

**LOCALIZAÇÃO E MAPEAMENTO  
EM TERRENOS IRREGULARES  
UTILIZANDO ROBÔS MÓVEIS**

DOUGLAS GUIMARÃES MACHARET

**LOCALIZAÇÃO E MAPEAMENTO  
EM TERRENOS IRREGULARES  
UTILIZANDO ROBÔS MÓVEIS**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

ORIENTADOR: PROF. MARIO FERNANDO MONTENEGRO CAMPOS

Belo Horizonte

Março de 2009

© 2009, Douglas Guimarães Macharet.  
Todos os direitos reservados.

Macharet, Douglas Guimarães  
G963l      Localização e mapeamento em terrenos irregulares  
              utilizando robôs móveis / Douglas Guimarães Macharet.  
— Belo Horizonte, 2009  
              xix, 94 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de  
Minas Gerais

Orientador: Prof. Mario Fernando Montenegro  
Campos

1. Robótica Móvel. 2. Navegação Inercial. 3. Filtro  
de Kalman. 4. Mapeamento. I. Título.

CDU 007.52



UNIVERSIDADE FEDERAL DE MINAS GERAIS  
INSTITUTO DE CIÊNCIAS EXATAS  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

## FOLHA DE APROVAÇÃO

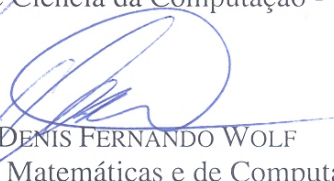
Localização e Mapeamento em Terrenos Irregulares Utilizando Robôs Móveis

**DOUGLAS GUIMARÃES MACHARET**

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:



PROF. MARIO FERNANDO MONTENEGRO CAMPOS - Orientador  
Departamento de Ciência da Computação - UFMG



PROF. DENIS FERNANDO WOLF  
Instituto de Ciências Matemáticas e de Computação - USP



PROF. GUILHERME AUGUSTO SILVA PEREIRA  
Departamento de Engenharia Elétrica - UFMG



PROF. LUIZ CHAIMOWICZ  
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 13 de março de 2009.



*Dedico este trabalho à minha família que nunca deixou de me apoiar.*

# Agradecimentos

Primeiramente agradeço a Deus por essa grande oportunidade, e por ter me dado força e sabedoria para vencer mais essa etapa.

Agradeço ao meus pais e avós por sempre me apoiarem em todos momentos da minha vida e pelas palavras de incentivo, sem vocês nada disso seria possível. Keyla e Timóteo por me ajudarem sempre que foi necessário.

Ao meu orientador, professor Mario Campos, pela paciência e por todos os ensinamentos. Todo o aprendizado obtido durante o processo foi de imensurável valor para minha boa formação profissional.

Aos meus colegas do VeRLab, pelos momentos de descontração e importantes trocas de informações, além de sempre estarem dispostos a ajudar quando precisei, Victor, Dimas, Renato, Pedro e todos demais. Em especial Vilar por disponibilizar a biblioteca do Kalman e Armando, por sempre estarem prontos a ouvir e dar sugestões, e Wolmar pela ajuda na montagem e integração do sistema mecânico.

Aos demais colegas do DCC pelo companheirismo e pela troca de experiências e idéias, ao Synergia pela oportunidade e auxílio que permitiu que esse trabalho pudesse ser realizado.

A todos que de alguma forma participaram desse processo, muito obrigado.

*“Porque o Senhor dá a sabedoria, e da sua boca  
vem a inteligência e o entendimento.”*

(Provérbios 2:6)

# Resumo

É notável o aumento na utilização de robôs nas mais diversas áreas e aplicações, principalmente os denominados robôs móveis, aqueles com capacidade de se locomover através do ambiente em que estão inseridos. Dessa forma, é crescente também a demanda de prover maior autonomia aos robôs, diminuindo ou eliminando a necessidade de supervisão humana.

Uma premissa fundamental para que um robô móvel consiga executar suas tarefas é que ele possua conhecimento de sua localização. Ao se locomover em um ambiente interno a condição do piso é de pouca relevância, podendo ser considerado como plano. Entretanto, ao se navegar em um ambiente externo, diversos problemas surgem relacionados ao solo, como buracos, aclives/declives, dentre outros, e devem ser considerados para uma navegação segura.

Considerando isso, é proposto então neste trabalho o desenvolvimento de um sistema de localização para robôs móveis terrestres navegando em ambientes externos de terreno irregular, e mapeamento utilizando-se um *laser*.

Inicialmente, é apresentada a metodologia utilizada para a fusão das informações fornecidas por diversos sensores com o objetivo de permitir realizar a melhor estimativa possível da localização e orientação do robô em um dado instante de tempo, considerando que esse estará navegando em um ambiente não estruturado e de terreno irregular.

Dada a localização, o robô deve então ser capaz de, a partir dos dados coletados por um *laser rangefinder*, levantar uma nuvem de pontos que represente da maneira mais fiel possível o ambiente por onde o robô trafegou.

**Palavras-chave:** Robótica Móvel, Navegação Inercial, Filtro de Kalman, Mapeamento.

# Abstract

It is remarkable the increase in the use of robots in several areas and applications, especially mobile robots, those with capacity to move through the environment. Thus, it is also growing the demand to provide autonomy to the robots, reducing or eliminating the need for human supervision.

A basic thing for a mobile robot to be capable of perform its tasks is to have the knowledge of its location. When you move into an indoor environment the condition of the floor is of little relevance, as it may be considered plan, however, when you go outdoor, a number of problems are related to the terrain, such as holes, slopes, among others, and that should be considered for a safe navigation.

Considering that, this work proposes the development of a localization system for mobile robots navigating in outdoors environments with uneven terrain, and mapping using a laser.

Initially it shows the methodology used for the sensors fusion, allowing the best possible estimate of the location and orientation of the robot at a given moment, given that the robot is navigating in an unstructured environment with uneven terrain.

Given the location, the robot should be able to raise a point cloud that represents the most accurate possible the environment where the robot has travelled, using the data collected by a laser rangefinder.

**Keywords:** Mobile Robotics, Inertial Navigation, Kalman Filter, Mapping.

# Lista de Figuras

1.1	Manipuladores robóticos. . . . .	2
1.2	Robôs móveis. . . . .	2
1.3	Navegação do robô <i>Spirit</i> [Maimone et al., 2006]. . . . .	6
2.1	Exemplo de um mapeamento 2D utilizando um <i>laser</i> [Mázl & Preucil, 2000].	12
2.2	Montagens que utilizam espelhos giratórios. . . . .	13
2.3	<i>Laser</i> sobre plataforma giratória [Katz et al., 2006]. . . . .	14
2.4	<i>Lasers</i> dispostos ortogonalmente [Howard et al., 2004]. . . . .	15
2.5	Stanford Racing Team (Stanley) [Thrun et al., 2006]. . . . .	15
2.6	SLAM 2D [Bailey, 2002]. . . . .	16
2.7	Equipamento utilizado para o SLAM 2.5D [Mahon & Williams, 2003]. . . . .	17
2.8	SLAM 3D com uso de um helicóptero [Thrun et al., 2003]. . . . .	18
2.9	<i>Lasers</i> com planos de leitura defasados em 180° [Wulf et al., 2007]. . . . .	19
3.1	Diagrama geral da metodologia de mapeamento. . . . .	20
3.2	Referenciais do mundo. . . . .	22
3.3	Referencial da plataforma. . . . .	22
3.4	Ciclo geral do Filtro de Kalman. . . . .	23
3.5	Diagrama detalhado da etapa de localização. . . . .	28
3.6	Cálculo da variância dos dados do GPS baseado no número de satélites e velocidade do robô. . . . .	32
3.7	Cálculo do <i>bias</i> dos giroscópios. . . . .	33
3.8	Diagrama detalhado da etapa de mapeamento. . . . .	34
3.9	Esquema de leitura do <i>laser</i> . . . . .	35
3.10	Distância considerada entre um ponto e seus vizinhos. . . . .	36
3.11	Correção realizada para leituras do <i>laser</i> com a presença de pontos medidos erroneamente. . . . .	36
3.12	Transformações entre os referenciais envolvidos. . . . .	38

3.13	Transformação do referencial do mundo para o do robô. . . . .	39
3.14	Transformação do referencial do robô para o do <i>laser</i> . . . . .	40
4.1	Equipamentos utilizados no trabalho. . . . .	43
4.2	Pré-filtragem dos dados da IMU e da bússola. A primeira linha apresenta os dados recuperados a partir dos giroscópios, a segunda linha os dados dos acelerômetros, e finalmente a orientação obtida a partir da bússola . . . . .	45
4.3	Destaque da montagem do <i>laser</i> sobre o robô. . . . .	46
4.4	Cálculo da inclinação do <i>laser</i> . . . . .	47
4.5	Experimentos de rolagem utilizando-se uma rampa para validação do <i>roll</i> . . . . .	49
4.6	Experimentos de rolagem utilizando-se uma rampa para validação do <i>roll</i> . . . . .	49
4.7	Experimentos de subida/descida de uma rampa para validação do <i>pitch</i> . . . . .	50
4.8	Experimentos de subida/descida de uma rampa para validação do <i>pitch</i> . . . . .	51
4.9	Experimentos de subida/descida de uma rampa, cálculo da altura utilizando o <i>pitch</i> estimado. . . . .	51
4.10	Experimentos de subida/descida de uma rampa, reconstrução do caminho baseado na estimativa do <i>pitch</i> . . . . .	52
4.11	Imagem do sistema de Visão Computacional utilizado para validação do <i>yaw</i> . . . . .	53
4.12	Experimentos de giro sobre o eixo para validação do <i>yaw</i> . . . . .	53
4.13	Trajeto quadrado para validação do <i>yaw</i> (girando para a esquerda). . . . .	54
4.14	Trajeto quadrado para validação do <i>yaw</i> (girando para a direita). . . . .	54
4.15	Reconstrução da trajetória realizada pelo robô no Experimento 1. . . . .	57
4.16	Número de satélites visíveis durante a realização do Experimento 1. . . . .	57
4.17	Ângulos de orientação calculados para o Experimento 1. . . . .	58
4.18	Ângulos de orientação calculados para o Experimento 2. . . . .	58
4.19	Reconstrução da trajetória realizada pelo robô no Experimento 2. . . . .	59
4.20	Variação de altura obtida durante a realização do Experimento 2. . . . .	59
4.21	Comparativo entre a variação de altura estimada e o valor fornecido pelo GPS no tempo para o Experimento 2. . . . .	60
4.22	Variância das variáveis de localização durante o Experimento 2. . . . .	61
4.23	Reconstrução da trajetória realizada pelo robô no Experimento 3. . . . .	62
4.24	Ângulos de orientação calculados para o Experimento 3. . . . .	62
4.25	Valores observados durante a realização do Experimento 3 para os parâmetros número de satélites visíveis, velocidade, e variância calculada a partir do modelo proposto para as informações do GPS. . . . .	63
4.26	Momentos em que a baixa velocidade do robô levou à ocorrência do efeito de <i>random walk</i> . . . . .	64

4.27	Reconstrução da trajetória executada pelo robô no Experimento 3 com uma simulação de queda de sinal do GPS. Trajeto completo realizado e destaque do início e final da simulação de queda de sinal. . . . .	65
4.28	Covariância do erro durante a execução do Experimento 3 com simulação de queda de sinal do GPS. . . . .	65
4.29	Diferentes tipos de terrenos trafegados durante a realização do Experimento 4. . . . .	67
4.30	Reconstrução da trajetória realizada pelo robô no Experimento 4. . . . .	67
4.31	Ângulos de orientação calculados para o Experimento 4. . . . .	68
4.32	Local e visão de topo do mapeamento obtido em um ambiente interno. . .	71
4.33	Local e visão em perspectiva do mapeamento obtido em um ambiente interno.	72
4.34	Detalhes do terreno identificáveis através do mapeamento. . . . .	74
4.35	Detalhes do terreno identificáveis através do mapeamento. . . . .	75
4.36	Mapeamento completo obtido para o Experimento 1. . . . .	76
4.37	Mapeamento completo obtido para o Experimento 2. . . . .	77
4.38	Mapeamento completo obtido para o Experimento 3. . . . .	78
4.39	Mapeamento completo obtido para o Experimento 4. . . . .	79
A.1	Esquema da plataforma <i>Player</i> . . . . .	92



# Lista de Tabelas

4.1	Desvios padrão dos ângulos de orientação calculados para cada diferente tipo de terreno trafegado durante o Experimento 4. . . . .	69
4.2	Tabela comparativa entre distância reais e os valores calculados à partir da nuvem de pontos obtida em um mapeamento interno. . . . .	70
A.1	Especificações técnicas - <i>Pioneer</i> P3-AT. . . . .	92
A.2	Especificações técnicas - <i>laser SICK</i> LMS 200. . . . .	93
A.3	Especificações técnicas - <i>Crista IMU</i> . . . . .	93
A.4	Especificações técnicas - bússola digital KVH C-100. . . . .	94
A.5	Especificações técnicas - GPS Garmin 18x-5Hz. . . . .	94

# Lista de Algoritmos

1	KF . . . . .	24
2	UKF . . . . .	27

# Lista de Acrônimos

**DARPA** *Defense Advanced Research Project Agency*

**ECEF** *Earth-Centered, Earth-Fixed*

**EKF** *Extended Kalman Filter* (Filtro de Kalman Estendido)

**ENU** *East-North-Up*

**GPS** *Global Positioning System*

**ICP** *Iterative Closest Point*

**IMU** *Inertial Measurement Unit* (Unidade de Medição Inercial)

**KF** *Kalman Filter* (Filtro de Kalman)

**LTP** *Local Tangent Plane*

**MCL** *Monte Carlo Localization*

**NASA** *National Aeronautics and Space Administration*

**RPY** *Roll-Pitch-Yaw*

**SLAM** *Simultaneous localization and mapping*

**SP** *Sigma Points* (Pontos Sigma)

**SPKF** *Sigma-Point Kalman Filter*

**UKF** *Unscented Kalman Filter* (Filtro de Kalman *Unscented*)

**UT** *Unscented Transform* (Transformação *Unscented*)

**UTM** *Universal Transverse Mercator*

# Lista de Símbolos

- $\{L\}$  Referencial do *laser*
- $\{R\}$  Referencial do robô
- $\{W\}$  Referencial do mundo
- $a_x$  Aceleração no eixo X
- $a_y$  Aceleração no eixo Y
- $a_z$  Aceleração no eixo Z
- $g_x$  Velocidade angular ao redor do eixo X
- $g_y$  Velocidade angular ao redor do eixo Y
- $g_z$  Velocidade angular ao redor do eixo Z
- $g$  Resultante das acelerações nos três eixos
- $\mathbf{g}$  Aceleração da gravidade
- $n$  Número de satélites visíveis
- $r$  Distância de um ponto medido pelo *laser*
- $v$  Velocidade linear do robô
- $\vec{x}$  Vetor de estados
- $\vec{u}$  Vetor de controle
- $\vec{z}$  Vetor de observação
- $A_t$  Matriz de transição
- $B_t$  Matriz de controle

$C_t$  Matriz de observação

$K_t$  Ganho de *Kalman*

$R_t$  Matriz de covariância de  $\varepsilon_t$

$Q_t$  Matriz de covariância de  $\nu_t$

$\phi$  Ângulo de rolamento (*roll*)

$\theta$  Ângulo de arfagem (*pitch*)

$\psi$  Ângulo guinada (*yaw*)

$\varepsilon_t$  Ruído do processo

$\nu_t$  Ruído da observação

$\rho$  Ângulo entre um ponto medido na leitura do *laser* e a origem do referencial do *laser*

$\sigma$  Ângulo de inclinação do *laser* em relação ao robô

$\eta$  Valor limite de distância entre o ponto médio e seus vizinhos

# Sumário

<b>Agradecimentos</b>	<b>vi</b>
<b>Resumo</b>	<b>viii</b>
<b>Abstract</b>	<b>ix</b>
<b>Lista de Figuras</b>	<b>x</b>
<b>Lista de Tabelas</b>	<b>xiii</b>
<b>Lista de Algoritmos</b>	<b>xiv</b>
<b>Lista de Acrônimos</b>	<b>xv</b>
<b>Lista de Símbolos</b>	<b>xvi</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Contextualização . . . . .	1
1.2 Problema . . . . .	4
1.3 Motivação . . . . .	6
1.4 Contribuições . . . . .	7
1.5 Organização do Texto . . . . .	8
<b>2 Trabalhos Relacionados</b>	<b>9</b>
2.1 Sistemas de Navegação Inercial . . . . .	9
2.2 Mapeamento . . . . .	11
2.3 SLAM . . . . .	15
<b>3 Metodologia</b>	<b>20</b>
3.1 Localização . . . . .	21
3.1.1 Sistemas de Coordenadas . . . . .	21

3.1.2	Filtro de Kalman . . . . .	22
3.1.3	Cálculo da Atitude . . . . .	28
3.2	Mapeamento . . . . .	33
3.2.1	Leitura do <i>Laser</i> . . . . .	34
3.2.2	Filtragem da Leitura . . . . .	35
3.2.3	Transformações entre os Referenciais . . . . .	37
<b>4</b>	<b>Experimentos</b>	<b>42</b>
4.1	Planejamento . . . . .	42
4.2	Arcabouço Experimental . . . . .	43
4.2.1	Localização . . . . .	43
4.2.2	Mapeamento . . . . .	45
4.3	Localização . . . . .	48
4.3.1	Experimentos Internos . . . . .	48
4.3.2	Experimentos Externos . . . . .	55
4.4	Mapeamento . . . . .	69
4.4.1	Experimentos Internos . . . . .	69
4.4.2	Experimentos Externos . . . . .	73
<b>5</b>	<b>Conclusões e Trabalhos Futuros</b>	<b>80</b>
5.1	Conclusões . . . . .	80
5.2	Trabalhos Futuros . . . . .	82
	<b>Referências Bibliográficas</b>	<b>84</b>
	<b>Apêndice A Recursos Utilizados</b>	<b>91</b>
A.1	Plataforma de Programação . . . . .	91
A.2	Robô . . . . .	92
A.3	Sensor <i>Laser</i> . . . . .	93
A.4	Unidade de Medição Inercial . . . . .	93
A.5	Bússola Digital . . . . .	94
A.6	GPS . . . . .	94

# Capítulo 1

## Introdução

Com o avanço no desenvolvimento de veículos autônomos que navegam em ambientes externos, grande esforço tem sido feito para aumentar a capacidade dos robôs de tomarem suas próprias decisões. Entretanto, para que isso seja feito é necessário que questões mais básicas envolvidas no processo sejam abordadas, como conhecimento do ambiente (mapeamento) e a localização.

Este trabalho apresenta a concepção, desenvolvimento e implementação de um sistema de localização e mapeamento para robôs móveis terrestres navegando em ambientes externos e de terrenos irregulares.

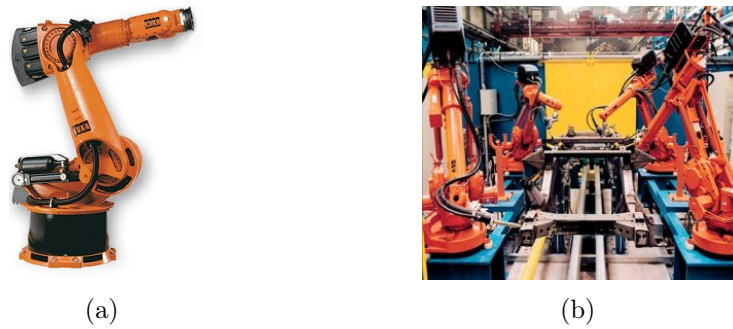
### 1.1 Contextualização

A robótica atualmente tem sua maior aplicação nas indústrias, e é representada pela utilização dos braços robóticos, ou manipuladores [Siegwart & Nourbakhsh, 2004]. As principais características de robôs nessa categoria são: operam dentro de um espaço definido, executam tarefas repetitivas, não podem mover-se de um local para outro, e possuem pouca (ou nenhuma) autonomia de decisão.

