distante do ponto final correto. Logo, a distância total percorrida calculada com o método não tem efetividade, haja visto que calcula com precisão suficiente a DTP, mas não chega perto do ponto final desejado. Isso também pode ser concluído com a Figura 4.23. Por outro lado, a abordagem LiDAR+IMU erra 7,3cm a mais no cálculo de odometria (8,7cm contra 1,4cm), mas atinge 0,26% de erro no ponto de chegada.

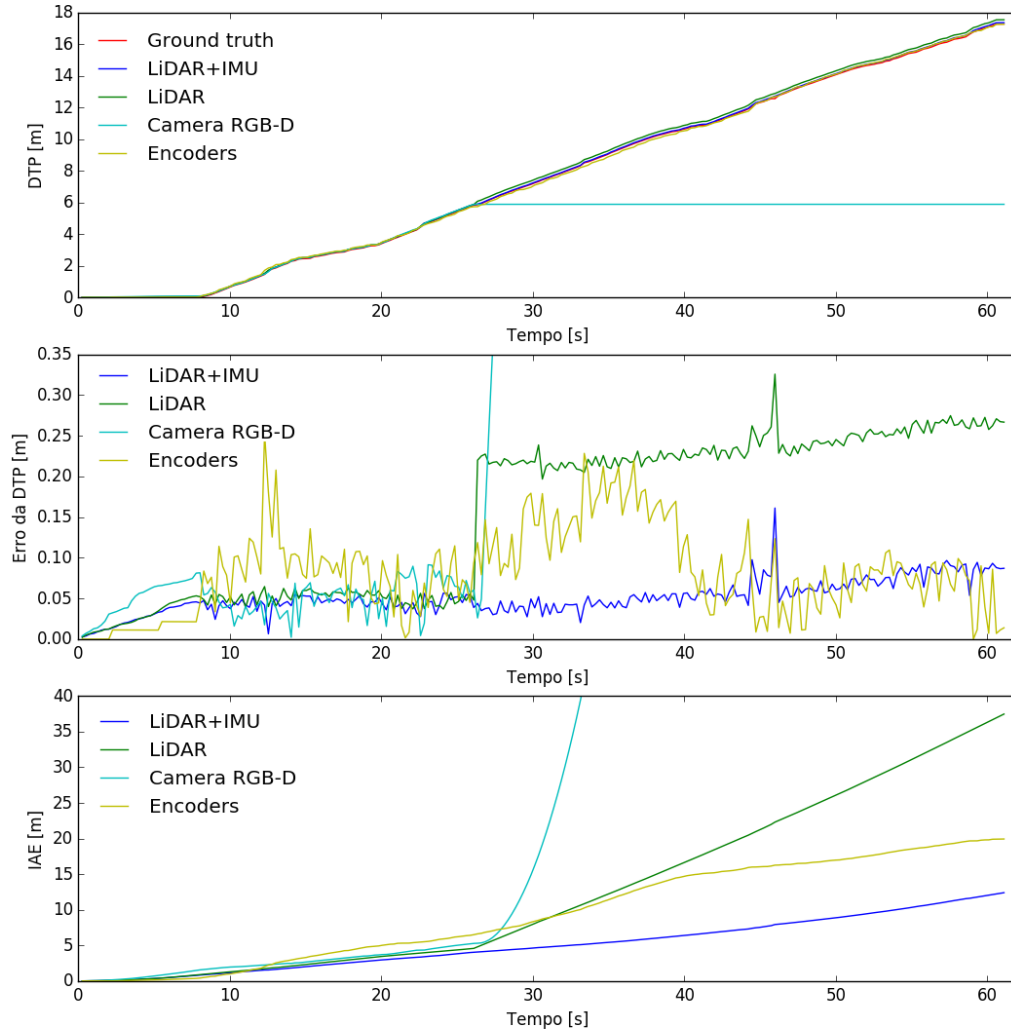


Figura 4.25: Teste 4 - Distância total percorrida, erros e integral do erro absoluto (IAE).

<i>Ground truth:</i> 17,279m	L+IMU	LiDAR	RGB-D	Encoder
Distância total percorrida (DTP) (m)	17,366	17,545	5,883	17,265
Erro de distância total percorrida (m)	0,087	0,267	11,396	0,014
% de erro em relação ao <i>ground truth</i>	0,50%	1,54%	65,95%	0,08%
Integral do erro absoluto (m)	12,396	37,484	829,732	19,934
Erro no ponto de chegada (m)	0,045	0,316	2,313	1,746
% de erro em relação ao <i>ground truth</i>	0,26%	1,83%	13,38%	10,11%

Tabela 4.14: Teste 4 - Tabela de erros de odometria.

4.4.2.2 Teste 5

Esse teste teve duração de 62,1s entre o momento de acionamento dos algoritmo e o encerramento dos mesmos após realizada a trajetória. A distância total percorrida computada pelo sistema de captura de movimento utilizada como *ground truth* foi de 15,605m.

A trajetória mostrada na Figura 4.26 apresenta curvas suaves em relação aos testes já aqui mostrados. Pela primeira vez o desenvolvimento usando câmera RGB-D conclui o teste sem interromper os cálculos de odometria. Nota-se que em geral as abordagens acompanham o traçado do *ground truth*, porém a abordagem com *encoders* se distancia, logo na primeira e, a com câmera RGB-D, logo na segunda curva. Ambas, porém, seguem próximas do formato da trajetória definido pelo sistema de validação.

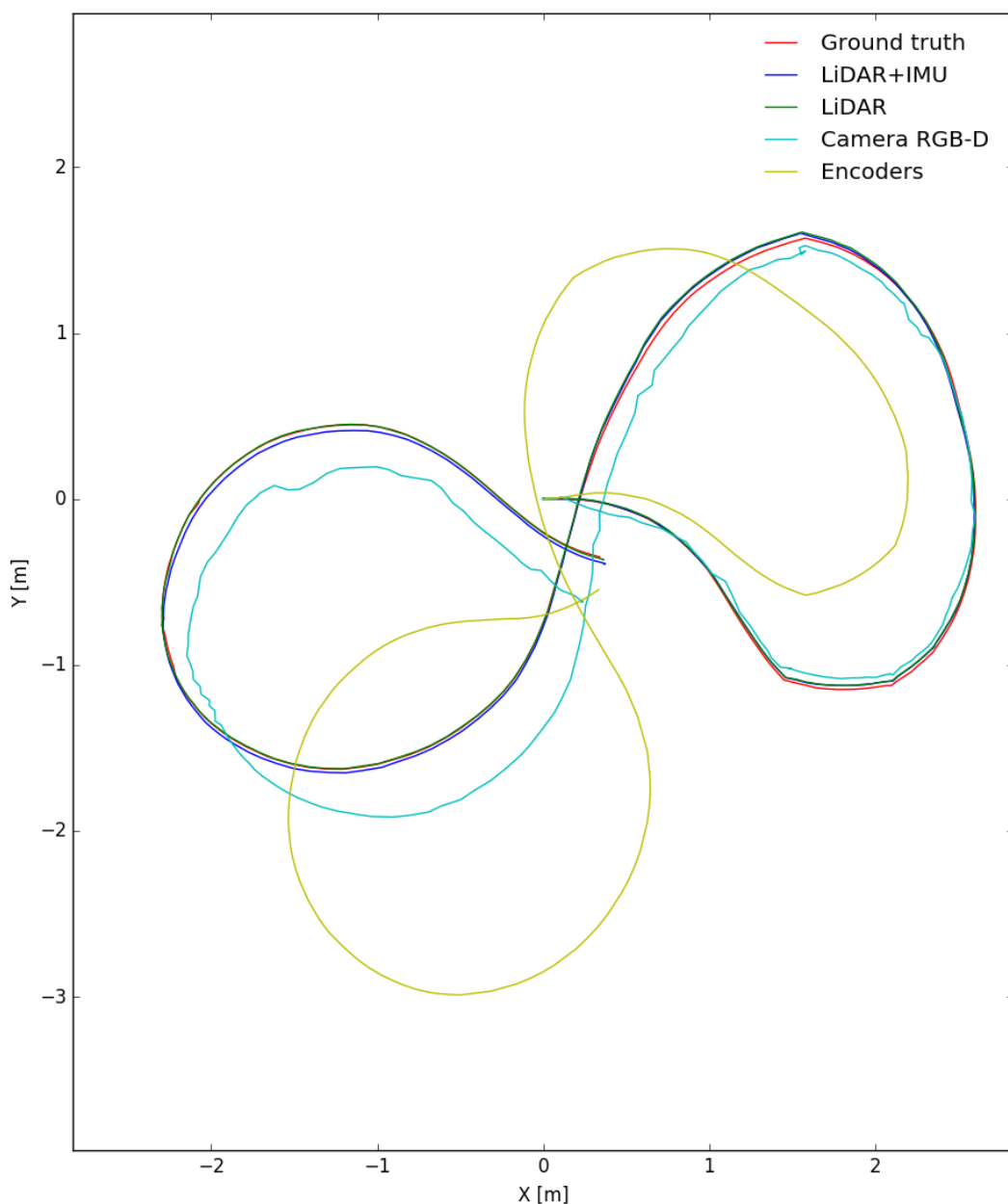


Figura 4.26: Teste 5 - Trajetória computada.