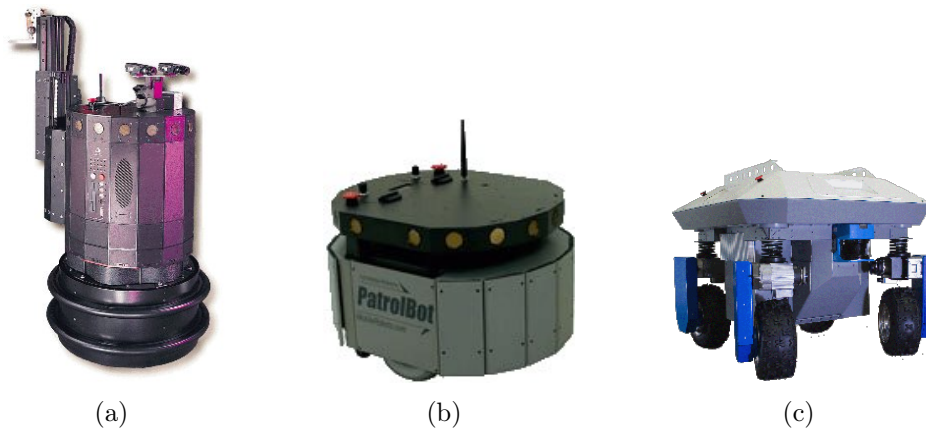
Uma área recente que tem obtido grandes avanços é a robótica móvel, responsável por estudar os robôs móveis, aqueles que podem locomover-se no ambiente em que estão inseridos. Dentre as principais aplicações, podemos citar a utilização desses robôs para a exploração de ambientes perigosos ou de difícil acesso para o ser humano. Alguns exemplos de robôs móveis são apresentados na Figura 1.2.

Os robôs móveis podem ser divididos em três categorias de acordo com a forma em que são controlados [Dudek & Jenkin, 2000], são elas:





**Figura 1.1.** 1.1(a) Manipulador robótico. 1.1(b) Utilização de manipuladores em uma linha de montagem.



**Figura 1.2.** Robôs móveis. 1.2(a) Nomad 200. 1.2(b) PatrolBot. 1.2(c) Seekur.

- **Teleoperados:** Os movimentos são completamente definidos através da intervenção de um operador.
- **Semi-autônomos:** As tarefas são definidas de uma forma geral por um operador e o robô deve executá-las sozinho.
- **Autônomos:** O robô é capaz de realizar tarefas e de tomar suas próprias decisões.

Uma das principais linhas de pesquisa na área da robótica móvel está ligada ao estudo de como prover uma maior autonomia aos robôs, principalmente nas questões ligadas à sua locomoção. Em robótica móvel, autonomia está intrinsecamente relacionada à navegação do robô, que consiste no robô ser capaz de locomover-se em um determinado ambiente e atingir determinado objetivo, desviando-se de possíveis obstáculos que possam ser encontrados durante o caminho.

Relacionados à navegação dos robôs podemos encontrar os paradigmas robóticos, ou, arquiteturas robóticas. Os paradigmas são definidos pela relação entre três primitivas básicas: *sentir*, *planejar* e *atuar*. Pode-se entender também que um paradigma é definido pela forma como os dados são coletados pelos sensores e processados [Murphy, 2000; Siegwart & Nourbakhsh, 2004; Thrun et al., 2005].

Três dos paradigmas mais utilizados são:

- **Hierárquico (Deliberativo):** O modelo mais antigo na robótica. O robô coleta os dados através dos sensores e constrói um modelo do mundo, feito isso ele então realiza o planejamento das próximas ações a serem tomadas e em seguida age.

Os principais problemas envolvidos nesse paradigma são a dificuldade em se construir e manter um bom modelo do mundo e a falta de um sistema que faça com que o robô saiba lidar com mudanças que ocorreram no mundo após seu planejamento.

- **Reativo:** Ignorando a etapa de planejamento, o robô atua de acordo com os dados coletados em determinado instante. Apesar de ser de fácil implementação o sistema se torna não determinístico (não é possível prever o exato local por onde o robô irá navegar), o que não é o ideal para determinados sistemas.
- **Híbrido:** Atualmente o paradigma mais utilizado se trata de uma fusão entre os paradigmas Deliberativo e Reativo. É realizada uma etapa de planejamento e conforme o robô atua no ambiente também está presente um sistema reativo, de forma a lidar com mudanças que possam ocorrer no ambiente.

Um dos fatores ao qual a escolha do paradigma está relacionada é o ambiente onde o robô irá trafegar. Existem duas grandes áreas de pesquisa envolvendo o ambiente onde o robô está inserido. Uma área aborda agentes que se locomovem principalmente no interior de edificações, a chamada navegação *indoor*. Já a outra área está preocupada com a locomoção de robôs em ambientes externos, como por exemplo campos, estradas e outros, essa sendo chamada de navegação *outdoor*.

Estando o robô em uma navegação *indoor* ou *outdoor* existem três perguntas fundamentais em robótica móvel e que devem ser respondidas pelo robô, “**Onde Estou?**” (localização), “**Aonde Vou?**” (objetivo) e “**Como Vou?**” (estratégia). Localização é a posição que o robô possui em relação ao ambiente e que pode ser calculada baseada em informações do próprio ambiente coletadas através de sensores. Ter sua localização conhecida é de fundamental importância e está diretamente ligada à navegação, uma vez que para o robô identificar se determinado objetivo foi alcançado é necessário saber

onde ele se encontra no momento. Entretanto, existe uma incerteza nos dados obtidos dos sensores, e que tende a prejudicar a estimativa de localização durante a navegação.

A forma mais conhecida de localização, e a mais usual para nós humanos, é a utilização de um mapa, sendo possível através dele sabermos o trajeto realizado e também traçar possíveis rotas. Porém, para que um mapa possa ser útil é necessário que o robô consiga se localizar nele.

Entretanto, nem sempre é possível se obter um mapa do ambiente e fornecê-lo para o robô, principalmente quando esse irá realizar uma navegação *outdoor*, o que faz com que em grande parte das vezes seja utilizado o paradigma reativo para sua navegação. Utilizando-se esse paradigma o robô não possui nenhum conhecimento prévio do mundo, mas apenas informações locais de onde ele está atuando no momento. Desta forma, ele não guarda nenhuma informação a respeito de caminhos já navegados, o que faz com que ele possa cometer novamente os mesmos erros caso tenha que navegar outra vez por um mesmo caminho problemático.

Considerando esse problema, seria interessante então que durante a navegação do robô por um ambiente ele fosse criando seu próprio mapa para ser utilizado posteriormente. Mas, para se fazer um mapa é necessário antes saber sua localização no ambiente.

Portanto, conclui-se que a utilização de um mapa poderia tornar mais eficiente a navegação do robô, porém, para se criar um mapa deve-se primeiramente conhecer a própria localização. É nesse contexto que este trabalho está inserido, deseja-se desenvolver um sistema de localização que permita ao robô criar um mapa do ambiente utilizando um *laser* durante a sua navegação em terrenos irregulares.

## 1.2 Problema

Durante a navegação de um robô em um ambiente interno, a condição do piso é de pouca (ou nenhuma) relevância para suas tomadas de decisões, pois não é esperado que se encontrem buracos, rochas ou outros tipos de imperfeições mais graves durante o trajeto. Portanto, em uma navegação *indoor*, faz-se necessário na maior parte das vezes, apenas o conhecimento da posição do robô em um plano 2D, sendo sua posição e orientação especificadas por meio de um vetor de configuração  $\begin{bmatrix} x & y & \psi \end{bmatrix}^T$ , onde  $\psi$  corresponde ao ângulo guinada (*yaw*) do veículo.

Já em um ambiente externo o relevo do terreno exerce grande influência na navegação do robô. Sem uma percepção do solo pelo qual está trafegando, o robô estará propenso a maiores falhas e problemas, uma vez que um ambiente externo é tipicamente

não estruturado e de terreno irregular.

Dentre os problemas encontrados em um ambiente externo, podemos citar como os principais:

- Buracos;
- Aclives/Declives;
- Diferentes tipos de terreno (brita, areia, asfalto, grama).

Estando em um ambiente *outdoor* o robô ao se locomover tende a ocupar posições dispostas nas três dimensões do espaço, além de poder assumir qualquer orientação ao longo dos três eixos de rotação, o que o leva a apresentar seis graus de liberdade. Sua configuração passa a ser representada pelo vetor  $\begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^T$ , onde  $\phi$  corresponde ao ângulo de rolamento (*roll*), e  $\theta$  ao ângulo de arfagem (*pitch*) do robô.

Portanto, o problema de localização a ser abordado está na determinação da posição e orientação do robô no espaço ao navegar em um ambiente externo.

Estando o robô localizado, o próximo problema a ser abordado é a maneira com a qual o mapa do terreno deve ser obtido. Será utilizado um *laser* fixado à frente do robô e direcionado ao solo, e que, conforme o robô movimentar-se, irá coletar informações do relevo à frente do robô.

Existem dois tipos principais de mapas, os mapas topológicos e os mapas métricos. Os mapas topológicos apenas representam a conectividade entre dois dados locais do ambiente e geralmente são representados por grafos, onde os locais em destaque são os nós e os caminhos entre eles as arestas. Já os mapas métricos consistem em uma representação quantitativa das propriedades geométricas do ambiente, gerando uma representação mais fiel do ambiente real.

Alguns problemas encontrados durante o mapeamento realizado por um robô navegando em um ambiente externo e de terreno irregular são:

- Imprecisão na localização;
- Complexidade do ambiente;
- Escalabilidade.

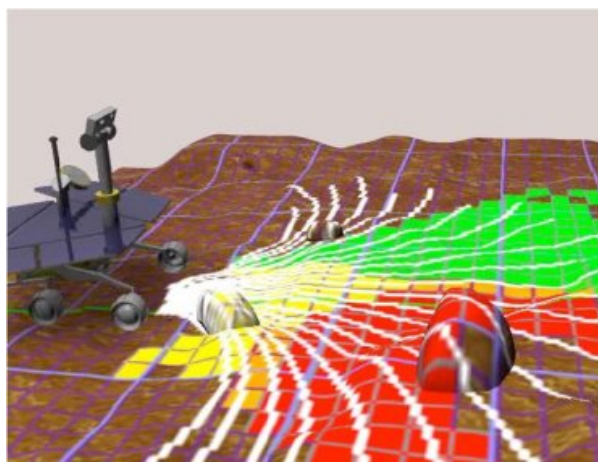
Assim, o problema relacionado ao mapeamento está no tratamento dos dados coletados abordando-se os principais problemas levantados, de forma a permitir a reconstrução de um modelo 3D do local por onde o robô trafegou com a maior qualidade possível.

## 1.3 Motivação

Atualmente, grande parte das pesquisas envolvendo robôs móveis tem como foco principal a necessidade de dar maior autonomia a robôs que navegam em ambientes externos, ou seja, fazendo com que cada vez menos haja alguma interferência humana auxiliando ou acompanhando a realização das tarefas designadas aos robôs. Um dos principais desafios nessa área está relacionada à navegação realizada por eles.

Como principais exemplos dessas aplicações, podemos citar o desafio realizado em 2005 pela *Defense Advanced Research Project Agency* (DARPA), onde os competidores deveriam desenvolver um veículo autônomo capaz de percorrer em segurança um trajeto de 132 milhas no deserto. Em [Buehler et al., 2007] é possível obter as informações de alguns dos veículos finalistas. Outra aplicação bem conhecida e de bastante sucesso consiste na exploração de Marte realizada pelos robôs *Spirit* e *Opportunity*.

Aprofundando em um dos exemplos citados, temos o robô *Spirit*, que se encontra em um ambiente completamente desconhecido e onde a teleoperação humana é difícil e lenta devido à distância. Apesar da tecnologia envolvida nesse projeto, é possível observar a simplicidade que envolve as tomadas de decisões relacionadas à navegação.



**Figura 1.3.** Navegação do robô *Spirit* [Maimone et al., 2006].

Na Figura 1.3 é mostrada a análise de terreno realizada pelo *Spirit*. Através dessa análise é feita uma classificação do terreno e então é decidido o melhor caminho a se tomar. Observa-se uma rocha na parte inferior da imagem, e na parte superior o caminho encontra-se livre para navegação. O robô age da seguinte forma: ele observa o caminho logo à sua frente e classifica qual a região mais segura para se trafegar. Feito isso, ele avança então até a região previamente detectada. Nessa nova posição o processo se repete, até que o objetivo final seja alcançado [Maimone et al., 2006].

Podemos assumir que o robô utiliza o paradigma híbrido, uma vez que ele consegue lidar com alterações inesperadas no ambiente (parte reativa) [Maimone et al., 2004], entretanto, a cada pequena locomoção realizada por ele existe a necessidade de se realizar um novo planejamento (parte deliberativa), ainda que esse planejamento seja apenas local [Biesiadecki & Maimone, 2006]. Esse tipo de navegação é simples e produz bons resultados, porém, os seguintes problemas podem ser levantados:

- Decisões locais;
- Lentidão na navegação;
- Maior gasto de energia;
- Re-análise de regiões já percorridas.

Logo, devido aos problemas levantados, surgiu a necessidade de se pesquisar meios que possam auxiliar na navegação do robô, permitindo assim que ele trafegue de uma maneira mais eficiente e rápida pelo ambiente.

Dessa forma, a idéia seria fazer com que o robô durante sua navegação realize uma reconstrução tridimensional do terreno por onde ele já passou, guardando a informação de todo o ambiente.

Esse mapa 3D servirá como uma “memória”, e permitirá posteriormente a utilização de algoritmos que otimizem a navegação do robô, fazendo com que ele evite trajetórias que levem a regiões já previamente classificadas como problemáticas, e com isso dando mais eficiência e velocidade à navegação. Além disso, esse mapa também poderia ser utilizado por outros robôs que fossem navegar pelo mesmo ambiente.

## 1.4 Contribuições

Este trabalho inicialmente realiza um estudo sobre como calcular a posição e orientação de um robô móvel navegando em terrenos irregulares. Sabendo então sua pose 6D é possível utilizar um *laser* para fazer uma leitura do terreno à frente do robô. Essas leituras devem ser unidas em um único mapa permitindo assim a reconstrução 3D do relevo por onde o robô passou.

Como dito anteriormente, uma das informações mais importantes em sistemas robóticos é a localização do agente. Logo, o cálculo da posição e orientação em terrenos irregulares já é um ganho importante e possui aplicação direta em sistemas de navegação de agentes autônomos, além disso, dotados dessa informação, é possível também

reconstituir a trajetória realizada pelo robô durante sua locomoção, permitindo assim realizar uma análise posterior da movimentação do veículo.

Outra contribuição encontra-se no modelo do ambiente gerado através da aquisição dos dados do *laser*. A qualidade do mapa é fundamental, uma vez que esse modelo poderá ter diversas aplicações, como: auxiliar no desvio de obstáculos encontrados durante a navegação; tornar mais eficiente a navegação do robô, permitindo que melhores trajetórias sejam traçadas para novas locomoções, evitando assim passar por locais já previamente classificados como intransitáveis; permitir a implementação de sistemas de reconhecimento de padrões e formas no ambiente navegado melhorando a localização do robô; entre outras.

## 1.5 Organização do Texto

A seguir é apresentada a estrutura textual deste documento. O Capítulo 2 apresenta uma análise crítica dos trabalhos encontrados na literatura que se relacionam com o tema aqui abordado. No Capítulo 3 é detalhada a metodologia desenvolvida para solução do problema apresentado. O Capítulo 4 exhibe os experimentos realizados para validação do sistema, e finalmente o Capítulo 5 sumariza os resultados obtidos e traça possíveis caminhos a serem tomados para extensão do projeto. O Apêndice A lista os equipamentos utilizados.

# Capítulo 2

## Trabalhos Relacionados

Sistemas que envolvem o cálculo da posição do robô ou que realizam o mapeamento do ambiente são temas bem pesquisados na área de Robótica e Visão Computacional. Logo, é possível se encontrar várias referências relacionadas ao tema deste trabalho.

Entretanto, essas áreas de pesquisa são de grande abrangência, fazendo com que as pesquisas sobre cada tema em particular (localização ou mapeamento) estejam focadas principalmente em determinada parte do processo, cada uma possuindo sua particularidade.

Nesta seção veremos as principais contribuições realizadas por trabalhos na literatura relacionados ao tema abordado.

### 2.1 Sistemas de Navegação Inercial

A localização de um robô é uma tarefa difícil, principalmente em terrenos irregulares. Como já é de conhecimento, a localização baseada somente nas informações de odometria produz resultados com alto nível de erro. Apesar disso, a odometria é vastamente utilizada para navegações em um plano (2D). Porém, sua utilização se torna extremamente limitada quando se deseja saber a posição no espaço (3D).

Em uma navegação *outdoor*, o robô pode sofrer variação em todas as seis dimensões de movimento: três dimensões de posição e três dimensões de orientação. Além disso, ainda existe um maior risco de se ocorrer derrapagens, fato que prejudica ainda mais a odometria.

Para amenizar os problemas inerentes à odometria, a solução é incorporar novos sensores ao sistema e realizar uma fusão dos dados coletados de maneira a se conseguir uma localização mais robusta e precisa.



Um dos sensores mais utilizados é uma Unidade de Medição Inercial (*Inertial Measurement Unit*, ou IMU), que na maioria dos casos possui três giroscópios e três acelerômetros, dispostos de maneira mutuamente ortogonal. O principal problema na utilização desse sensor está nos ruídos associados aos sinais medidos, como o *bias* que provoca, entre outros efeitos, o problema de deriva (ou *drift*) no processo de integração dos dados. Sistemas de navegação que usam esse sensor são conhecidos como sistemas de navegação inercial, e nesses sistemas é possível obter informações que permitem calcular também a inclinação do robô e assim a sua posição e orientação em todo o espaço [Titterton & Weston, 2004].

Existem várias técnicas que podem ser utilizadas para a realização da fusão de dados fornecidos por sensores. Em nosso caso específico, informações envolvidas na localização do robô, entre elas, a mais tradicionalmente utilizada para esse tipo de problema é o Filtro de Kalman (*Kalman Filter*, ou KF). Entretanto, abaixo também analisaremos outras técnicas encontradas na literatura.

Como exemplo de uma abordagem alternativa para esse problema, temos em [Ojeda & Borenstein, 2002] um sistema que faz uso de lógica *fuzzy* para estimar a atitude de um robô móvel. Entretanto, os experimentos apresentados exibiram resultados apenas para locomoções ocorridas em duas dimensões.

Em [Borenstein & Feng, 1996] é possível observar o uso direto dos dados do giroscópio, onde é assumido um *threshold*. Caso o valor fornecido pelo giroscópio seja maior que esse *threshold*, esse valor então passa a ser utilizado no cálculo da posição. Durante os experimentos realizados, o robô apenas movimenta-se em uma linha reta, o quê também não mostra a eficiência do método para a variação de movimentos nos três eixos.

Nesses últimos casos citados, os experimentos realizados não envolveram movimentos nos três eixos do robô, o que omite todo o erro inerente promovido por um sistema que realiza a fusão de odometria e navegação por meio de sensores inerciais.

Outra técnica interessante é a Filtragem Complementar, que permite o casamento de sinais confiáveis em diferentes faixas de frequência. Esta técnica foi utilizada com sucesso em processos de navegação aérea, conforme visto em [Iscold et al., 2007] e [Baveveldt & Klang, 1997], tendo como principais vantagens, o baixo custo computacional e a simplicidade do ajuste dos parâmetros do algoritmo. A principal dificuldade em utilizar essa técnica está na necessidade de se construir o sistema utilizando-se sensores que forneçam informações em frequências complementares.

Utilizando-se apenas os dados da IMU e odometria é possível estimar a posição relativa do robô. Para se obter a posição absoluta é necessária à inserção de um novo sensor ao sistema, um receptor do sistema de *Global Positioning System* (GPS).

Podemos encontrar em [Sukkarieh et al., 1999] e [Liu et al., 2005] um sistema composto por uma IMU e também um GPS. A fusão dos dados dos sensores foi feita utilizando-se um KF padrão, que é utilizado para sistemas lineares. Também temos o uso de IMU/GPS em [Walchko & Mason, 2002], onde foi utilizado o Filtro de Kalman Estendido (*Extended Kalman Filter*, ou EKF), e em [Li et al., 2006] com a utilização do *Sigma-Point Kalman Filter* (SPKF). Em ambos os casos os filtros utilizados são uma implementação do KF para sistemas não-lineares.