A diferença entre as outras duas abordagens apenas é notada na Figura 4.27. Por meio dessa figura, pode-se concluir que a abordagem que utiliza apenas o LiDAR supera a que faz uso do LiDAR em conjunto com a unidade de medição inercial no que diz respeito à posição, mas mantém desempenho abaixo se comparada em relação à orientação.

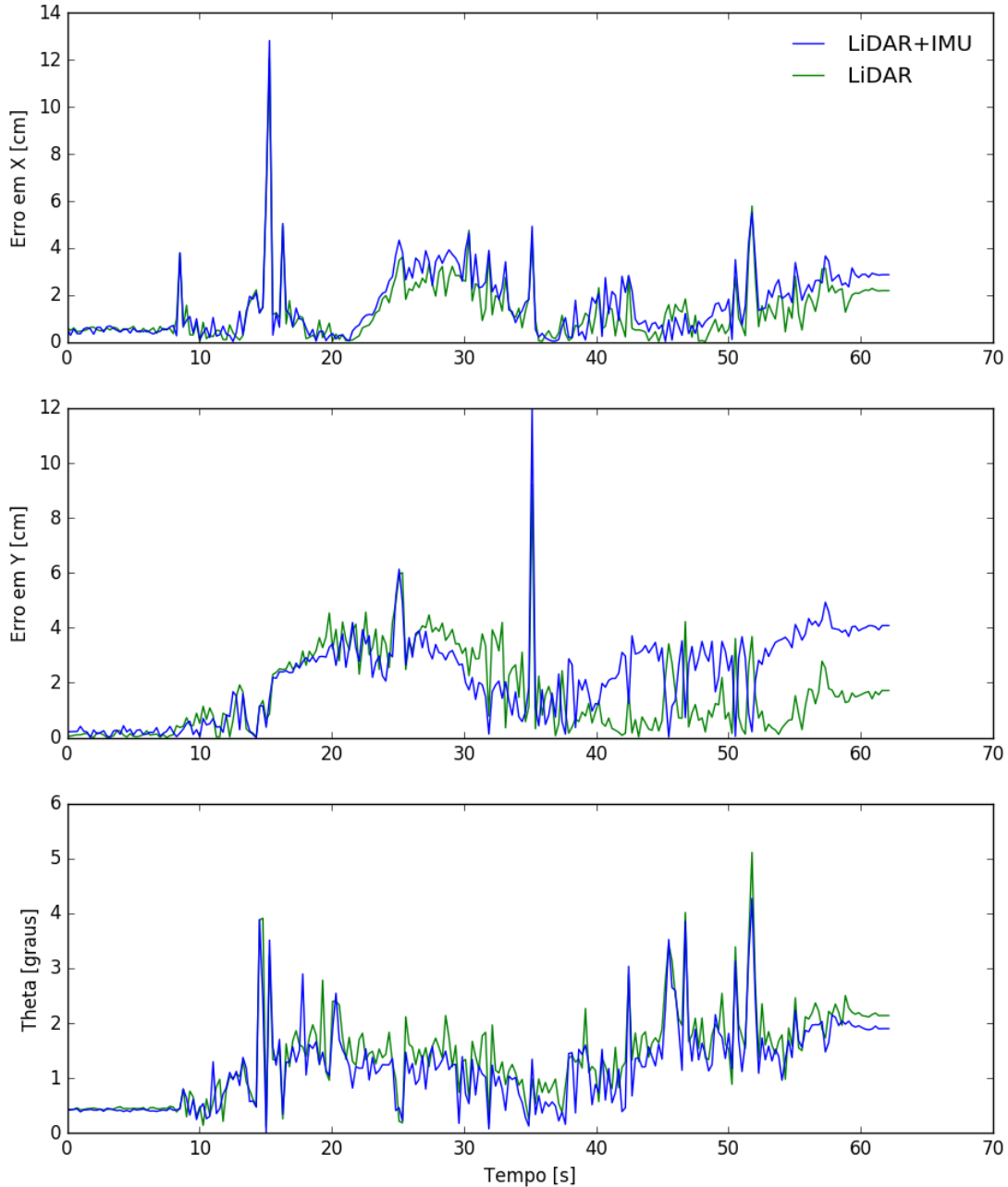


Figura 4.27: Teste 5 - Erros computados - LiDAR vs. LiDAR+IMU.

Essa diferença pode ser analisada de forma mais precisa na Tabela 4.15. Nela observa-se que apesar da diferença entre os métodos existir, a magnitude da mesma não apresenta valores significativos. Nos erros em x a diferença é de 0,7cm, 0,3cm e 0,1cm para erro máximo, médio e desvio padrão, respectivamente. Em y o erro máximo traz diferença de

2,8cm entre as abordagens, enquanto o erro médio diverge em apenas 0,6cm e o desvio padrão empata. Considerando as integrais do erro absoluto a diferença é maior, visto que representa a soma de todos os erros calculados para cada eixo. Em se tratando da orientação, o IMU influencia mais uma vez positivamente, e a abordagem que o utiliza tem ligeira vantagem sobre a que usa apenas o LiDAR.

x (m)				
	L+IMU	LiDAR	RGB-D	Encoder
Erro máximo	0,128	0,121	0,416	1,445
Erro médio	0,016	0,013	0,099	0,495
Desvio padrão (σ)	0,014	0,013	0,113	0,517
Integral do erro absoluto (IAE)	3,996	3,248	24,619	122,713

y (m)				
	L+IMU	LiDAR	RGB-D	Encoder
Erro máximo	0,120	0,092	0,426	1,940
Erro médio	0,022	0,016	0,138	0,711
Desvio padrão(σ)	0,015	0,015	0,130	0,594
Integral do erro absoluto (IAE)	5,416	4,055	34,169	176,323

θ (°)				
	L+IMU	LiDAR	RGB-D	Encoder
Erro máximo	4,3	5,1	354,3	394,3
Erro médio	1,2	1,4	4,5	125,3
Desvio padrão(σ)	0,7	0,8	22,4	111,9
Integral do erro absoluto (IAE)	297,4	345,2	1104,9	31078,8

Tabela 4.15: Teste 5 - Tabela de erros na trajetória.

Nos gráficos apresentados na Figura 4.28 observa-se como os desenvolvimentos com *encoders* destoam das outras abordagens no cálculo da distância total percorrida neste teste. É interessante notar que mesmo com erros de posição e orientação superiores às abordagens LiDAR e LiDAR+IMU, a implementação com câmera RGB-D acumula menos erros até os últimos instantes do teste, quando tem um pico e passa a errar mais ao final do experimento. Com isso, mesmo apresentando menor acurácia no cálculo da DTP, a IAE da câmera RGB-D é a menor entre as abordagens.

Na Tabela 4.16 é possível constatar-se isso, além de verificar que as abordagens com LiDAR praticamente empatam com apenas 0,1cm de diferença no cálculo de odometria. O erro no ponto de chegada também é muito parecido: 0,32% do LiDAR+IMU contra 0,18% do LiDAR.

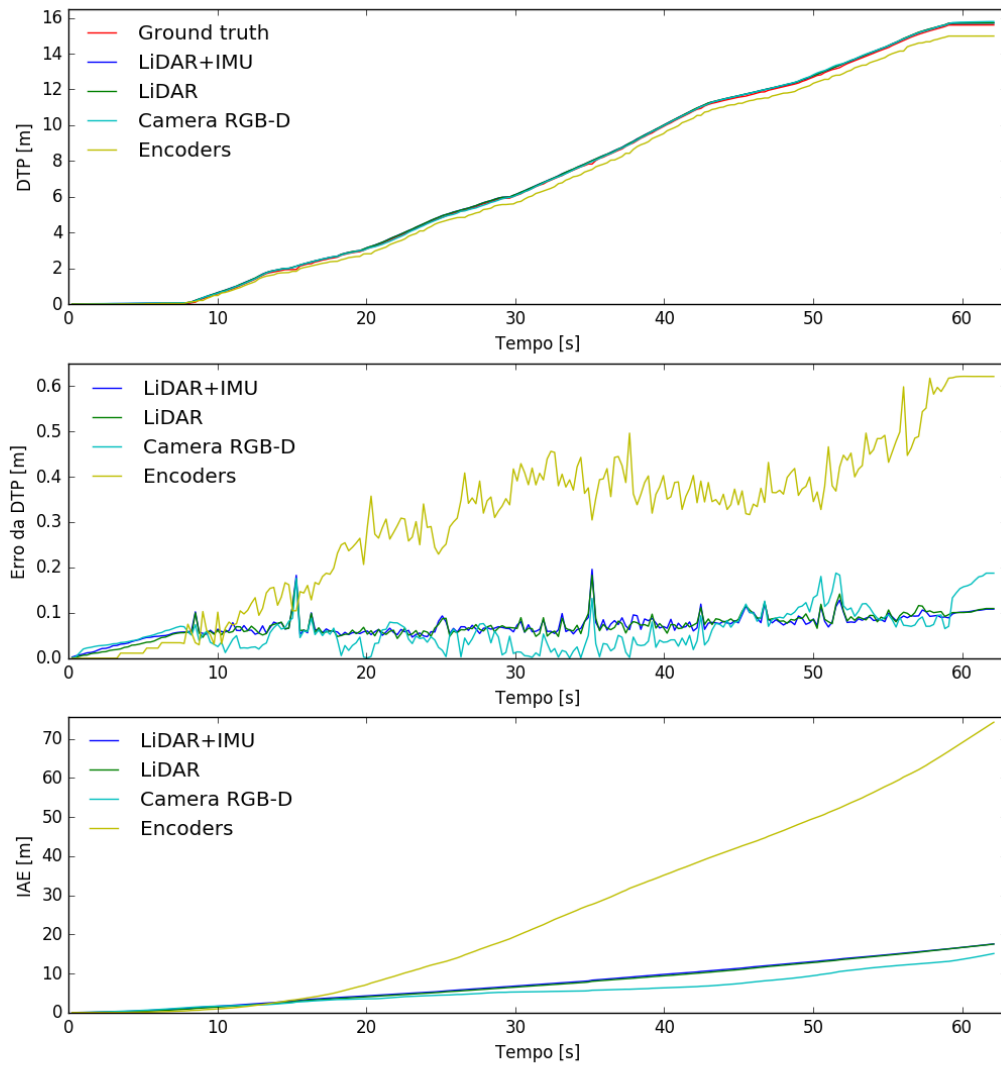


Figura 4.28: Teste 5 - Distância total percorrida, erros e integral do erro absoluto (IAE).