Em todos os casos, os resultados para a estimativa de posição são bons, entretanto o principal foco está em realizar a localização no plano (locomoção em estradas), ou seja, não são fornecidas maiores informações a respeito do *roll* e *pitch* durante a navegação do robô, apenas valores de *yaw*.

Já em [Zhang et al., 2005] temos a adição de um novo sensor, uma bússola digital, útil principalmente no cálculo do ângulo de guinada do robô. A fusão das informações de IMU/GPS/bússola foi realizada com o uso do Filtro de Kalman *Unscented* (*Unscented Kalman Filter*, ou UKF), também uma implementação do KF para sistemas não-lineares. Nenhuma informação direta a respeito do *roll* e *pitch* do veículo é fornecida, entretanto, é possível observar que o modelo utilizado trata dessas informações. Portanto, assim como o cálculo da posição 2D produziu bons resultados com o uso do valor de *yaw* entende-se que os valores produzidos para *roll* e *pitch* também devem ser precisos.

## 2.2 Mapeamento

O mapeamento de áreas utilizando-se robôs móveis tem sido um tema bastante estudado. Mapear uma área consiste em obter uma representação espacial do ambiente onde o robô está. Essas representações (mapas) são comumente utilizadas durante a navegação do robô [Thrun, 2002].

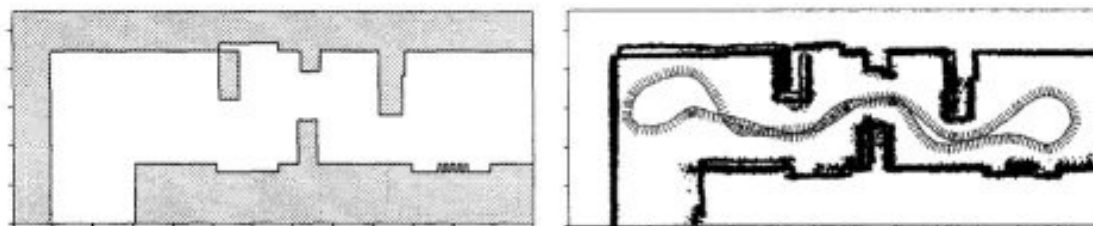
Existem duas principais vertentes no mapeamento realizado utilizando-se um *laser*: o mapeamento 2D que visa uma representação no formato de uma planta-baixa do local, e o mapeamento 3D que tem como objetivo uma reconstrução 3D do ambiente.

Mapeamentos 2D são geralmente utilizados em ambientes *indoor*, ou seja, ambientes com estruturas bem definidas, por exemplo paredes. Em [Mázl & Preucil, 2000] temos um exemplo dessa forma de mapeamento, e como já dito o mapa é uma representação no formato de uma planta-baixa do local por onde o robô trafegou. Esse tipo de mapeamento é mais comumente utilizado como um meio e não como um fim, tendo papel fundamental em técnicas de localização, como por exemplo através da utilização

de filtros de partículas [Kwok et al., 2004].

Entretanto, a utilização de filtros de partículas não é uma solução interessante para a localização no espaço (3D), uma vez que o custo computacional dessa técnica é proporcional ao número de partículas sendo utilizadas.

A Figura 2.1 apresenta um corredor e o mapa 2D obtido com a utilização de um *laser* [Mázl & Preucil, 2000].



**Figura 2.1.** Exemplo de um mapeamento 2D utilizando um *laser* [Mázl & Preucil, 2000].

Ao analisarmos as técnicas existentes de mapeamento 3D, existem várias particularidades que devem ser abordadas, como por exemplo: tipo de sensores, quantidade e como estão dispostos sobre o robô.

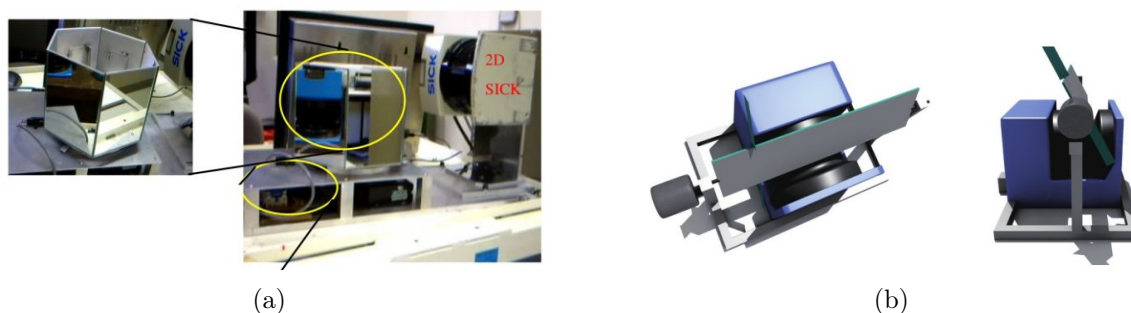
Atualmente existem sensores capazes de já obterem informações 3D do ambiente, entretanto ainda possuem um custo extremamente elevado, chegando ao valor de \$75.000 [Velodyne, 2008]. Por isso, a maior parte das aplicações utiliza os chamados *lasers* 2D.

Na literatura temos abordagens de mapeamento baseadas em Visão Computacional, como [Kurusu et al., 2005], onde um *laser* emite um feixe visível no formato de uma circunferência. Em seguida, uma câmera omnidirecional detecta o feixe possibilitando então o mapeamento, tendo como principal aplicação o mapeamento de túneis, ou [Thompson & Kagami, 2004; Goldberg et al., 2002], onde é utilizada visão estéreo e o mapeamento não possui como principal objetivo a reconstrução do relevo mas sim um auxílio à navegação.

Porém, a abordagem mais comumente utilizada consiste em utilizar um *laser rangefinder*. O tipo de *laser* tradicionalmente utilizado retorna informações 2D (plano de leitura), possuindo um campo de visão de 180°. A principal diferença na sua utilização está na forma em que é feita sua montagem sobre o robô.

Em [Ng, 2005] é mostrada uma montagem que utiliza um *laser* 2D fixo e deitado lateralmente, de forma que esse realiza leituras no plano vertical, e um conjunto de espelhos montados em uma estrutura hexagonal giratória, permitindo assim ao sistema adquirir leituras 3D do ambiente mesmo com o robô parado, as informações retornadas

consistem em planos perpendiculares ao solo e se estendem à frente do robô. Em [Ryde & Hu, 2007] é utilizado um *laser* deitado e direcionado para cima - realizando leituras no plano horizontal - e com um espelho giratório posicionado bem à frente de seu campo de visão, dessa forma, durante o movimento do espelho é possível obter informações 3D do ambiente onde o robô está inserido.



**Figura 2.2.** Montagens que utilizam espelhos giratórios à frente do *laser* para a realização de leituras 3D. 2.2(a) Espelho hexagonal [Ng, 2005]. 2.2(b) Espelho plano [Ryde & Hu, 2007].

Uma forma de montagem bastante utilizada consiste em se ter o *laser* sobre uma estrutura móvel (giratória), permitindo assim variar a angulação do plano de leitura e dessa maneira obter informações em três dimensões.

Esse tipo de montagem foi utilizado em [Katz et al., 2006], onde o *laser* está posicionado sobre o teto de um automóvel em uma base giratório e é capaz de capturar informações 180° na horizontal (campo de visão do *laser*), e o *laser* gira 180° sobre o seu eixo (Figura 2.3). O sistema funciona da seguinte maneira: o veículo move-se uma determinada distância e pára, é feita então a leitura 3D do ambiente, ele então se move mais uma determinada distância e o processo se repete por todo o ambiente que se deseja mapear. As leituras são unificadas depois através do algoritmo *Iterative Closest Point* (ICP) [Besl & McKay, 1992]. Apesar de produzir bons resultados esse tipo de captura faz com que a locomoção do veículo seja muito lenta. Montagem semelhante também foi utilizada em [Surmann et al., 2003] para a realização de um mapeamento *indoor*.

Outra forma de se montar o sistema consiste na utilização de dois *lasers* montados ortogonalmente, sendo que um *laser* realiza a leitura no plano horizontal e o outro no plano vertical.



**Figura 2.3.** *Laser sobre plataforma giratória* [Katz et al., 2006].

Em [Howard et al., 2004] podemos observar esse tipo de montagem. O mapeamento é realizado unicamente baseado nas leituras feitas pelo *laser* posicionado verticalmente. A localização do robô é feita através de uma fusão sensorial entre IMU/GPS e um filtro de partículas. É esperado que o ambiente por onde o robô irá trafegar seja parcialmente conhecido, ou seja, contenha algumas características como prédios e objetos estáticos como partes de um mapa prévio. Essas características são então inseridas em um mapa (que não necessita ser muito detalhado) que é fornecido ao robô. O *laser* horizontal é utilizado justamente para a execução do filtro de partículas, no caso o algoritmo usado é o *Monte Carlo Localization* (MCL). A principal desvantagem nessa abordagem está na necessidade de se ter um mapa prévio do ambiente, mesmo que não seja um mapa detalhado. Outro problema encontrado é que apesar de a técnica ter sido desenvolvida para mapeamento *outdoor*, é feita à suposição de que a altura do robô é sempre constante. Além disso, como é utilizado apenas o *laser* vertical o mapa 3D gerado ao final não fornece nenhuma informação a respeito do relevo por onde o robô transitou. A Figura 2.4 exibe a montagem utilizada nesse trabalho.

Como visto nessa seção, os sistemas utilizados para mapeamentos de ambiente através de *laser* se diferenciam bastante em relação à disposição e números de *lasers* utilizados.

Neste trabalho, a montagem consiste em um *laser* fixado à frente do robô e direcionado para o solo e obtém informações 3D baseado na locomoção do robô. Montagens similares à que será utilizada neste trabalho podem ser encontradas na literatura principalmente em sistemas que realizam o mapeamento de forma contínua, ou seja, sistemas onde a navegação do robô não deve ser influenciada.

Como exemplo, podemos citar principalmente automóveis autônomos, como é o caso do veículo Stanley da Universidade de Stanford (Figura 2.5).



**Figura 2.4.** Lasers dispostos ortogonalmente [Howard et al., 2004].



**Figura 2.5.** Stanford Racing Team (Stanley) [Thrun et al., 2006].

## 2.3 SLAM

As Seções 2.1 e 2.2 apresentaram técnicas individuais para sistemas de navegação e mapeamento. Entretanto, uma abordagem que tem se difundido bastante atualmente é a técnica de *Simultaneous localization and mapping* (SLAM). Problemas que envolvem SLAM surgem quando o robô não tem acesso a um mapa do ambiente e também não tem conhecimento da sua posição, logo, durante sua navegação, o robô deve realizar um mapeamento do ambiente onde está e simultaneamente utilizar esse mapa para calcular sua posição [Thrun et al., 2005; Durrant-Whyte & Bailey, 2006].

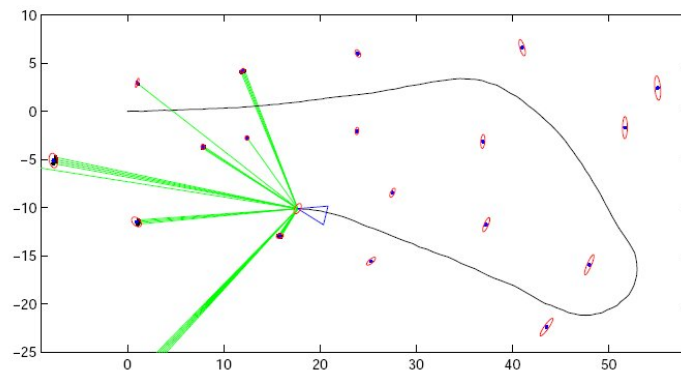
O problema de localização e mapeamento simultâneos é um problema difícil e ainda não está totalmente resolvido, entretanto, existem boas soluções já empregadas em navegações de diferentes tipos de veículos autônomos, como robô aéreos, terrestres e subaquáticos, produzindo resultados bem satisfatórios.

Os principais problemas presentes na utilização dessa técnica estão relacionados ao custo computacional e ao correto relacionamento que deve ser feito entre os novos marcos observados e os já adicionados ao mapa, uma vez que se essa associação for feita

de maneira incorreta o algoritmo de SLAM sendo executado não produzirá os resultados esperados. Já o custo computacional limita a utilização dessa técnica a ambientes de porte moderado, fazendo com que um dos próximos desafios a ser superado seja uma solução de SLAM que permita sua aplicação em grandes ambientes, principalmente em locais onde soluções que utilizam GPS não estão disponíveis [Bailey & Durrant-Whyte, 2006].

Podemos separar as formas de realização de SLAM em três grandes divisões: SLAM 2D, SLAM 2.5D e SLAM 3D. Como o próprio nome sugere, na realização de um SLAM 2D são feitas localização e mapeamento em duas dimensões, enquanto em um SLAM 2.5D é feita uma localização bidimensional, porém é gerado um mapa tridimensional. Já em um SLAM 3D o mapa 3D gerado é o mesmo utilizado para a localização 3D ou 6D do robô.

A Figura 2.6 retirada de [Bailey, 2002] exemplifica a execução de SLAM em duas dimensões. Durante a navegação do veículo ele identifica novos marcos e os associa aos previamente existentes no mapa. Assim, sabendo a posição de cada *landmark* no mapa e também a própria posição, é possível estimar de uma maneira mais precisa sua posição enquanto ele se move. Em [Diosi & Kleeman, 2004] é apresentado um SLAM 2D baseado na fusão das leituras realizadas por sonares e um *laser*.



**Figura 2.6.** SLAM 2D [Bailey, 2002].

O trabalho [Mahon & Williams, 2003] apresenta a realização de um SLAM 2.5D. O sistema é composto por dois *lasers* ortogonais, onde os dados utilizados para a estimativa da posição são obtidos pelo *laser* que realiza as leituras no plano horizontal, e o *laser* orientado verticalmente fornece as informações para a criação do mapa 3D. Como essa técnica realiza o mapeamento 3D independente das informações retornadas pelo SLAM surge o problema de que após ocorrer a correção da posição baseado no SLAM 2D, essa correção não é propagada para os pontos 3D previamente coletados. A Figura 2.7 apresenta o sistema utilizado nesse trabalho.





**Figura 2.7.** Equipamento utilizado para o SLAM 2.5D [Mahon & Williams, 2003].

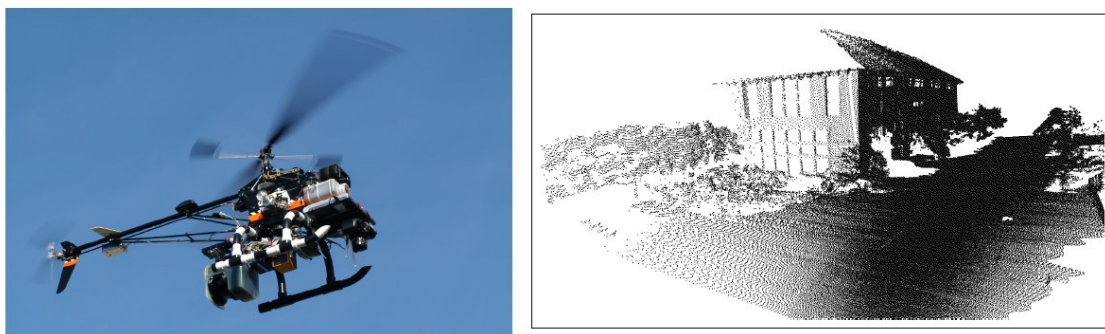
Apesar de um mapa 2D ser suficiente para a localização, a extensão para um mapa 3D do ambiente permite a extração de mais informações. Além disso, uma boa representação 3D do mundo permite a um usuário remoto um melhor entendimento do local por onde o robô está trafegando. As principais dificuldades encontradas ao se utilizar um SLAM 3D estão relacionadas à grande quantidade de dados que passa a estar envolvida no processo, assim como o alto custo computacional para a manipulação desses dados.

As principais diferenças nas montagens vistas na Seção 2.2 também se aplicam à técnica de SLAM 3D, uma vez que inicialmente é feito um mapa do ambiente e em seguida o cálculo da posição baseado nesse mapa. A seguir serão abordados exemplos de diferentes tipos de montagens.

Em [Thrun et al., 2003] a montagem realizada é constituída de um sensor *laser* que se encontra afixado na parte inferior de um modelo de helicóptero, as leituras realizadas pelo *laser* são ortogonais ao plano de locomoção do aeromodelo. É realizado um SLAM probabilístico, onde as leituras realizadas pelo *laser* são alinhadas e a posição do helicóptero é re-calculada simultaneamente. A técnica utilizada apresentou bons resultados, entretanto, sua utilização fica restrita ao helicóptero não passar mais de uma vez pelo mesmo local, o quê é uma premissa indesejável para a locomoção de um robô. A Figura 2.8 exhibe a montagem utilizada e o mapeamento obtido como resultado.

O sistema mais encontrado na literatura consiste em um robô equipado com um *laser* orientado na direção de seu deslocamento, esse *laser* encontra-se sobre uma plataforma móvel giratória, que se movimenta ao longo do eixo de *pitch*. Em sistemas desse tipo a obtenção dos dados ocorre da seguinte maneira, o robô move-se ao longo de uma determinada distância e pára, nesse momento o *laser* é rotacionado para a coleta





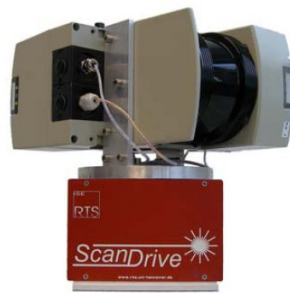
**Figura 2.8.** SLAM 3D com uso de um helicóptero [Thrun et al., 2003].

das informações em uma região especificada. Esses novos dados são alinhados com os dados previamente coletados e fornecem uma melhora na estimativa da posição atual do robô.

Um trabalho extenso sobre essa forma de se realizar SLAM 3D pode ser encontrado em [Nüchter et al., 2007]. Os resultados obtidos para o mapeamento 3D da área são muito bons, entretanto, apesar de ter sido realizada uma navegação *outdoor*, o terreno por onde o robô trafegou durante os experimentos não apresentava grandes variações. Mesmo sendo apresentada a técnica utilizada para a melhoria no cálculo da posição do robô os resultados a respeito da localização do robô se reduziram à apresentação de poucas informações a respeito da posição do robô e nenhuma informação sobre a orientação do mesmo, o que dificulta a avaliação do método para o cálculo da pose do robô.

Visando superar esse problema da falta de uma avaliação quantitativa do desempenho da técnica de SLAM 3D, em [Wulf et al., 2007] é apresentada uma metodologia de *benchmarking* para algoritmos de SLAM, permitindo assim realizar uma comparação entre os resultados obtidos pelo SLAM e valores de referência (*ground truth*). A montagem utilizada para a experimentação consiste em dois *lasers* acoplados com uma diferença de  $180^\circ$  entre seus planos de leitura sobre uma plataforma giratória, como pode ser visto na Figura 2.9. Os dados obtidos permitem avaliar de uma melhor maneira a eficiência da técnica utilizada, entretanto, os valores exibidos são absolutos, não havendo uma distinção para cara eixo de localização e orientação.

Os autores de [Davison et al., 2004] apresentaram uma técnica que utiliza uma câmera com um campo de visão maior que as câmeras convencionais para a realização de um SLAM 3D. Conforme a câmera se movimenta, pontos característicos da imagem são guardados em um mapa e pelo movimento desses pontos é possível detectar a posição relativa da câmera em três dimensões. Um bom resultado foi obtido, entretanto, os experimentos foram realizados em um ambiente controlado e a câmera não estava presa



**Figura 2.9.** *Lasers* com planos de leitura defasados em  $180^\circ$  [Wulf et al., 2007].

sobre um robô, mas sendo manipulada manualmente. Devido a isso não é possível saber o real comportamento que o sistema teria ao ser utilizado em um SLAM 3D de um robô em um ambiente *outdoor*. No caso deste trabalho o foco principal apresentado é realmente a localização da câmera, sendo que o mapa gerado consiste apenas dos marcos detectados.