<i>Ground truth:</i> 15,605m	L+IMU	LiDAR	RGB-D	Encoder
Distância total percorrida (DTP) (m)	15,714	15,715	15,793	14,985
Erro de distância total percorrida (m)	0,108	0,110	0,188	0,621
% de erro em relação ao <i>ground truth</i>	0,69%	0,70%	1,20%	3,98%
Integral do erro absoluto (m)	17,556	17,511	15,093	74,261
Erro no ponto de chegada (m)	0,050	0,028	0,290	0,198
% de erro em relação ao <i>ground truth</i>	0,32%	0,18%	1,86%	1,27%

Tabela 4.16: Teste 5 - Tabela de erros de odometria.

4.4.2.3 Teste 6

Esse teste teve duração de 64,7s entre o momento de acionamento dos algoritmo e o encerramento dos mesmos após realizada a trajetória. A distância total percorrida computada pelo sistema de captura de movimento utilizada como *ground truth* foi de 15,447m.

No Teste 6, novamente a abordagem com câmera RGB-D conclui os cálculos da trajetória (Figura 4.29). De forma semelhante ao teste anterior, mesmo com curvas mais suaves em relação aos outros testes, o desenvolvimento com *encoders* tem dificuldades em reproduzir estas curvas traçadas pelo robô. Dessa forma, a implementação com câmera RGB-D supera a com *encoders* e as com LiDAR superam ambas. A diferença entre as curvas azul e verde é imperceptível na Figura 4.29, mas pode ser primeiramente observada na Figura 4.30.

Na Tabela 4.17 são apresentados os erros entre as abordagens de melhor desempenho, destacando-se mais uma vez que estas têm mesma magnitude. Mesmo assim, verifica-se que o LiDAR+IMU acumula menor erro em todos os quesitos. É relevante destacar que mesmo completando a trajetória, o uso da câmera RGB-D acumula 10x, 5x e 7x mais erros no cálculo de posição x , y e orientação θ , se comparado à implementação que combina LiDAR com IMU.

x (m)				
	L+IMU	LiDAR	RGB-D	Encoder
Erro máximo	0,034	0,044	0,290	2,762
Erro médio	0,011	0,016	0,113	0,821
Desvio padrão (σ)	0,009	0,011	0,078	0,846
Integral do erro absoluto (IAE)	2,743	4,006	29,061	211,758
y (m)				
	L+IMU	LiDAR	RGB-D	Encoder
Erro máximo	0,090	0,079	0,521	1,888
Erro médio	0,026	0,028	0,139	0,541
Desvio padrão(σ)	0,023	0,022	0,155	0,541
Integral do erro absoluto (IAE)	6,627	7,288	35,850	139,544
θ (°)				
	L+IMU	LiDAR	RGB-D	Encoder
Erro máximo	4,0	3,7	355,7	405,3
Erro médio	1,1	1,2	7,2	127,4
Desvio padrão(σ)	0,7	0,7	37,7	107,2
Integral do erro absoluto (IAE)	275,5	298,2	1863,4	32862,8

Tabela 4.17: Teste 6 - Tabela de erros na trajetória.

Em relação ao cálculo de odometria, a Figura 4.31 possibilita a análise de que a

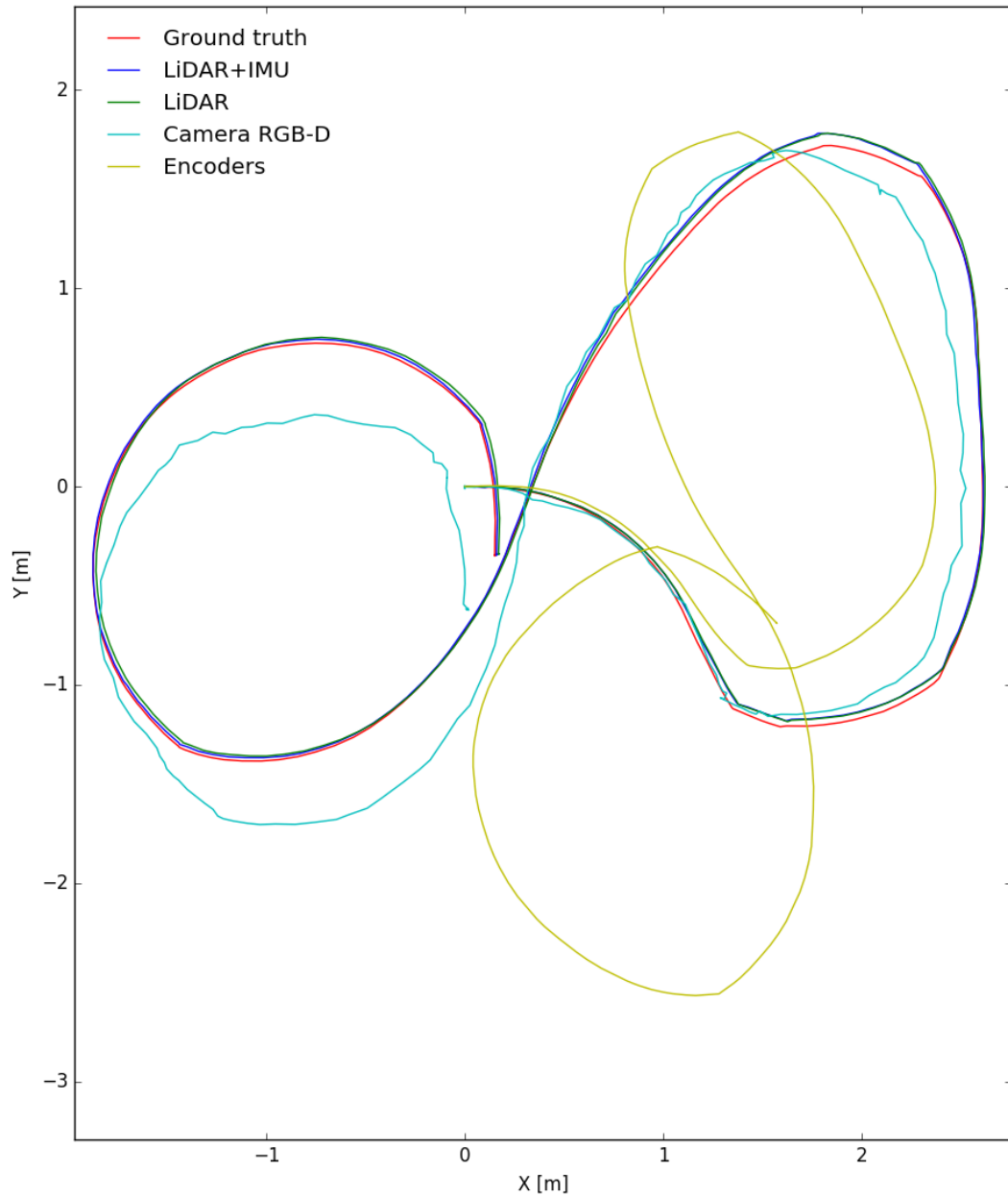


Figura 4.29: Teste 6 - Trajetória computada.

implementação com *encoders* destoa das demais. Além disso, a abordagem com câmera RGB-D varia mais o erro ao longo do teste, causando maior acúmulo, se comparado ao acumulado pelas abordagens com LiDAR.

A diferença entre estas implementações somente pode ser analisada na Tabela 4.18, na qual é possível observar que o LiDAR possui menor erro no cálculo de distância total percorrida, porém o LiDAR+IMU obtém maior precisão em relação ao ponto de chegada. De toda forma, a diferença que define os melhores desempenhos é pequena: 1,6cm na DTP e 1,2cm no ponto de chegada. Em resumo, as duas abordagens com LiDAR atingem erros menores que 1cm por metro percorrido e acurácia de pelo menos 99,88% no ponto de chegada.