Após a análise das principais formas de solução para esse problema, é possível então classificar esse trabalho. São realizadas duas etapas (localização e mapeamento), onde a localização é baseada em navegação inercial, e é calculada utilizando o UKF. Para a etapa de mapeamento será utilizado um sensor *laser rangefinder*, fixo, posicionado à frente do robô e direcionado ao solo. Um maior detalhamento do sistema mecânico utilizado será apresentado no Capítulo 4.

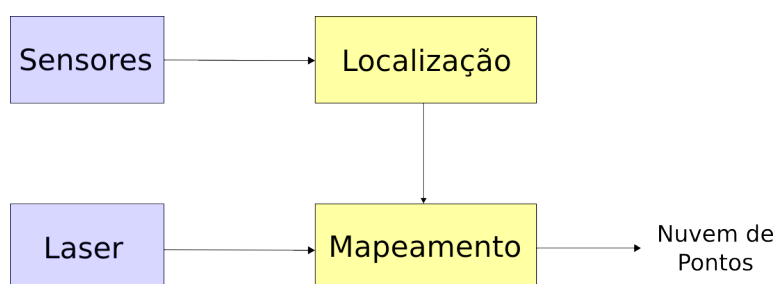
# Capítulo 3

## Metodologia

Este capítulo apresenta a metodologia proposta para a reconstrução do relevo a partir de um robô móvel que navega em terrenos irregulares. O sistema é composto por duas etapas fundamentais:

- **Localização:** Dado um conjunto de sensores ser capaz de gerar uma estimativa precisa para a pose (localização e orientação) do robô em um dado instante de tempo;
- **Mapeamento:** Dada uma localização precisa do robô, a próxima etapa consiste então em a partir das leituras do *laser* gerar um modelo 3D do relevo.

A Figura 3.1 apresenta um diagrama geral do sistema.



**Figura 3.1.** Diagrama geral da metodologia de mapeamento.

O capítulo se encontra organizado da seguinte forma, inicialmente a Seção 3.1 apresenta as técnicas utilizadas para o cálculo da pose, em seguida, é possível encontrar na Seção 3.2 o detalhamento da etapa de mapeamento.

## 3.1 Localização

Nesta seção, inicialmente são descritos os sistemas de coordenadas envolvidas no sistema, em seguida é abordado o UKF, método utilizado para a fusão sensorial e estimação do cálculo da atitude, e finalmente é apresentado como o cálculo da atitude é realizado utilizando-se o estimador de estados visto.

### 3.1.1 Sistemas de Coordenadas

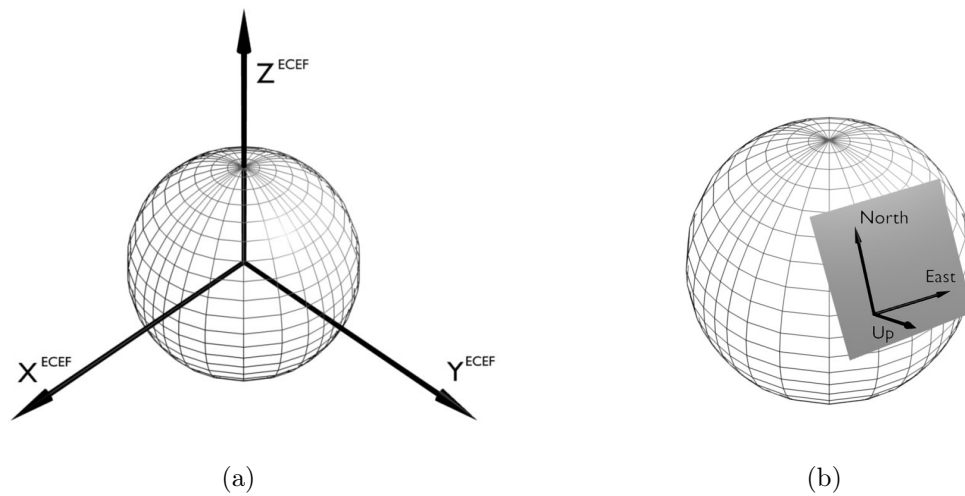
Para se descrever a posição e orientação de um corpo no espaço sempre é necessário lidar com sistemas de coordenadas, e conseqüentemente transformações entre os sistemas envolvidos. Um sistema de coordenadas pode ser entendido como um referencial padrão ao qual determinados pontos estão especificados [Craig, 1989].

Durante a navegação do robô é possível identificarmos três sistemas de referência envolvidos, referencial do mundo: referencial local e o referencial do robô.

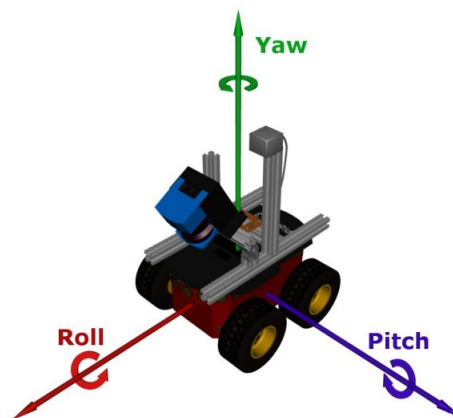
Como estaremos lidando com a posição absoluta do robô na Terra é necessário lidar com um sistema de coordenadas que permita essa representação, no caso o *Earth-Centered, Earth-Fixed* (ECEF). Esse sistema possui sua origem no centro da Terra, sendo que o  $Z$  passa pelo Pólo, e gira junto com a Terra.

Para o referencial local estaremos usando o sistema de referência *Local Tangent Plane* (LTP), que é um modelo onde a Terra é considerada plana. Adotou-se esse sistema uma vez que a locomoção realizada pelo robô é pequena e não sofrerá influência do formato esférico da superfície da Terra. Dentro do sistema LTP será usado o sistema de coordenadas locais *East-North-Up* (ENU), uma vez que a medida em  $Z$  é medida positivamente para cima nesse sistema e possui valor zero no instante inicial da navegação do robô. Tais referenciais estão representados na Figura 3.2.

O sistema de referência *Roll-Pitch-Yaw* (RPY) está fixo ao corpo do robô (porém móvel em relação ao mundo). O eixo  $X$  (*roll*) deve apontar para a direção nominal de movimento do robô (frente), o eixo  $Y$  (*pitch*) é ortogonal ao eixo  $X$  e aponta para o lado esquerdo do veículo, e a partir da regra da mão direita obtém-se o eixo  $Z$  (*yaw*) apontando para cima, obtendo então um valor positivo de orientação a partir de um giro para a esquerda, Figura 3.3. Essa escolha difere dos sistemas normalmente utilizados na literatura em que o eixo  $Z$  está direcionado para baixo [Grewal et al., 2007].



**Figura 3.2.** Referenciais do mundo 3.2(a) ECEF. 3.2(b) ENU.



**Figura 3.3.** Referencial da plataforma.

## 3.1.2 Filtro de Kalman

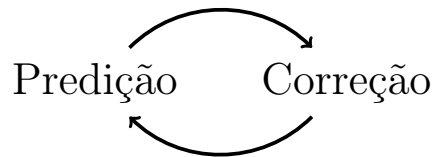
### 3.1.2.1 Introdução

O KF é um dos estimadores de estados mais utilizados atualmente, e provavelmente a técnica de implementação de Filtro de Bayes melhor estudada. O KF consiste em um método recursivo de estimação de estados de um sistema dinâmico sobre a influência de ruído [Choset et al., 2005; Thrun et al., 2005; Welch & Bishop, 1995]. Além disso, o KF é uma ferramenta poderosa também para a fusão de informações obtidas por vários sensores. Consiste em um estimador ótimo para sistemas lineares, e funciona através da minimização do erro quadrático médio.

O KF surgiu a partir dos trabalhos de Peter Swerling [Swerling, 1958] e Rudolf Emil Kalman [Kalman, 1960]. É creditado à Stanley F. Schmidt, pesquisador do Centro de Pesquisa AMES da *National Aeronautics and Space Administration* (NASA), a primeira implementação do KF, essa sendo utilizada no problema de estimação de trajetórias e controle do programa Apollo, levando à sua incorporação em seu computador de navegação [Grewal & Andrews, 2001].

Atualmente, a utilização do KF ocorre nas mais diversas áreas, dentre elas podemos citar sistemas de navegação autônomos, controle de processos químicos, processamento e análise de sinais e imagens, economia, etc.

A execução do KF é um processo recursivo, constituído basicamente de duas etapas fundamentais, Figura 3.4. No primeiro passo temos a etapa de Predição, onde o novo estado é calculado a partir do modelo do sistema, assim como a estimativa da sua incerteza. Em seguida, ocorre a etapa de Correção (atualização da medição), onde o estado estimado na etapa anterior é corrigido utilizando os dados obtidos por uma nova medição realizada pelos sensores.



**Figura 3.4.** Ciclo geral do Filtro de Kalman.

### 3.1.2.2 Sistemas Lineares

A facilidade de implementação do KF para sistemas lineares é uma das principais razões de sua grande utilização.

O KF aborda o problema de estimar o estado  $x \in \mathfrak{R}^n$ , representado pela Equação (3.1), e uma observação  $z \in \mathfrak{R}^m$ , dada pela Equação (3.2),

$$\vec{x}_t = A_t \vec{x}_{t-1} + B_t \vec{u}_t + \vec{\varepsilon}_t, \quad (3.1)$$

$$\vec{z}_t = C_t \vec{x}_t + \vec{v}_t. \quad (3.2)$$

O parâmetro  $A_t$  é uma matriz quadrada de tamanho  $n \times n$ , onde  $n$  é a dimensão do vetor de estados  $\vec{x}_t$ , e representa a matriz de transição.  $B_t$  é a matriz de controle e possui tamanho  $n \times m$ , sendo  $m$  a dimensão do vetor de controle  $\vec{u}_t$ . Já  $C_t$  é a matriz de observação, de tamanho  $k \times n$ , com  $k$  sendo a dimensão do vetor de observação  $\vec{z}_t$ .

Os vetores  $\vec{\varepsilon}_t$  e  $\vec{\nu}_t$  representam respectivamente o ruído do processo e o ruído da observação. Ambos são considerados como ruído branco Gaussiano com média zero, com  $R_t$  sendo a matriz de covariância de  $\vec{\varepsilon}_t$ , e  $Q_t$  a matriz de covariância de  $\vec{\nu}_t$ .

Aqui,  $\vec{x}_t$  e  $\vec{x}_{t-1}$  são os vetores de estado, e  $\vec{u}_t$  é o vetor de controle, sendo todos esses em um determinado instante  $t$  de tempo. Tanto o vetor de estados quanto o vetor de controle são considerados vetores coluna e possuem o formato apresentado como:

$$\vec{x}_t = \begin{bmatrix} x_{1,t} \\ x_{2,t} \\ \vdots \\ x_{n,t} \end{bmatrix}, \quad \text{e}, \quad \vec{u}_t = \begin{bmatrix} c_{1,t} \\ c_{2,t} \\ \vdots \\ c_{m,t} \end{bmatrix}. \quad (3.3)$$

O KF é um processo recursivo e constituído por duas etapas bem definidas, sendo elas:

- **Predição:** Responsável por calcular as melhores estimativas do estado atual e da covariância do erro para esse estado (*a priori*);
- **Correção:** Incorpora à estimativa *a priori* novos dados fornecidos pelos sensores, de forma a gerar uma melhor estimativa do estado atual (*a posteriori*).

A partir dos modelos apresentados, é possível obter o Algoritmo 1 do KF para sistemas lineares.

---

**Algoritmo 1** KF(  $\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$  )

---

- 1:  $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$
  - 2:  $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$
  - 3:  $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$
  - 4:  $\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$
  - 5:  $\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$
  - 6: **return**  $\mu_t, \Sigma_t$
- 

As variáveis  $\bar{\mu}_t$  e  $\mu_t$  são vetores que representam os estados *a priori* e *a posteriori* respectivamente. Da mesma forma temos duas matrizes de covariância que nos fornecem o grau de confiabilidade das estimativas geradas na predição ( $\bar{\Sigma}_t$ ) e correção ( $\Sigma_t$ ).

A variável  $K_t$  é chamada de *ganho de Kalman*, e especifica o grau de influência em que a observação mais recente é então incorporada na estimativa do novo estado.

### 3.1.2.3 Sistemas Não-Lineares

O correto funcionamento do KF está ligado a duas premissas básicas: que as observações são funções lineares do estado e que as transições entre os estados também são representadas por funções lineares.

O KF é um estimador ótimo para sistemas lineares, entretanto, a maior parte das aplicações em robótica são constituídas por sistemas não lineares, como é o caso desse trabalho. Nesta parte estaremos discutindo as duas principais adaptações do KF encontradas na literatura para a solução do problema da restrição de linearidade, o EKF e o UKF.

Inicialmente será analisado o Filtro de Kalman Estendido. A primeira diferença em relação ao KF está no fato de ocorrer um relaxamento na restrição de linearidade, onde o estado e a observação passam agora a ser calculados a partir de duas funções não lineares  $g(\bullet)$  (Equação (3.4)) e  $h(\bullet)$  (Equação (3.5)), respectivamente, entretanto, essas funções devem ser diferenciáveis em relação à variável de estado,

$$\vec{x}_t = g(\vec{u}_t, \vec{x}_{t-1}) + \vec{\varepsilon}_t, \quad (3.4)$$

$$\vec{z}_t = h(\vec{x}_t) + \vec{v}_t. \quad (3.5)$$

As funções  $g(\bullet)$  e  $h(\bullet)$  não podem ser usadas sobre as matrizes de covariância diretamente, mas antes devem ser calculadas as matrizes de derivadas parciais (jacobianas) ao redor do estado estimado. Temos então as jacobianas para o cálculo do estado (Equação (3.6)) e da observação (Equação (3.7)):

$$\nabla g_t = \frac{\partial g(\vec{u}_t, \vec{x}_{t-1})}{\partial \vec{x}_{t-1}}, \quad (3.6)$$

$$\nabla h_t = \frac{\partial h(\vec{x}_t)}{\partial \vec{x}_t}. \quad (3.7)$$

Mesmo sendo bastante utilizado, o EKF não é um estimador ótimo. Além disso, possui como um dos principais problemas apenas a propagação da média através de funções não lineares, o que pode gerar uma grande ineficiência quando utilizado para funções extremamente não lineares [Julier & Uhlmann, 1996], o cálculo da jacobiana também pode se tornar uma tarefa difícil dependendo da complexidade das funções envolvidas.

Considerado essas dificuldades, uma nova especialização do KF para sistemas não lineares foi desenvolvida, o UKF [Julier & Uhlmann, 1997].



O UKF faz uso de uma técnica de amostragem determinística responsável por calcular as estatísticas de uma variável aleatória que sofre uma transformação não linear [Julier & Uhlmann, 1997], essa técnica é chamada de Transformação *Unscented* (*Unscented Transform*, ou UT). A partir da UT é gerado então um conjunto de pontos de amostragem (estados) ao redor da média, esses pontos são chamados de Pontos Sigma (*Sigma Points*, ou SP).

A partir de uma gaussiana  $n$ -dimensional de média  $\mathbf{v}$  (estaremos representando essa variável por um vetor de estados  $\vec{x}$  de dimensão  $n$ ) e possuindo covariância  $\mathbf{P}$ , temos  $2n + 1$  pontos Sigma que são obtidos a partir das Equações (3.8):

$$\mathcal{X}^{[0]} = \vec{x}, \quad (3.8a)$$

$$\mathcal{X}^{[i]} = \vec{x} + \left( \sqrt{(n + \lambda)P} \right)_{[i]} \quad \text{para } i = 1, \dots, n, \quad (3.8b)$$

$$\mathcal{X}^{[i]} = \vec{x} - \left( \sqrt{(n + \lambda)P} \right)_{[i-n]} \quad \text{para } i = n+1, \dots, 2n. \quad (3.8c)$$

A notação  $[i]$  significa a  $i$ -ésima coluna da matriz resultante do cálculo de  $\sqrt{(n + \lambda)P}$ . O fator  $\lambda$  é definido de acordo com a Equação (3.9),

$$\lambda = \alpha^2(n + \kappa) - n, \quad (3.9)$$

onde  $\alpha$  e  $\kappa$  são fatores de escala que determinam a distância em que os pontos Sigma estarão dispersos da média. Geralmente assume-se para  $\alpha$  um valor positivo pequeno (*e.g.*,  $10^{-3}$ ), já  $\kappa$  costuma ter seu valor fixado em  $\kappa = 0$ .

Cada ponto Sigma possui associado a si dois pesos, representados por  $w_m^{[i]}$  e  $w_c^{[i]}$ , calculados a partir das Equações (3.10), com  $\beta = 2$  ótimo para distribuições gaussianas [Wan & van der Merwe, 2000]:

$$w_m^{[0]} = \frac{\lambda}{n + \lambda}, \quad (3.10a)$$

$$w_c^{[0]} = \frac{\lambda}{n + \lambda} + (1 - \alpha^2 + \beta), \quad (3.10b)$$

$$w_m^{[i]} = w_c^{[i]} = \frac{1}{2(n + \lambda)} \quad \text{para } i = 1, \dots, 2n. \quad (3.10c)$$

Os pontos Sigma calculados são então propagados por uma função  $f(\bullet)$  (Equação (3.11)), resultando em uma única estimativa ponderada pelos pesos previamente calculados,

$$\Upsilon^{[i]} = f(\mathcal{X}^{[i]}). \quad (3.11)$$

O novo estado ( $\vec{x}'$ ) e sua covariância ( $P'$ ) podem então ser calculados pela Equação (3.12) e Equação (3.13), respectivamente,

$$\vec{x}' = \sum_{i=0}^{2n} w_m^{[i]} \Upsilon^{[i]}, \quad (3.12)$$

$$P' = \sum_{i=0}^{2n} w_c^{[i]} (\Upsilon^{[i]} - \vec{x}') (\Upsilon^{[i]} - \vec{x}')^T. \quad (3.13)$$

Dado o modelo matemático é possível então apresentar o Algoritmo 2 para o UKF.

---

**Algoritmo 2** UKF(  $\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$  )

---

- 1:  $\mathcal{X}_{t-1} = (\mu_{t-1} \quad \mu_{t-1} \pm \sqrt{(n+\lambda)\Sigma_{t-1}})$
  - 2:  $\bar{\mathcal{X}}_t^* = g(\mathcal{X}_{t-1}, u_t)$
  - 3:  $\bar{\mu}_t = \sum_{i=0}^{2n} w_m^{[i]} \bar{\mathcal{X}}_t^{*[i]}$
  - 4:  $\bar{\Sigma}_t = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\mathcal{X}}_t^{*[i]} - \bar{\mu}_t) (\bar{\mathcal{X}}_t^{*[i]} - \bar{\mu}_t)^T + R_t$
  - 5:  $\bar{\mathcal{X}}_t = (\bar{\mu}_t \quad \bar{\mu}_t \pm \sqrt{(n+\lambda)\bar{\Sigma}_t})$
  - 6:  $\bar{\mathcal{Z}}_t = h(\bar{\mathcal{X}}_t)$
  - 7:  $\hat{z}_t = \sum_{i=0}^{2n} w_m^{[i]} \bar{\mathcal{Z}}_t^{[i]}$
  - 8:  $S_t = \sum_{i=0}^{2n} w_c^{[i]} (\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t) (\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)^T + Q_t$
  - 9:  $K_t = \left[ \sum_{i=0}^{2n} w_c^{[i]} (\bar{\mathcal{X}}_t^{[i]} - \bar{\mu}_t) (\bar{\mathcal{Z}}_t^{[i]} - \hat{z}_t)^T \right] S_t^{-1}$
  - 10:  $\mu_t = \bar{\mu}_t + K_t(z_t - \hat{z}_t)$
  - 11:  $\Sigma_t = \bar{\Sigma}_t - K_t S_t K_t^T$
  - 12: **return**  $\mu_t, \Sigma_t$
- 

Tanto o UKF quanto o EKF possuem a mesma complexidade assintótica, e para sistemas não lineares o UKF produz resultados tão bons ou melhores que o *EKF* [Thrun et al., 2005; Orderud, 2005]. Outra vantagem do UKF é que esse não necessita da computação das jacobianas, aumentando assim sua facilidade de implementação.