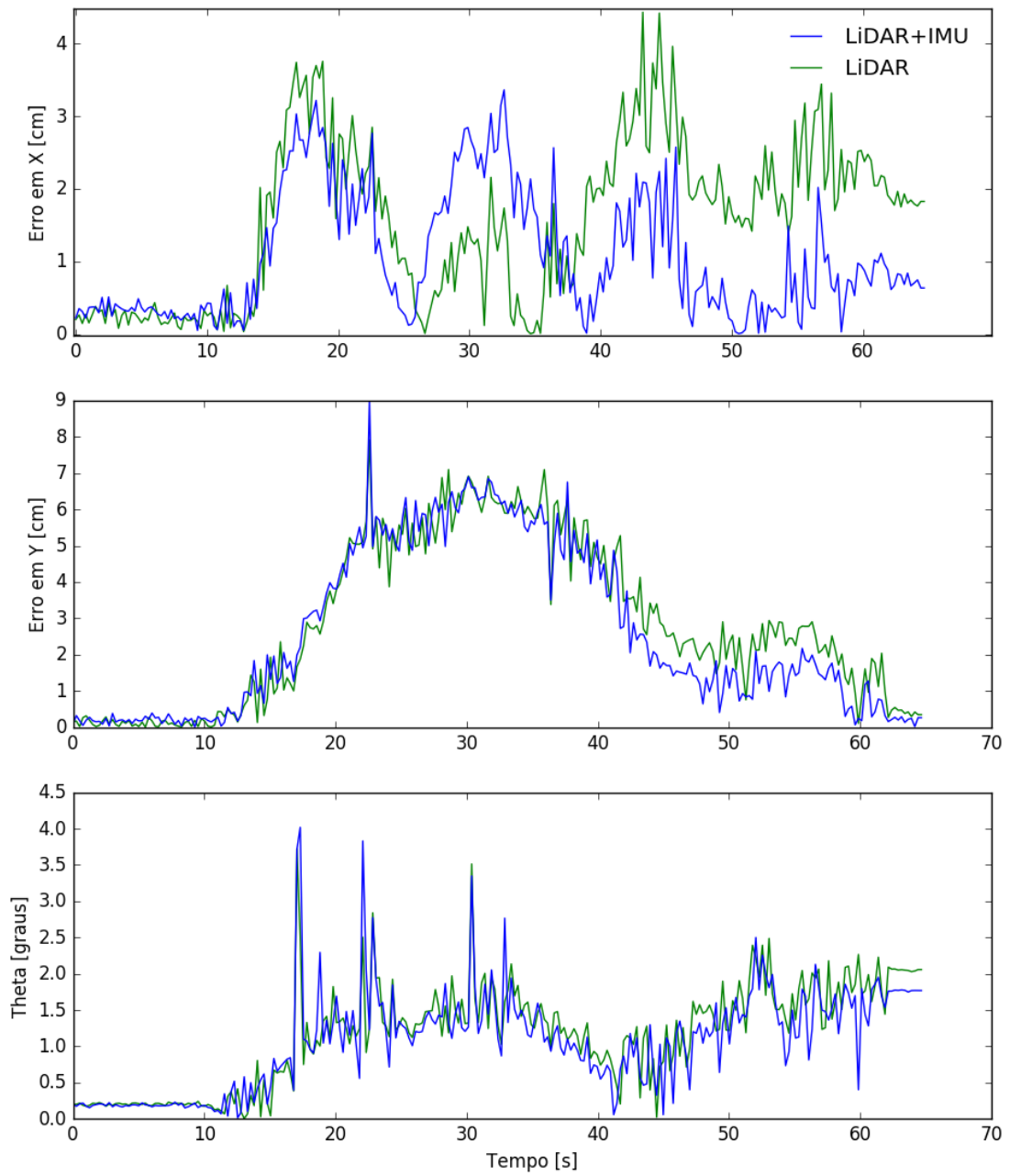


Figura 4.30: Teste 6 - Erros computados - LiDAR vs. LiDAR+IMU.

<i>Ground truth:</i> 15,447m	L+IMU	LiDAR	RGB-D	Encoder
Distância total percorrida (DTP) (m)	15,580	15,564	15,637	14,595
Erro de distância total percorrida (m)	0,132	0,116	0,190	0,853
% de erro em relação ao <i>ground truth</i>	0,86%	0,75%	1,23%	5,52%
Integral do erro absoluto (m)	19,620	19,030	23,415	83,487
Erro no ponto de chegada (m)	0,007	0,019	0,309	1,462
% de erro em relação ao <i>ground truth</i>	0,04%	0,12%	2,00%	9,46%

Tabela 4.18: Teste 6 - Tabela de erros de odometria.

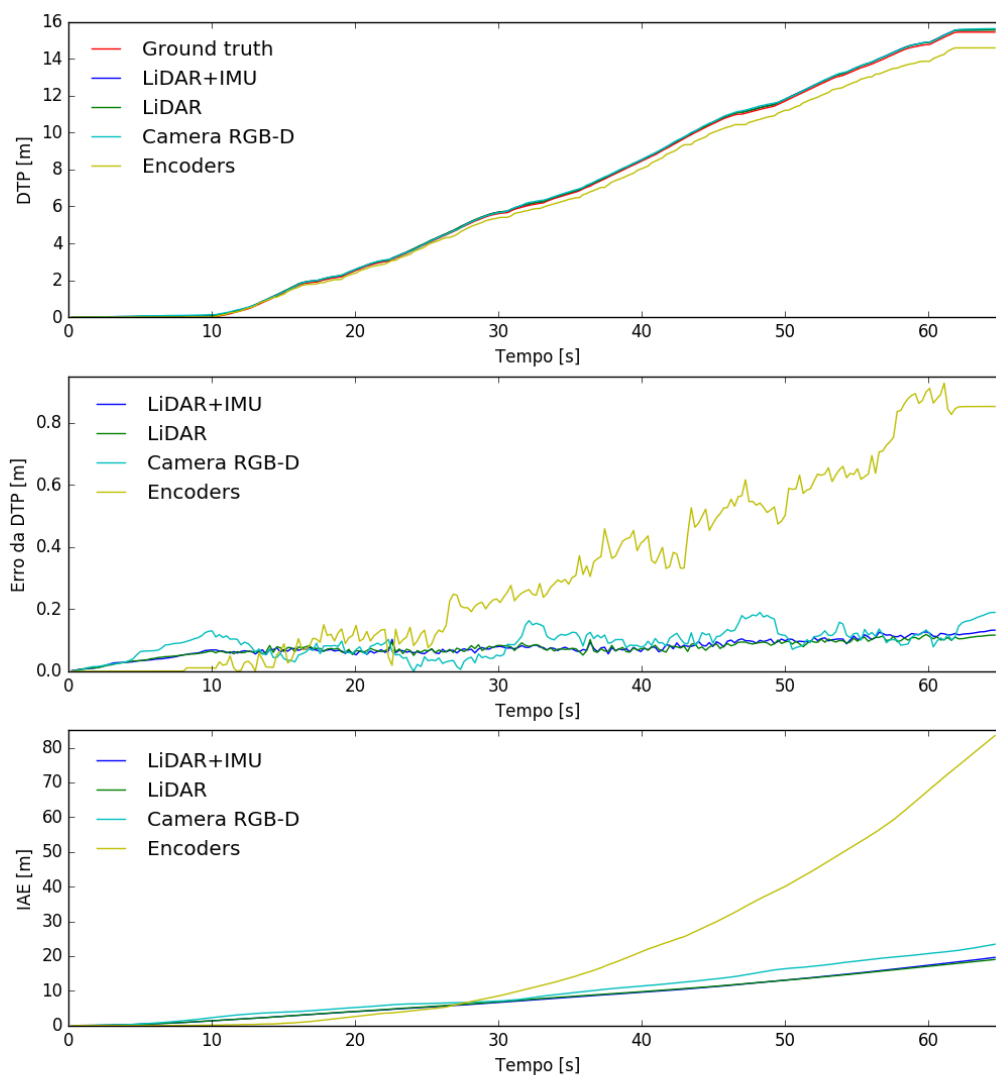


Figura 4.31: Teste 6 - Distância total percorrida, erros e integral do erro absoluto (IAE).

4.4.3 Análise geral

Os testes realizados geraram informações para avaliar o desempenho das técnicas de odometria aplicadas em cada teste. Com as informações levantadas é possível analisar de forma geral o desempenho das implementações. Na Figura 4.32 é possível verificar que o método com LiDAR+IMU apresenta, em todos os testes, precisão superior a 99% no cálculo da distância total percorrida pelo robô. As abordagens que fazem uso do LiDAR (sem IMU) e *encoders* atingem pelo menos 95% e 93% de precisão, respectivamente. A implementação com câmera RGB-D apresenta a maior variação na precisão da odometria. Isso se deve basicamente às interrupções de cálculo que impediram o método de computar as trajetória descritas por inteiro.

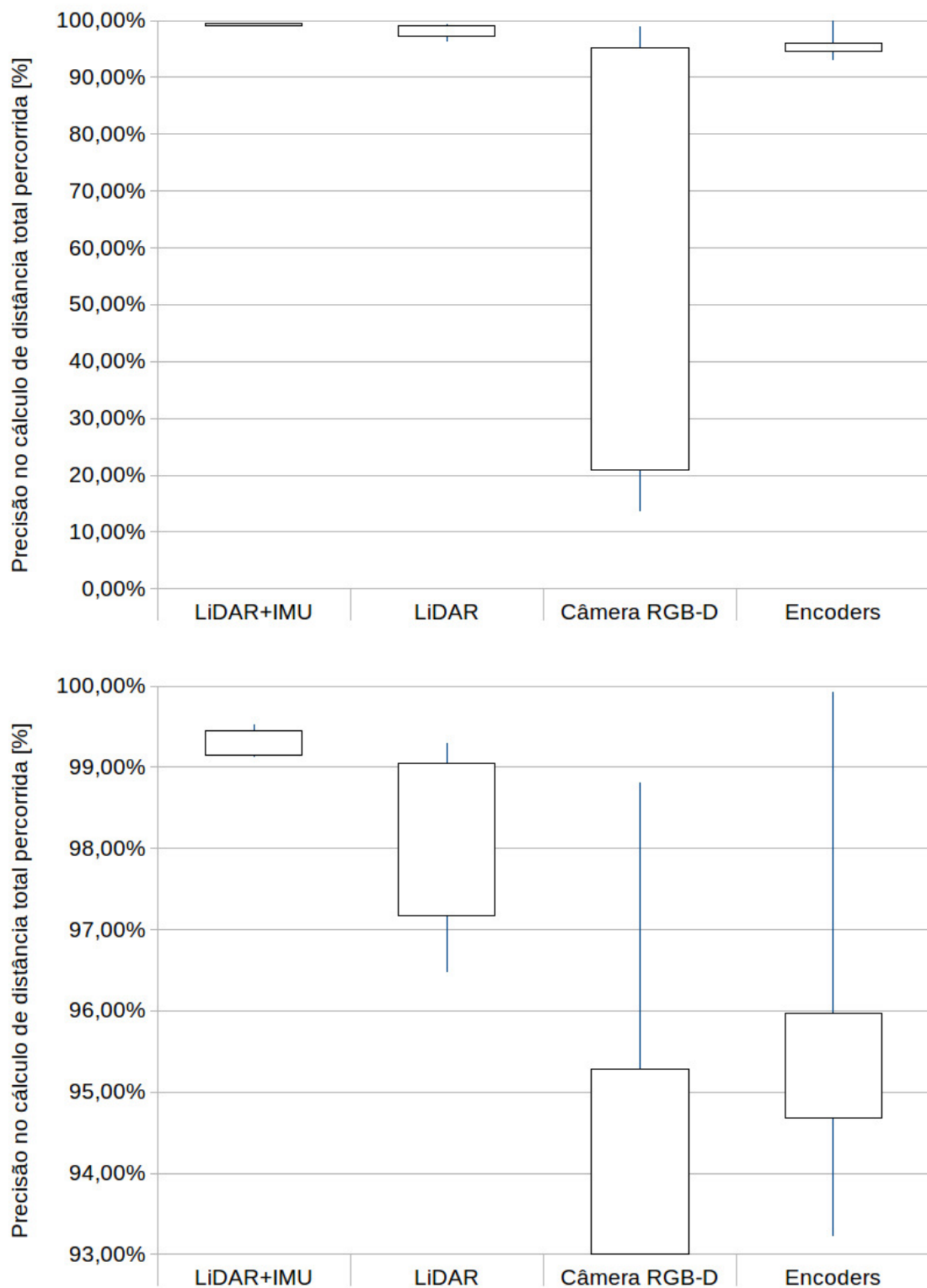


Figura 4.32: Comparação da precisão do cálculo odométrico feito por cada método (acima), mesmo gráfico porém ajustado para melhor visualização (abaixo).

A precisão no ponto de chegada (Figura 4.33) complementa a métrica da precisão da odometria avaliando se a distância calculada ainda levou ao destino correto. O método que combina LiDAR e IMU apresenta novamente precisão superior a 99% para todos os testes

realizados. O desenvolvimento com LiDAR apresenta valores superiores a 95%, enquanto os *encoders* proporcionam precisão de 80%. Apesar da câmera RGB-D apresentar valores entre aproximadamente 86 e 98%, é necessário destacar que as interrupções logo no início dos testes acabaram contribuindo para que o ponto de chegada encontrado fosse próximo do real. Por isso, se faz necessária a avaliação em conjunto das duas métricas apresentadas.

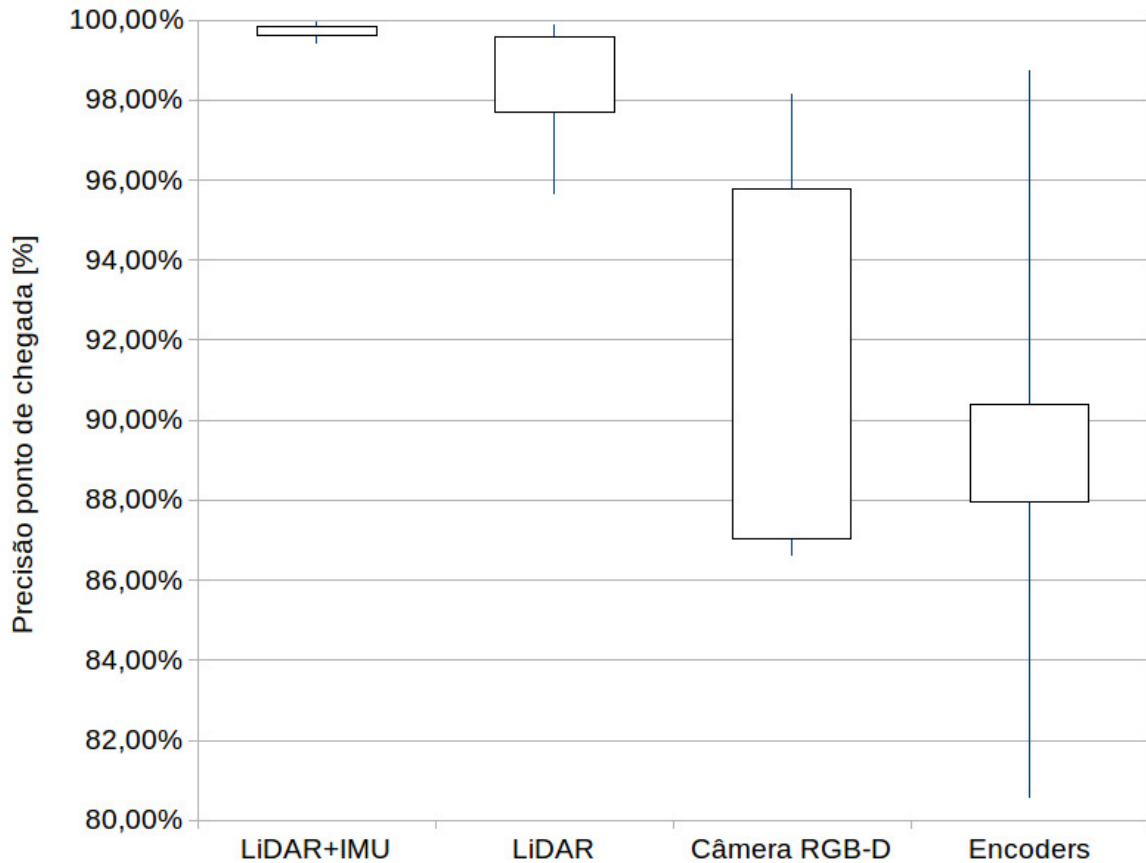


Figura 4.33: Precisão no ponto de chegada do robô após finalizado cada teste.

4.4.4 Considerações sobre implementação com *encoders*

A maior dificuldade encontrada pela abordagem implementada com *encoders* foi acompanhar corretamente os movimentos curvos descritos pelo robô. Como já vastamente discutido na literatura, os erros computados durante esse movimentos se devem ao escorregamento das rodas: as rodas escorregam, o sensor faz a leitura do giro, o método implementado computa o movimento, mas ele não é de fato realizado. O problema descrito é notado de forma ainda mais clara em curvas bruscas ou giros em torno do próprio eixo.

Apesar dessa dificuldade, o método ainda conseguiu proporcionar precisão superior a 93% no cálculo da odometria em todos os testes realizados.

4.4.5 Considerações sobre implementação com câmera RGB-D

O método que utiliza as imagens geradas pela câmera RGB-D para calcular a odometria completou um terço dos testes realizados. Nos outros testes houve a interrupção dos cálculos causando grandes erros nos quesitos avaliados. Primeiramente foram levantadas duas hipóteses:

- determinados movimentos são tão rápidos que não há características marcantes suficientes entre dois *frames* para que seja calculada a transformada;
- falta de capacidade de processamento para realizar os cálculo.

Ambas hipóteses são plausíveis, mas mesmo nos testes com restrição de velocidade ainda houve interrupção (Teste 4, Sub-subseção 4.4.2.1), o que indicou maior possibilidade de falha no processamento. Dessa forma, foi realizada investigação mais aprofundada nessa hipótese. Após iniciada pesquisa, outra hipótese não antes mencionada é que o alcance do sensor não foi suficiente. De acordo com Labbe (2018), para sensores como a câmera RGB-D, além de o algoritmo de odometria do pacote *rtabmap_ros* necessitar de pelo menos 100 características marcantes por imagem, é necessário que elas se localizem entre 0,4 e 4m.