Devido às vantagens verificadas optou-se por utilizar nesse trabalho o UKF para a fusão dos dados e cálculo da posição do robô.

### 3.1.3 Cálculo da Atitude

Na seção anterior foi apresentado de forma geral o funcionamento do UKF. Portanto, agora é necessário determinar os modelos específicos envolvidos a serem utilizados para esse trabalho.

A Figura 3.5 apresenta o diagrama detalhado desta etapa. Primeiramente ocorre a filtragem dos sinais da IMU e da bússola, em seguida ocorre a fusão dos dados utilizando o UKF.

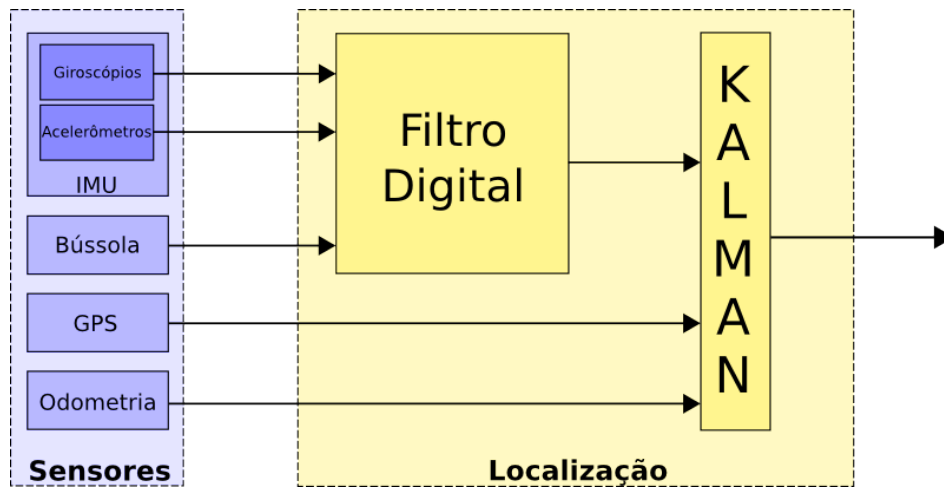


Figura 3.5. Diagrama detalhado da etapa de localização.

Decidiu-se que o vetor de estados deveria conter apenas as informações de localização  $(x, y, z)$  e orientação  $(\phi, \theta, \psi)$ , tendo então a forma apresentada na Equação (3.14),

$$\vec{x} = \begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^T. \quad (3.14)$$

A partir de um dado estado deve ser fornecido o vetor de controle para que seja feita a predição do próximo estado. O vetor de controle a ser utilizado contém as informações de  $g_x$ ,  $g_y$  e  $g_z$ , que correspondem às velocidades angulares em torno dos eixos de rolamento, arfagem e guinada, respectivamente, e são obtidas pelos giroscópios da IMU, além do valor da velocidade linear  $v$  fornecido pela odometria.

Temos então o vetor de entrada do modelo apresentado na Equação (3.15),

$$\vec{u} = \begin{bmatrix} g_x & g_y & g_z & v \end{bmatrix}^T. \quad (3.15)$$

Os valores  $g_x$ ,  $g_y$ ,  $g_z$  e  $v$  são recuperados no instante de tempo  $t$ . Utilizando então as informações passadas através do vetor de controle é possível calcular a estimativa do estado atual a partir das Equações (3.16), são elas:

$$x_t = x_{t-1} + \cos \psi_t \cos \theta_t v \Delta t, \quad (3.16a)$$

$$y_t = y_{t-1} + \sin \psi_t \cos \theta_t v \Delta t, \quad (3.16b)$$

$$z_t = z_{t-1} + \sin \theta_t v \Delta t, \quad (3.16c)$$

$$\phi_t = \phi_{t-1} + (g_x + g_y \tan \theta_{t-1} \sin \phi_{t-1} + g_z \tan \theta_{t-1} \cos \phi_{t-1}) \Delta t, \quad (3.16d)$$

$$\theta_t = \theta_{t-1} + (g_y \cos \phi_{t-1} - g_z \sin \phi_{t-1}) \Delta t, \quad (3.16e)$$

$$\psi_t = \psi_{t-1} + \left( g_y \frac{\sin \phi_{t-1}}{\cos \theta_{t-1}} - g_z \frac{\cos \phi_{t-1}}{\cos \theta_{t-1}} \right) \Delta t. \quad (3.16f)$$

O cálculo de  $\psi_t$  apresenta pontos de singularidade para  $\theta_{t-1} = \pm\pi/2$ , entretanto, esse fato não será relevante no contexto desse trabalho, uma vez que o robô não deve obter tais valores de *pitch* durante sua navegação.

O próximo passo a ser realizado é a etapa de correção, que ocorre a partir dos valores presentes no vetor de observações. As informações fornecidas são  $x_{GPS}$  e  $y_{GPS}$  obtidos do GPS, os valores das acelerações nos três eixos locais dados pelos acelerômetros da IMU, e também o valor absoluto de orientação  $\psi_b$  fornecido pela bússola.

A bússola retorna valores no intervalo  $[0, 2\pi[$ , onde o 0 representa o Norte magnético da Terra e os valores aumentam no sentido horário, porém, no sistema geométrico convencional é considerado que um giro no sentido anti-horário deve produzir valores positivos de orientação, logo, temos então que  $\psi_b = \pi/2 - \psi_{br}$ , onde  $\psi_{br}$  significa o valor *raw* de orientação retornado pela bússola.

Além disso, outro fator relacionado à bússola e que deve ser tratado é o fato de que o Norte magnético referenciado pela bússola não necessariamente é o Norte geográfico da Terra (*true north*). Essa diferença entre os valores que o campo magnético da Terra sofre é chamada de variação ou declinação magnética, e está diretamente ligada ao tempo e localização no globo terrestre [Everett, 1995]. Para obtermos o valor do Norte real, basta realizarmos o cálculo Norte Real = Norte Magnético - Variação, onde o valor da variação para localização onde ocorreu o desenvolvimento desse trabalho foi obtido em [NOAA Satellite and Information Service, 2008], e possui valor de  $\approx 21,7^\circ$ .

O vetor de observação é representado pela Equação (3.17),

$$\vec{z} = \begin{bmatrix} x_{GPS} & y_{GPS} & a_x & a_y & a_z & \psi_b \end{bmatrix}^T. \quad (3.17)$$

Os valores estimados para  $x, y$  e  $\psi$  na etapa de predição são avaliados diretamente pelos valores da observação  $x_{GPS}, y_{GPS}$  e  $\psi_b$ . Os valores de  $\phi, \theta$  são corrigidos através do cálculo de  $\phi_a$  e  $\theta_a$ , obtidos a partir das Equações (3.19) e (3.18), respectivamente,

$$\theta_a = \arcsin\left(\frac{a_x}{g_r}\right), \quad (3.18)$$

$$\phi_a = \arcsin\left(\frac{a_y}{-g_r \cos \theta_a}\right). \quad (3.19)$$

Os cálculos baseiam-se no vetor aceleração da gravidade, entretanto durante experimentos realizados observou-se que as variáveis  $a_x$  e  $a_y$  podem obter valores maiores que a aceleração da gravidade ( $g \approx 9,78 \text{ m/s}^2$ ) caso o terreno onde o robô trafega seja extremamente acidentado, o que gera uma inconsistência trigonométrica nas Equações (3.18) e (3.19). Para solucionar tal problema optou-se por realizar os cálculos utilizando como valor do vetor aceleração da gravidade a resultante das acelerações,  $g_r = \sqrt{a_x^2 + a_y^2 + a_z^2}$ .

Na literatura, sistemas que integram IMU/GPS podem ser classificados como *tightly* ou *loosely coupled* [Grewal et al., 2007]. Em sistemas do tipo *loosely coupled* tanto a IMU quanto o GPS são tratados como sistemas de navegação independentes, as informações de cada um dos sensores são processadas separadamente em diferentes filtros para em seguida ser realizada a fusão. Já sistemas *tightly coupled* são aqueles que utilizam apenas um filtro que recebem como entrada os dados fornecidos por ambos os sensores. O sistema utilizado nesse trabalho pode então ser classificado como *tightly coupled*.

Aparelhos de GPS são amplamente utilizados em veículos aéreos, e nesses casos dificilmente ocorrem interrupções de sinal (sendo que o GPS precisa de no mínimo quatro satélites para que a posição possa ser calculada). Entretanto, ao utilizar esse equipamento em veículos terrestres, problemas de queda de sinal tendem a ser mais constantes, decorrentes das características do ambiente onde o veículo está trafegando e que podem obstruir esse sinal.

Os veículos onde também esses aparelhos são utilizados possuem um fator em comum que é se deslocarem a uma velocidade da grandeza de dezenas ou centenas de quilômetros por hora. Quando um GPS se encontra se movimentando a baixas velocidades (ou parado) surge um problema denominado *random walk*, onde valores aleatórios dentro de um raio passam a ser observados.

Os dados utilizados na etapa de correção e que são obtidos a partir do GPS são extremamente dependentes do ambiente, uma vez que determinados obstáculos (árvores, prédios, túneis, etc) podem impedir o aparelho de receber o sinal dos satélites.

Para que o sistema tenha certa tolerância a falhas do GPS (perda de sinal e *random walk*), é proposto então um modelo para o cálculo dinâmico da variância do GPS. Optou-se por desenvolver um modelo próprio uma vez que as informações de qualidade fornecidas pelo GPS não se mostraram confiáveis.

As variâncias individuais das informações obtidas a partir do GPS ( $x_{GPS}, y_{GPS}$ ) serão calculadas em cada instante de tempo e fornecidas à matriz de covariância, onde as variâncias serão iguais para as duas informações e são inversamente proporcionais ao número de satélites e à velocidade do robô em um dado momento.

A parcela da variância associada à influência do números de satélites ( $\mathcal{C}_n$ ) e velocidade ( $\mathcal{C}_v$ ) são calculadas separadamente através das Equações (3.20) ( $n$  sendo o número de satélites visíveis) e (3.21) ( $v$  sendo a velocidade linear do robô) respectivamente, e depois somadas para se obter o valor da variância dos valores obtidos do GPS e que devem ser atribuídas à matriz de covariância, Equação (3.22), onde:

$$\mathcal{C}_n = \begin{cases} \frac{P_n \times M_n}{n} & \text{se } n > 3, \\ \infty & \text{caso contrário} \end{cases}, \quad (3.20)$$

$$\mathcal{C}_v = \begin{cases} \frac{P_v \times M_v}{v} & \text{se } \geq v_{min}, \\ \infty & \text{caso contrário} \end{cases}, \quad (3.21)$$

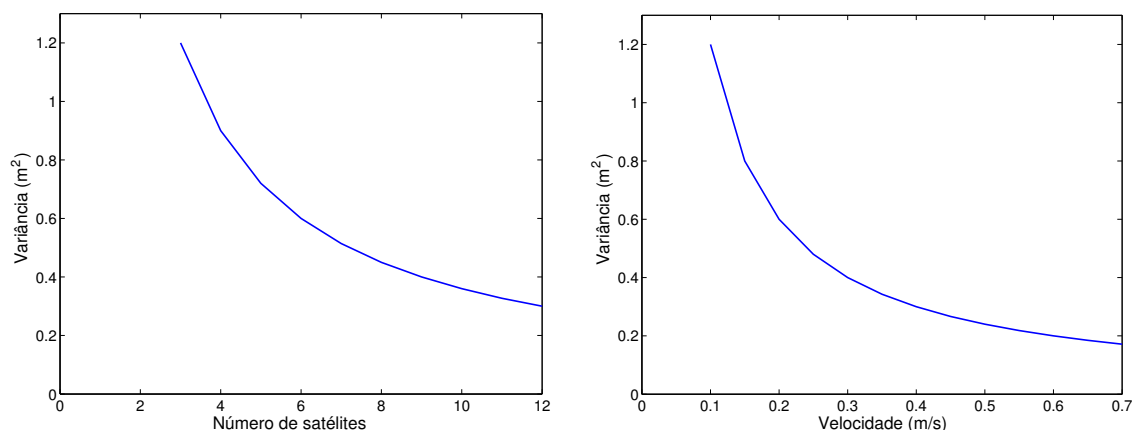
$$\mathcal{C}_{GPS} = \mathcal{C}_n + \mathcal{C}_v. \quad (3.22)$$

Temos que  $P_n + P_v = 1$ , e representam a porcentagem de influência no sistema que se deseja atribuir a cada fator (número de satélites, velocidade), e  $M_n$  e  $M_v$  representam valores médios esperados no sistema para esses mesmos fatores.

A Figura 3.6 mostra o comportamento da variância que será atribuída a  $x_{GPS}$  e  $y_{GPS}$  com a variação do número de satélites e velocidade do robô.

Caso o GPS tenha em sua linha de visada zero satélites os valores obtidos devem ser desconsiderados, logo, é colocado um valor muito alto de variância e o UKF tende a dar um peso muito baixo para esses novos dados coletados. Da mesma forma, caso o robô esteja desempenhando uma velocidade muito baixa (ou parado) também deve-se aumentar a variância, diminuindo-se assim a influência do efeito de *random walk*.

Os valores de variância para os demais sensores foram considerados constantes durante a navegação do robô. Além disso, o ajuste da matriz de covariância foi realizado de forma empírica.



**Figura 3.6.** Cálculo da variância dos dados do GPS baseado no número de satélites e velocidade do robô.

### 3.1.3.1 Pré-Filtragem

Os valores de velocidade angular e aceleração retornados pela IMU e a orientação absoluta obtida através da bússola possuem um alto grau de ruído, o que acaba por diminuir a precisão do cálculo da posição. Assim, antes dos valores serem utilizados no UKF eles passam por um filtro digital, reduzindo assim a quantidade de ruído inserido no sistema.

Não deve ser realizada uma filtragem agressiva nesse primeiro momento, uma vez que há uma dificuldade em diferenciar o que efetivamente é o ruído do sinal e o que faz parte da movimentação do robô (devido ao terreno irregular).

### 3.1.3.2 Tratamento do *Bias*

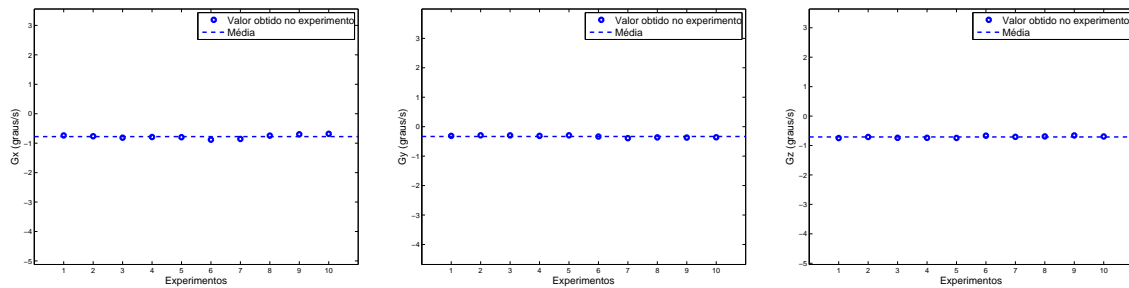
Um dos grandes problemas encontrados na utilização de sensores inerciais está no *bias* (ou polarização) presente nos sinais, principalmente nos sinais dos giroscópios, que provoca, entre outros efeitos, o problema de deriva no processo de integração dos dados.

O *bias* consiste em um erro de um determinado valor presente nos dados dos sensores. Nos giroscópios, o *bias* consiste em um valor de leitura não nula mesmo quando nenhum movimento angular está sendo realizado. Para solucionar esse problema deve ser realizado um processo de calibração, para que, depois de identificado, esse valor possa ser compensado posteriormente na etapa de cálculo da atitude. Entretanto, não é possível solucionar totalmente esse problema, uma vez que no caso mais geral o *bias* não é fixo, podendo variar devido a mudanças de temperatura ou ao se desligar e religar o equipamento.

Para estimar-se esse valor para cada um dos giroscópios presentes na IMU o seguinte método de calibração é proposto. A plataforma onde o sensor se encontra deve estar em repouso e a IMU alinhada paralelamente em relação ao solo, de forma que os valores retornados pelos acelerômetros nos eixos X e Y estejam com média zero.

Como os valores de *bias* podem variar foram realizados um total de dez experimentos divididos em dois grupos, onde cada grupo de cinco experimentos foi realizado em dias diferentes. Em seguida, para cada experimento é calculada a média dos valores para cada giroscópio, e finalmente encontra-se a média entre todos os experimentos realizados, e essa média é o valor de *bias* que deve ser retirado do sinal.

Os dados obtidos durante a etapa de calibração são apresentados na Figura 3.7.



**Figura 3.7.** Cálculo do *bias* dos giroscópios.

É possível observar que os valores de *bias* encontrados para os giroscópios do eixos X e Z chegam quase ao valor de  $1^\circ/s$ , o quê pode levar a grandes erros de deriva durante a etapa de integração.

## 3.2 Mapeamento

Conforme apresentado nas Seções 2.2 e 2.3, diferentes soluções para o problema de mapeamento podem ser encontradas na literatura. Entretanto, grande parte das abordagens é feita com o robô parado e o *laser* girando de forma a criar uma leitura tridimensional da região, em seguida o robô se desloca um determinado trecho e uma nova leitura é feita, de forma que uma parte dessa nova região analisada seja correspondente a alguma parte da leitura anterior. Essa abordagem, onde robô anda e pára consecutivamente, não é muito interessante, prejudicando assim a eficiência da navegação. Entretanto, bons resultados de mapeamento podem ser alcançados, onde comumente é utilizado o algoritmo ICP, que baseia-se no fato de se realizar várias leituras de uma mesma região e em seguida essas leituras são então alinhadas.



Nesse trabalho é utilizado um único *laser* e que permanecerá fixo ao robô (estático), esse realiza uma leitura 2D do ambiente e a tridimensionalidade é conseguida através da locomoção do robô. Esse tipo de abordagem é amplamente utilizada para a navegação de automóveis autônomos [Buehler et al., 2007], entretanto devido à grande quantidade de ruído presente nesse tipo sistema costuma-se utilizar vários *lasers* para uma melhor detecção do solo, entretanto, apesar disso, acredita-se não ser possível a recuperação de um modelo 3D preciso do mundo a partir apenas dessas informações ruidosas [Stavens & Thrun, 2006].

Logo, nas próximas seções são apresentadas as etapas realizadas visando a melhoria da qualidade do mapa obtido durante a navegação do robô, permitindo assim sua utilização em diversas aplicações.

Inicialmente, é explicado o processo de coleta e o formato dos dados retornados a partir de uma leitura realizada pelo sensor *laser rangefinder*. Em seguida, é apresentada a filtragem realizada sobre essa leitura. Após isso, são apresentadas as transformações de referencial utilizadas para que as leituras locais do *laser* estejam dispostas em um modelo único no referencial global.

A Figura 3.8 apresenta a seqüência de passos presentes durante esta etapa.

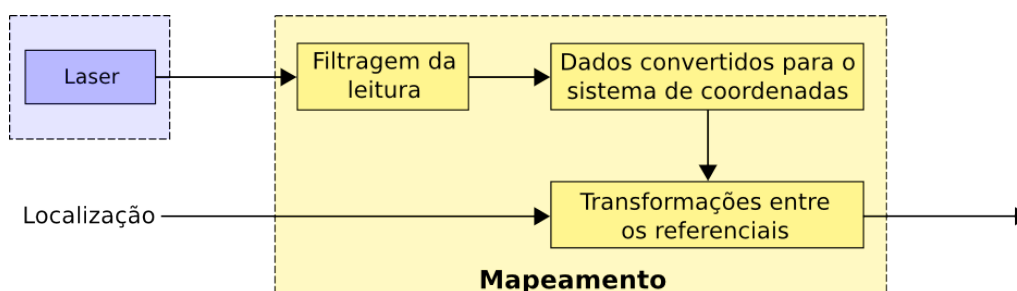
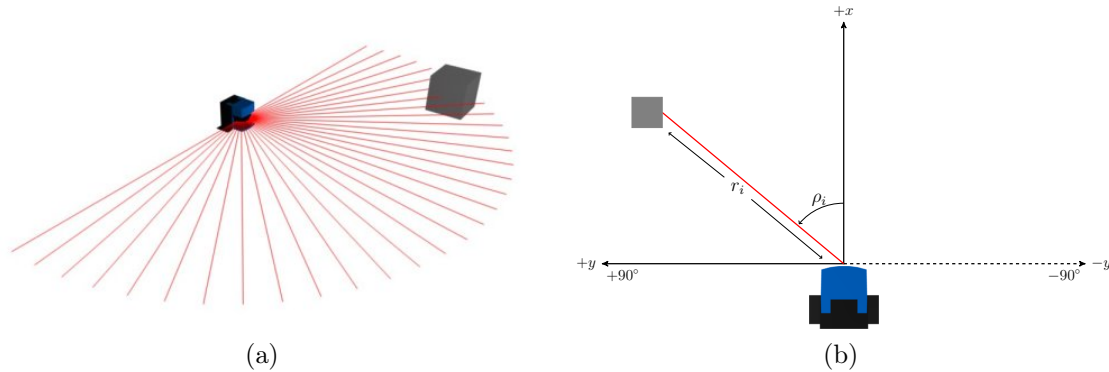


Figura 3.8. Diagrama detalhado da etapa de mapeamento.

### 3.2.1 Leitura do *Laser*

Após realizada a leitura do *laser*, temos como resultado um vetor contendo duplas no formato  $(\rho_i, r_i)$ , onde  $\rho$  representa a angulação onde determinado valor foi obtido e encontra-se no intervalo  $[-\pi, \pi]$ ,  $r$  a distância em metros do *laser* a qual se encontra o obstáculo detectado, e  $i$  a  $i$ -ésima posição do vetor.

O tamanho do vetor está diretamente relacionado à resolução angular utilizada durante a leitura, podendo ter 361 valores (resolução de  $0,5^\circ$ ) ou 181 valores (resolução de  $1^\circ$ ). A Figura 3.9 apresenta o esquema de uma leitura realizada pelo *laser*, permitindo assim uma melhor compreensão desses valores.



**Figura 3.9.** Esquema de leitura do *laser*. 3.9(a) Representação da detecção de um obstáculo. 3.9(b) Variáveis envolvidas na leitura.

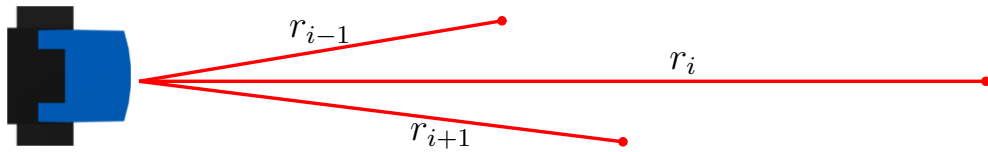
### 3.2.2 Filtragem da Leitura

Para esse trabalho, optou-se por utilizar a maior resolução disponível para o *laser*, 1 mm, o que faz com que o alcance máximo de leitura seja de  $\approx 8$  m, logo, valores acima desse limite em sua maioria representam leituras incorretas. Entretanto, é possível observar dois casos em que esse fato ocorre e que necessitam de um tratamento específico.

Após uma leitura completa, temos então um conjunto de valores de distância que foram obtidos a partir da reflexão do feixe de *laser* em um obstáculo. Entretanto, não necessariamente o valor obtido na leitura é válido. Isso ocorre devido a alguns problemas que podem surgir durante a realização da leitura, dentre eles podemos citar, a incidência do feixe sobre uma quina (o valor retornado pode estar no intervalo entre a distância do início e do final da quina), incidência da luz do sol diretamente sobre a frente do equipamento (nenhum valor é retornado uma vez que o valor medido é considerado inválido), influência na reflexão do feixe pelo objeto (o valor retornado se encontra próximo ao limite porém esse valor está acima do esperado). Serão tratados o primeiro e último casos citados.

A diferença entre as distâncias medidas por dois pontos vizinhos de uma mesma leitura pode ser grande (tendo como limite o alcance máximo do *laser*), entretanto, ao se considerar três pontos vizinhos podemos assumir que o valor do ponto central não deverá se encontrar muito distante de seus pontos vizinhos, isso porque considerando que os pontos possuem uma separação angular entre si de  $0,5^\circ$  (resolução angular do *laser*), então a distância entre três pontos consecutivos é de  $\approx 14$  cm (considerando o alcance máximo de  $\approx 8$  m).

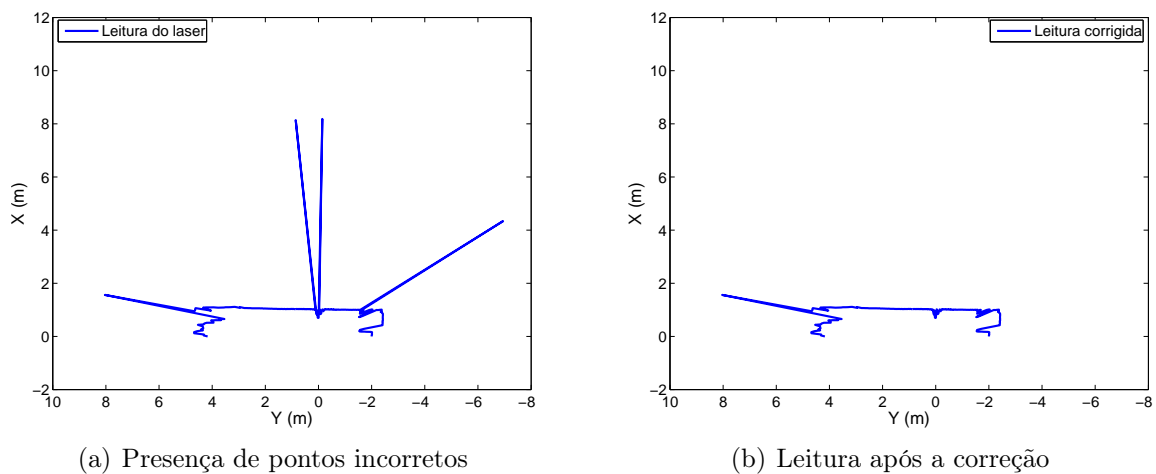
Dessa forma, é estipulado um valor limite para a diferença entre um ponto e seus vizinhos ( $\eta$ ) (Figura 3.10). Caso a diferença para ambos vizinhos esteja acima desse valor então será considerado que o valor de distância para aquele ponto é dado pela média dos valores medidos de seus vizinhos (Equação (3.23)). Com isso, visamos reduzir possíveis erros que possam ocorrer pelas situações previamente descritas.



**Figura 3.10.** Distância considerada entre um ponto e seus vizinhos.

$$\hat{r}_i = \frac{r_{i-1} + r_{i+1}}{2}, \quad \text{se } |r_i - r_{i-1}| > \eta \text{ e } |r_i - r_{i+1}| > \eta \quad (3.23)$$

A Figura 3.11(a) apresenta um exemplo de leitura do *laser* com a presença de pontos medidos incorretamente, e na Figura 3.11(b) temos o resultado da leitura após a execução da filtragem proposta, os pontos foram ligados e os eixos variam entre os gráficos para uma melhor visualização da correção realizada.



(a) Presença de pontos incorretos

(b) Leitura após a correção

**Figura 3.11.** Correção realizada para leituras do *laser* com a presença de pontos medidos erroneamente.

É possível observar que a filtragem conseguiu eliminar os pontos com medições incorretas, permitindo assim deixar o mapa com mais detalhes do ambiente, entretanto um ponto ainda persistiu devido ao limiar escolhido ( $\eta = 6$  m).

A plataforma *Player* utilizada retorna um valor acima do alcance máximo citado caso nenhum obstáculo seja encontrado dentro do alcance máximo do *laser*, logo, após a execução do passo anterior caso algum ponto ainda possua um valor de distância acima de 8 m ele deve então ser descartado.

Essa etapa possui como principal objetivo diferenciar e identificar pontos cujo feixe de *laser* efetivamente incidiu sobre um obstáculo (porém teve sua reflexão influenciada pelo objeto) de pontos que não representam uma leitura real, e com isso aumentar a quantidade de informações do mapa reduzindo a influência do ambiente sobre as leituras *laser*.

### 3.2.3 Transformações entre os Referenciais

Na geração do mapa temos basicamente três diferentes referenciais envolvidos, são eles, o referencial do mundo  $\{W\}$  (referencial fixo ao plano da Terra LTP) que não sofre nenhuma variação, o referencial que está fixo ao corpo do robô  $\{R\}$  (RPY) e por último o referencial fixo ao *laser*  $\{L\}$ , sendo esses dois últimos referenciais móveis.

No caso deste projeto temos um robô móvel que durante sua navegação irá possuir diferentes posições e orientações em relação ao referencial fixo ao plano da Terra, dessa forma, é necessário realizar as transformações dos pontos no plano de leitura do *laser* para o referencial do mundo.

A transformação entre dois referenciais é definida por um conjunto de translações e rotações ocorridas nos três eixos e que são representadas pelos seguintes operadores:

$$D_x(x) = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad D_y(y) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad D_z(z) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$R_x(\alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha & 0 \\ 0 & \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad R_y(\beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

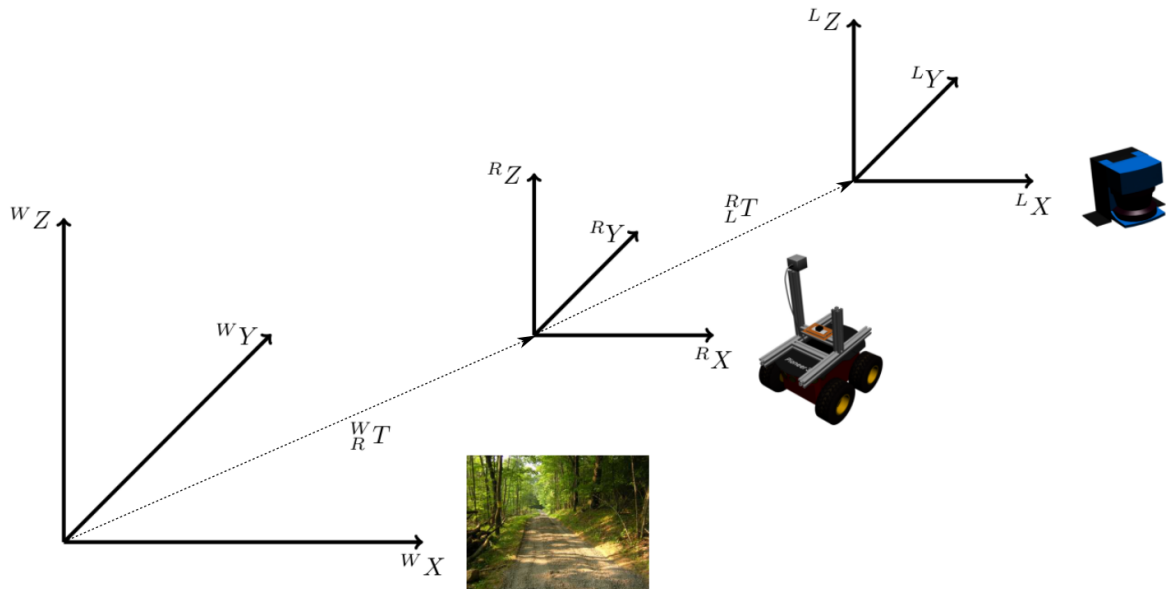
$$R_z(\gamma) = \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 & 0 \\ \sin \gamma & \cos \gamma & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

onde  $D$  é o operador de translação em relação a um eixo, já  $R$  é o operador de rotação ao redor de um eixo.

A partir desses operadores é possível então obter o operador  $T$ , responsável por representar em uma única matriz um conjunto de translações ( $D$ ) e rotações ( $R$ ), sendo definido então pela Equação (3.24),

$$T = \left[ \begin{array}{ccc|c} R & & & D \\ \hline 0 & 0 & 0 & 1 \end{array} \right]. \quad (3.24)$$

Existem duas transformações básicas e que podem ser definidas, a transformação do referencial do mundo para o robô ( ${}^W_R T$ ) e do referencial do robô para o do *laser* ( ${}^R_L T$ ). A Figura 3.12 apresenta em linhas gerais as transformações entre os referenciais presentes no sistema, e que serão detalhadas nas próximas seções.



**Figura 3.12.** Transformações entre os referenciais envolvidos.

É necessário, entretanto, que primeiramente os dados sejam convertidos de informações de distância para o sistema cartesiano. Realizada a leitura do *laser*, o vetor é percorrido e cada dupla  $i$  sofre então uma transformação geométrica, de forma que possa ser representada como coordenadas cartesianas referentes ao plano de leitura do *laser*.

Essa transformação é realizada através dos cálculos vistos na Equação (3.25),

$$\begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} = r_i \begin{bmatrix} \cos \rho_i \\ \sin \rho_i \\ 0 \end{bmatrix}. \quad (3.25)$$

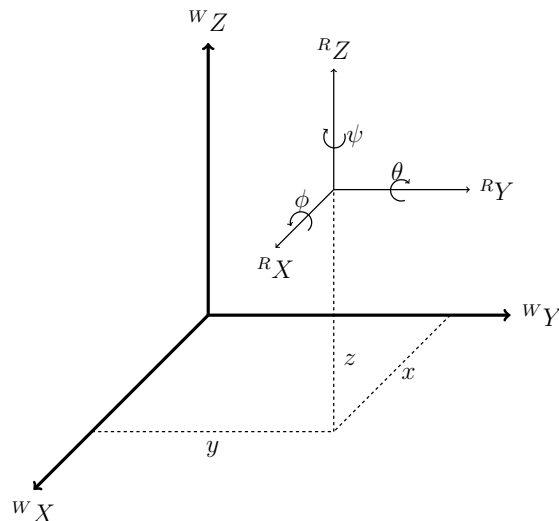
Inicialmente  $z$  é colocado com valor 0 pois o ponto encontra-se no sistema de referência do plano de leitura do *laser*.

### 3.2.3.1 Transformação do Referencial do Mundo para o do Robô

O referencial do mundo é fixo e considerado no ponto inicial da navegação do robô, é importante ressaltar que caso o robô no momento inicial esteja posicionado em um terreno inclinado o referencial do mundo não possuirá essa rotação.

O referencial do robô encontra-se no centro geométrico, porém com altura zero, ou seja, encontra-se no ponto onde suas rodas tocam o solo. Por trafegar em terrenos irregulares, durante sua locomoção o robô irá sofrer variações de translação e rotação em todos os eixos, logo, sua configuração é dada pelo vetor  $\begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix}^T$ , os ângulos de orientação consistem nas rotações em torno dos eixos  ${}^W X$ ,  ${}^W Y$  e  ${}^W Z$ , respectivamente.

A Figura 3.13 exemplifica essa mudança de referencial.



**Figura 3.13.** Transformação do referencial do mundo para o do robô.

Primeiramente deve-se realizar as rotações e em seguida as translações. Utilizando ângulos fixo, temos que realizar uma rotação ao redor do eixo  $X$ , em seguida do eixo  $Y$  e então do eixo  $Z$ .

Logo, considerando um ponto  ${}^R P$  no referencial do robô, a transformação necessária para representá-lo no referencial do mundo é dada pela Equação (3.26),

$${}^W P = {}^W_R T {}^R P \text{ onde,} \quad (3.26)$$

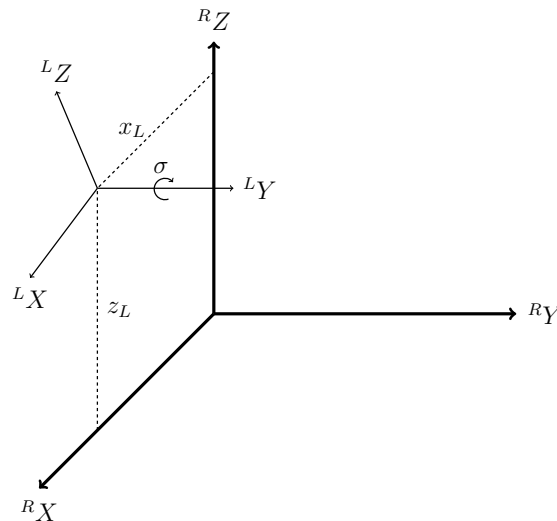
$${}^W_R T = \begin{bmatrix} \cos \theta \cos \psi & -\cos \phi \sin \psi + \sin \phi \sin \theta \cos \psi & \sin \phi \sin \psi + \cos \phi \sin \theta \cos \psi & x \\ \cos \theta \sin \psi & \cos \phi \cos \psi + \sin \phi \sin \theta \sin \psi & -\sin \phi \cos \psi + \cos \phi \sin \theta \sin \psi & y \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta & z \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

### 3.2.3.2 Transformação do Referencial do Robô para o do *Laser*

Conforme visto na Seção 3.2.3.1 o referencial do robô encontra-se no seu centro à altura do solo. Logo, como o *laser* utilizado para a varredura encontra-se deslocado em relação a esse referencial, é necessário uma transformação para o referencial do *laser*. O referencial do *laser* é fixo em relação ao robô, ou seja, os valores utilizados para a translação e rotação são constantes, não sofrendo alterações durante a navegação.

O *laser* encontra-se deslocado em relação ao eixo  ${}^R X$  (mais à frente do robô) e ao eixo  ${}^R Z$  (altura) do robô. Além disso, ele encontra-se inclinado para frente (rotação ao redor do eixo  ${}^R Y$ ) em direção ao solo.

A Figura 3.14 exemplifica essa mudança de referencial.



**Figura 3.14.** Transformação do referencial do robô para o do *laser*.

Logo, considerando um ponto  ${}^L P$ , que está no referencial do *laser* e foi obtido pela Equação (3.25), a transformação necessária para representá-lo no referencial do robô é dada pela Equação (3.27),

$${}^R P = {}^R T^L P \text{ onde,} \quad (3.27)$$

$${}^R T^L = \begin{bmatrix} \cos \sigma & -\sin \sigma & 0 & x_L \\ \sin \sigma & \cos \sigma & 0 & 0 \\ 0 & 0 & 1 & z_L \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$



# Capítulo 4

## Experimentos

Neste capítulo são apresentados os resultados obtidos utilizando-se a metodologia apresentada no capítulo anterior. A parte de experimentos divide-se em duas seções específicas para experimentos de localização e mapeamento, possibilitando assim um maior detalhamento de cada etapa separadamente.

Todos os experimentos foram realizados utilizando-se o sistema real e de forma “*offline*” para pós-processamento, ou seja, durante a navegação o robô coleta e salva todos os dados fornecidos pelos sensores para que depois seja realizada a manipulação sobre essas informações.

### 4.1 Planejamento

Os experimentos possuem como principal objetivo a validação da metodologia proposta, ou seja, medir a precisão com que a localização é feita, assim como a confiabilidade do mapa gerado.

Na parte de localização inicialmente são apresentados os experimentos realizados em um ambiente controlado e *indoor*, visando verificar-se principalmente a precisão no cálculo dos ângulos de orientação. Essa etapa é fundamental uma vez que a orientação do robô possui influência direta na localização, e conseqüentemente na qualidade do mapa que será feito posteriormente. Durante essa etapa, também deve-se verificar a precisão no cálculo do deslocamento do robô no eixo  $Z$ , uma vez que esse parâmetro não possui nenhum valor de correção e é estimada com base apenas no ângulo de *pitch* e na velocidade do robô.

Para os experimentos internos o robô foi programado a executar um deslocamento por uma determinada distância fixa, de forma a fazer com que as repetições fossem semelhantes.

Em seguida são apresentados os experimentos realizados de forma *outdoor*, principalmente para verificar-se a localização absoluta utilizando-se o GPS. Durante os experimentos externos o robô navegou pelo ambiente guiado através de um *joystick*.

Da mesma forma, a etapa de mapeamento deve inicialmente ser testada em um ambiente controlado e de relevo plano onde são considerados então todos os sensores (exceto GPS), finalmente passa-se à etapa de experimentos em ambientes externos.

## 4.2 Arcabouço Experimental

### 4.2.1 Localização

O principal foco desse trabalho na etapa de localização está em realizar a navegação em ambientes externos e que possuam terrenos irregulares, logo, é necessário que o robô esteja equipado com sensores que forneçam todos os dados necessários que permitam um bom cálculo da atitude do veículo durante a realização de sua trajetória.