Segundo Aqel *et al.* (2016) *apud* Gonzalez *et al.* (2013); Nagatani *et al.* (2010); Nourani-Vatani e Borges (2011); Yu *et al.* (2011), os principais desafios de sistemas de odometria visual são relacionados com custo computacional, iluminação e condição de imagem. Como os testes foram realizados com iluminação adequada, a câmera RGB-D fornece imagens com qualidade e é amplamente usado na comunidade robótica, isso é mais um indício que faltou capacidade de processamento.

Ademais, Lhotský (2016) afirma que o minicomputador Intel NUC (mesmo modelo do EspeleoRobô) é capaz de processar o RTAP-map, porém com faixa de carga de processamento da CPU (Unidade Central de Processamento) em 95-100%. Apesar do autor estar referindo-se ao processo completo de mapeamento, é válido destacar que o método usado pelo presente trabalho também utiliza mapas locais para calcular a odometria visual.

Apesar de não ser conclusivo, pode-se inferir que a causa da interrupção dos cálculos está relacionada com a complexidade do algoritmo implementado para extrair características das imagens e calcular a odometria do robô.

4.4.6 Considerações sobre implementação com sensor LiDAR e LiDAR+IMU

Os métodos que utilizaram sensor LiDAR nos desenvolvimentos apresentaram resultados de alta precisão e todos os quesitos avaliados. A utilização da unidade de medição

inercial para se extrair a rotação do robô influenciou na melhora da capacidade da implementação LiDAR+IMU em descrever movimentos curvos bruscos. Consequentemente, a IMU melhorou os resultados gerais em relação ao cálculo de distância total percorrida.

Apenas no Teste 5, a abordagem sem IMU apresenta melhores resultados no cálculo de posição (x, y) . Nessa ocasião, a adição da IMU não acarretou melhoria no cálculo de posição, apesar de melhorar os resultados de orientação. Essa diferença de resultado apenas é possível, visto que a adição da IMU é feita sem o procedimento de fusão de dados. É válido destacar ainda que as diferenças notadas no erro médio são de apenas 0,3cm e 0,6cm, em x e y , respectivamente.

Capítulo 5

Conclusões

Nesta dissertação foi apresentada a implementação e avaliação de técnicas de odometria para robôs móveis. Especificamente, as técnicas mostradas foram desenvolvidas utilizando a plataforma do EspeleoRobô, um dispositivo robótico móvel terrestre que possui múltiplos mecanismos e modos de locomoção. Testes foram executados em laboratório e a validação dos resultados foi realizada utilizando um sistema de captura de movimento com câmeras Vicon.

É possível concluir que a abordagem de odometria com leitura da velocidade dos eixos do EspeleoRobô por meio de *encoders* traz imprecisões e acúmulo de erros com o tempo. A maior parte dos erros são computados em movimentos curvos, quando há o escorregamento das rodas.

A abordagem com câmera RGB-D não trouxe resultados praticáveis, visto que em movimentos rápidos e até mesmo em movimentos lentos, a referência é perdida e os cálculos interrompidos. Isso se deve provavelmente ao alto processamento demandado pelo algoritmo, causando o não término dos cálculos em determinado ponto. Apesar de ainda não ser possível afirmar com absoluta certeza a causa das interrupções, o desenvolvimento desta dissertação leva a concluir que esta técnica não é uma opção viável para os objetivos propostos pelo projeto do EspeleoRobô. Isso se deve ao fato de que a plataforma robótica possui inúmeras funcionalidades, e o processamento de imagens para o cálculo de odometria contribui para a diminuição da capacidade de processamento do robô. Além disso, as limitações do sensor como o alcance de até 4m e a necessidade de boa iluminação também podem ser um empecilho, diminuindo, no mínimo, a versatilidade da técnica para diversos ambientes de operação.

A implementação utilizando somente sensor LiDAR mostrou dificuldades em movimentos bruscos e rotações em torno do próprio eixo. Enquanto a implementação que une esse sensor com uma IMU apresentou resultados positivos consistentes em todas as situações testadas. Não houve em nenhum teste erros significativos na trajetória ou no ponto de chegada calculados e a precisão destes cálculos é superior a 99% em todos os testes realizados.

A odometria *online* calculada com essa última implementação pode ser testada para atividades realizadas com o EspeleoRobô em dutos, auxiliando a operação com informações de distância total percorrida. Algumas dificuldades são esperadas como, por exemplo, a uniformidade das estruturas dos dutos. De toda forma, é necessário observar o comportamento do método e, caso apresente dificuldades, seja investigada a causa e apresentada solução.

O método também se mostrou promissor para ser testado em atividades de inspeção e monitoramento de ambientes estruturados confinados, de difícil acesso, como galerias de barragem. Do mesmo desenvolvimento, pode-se também determinar a localização do robô relativa ao ponto inicial da operação.

5.1 Contribuições da dissertação

A seguir estão descritas as contribuições do trabalho.

5.1.1 Integração Vicon-ROS

A validação dos resultados experimentais foi realizada com sistema de captura de movimentos com câmeras Vicon. Primeiramente as 8 câmeras foram devidamente dispostas no laboratório e configuradas para melhor identificação das esferas refletivas no ambiente do laboratório.

Para que fosse possível a implementação da validação foi realizada a integração do *software* proprietário da marca (Vicon Tracker) com a plataforma de prototipagem de robôs (ROS) por meio do pacote *vicon*. Essa integração está pronta e disponível para os demais usuários e desenvolvedores do Laboratório de Robótica do Instituto Tecnológico Vale.

5.1.2 Estrutura de testes para odometria do EspeleoRobô

Além da integração Vicon-ROS, foi desenhado um esquema de testes próprio para o EspeleoRobô e códigos para manipular valores dos sensores. Essa estrutura possibilitará futuras avaliações de técnicas de odometria, conforme necessidade, seja para testar novos algoritmos ou novos sensores. Os códigos desenvolvidos estão todos disponibilizados para uso do laboratório.

5.1.3 Odometria *online*

Foram implementadas quatro técnicas de odometria que podem ser usadas para o cálculo *online* e os códigos desenvolvidos também estão disponíveis. A técnica de odometria com sensores LiDAR e IMU, implementada com algoritmos do pacote *laser_scan_matcher*

obteve precisão superior a 99% no cálculo de odometria em todos os testes realizados, além de também calcular com eficiência a trajetória do robô, por meio da posição (x, y) e orientação (θ) .

Os desenvolvimentos que utilizam sensores LiDAR e câmera RGB-D podem ser aplicados em outros dispositivos e se fazem ainda mais relevantes quando aplicados a robôs que utilizam diferentes mecanismos de locomoção, como o EspeleoRobô, pois independem desses mecanismos para realizar os cálculos.

5.1.4 Publicações e apresentações

Durante a construção desta dissertação foram desenvolvidos trabalhos no sentido de contribuir com a comunidade acadêmica em relação ao tema robótica na mineração:

- Avaliação de técnicas de odometria aplicadas a um dispositivo robótico móvel (COTA *et al.*, 2018).

Artigo contendo os resultados preliminares desta dissertação, publicado e apresentado no XXII Congresso Brasileiro de Automática, em João Pessoa-PB, 2018.

- *Multi-terrain inspection robotic device and methods for configuring and guiding the same* (FREITAS *et al.*, 2018).

Participação no pedido de patente do EspeleoRobô depositado no *United States Patent and Trademark Office*.

- *Current trends in Vale's mining robotics operations* (COTA *et al.*, 2018).

Resumo sobre tendências atuais de dispositivos robóticos em operações de mineração publicado e apresentado *Aachen International Mining Symposia - Mines of the Future*, em Aachen, Alemanha, 2018.

- *Espeleorobo: A novel method for caves and confined environment inspection* (ROCHA *et al.*, 2018).

Resumo sobre o EspeleoRobô publicado e apresentado *Aachen International Mining Symposia - Mines of the Future*, em Aachen, Alemanha, 2018.

- Dispositivo de Monitoramento Remoto de Cavidades - EspeleoRobô (COTA *et al.*, 2017b).

Artigo descrevendo o EspeleoRobô, suas funcionalidades e detalhando melhorias implementadas pelo Laboratório de Robótica do Instituto Tecnológico Vale. Foi publicado e apresentado no XIII Simpósio Brasileiro de Automação Inteligente, em Porto Alegre-RS, 2017.

- Robótica na Mineração (COTA *et al.*, 2017a).

Artigo publicado e apresentado na ABM Week 2017, descrevendo o EspeleoRobô e demais dispositivos robóticos do Instituto Tecnológico Vale.

Essa publicação também se derivou em uma apresentação, no formato de palestra, sobre o tema, que foi apresentada na *Semana de Engenharia Mecatrônica 2017 da Universidade Federal de São João del-Rei*, em Ouro Branco-MG; na *Semana de Engenharia Mecânica e de Materiais 2017 da Universidade Federal do Sul e Sudeste do Pará*, em Marabá-PA; e na *EXPOSIBRAM 2017*, em Belo Horizonte-MG.

5.2 Sugestões de trabalhos futuros

Como proposta de desenvolvimento futuro, são sugeridas outras combinações de sensores e a fusão de dados com abordagem probabilística, como um observador de estados ótimo utilizando, por exemplo, Filtro de Kalman. A fusão de *encoder* com IMU pode melhorar os resultados já apresentados, reduzindo os efeitos do escorregamento das rodas e melhorando o desempenho da técnica em movimentos curvos. A fusão *encoders*+LiDAR+IMU também pode melhorar ainda mais os resultados apresentados e pode ser uma solução interessante caso a uniformidade das estruturas dos dutos atrapalhe as leituras do sensor LiDAR.

Na abordagem LiDAR+IMU pode ser implementada a técnica de inicialização com funções de custo, para otimizar os resultados em relação a orientação. Além disso, para melhor representar os erros de orientação encontrados pode-se utilizar a distribuição Von Mises enquadrando os erros encontrados entre 0° e 180°.

Em relação à câmera RGB-D, como já mencionado na Subseção 4.4.3, o sensor não se mostrou o mais adequado para as condições atuais do EspeleoRobô. Apesar disso, considerando futuras melhorias na capacidade de processamento do robô, bem como a aplicação dos conhecimentos gerados por esta dissertação em outros projetos do laboratório, são sugeridos novos testes com o sensor, visando determinar de forma definitiva as causas da interrupção dos cálculos e buscando outros tipos de algoritmos, como baseados em *optical flow*, por exemplo. Por fim, com o bom funcionamento de uma abordagem com a câmera RGB-D, seria possível a implementação de uma técnica com a fusão de todos os sensores disponíveis: *encoders*+LiDAR+IMU+câmera RGB-D.

Buscando a padronização para melhor avaliação das implementações testadas em laboratório, é sugerido o desenvolvimento de um sistema automático para realização dos experimentos de odometria. Neste sentido, pode-se usar o sistema de captura de movimento como entrada em um controle PID, baseado em *waypoint-tracking*, de forma que o robô siga trajetórias preestabelecidas dentro do laboratório. Testes com outros mecanismos e modos de locomoção também devem ser executados para avaliar qual o impacto no

desempenho e precisão dos métodos implementados.

São recomendados testes de campo em dutos localizados em áreas de mineração, com os quais seria possível validar a capacidade das técnicas em calcular a distância total percorrida, auxiliando nas atividades de inspeção. Testes de campo em ambientes mais representativos também são indicados para verificar o desempenho em locais reais de operação.

Por fim, deve ser avaliado o impacto do relevo do terreno para iniciar os desenvolvimentos considerando a odometria 3D, com o intuito de propiciar um cálculo preciso de odometria para ambientes não planares e não controlados como cavernas.

Referências Bibliográficas

- ACHTELIK, M., BACHRACH, A., HE, R., et al.. “Stereo vision and laser odometry for autonomous helicopters in GPS-denied indoor environments”. SPIE, 2009.
- ALLALI, S., BENCHAIÏBA, M. “Overview of search and rescue from robotics to wireless sensors and robots networks”. Em: *Emergency and Disaster Management: Concepts, Methodologies, Tools, and Applications*, IGI Global, pp. 1212–1231, 2019.
- AQEL, M. O., MARHABAN, M. H., SARIPAN, M. I., et al.. “Review of visual odometry: types, approaches, challenges, and applications”, *SpringerPlus*, v. 5, n. 1, pp. 1897, 2016.
- ARRAS, K. O., VESTLI, S. J. “Hybrid, high-precision localisation for the mail distributing mobile robot system MOPS”. Em: *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, v. 4, pp. 3129–3134. IEEE, 1998.
- ATHERTON, J. A., HARDIN, B., GOODRICH, M. A. “Coordinating a Multi-Agent Team using a Multiple Perspective Interface Paradigm.” Em: *AAAI Spring Symp.: To Boldly Go Where No Human-Robot Team Has Gone Before*, pp. 47–51, 2006.
- BASACA-PRECIADO, L. C., SERGIYENKO, O. Y., RODRÍGUEZ-QUINONEZ, J. C., et al.. “Optical 3D laser measurement system for navigation of autonomous mobile robot”, *Optics and Lasers in Engineering*, v. 54, pp. 159–169, 2014.
- BAY, H., ESS, A., TUYTELAARS, T., et al.. “Speeded-up robust features (SURF)”, *Computer vision and image understanding*, v. 110, n. 3, pp. 346–359, 2008.
- BEISIEGEL, V. D. R., BERNARDELLI, A. L., DRUMMOND, N. F., et al.. “GEOLOGIA E RECURSOS MINERAIS DA SERRA DOS CARAJÁS”, *Revista Brasileira de Geociências*, v. 3, pp. 215, 1973.
- BLAIS, F. “Review of 20 years of range sensor development”, *Journal of electronic imaging*, v. 13, n. 1, pp. 231–244, 2004.

- BLANA, D. *Hand motion capture*, 2018. <https://medium.com/the-quest-for-a-life-like-prosthetic-hand/hand-motion-capture-e7dd2c2205d7>.
- BROGGI, A., BUZZONI, M., DEBATTISTI, S., et al.. “Extensive tests of autonomous driving technologies”, *IEEE Transactions on Intelligent Transportation Systems*, v. 14, n. 3, pp. 1403–1415, 2013.
- BRUNO, S., OUSSAMA, K. “Springer handbook of robotics”. 2008.
- BURKE, J. L., MURPHY, R. R., ROGERS, E., et al.. “Final report for the DARPA/NSF interdisciplinary study on human-robot interaction”, *IEEE Trans. on Systems, Man, and Cybernetics, Part C (App. and Reviews)*, v. 34, n. 2, pp. 103–112, 2004.
- CENSI, A. “An ICP variant using a point-to-line metric”. Em: *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pp. 19–25. IEEE, 2008.
- CHABLAT, D., VENKATESWARAN, S., BOYER, F. “Dynamic Model of a Bio-Inspired Robot for Piping Inspection”. Em: *ROMANSY 22–Robot Design, Dynamics and Control*, Springer, pp. 42–51, 2019.
- CHANG, F.-R., WANG, L.-S., TU, C.-H., et al.. “Global positioning system (GPS) Compass”. jul. 7 1998. US Patent 5,777,578.
- CHENAVIER, F., CROWLEY, J. L. “Position estimation for a mobile robot using vision and odometry”. Em: *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, pp. 2588–2593. IEEE, 1992.
- CHUM, O., MATAS, J. “Optimal randomized RANSAC”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 30, n. 8, pp. 1472–1482, 2008.
- COTA, E., TORRE, M. P., FERREIRA, J., et al.. “Robótica na Mineração”, *Anais da ABM Week 2017, São Paulo*, 2017a.
- COTA, E., TORRE, M., ROCHA, F., et al.. “Dispositivo de monitoramento remoto de cavidades - EspeleoRobô”. Em: *XIII Simpósio Brasileiro de Automação Inteligente*, 2017b.
- COTA, E., ROCHA, F., AZPÚRUA, H., et al.. “Avaliação de Técnicas de odometria aplicadas a um dispositivo robótico móvel”, *XXII Congresso Brasileiro de Automática*, 2018.

- COTA, E., TORRE, M. P., FERREIRA, J., et al.. “Current trends in Vale’s mining robotics operations”. Em: *First International Conference Mines of the Future, 2018, Aachen, Germany*. Rheinisch-Westfälische Technische Hochschule Aachen, 2018.
- CUNHA, W., ATZINGEN, N., CRESCENCIO, G. “Estudos espeleológicos na serra dos Carajás, município de Paraupabas, PA”. 2007.
- DRYANOVSKI, I., MORRIS, W., CENSI, A. “Laser scan matcher”. 2014.
- DUDEK, G., JENKIN, M. *Computational principles of mobile robotics*. Cambridge university press, 2010.
- DYNAMICS, B. “RHex”. http://www.bostondynamics.com/robot_rhex.html, 2017. acessado em 30/01/2017.
- ENDRES, F., HESS, J., ENGELHARD, N., et al.. “An evaluation of the RGB-D SLAM system”. Em: *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 1691–1696. IEEE, 2012.
- FIGUEIREDO, L. A. V. D., RASTEIRO, M. A., RODRIGUES, P. C. “Legislação Para a Proteção Do Patrimônio Espeleológico Brasileiro : Mudanças , Conflitos E O Papel da Sociedade Civil”, *Espeleo-Tema*, v. 21, n. 1, pp. 49–65, 2010.
- FISCHLER, M. A., BOLLES, R. C. “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography”. Em: *Readings in computer vision*, Elsevier, pp. 726–740, 1987.
- FRAUNDORFER, F., SCARAMUZZA, D. “Visual odometry: Part ii: Matching, robustness, optimization, and applications”, *IEEE Robotics & Automation Magazine*, v. 19, n. 2, pp. 78–90, 2012.
- FREITAS, G., ROCHA, F., TORRE, M. P., et al.. “Multi-terrain inspection robotic device and methods for configuring and guiding the same”. 2018. No PCT/BR2018/050025.
- GONZALEZ, R., RODRIGUEZ, F., GUZMAN, J. L., et al.. “Control of off-road mobile robots using visual odometry and slip compensation”, *Advanced Robotics*, v. 27, n. 11, pp. 893–906, 2013.
- GRISSETTI, G., GRZONKA, S., STACHNISS, C., et al.. “Efficient estimation of accurate maximum likelihood maps in 3D”. Em: *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pp. 3472–3478. IEEE, 2007.