Para o cálculo da sua atitude o robô utiliza os dados fornecidos por uma IMU, um GPS, uma bússola digital, além de informações da Odometria do robô, a Figura 4.1 apresenta a disposição dos sensores sobre a plataforma.

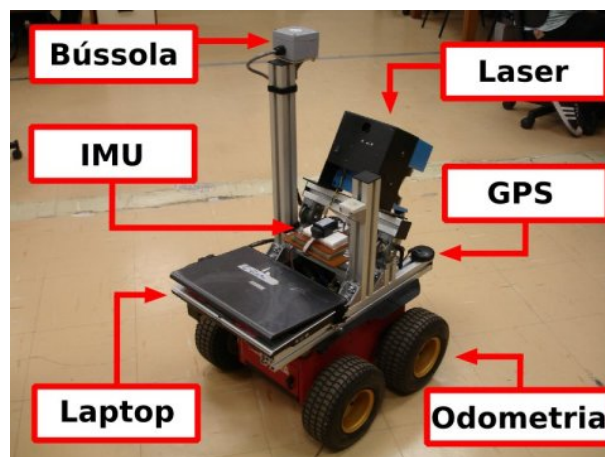


Figura 4.1. Equipamentos utilizados no trabalho.

A IMU (Cloud Cap Crista IMU) possui três giroscópios mutuamente ortogonais que fornecem a medida das velocidades angulares do robô em torno dos três eixos, e três acelerômetros também ortogonais que informam os valores das acelerações lineares existentes nos três eixos do referencial do robô. De forma a simplificar os cálculos assume-se que a IMU encontra-se exatamente no centro de rotação do robô, o que não necessariamente é verdadeiro. Como a diferença é pequena e a velocidade de

locomoção do robô é baixa a abordagem adotada funciona bem. Também é assumido que o referencial fixo ao corpo do robô é equivalente ao referencial da IMU.

Como a IMU está montada sobre uma base rígida (o corpo do robô) o sistema aqui utilizado pode então ser considerado como sendo do tipo *strapdown* [Titterton & Weston, 2004].

A partir do GPS (Garmin 18x-5Hz) é possível obter os valores absolutos da posição geográfica do robô, essas informações serão utilizadas no formato *Easting* e *Northing* (sistema de coordenadas *Universal Transverse Mercator* (UTM)), e são dadas em metros.

A bússola digital (KVH C-100) é responsável por fornecer valores absolutos para a orientação do robô de acordo com o norte magnético da Terra, ela está disposta sobre uma plataforma um pouco distante do robô para evitar influências magnéticas provindas desse.

O robô utilizado é um modelo voltado para aplicações externas e possui uma boa capacidade de carga (Pioneer P3-AT), permitindo assim a montagem de uma plataforma em alumínio onde os sensores utilizados estão dispostos. A coleta dos dados é realizada por um *laptop*, também responsável por enviar os comandos ao robô.

As frequências de amostragem dos equipamentos utilizados são para a IMU, GPS, bússola e sistema robótico (robô) respectivamente 20Hz, 5Hz, 10Hz e 20Hz. Durante a coleta dos dados um *timestamp* é associado a cada nova leitura realizada.

Demais detalhes técnicos dos equipamentos utilizados no projeto podem ser vistos no Apêndice A.

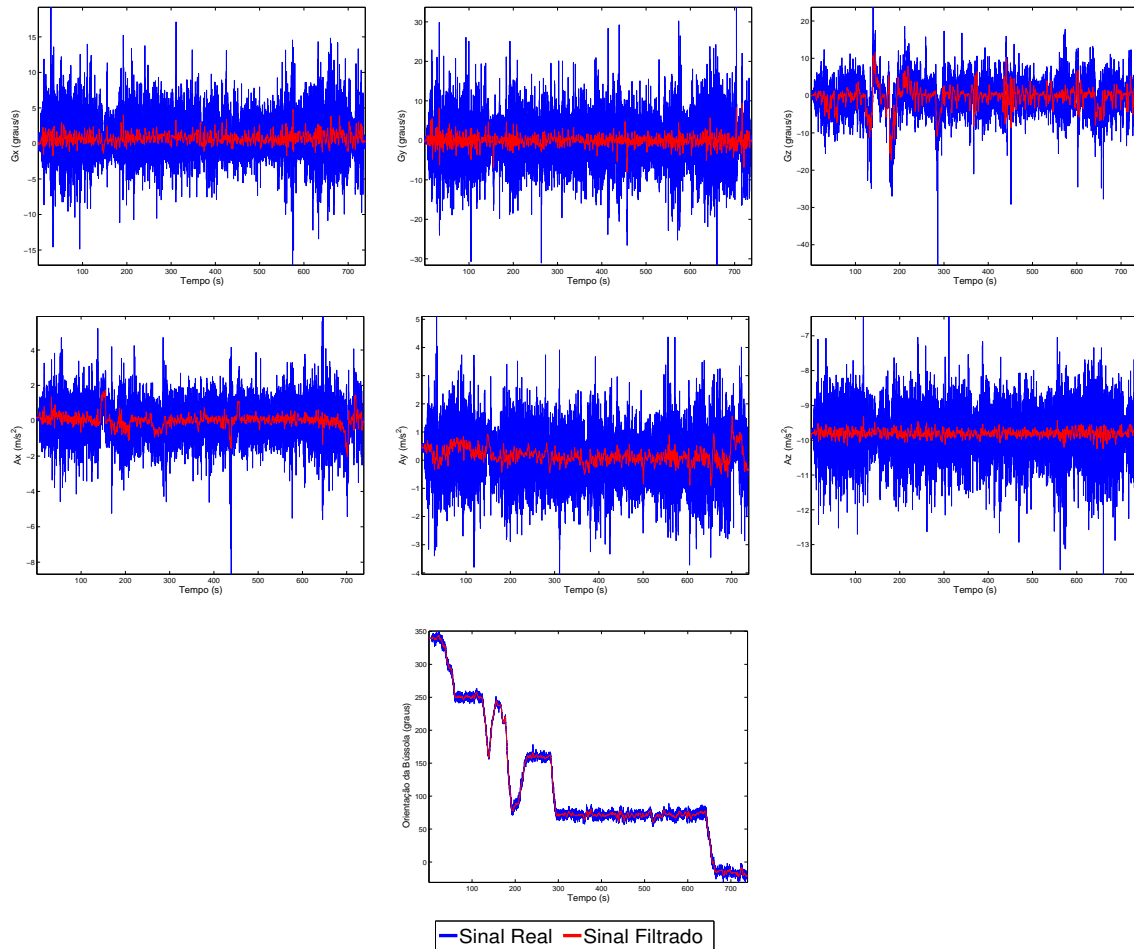
#### 4.2.1.1 Calibração

A seguir são apresentados os valores específicos para a instanciação experimental aqui utilizada e atribuídos a determinados parâmetros apresentados na seção anterior.

O teorema da amostragem (Nyquist) nos diz que a máxima frequência representável deve ser metade da frequência de amostragem do sinal. Logo, os valores de corte deveriam ser escolhidos nos intervalos de 0 a 10Hz (IMU), e de 0 a 5Hz (bússola). Os valores foram escolhidos de forma empírica após uma análise espectral do sinal.

Foi utilizado um filtro Butterworth passa-baixa de segunda ordem com frequência de corte de 5Hz (IMU) e 2,5Hz (bússola).

A Figura 4.2 apresenta os dados coletados durante a execução de um experimento realizado em um ambiente externo, onde é possível observar os perfis dos sinais retornados pela IMU e pela bússola, e que os resultados obtidos a partir da filtragem são bastante satisfatórios conseguindo reduzir a dispersão dos dados.



**Figura 4.2.** Pré-filtragem dos dados da IMU e da bússola. A primeira linha apresenta os dados recuperados a partir dos giroscópios, a segunda linha os dados dos acelerômetros, e finalmente a orientação obtida a partir da bússola

Para a etapa adaptativa da variância do GPS decidiu-se por dar um peso maior no sistema para o número de satélites em visada, então temos  $P_n = 0,6$  e  $P_v = 0,4$ . Já os valores médios esperados específicos para o sistema aqui utilizado são  $M_n = 6$  e  $M_v = 0,3$  (apesar de nominalmente o robô desempenhar velocidades de até  $0,7$  m/s durante a realização dos experimentos verificou-se que o robô não alcançava valores maiores que  $0,6$  m/s).

## 4.2.2 Mapeamento

Apesar de experimentalmente terem sido observados bons resultados obtidos para a localização calculada através da metodologia proposta descrita na Seção 3.1.3, há a presença residual de uma pequena quantidade de ruído nos ângulos de orientação. Esse ruído acaba por influenciar a etapa de transformações entre os referenciais, fazendo com

que a reconstrução não represente fielmente o mundo real.

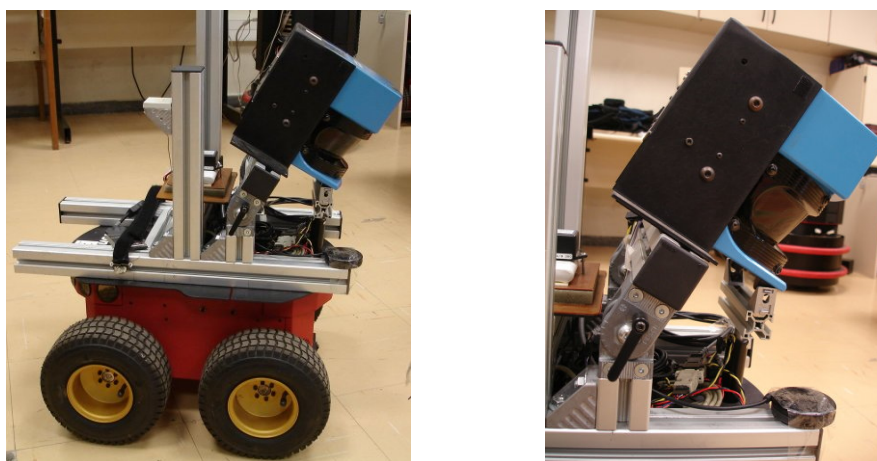
Portanto, uma filtragem é realizada sobre essas informações, tendo como principal objetivo uma suavização das estimativas de rotação do robô, e conseqüentemente diminuindo a variação entre leituras consecutivas do *laser*, para isso, foi utilizando um filtro Butterworth passa-baixa de segunda ordem com frequência de corte de 1 Hz.

Visando reduzir também uma quantidade desnecessária de dados de uma mesma posição, as leituras serão analisadas apenas se corresponderem a uma nova posição no espaço. Dessa forma, caso o robô possua velocidade linear e angular nulas as leituras são descartadas.

Para a coleta das informações do relevo é utilizado um sensor *laser*, que é um sensor do tipo *rangefinder*. Sensores desse tipo são utilizados para medir a distância em que determinado obstáculo está do local do sensor. No caso do *laser* essa medida é realizada através de um espelho giratório e que viabiliza a leitura em um campo de 180°.

Como nosso principal interesse está em mapear o relevo do ambiente, o *laser* está posicionado na parte frontal do robô e com uma certa inclinação, ou seja, direcionado ao solo. Essa inclinação é fixa e definida anteriormente à navegação do robô.

A Figura 4.3 destaca a montagem do *laser* sobre o robô. É utilizada uma base girante fixa que permite ao *laser* estar inclinado em direção ao solo.



**Figura 4.3.** Destaque da montagem do *laser* sobre o robô.

Nesse trabalho não será abordado qual a melhor estrutura de dados a ser utilizada para armazenar o mapa gerado. Todos os pontos obtidos após a etapa de mapeamento são salvos de forma sequencial em um arquivo de texto.

### 4.2.2.1 Calibração

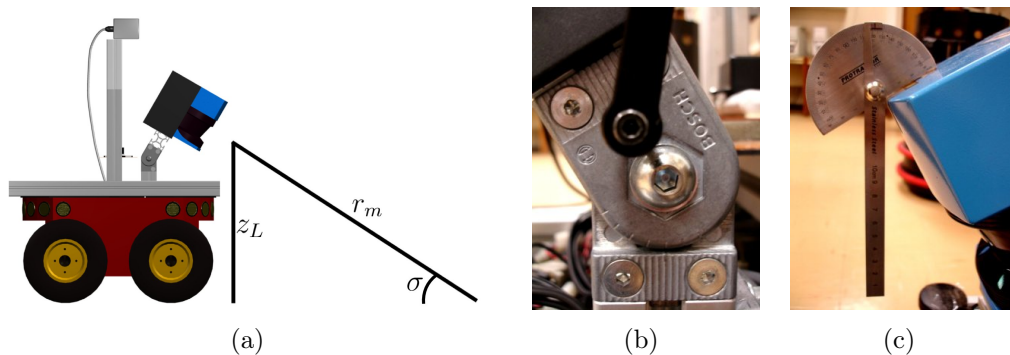
O conhecimento da inclinação do *laser* é de grande importância para a geração do mapa, e como não se dispunha de nenhum equipamento que permitisse uma medição direta e precisa dessa inclinação foram utilizados três métodos distintos para se estimar a inclinação do *laser*.

O primeiro método consiste em uma calibração via *software* utilizando os dados do *laser*. O robô deve ser colocado em uma região que possua o piso plano, onde são obtidas diversas medidas de uma mesma área que se encontra à frente do robô. A partir dessas medidas é possível calcular a distância média  $r_m$  em que o *laser* está do piso. A altura  $z_L$  que o *laser* se encontra em relação ao solo é medida diretamente, e permite realizar o cálculo da inclinação do *laser* pela Equação (4.1),

$$\sigma = \arcsin\left(\frac{z_L}{r_m}\right). \quad (4.1)$$

As outras duas formas de se obter a inclinação do *laser* consistem na medição direta. A primeira medição é feita utilizando-se a própria escala presente na base que prende o *laser* ao robô, essa escala possui marcações a cada  $15^\circ$ , o que produz uma boa estimativa para inclinações múltiplas desse valor, mas não para valores intermediários. A outra forma de se medir a inclinação foi utilizando-se um transferidor apoiado sobre o *laser* e uma régua posicionada verticalmente, dessa forma basta medir a variação do valor medido em relação à  $90^\circ$ .

Cada um dos três métodos possui erros inerentes a si, logo, é calculada a média dos três valores, obtendo-se então um valor com uma boa precisão. A Figura 4.4 destaca cada um desses métodos.



**Figura 4.4.** Cálculo da inclinação do *laser* 4.4(a) Calibração baseada nos dados do *laser*. 4.4(b) Escala demarcada na base de sustentação. 4.4(c) Utilização de um esquadro.

Após a realização da calibração foi possível obter os seguintes valores:  $x_L = 0,25$  m,  $z_L = 0,53$  m e com um ângulo  $\sigma \approx -30^\circ$  de rotação ao redor do eixo  $^R Y$ .

## 4.3 Localização

### 4.3.1 Experimentos Internos

Os experimentos iniciais realizados de forma *indoor* não farão uso do GPS, ou seja, toda informação de localização diz respeito a uma localização relativa do robô baseada apenas nos ângulos de orientação calculados através da metodologia proposta e da velocidade do robô.

O ambiente onde esses experimentos foram realizados possui o piso plano e variações de relevo (rampas) de valores conhecidos.

#### 4.3.1.1 Cálculo do *Roll*

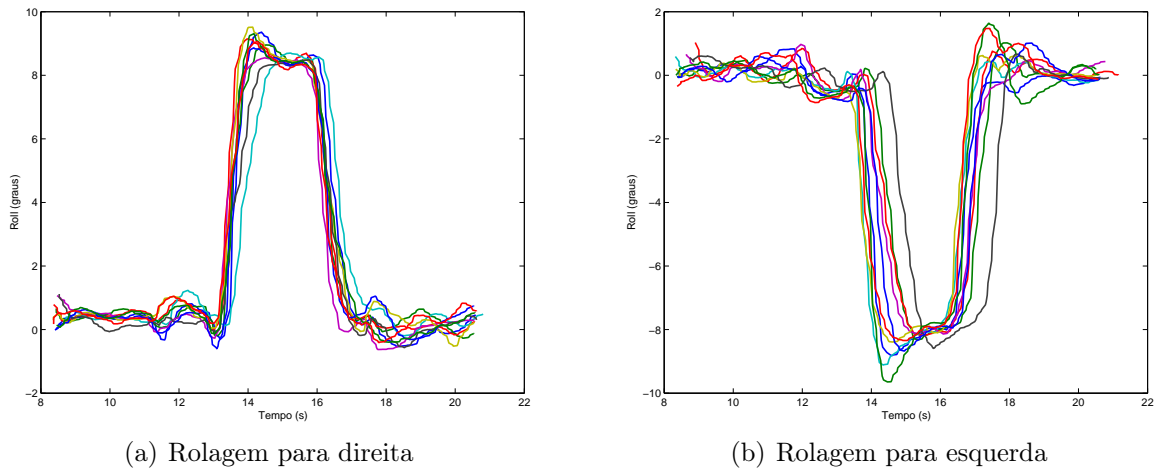
Para se validar o cálculo de  $\phi$ , uma pequena rampa de altura conhecida foi disposta em um local com o piso plano e de forma que o robô passasse apenas com duas rodas sobre ela, o que gera um movimento lateral e conseqüentemente uma variação no ângulo de rolagem. Conhecendo a altura da rampa é possível obter que ao passar na rampa  $\phi \approx \pm 8^\circ$ .

O experimento foi executado um total de vinte vezes, sendo que dez vezes o robô sofreu uma inclinação para a direita (valores positivos de  $\phi$ ), e dez vezes o robô fez um rolamento para a esquerda (valores negativos).

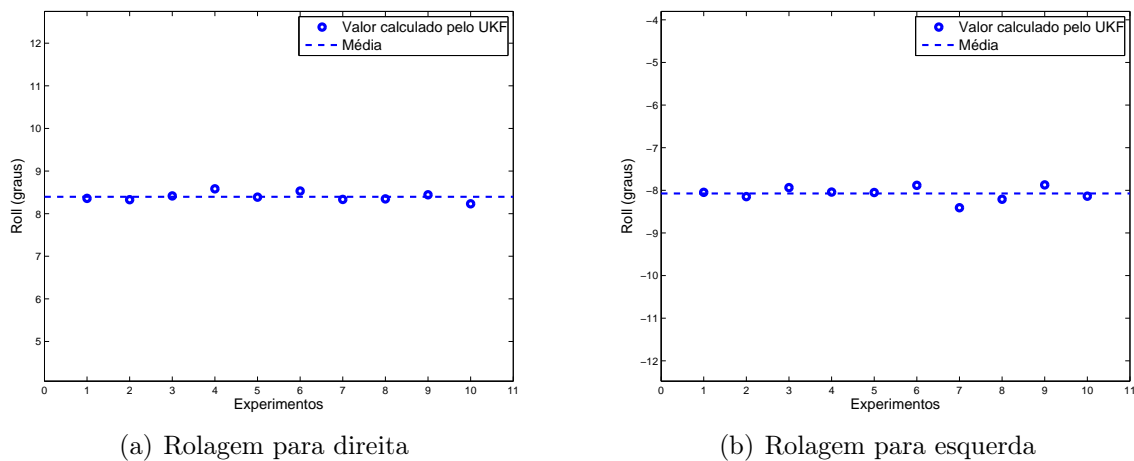
A Figura 4.5 apresenta os dados obtidos durante os experimentos. É possível observar que ao se realizar o movimento de rolagem temos um pico no valor de  $\phi$ , isso se deve principalmente ao movimento brusco realizado pelo robô nesse momento, após isso temos então que o valor se estabiliza próximo ao esperado. Após a estabilização a média é calculada.

A Figura 4.6 apresenta a média para cada experimento após o robô já estar sobre a rampa, assim como também a média geral de todas as execuções. Para valores de giro para a direita foi obtido o valor médio de  $\bar{\phi}_d = 8,39^\circ$ , já para os giros à esquerda encontrou-se  $\bar{\phi}_e = -8,07^\circ$ .

Os valores encontrados foram satisfatórios e a metodologia se mostrou estável, recuperando-se mesmo após um movimento brusco como o visto no início do movimento.