- HAHNEL, D., BURGARD, W., FOX, D., et al.. “An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements”. Em: *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, v. 1, pp. 206–211. IEEE, 2003.
- HESCH, J. A., KOTTAS, D. G., BOWMAN, S. L., et al.. “Camera-IMU-based localization: Observability analysis and consistency improvement”, *The International Journal of Robotics Research*, v. 33, n. 1, pp. 182–201, 2014.
- HILL, S. G., BODT, B. “A field experiment of autonomous mobility: operator workload for one and two robots”. Em: *Pro. of the ACM/IEEE int. conf. on Human-robot interaction*, pp. 169–176. ACM, 2007.
- HONEYWELL. “HALL EFFECT SENSING AND APPLICATION”. 2017.
- HONEYWELL. “Magnetoresistive Sensors - Industry: Position and Solid State Sensing”, *downloaded from/sensing. honeywell. com/index.cfm/ci—id/140970/la—id/1/document/1/re—id/0*, v. 3, 2011.
- KOMSUOGLU, H., MAJUMDAR, A., AYDIN, Y. O., et al.. “Characterization of dynamic behaviors in a hexapod robot”. Em: *Experimental Robotics*, pp. 667–684. Springer, 2014.
- LABBE, M. *Kinect mapping lost odometry*, 2018. <https://github.com/introlab/rtabmap/wiki/Kinect-mapping#lostodometry>.
- LABBE, M., MICHAUD, F. “Appearance-based loop closure detection for online large-scale and long-term operation”, *IEEE Transactions on Robotics*, v. 29, n. 3, pp. 734–745, 2013.
- LABBÉ, M., MICHAUD, F. “Online global loop closure detection for large-scale multi-session graph-based slam”. Em: *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pp. 2661–2666. IEEE, 2014.
- LAI, K., BO, L., REN, X., et al.. “Sparse distance learning for object recognition combining rgb and depth information”. Em: *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 4007–4013. IEEE, 2011.
- LHOTSKÝ, V. “An Integrated Approach to Multi-Robot Exploration of an Unknown Space”, *CTU in Prague, FEL, Dept. of Cybernetics*, 2016.
- LI, J., ZHONG, R., HU, Q., et al.. “Feature-based laser scan matching and its application for indoor mapping”, *Sensors*, v. 16, n. 8, pp. 1265, 2016.

- LU, F., MILIOS, E. “Robot pose estimation in unknown environments by matching 2d range scans”, *Journal of Intelligent and Robotic systems*, v. 18, n. 3, pp. 249–275, 1997.
- MACCORMICK, J. “How does the Kinect work”, *Presentert ved Dickinson College*, v. 6, 2011.
- MAIMONE, M., CHENG, Y., MATTHIES, L. “Two years of visual odometry on the mars exploration rovers”, *Journal of Field Robotics*, v. 24, n. 3, pp. 169–186, 2007.
- MANDOW, A., MARTINEZ, J. L., MORALES, J., et al.. “Experimental kinematics for wheeled skid-steer mobile robots”. Em: *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pp. 1222–1227. IEEE, 2007.
- MARTÍNEZ, J. L., MANDOW, A., MORALES, J., et al.. “Approximating kinematics for tracked mobile robots”, *The International Journal of Robotics Research*, v. 24, n. 10, pp. 867–878, 2005.
- MONTEIRO, F. A. D. “PROJETO DE IMPLANTAÇÃO DE CONHECIMENTOS, PRÁTICAS E LEGISLAÇÃO PERTINENTES À ESPELEOLOGIA NA SUPERINTENDÊNCIA ESTADUAL DO MEIO AMBIENTE – SEMACE”. Em: *Anais do 31º Cong. Brasileiro de Espeleologia*, pp. 263–267, 2011.
- NAGATANI, K., IKEDA, A., ISHIGAMI, G., et al.. “Development of a visual odometry system for a wheeled robot on loose soil using a telecentric camera”, *Advanced Robotics*, v. 24, n. 8-9, pp. 1149–1167, 2010.
- NISTÉR, D., NARODITSKY, O., BERGEN, J. “Visual odometry”. Em: *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, v. 1, pp. I–I. Ieee, 2004.
- NOURANI-VATANI, N., BORGES, P. V. K. “Correlation-based visual odometry for ground vehicles”, *Journal of Field Robotics*, v. 28, n. 5, pp. 742–768, 2011.
- PEI, Y., KLEEMAN, L. “A novel odometry model for wheeled mobile robots incorporating linear acceleration”. Em: *Mechatronics and Automation (ICMA), 2017 IEEE International Conference on*, pp. 1396–1403. IEEE, 2017.
- PENTZER, J., BRENNAN, S., REICHARD, K. “Model-based Prediction of Skid-steer Robot Kinematics Using Online Estimation of Track Instantaneous Centers of Rotation”, *Journal of Field Robotics*, v. 31, n. 3, pp. 455–476, 2014.

- PREDKO, M., MC COMB, G. *Robot Builders Bonanza*. Mcgraw-hill Education-Europe, 2011.
- QUIGLEY, M., FAUST, J., FOOTE, T., et al.. “ROS: an open-source Robot Operating System”. 2009.
- REZENDE NETO, A., MAGAGNIN JUNIOR, A., NEIVA, E., et al.. “Sistema de medição de campo magnético baseado no efeito hall e arduino”, *Monografia. Dep. Acadêmico de Eletrônica. Universidade Tecnológica Federal do Paraná, Curitiba*, 2010.
- ROCHA, F., TORRE, M. P., GARCIA, G., et al.. “Espeleorobo: A novel method for caves and confined environment inspection”. Em: *First International Conference Mines of the Future, 2018, Aachen, Germany*. Rheinisch-Westfälische Technische Hochschule Aachen, 2018.
- ROCHA, F. “Análise de mobilidade de um dispositivo robótico para inspeção remota de cavidades”, 2018.
- RUBLEE, E., RABAUD, V., KONOLIGE, K., et al.. “ORB: An efficient alternative to SIFT or SURF”. Em: *Computer Vision (ICCV), 2011 IEEE international conference on*, pp. 2564–2571. IEEE, 2011.
- RUCHKYS, U., TRAVASSOS, L., RASTEIRO, M., et al.. “Patrimônio espeleológico em rochas ferruginosas: propostas para sua conservação no Quadrilátero Ferrífero, Minas Gerais”. 2015.
- SAHBEL, A., ABBAS, A., SATTAR, T. “System Design and Implementation of Wall Climbing Robot for Wind Turbine Blade Inspection”, 2019.
- SHOTTON, J., FITZGIBBON, A., COOK, M., et al.. “Real-time human pose recognition in parts from single depth images”. Em: *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pp. 1297–1304. Ieee, 2011.
- SICILIANO, B., SCIAVICCO, L., VILLANI, L., et al.. *Robotics: modelling, planning and control*. Springer Science & Business Media, 2010.
- SMISTAD, E., FALCH, T. L., BOZORGI, M., et al.. “Medical image segmentation on GPUs—A comprehensive review”, *Medical image analysis*, v. 20, n. 1, pp. 1–18, 2015.
- SPENKO, M., BUERGER, S., IAGNEMMA, K. *The DARPA Robotics Challenge Finals: Humanoid Robots To The Rescue*, v. 121. Springer, 2018.

- STEINBRÜCKER, F., STURM, J., CREMERS, D. “Real-time visual odometry from dense RGB-D images”. Em: *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pp. 719–722. IEEE, 2011.
- STUFF, A. “Hokuyo Range”. <https://autonomoustuff.com/product/hokuyo-utm-301x/>, 2018. acessado em 25/04/2018.
- SURMANN, H., NÜCHTER, A., HERTZBERG, J. “An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments”, *Robotics and Autonomous Systems*, v. 45, n. 3-4, pp. 181–198, 2003.
- VALE, S. “S11D é o maior investimento privado realizado no Brasil nesta década”, *Disponível em:*, v. <http://www.vale.com/hotsite/PT/Paginas/s11d-maior-investimento-privado-realizado-brasil-nesta-decada.aspx>, 2018.
- WALLHOFF, F., RUB, M., RIGOLL, G., et al.. “Surveillance and activity recognition with depth information”. Em: *Multimedia and Expo, 2007 IEEE International Conference on*, pp. 1103–1106. IEEE, 2007.
- WANG, C., LV, W., LI, X., et al.. “Terrain Adaptive Estimation of Instantaneous Centres of Rotation for Tracked Robots”, *Complexity*, v. 2018, 2018.
- WANG, T., WU, Y., LIANG, J., et al.. “Analysis and experimental kinematics of a skid-steering wheeled robot based on a laser scanner sensor”, *Sensors*, v. 15, n. 5, pp. 9681–9702, 2015.
- WULF, O., ARRAS, K. O., CHRISTENSEN, H. I., et al.. “2D mapping of cluttered indoor environments by means of 3D perception”. Em: *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, v. 4, pp. 4204–4209. IEEE, 2004.
- YU, Y., PRADALIER, C., ZONG, G. “Appearance-based monocular visual odometry for ground vehicles”. Em: *Advanced Intelligent Mechatronics (AIM), 2011 IEEE/ASME International Conference on*, pp. 862–867. IEEE, 2011.
- ZHANG, L., CURLESS, B., SEITZ, S. M. “Rapid shape acquisition using color structured light and multi-pass dynamic programming”. Em: *3D Data Processing Visualization and Transmission, 2002. Proceedings. First International Symposium on*, pp. 24–36. IEEE, 2002.