**Figura 4.5.** Experimentos de rolagem utilizando-se uma rampa para validação do *roll*.



**Figura 4.6.** Experimentos de rolagem utilizando-se uma rampa para validação do *roll*.

Um fato interessante a ser destacado é que, conforme visto na Seção 3.1.3, o ângulo de *roll* ( $\phi$ ) não exerce nenhuma influência no cálculo da posição do robô. Logo, a precisão desse parâmetro está relacionado apenas à qualidade do mapeamento final gerado.

#### 4.3.1.2 Cálculo do *Pitch*

A precisão no cálculo do ângulo  $\theta$  é de fundamental importância para uma boa localização do robô, uma vez que o cálculo da variação na altura (eixo Z) do robô depende exclusivamente de  $\theta$  e da velocidade linear.

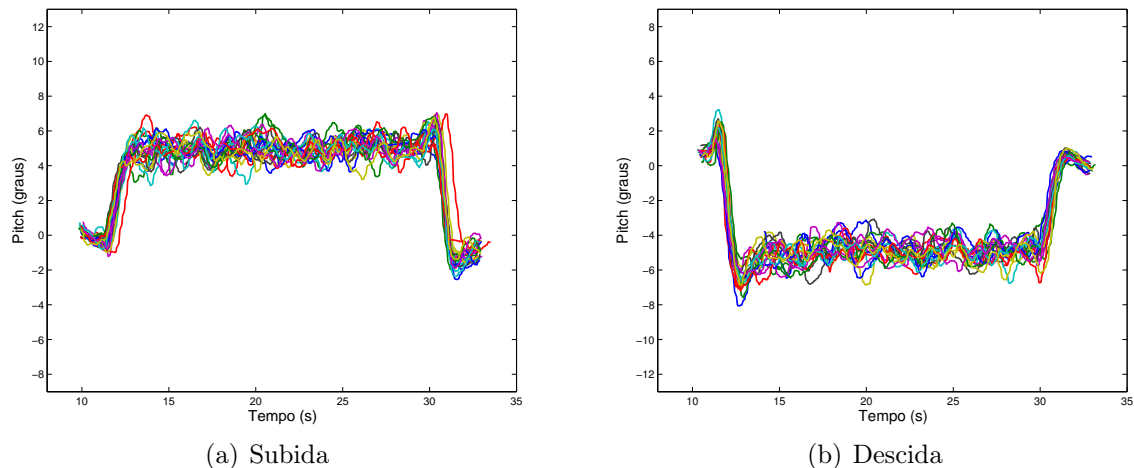


O experimento consistiu em fazer com que o robô realizasse um deslocamento de uma determinada distância, onde inicialmente o robô percorre um plano sem variação de nível, em seguida inicia a subida por uma rampa de inclinação conhecida, e por fim alcança novamente um plano de inclinação nula. Quando o robô estiver totalmente sobre a rampa ele deve possuir um ângulo de inclinação de  $\theta \approx \pm 5^\circ$ .

Da mesma forma o robô também percorre o caminho inverso descendo a rampa, para que se possa verificar o cálculo da inclinação tanto para valores positivos quanto negativos de *pitch*.

Para cada formato de experimento foram realizadas vinte repetições, permitindo assim eliminar falsos resultados devidos a possíveis erros espúrios durante a realização dos experimentos.

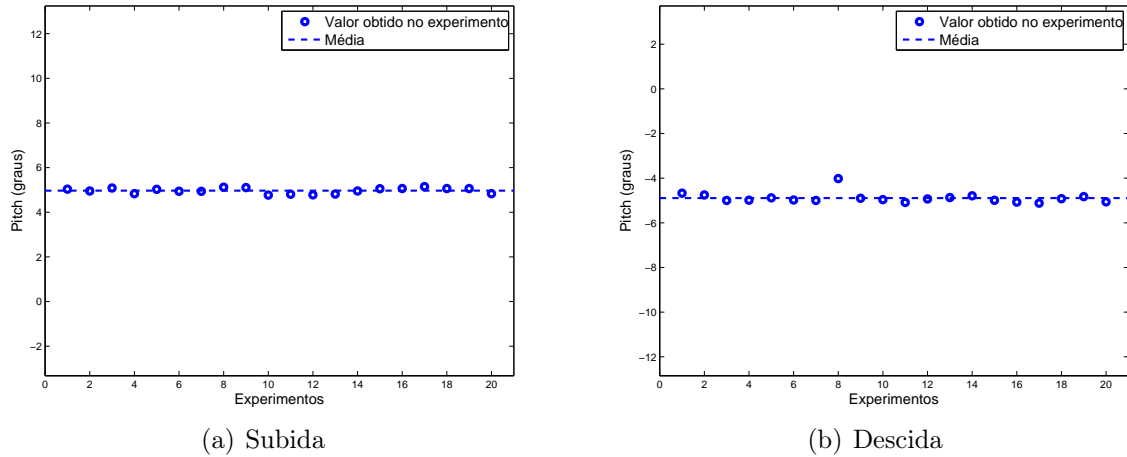
A Figura 4.7 apresenta a variação no tempo do ângulo  $\theta$  do robô. Os picos nas extremidades dos gráficos são devidos principalmente a imperfeições na entrada e saída da rampa. Como pode-se observar o valor é ruidoso, logo, para a análise foi utilizado o valor médio de  $\theta$  durante o experimento.



**Figura 4.7.** Experimentos de subida/descida de uma rampa para validação do *pitch*.

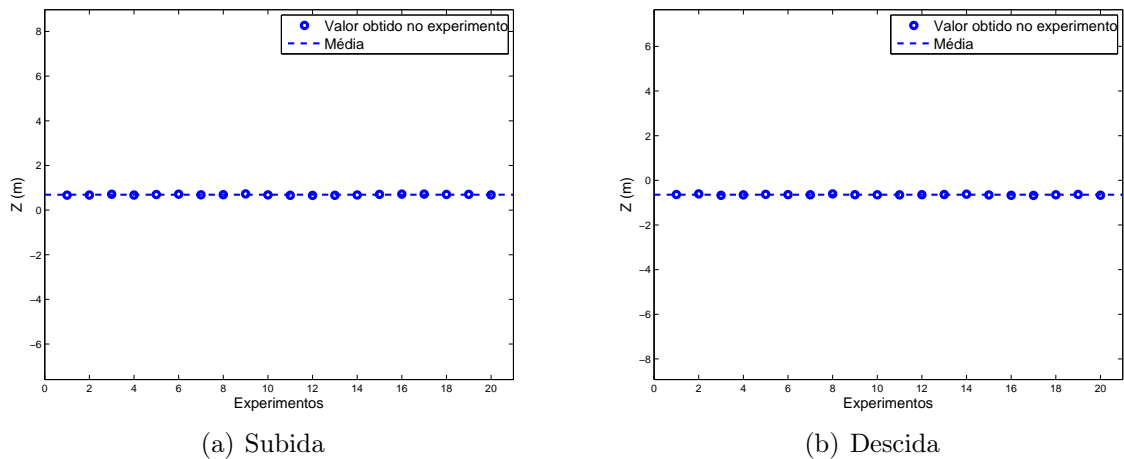
Cada ponto na Figura 4.8 representa a média do valor de  $\theta$  que o robô teve enquanto estava completamente sobre a rampa ( $15 < t < 30$ ). Os experimentos foram executados várias vezes para que não houvessem falsas conclusões devido a algum problema na realização dos experimentos, o que entende-se foi a razão de no experimento oito da Figura 4.8(b) foi encontrado um valor tão destoante dos demais.

Os valores obtidos estavam bem próximos ao valor referência, sendo que a média obtida para todos os experimentos foi  $\bar{\theta}_s = 4,97^\circ$  e  $\bar{\theta}_d = -4,89^\circ$ .



**Figura 4.8.** Experimentos de subida/descida de uma rampa para validação do *pitch*.

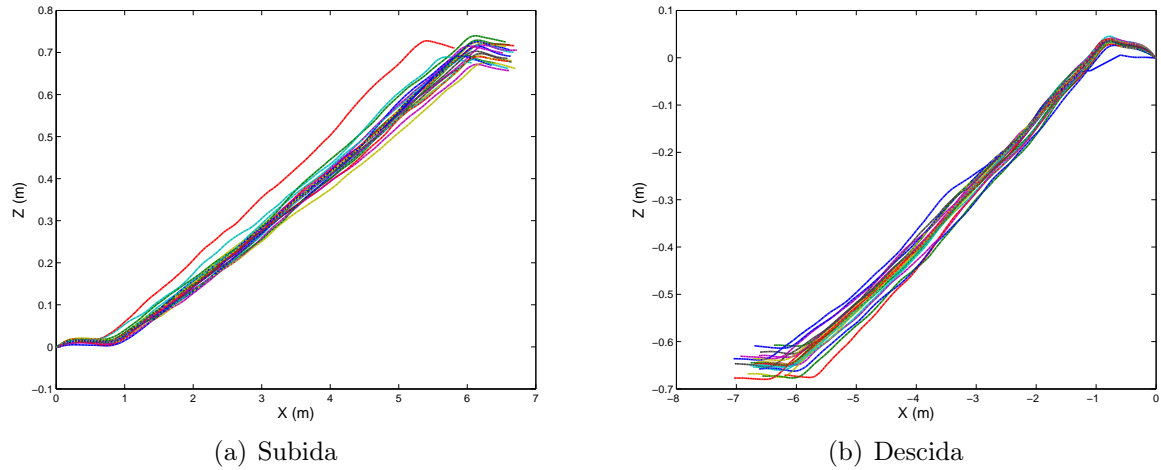
A Figura 4.9 apresenta os valores calculados para a variação no eixo  $Z$  sofrida pelo robô durante o seu deslocamento. A altura ( $h$ ) real da rampa utilizada nos experimentos é  $h \approx 65$  cm, já a média da variação de altura encontrada durante os experimentos foi  $\bar{h}_s = 69,06$  cm e  $\bar{h}_d = 64,50$  cm.



**Figura 4.9.** Experimentos de subida/descida de uma rampa, cálculo da altura utilizando o *pitch* estimado.

Apesar do ângulo de *pitch* calculado para a subida ter se aproximado mais do valor esperado, o erro na variação de altura calculada nesse caso foi maior que o erro para a descida. A principal razão encontrada para isso pode estar na execução dos experimentos, uma vez que o robô não efetuava exatamente a mesma trajetória retilínea, o que faz com que uma pequena distância a mais no caminho percorrido também influencie na altura calculada.

A partir desses dados é possível então reconstituir a trajetória realizada pelo robô, sendo apresentada na Figura 4.10.



**Figura 4.10.** Experimentos de subida/descida de uma rampa, reconstrução do caminho baseado na estimativa do *pitch*.

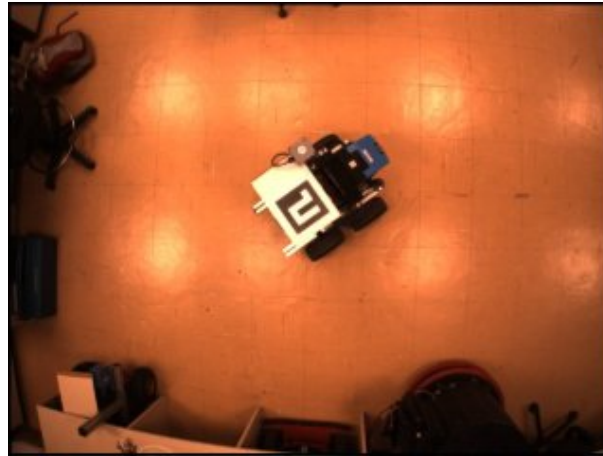
#### 4.3.1.3 Cálculo do Yaw

A orientação  $\psi$  do robô exerce grande influência na localização no plano XY, principalmente se houverem poucas variações de relevo (ambiente plano), onde pequenos erros em  $\psi$  podem gerar grandes erros de localização. Logo, foram realizados dois experimentos para verificação do cálculo do ângulo de *yaw*.

O primeiro experimento consiste em o robô realizar um giro completo em torno do próprio eixo (*turn-in-place*). Deve-se então comparar os valores dados pela odometria, pela metodologia e por um sistema de localização baseado em câmeras dispostas sobre o robô [Garcia et al., 2007], esse sistema possui uma boa precisão e seu valor será considerado o *ground truth* para efeitos de comparação.

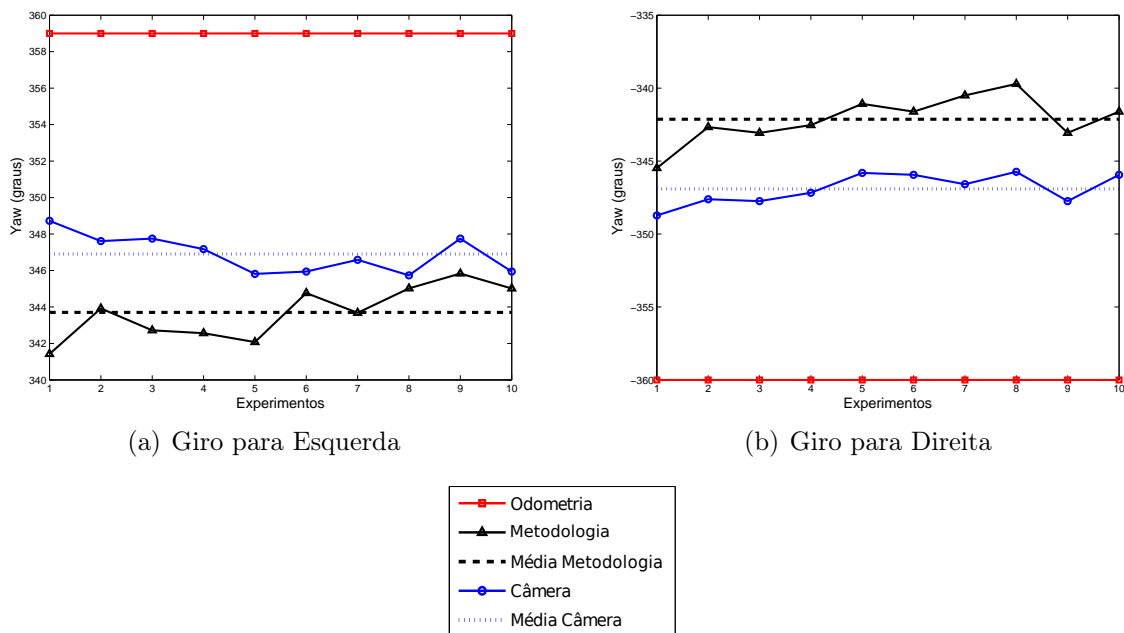
O sistema baseado em Visão Computacional realiza seus cálculos a partir de determinados marcos específicos presentes na cena e utilizando-se a biblioteca ARTToolkit [Wagner & Schmalstieg, 2007]. A Figura 4.11 mostra uma imagem do robô obtida através desse sistema onde é possível também se observar o marco utilizado para a localização.

O robô deverá realizar giros tanto para a esquerda quanto para direita, verificando-se assim o cálculo para valores positivos e negativos de *yaw*. Os valores encontrados durante esses experimentos estão dispostos nos intervalos  $[0,360]$  (giros para esquerda) e  $[0,-360]$  (giros para direita), escolheu-se esses intervalos para facilitar a comparação entre a distância encontrada e a distância alvo desejada.



**Figura 4.11.** Imagem do sistema de Visão Computacional utilizado para validação do *yaw*.

Para cada valor de ângulo específicos os experimentos foram executados dez vezes, permitindo assim melhor observar o comportamento do método. A Figura 4.12 apresenta os valores obtidos.

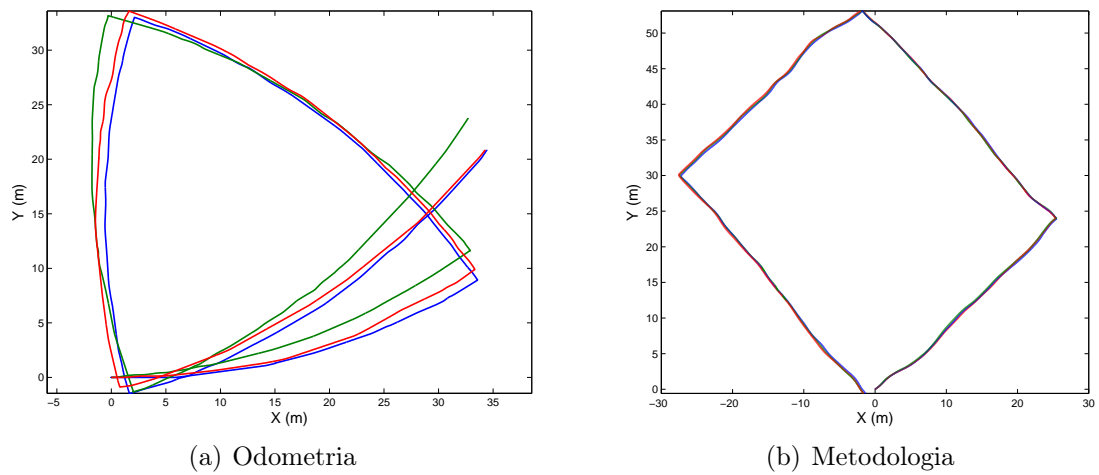


**Figura 4.12.** Experimentos de giro sobre o eixo para validação do *yaw*.

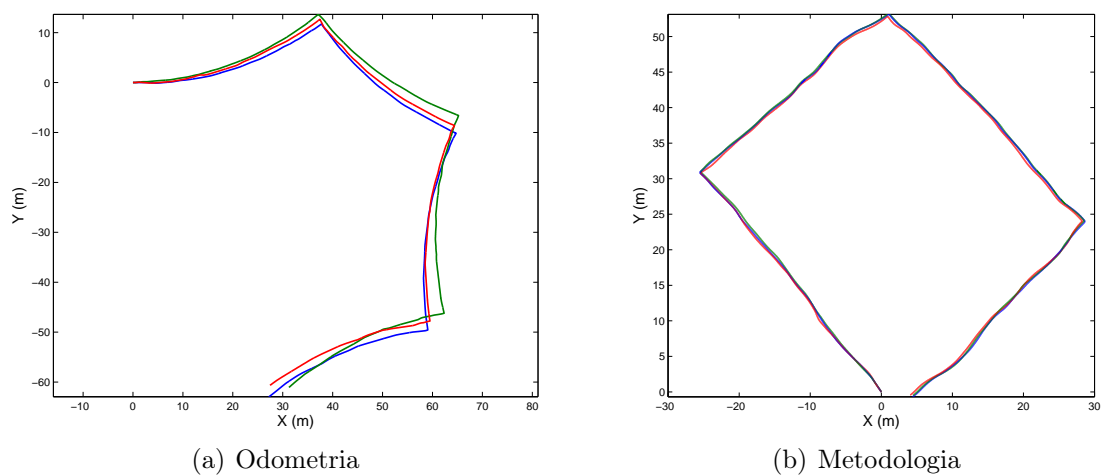
É possível observar que o erro presente no valor fornecido pela odometria supera os  $10^\circ$ . Já os valores encontrados pela metodologia se mostraram bem satisfatórios, obtendo-se as seguintes médias  $\bar{\psi}_{MET} = 343,70^\circ$  e  $\bar{\psi}_{CAM} = 346,68^\circ$  (giro para esquerda), e  $\bar{\psi}_{MET} = -342,15^\circ$  e  $\bar{\psi}_{CAM} = -346,90^\circ$  (giro para direita). Podemos concluir então, que essa pequena diferença entre os valores calculados deve-se princi-

palmente ao ruído presente nas leituras dos giroscópios e da bússola (devido principalmente a interferências magnéticas do local onde os experimentos ocorreram).

Para se verificar então a importância de se ter um valor de orientação preciso é proposto então o segundo experimento, que consiste em o robô realizar uma trajetória semelhante à forma de um quadrado em um ambiente interno e sem grandes imperfeições no piso. O trajeto a ser percorrido pelo robô foi demarcado no piso (para que durante as repetições o mesmo trajeto fosse percorrido), e possui comprimento conhecido. O principal objetivo é verificar o grande erro presente nas informações da odometria e como uma melhoria na informação de  $\psi$  aumenta significativamente a precisão da localização. O trajeto foi percorrido um total de seis vezes, onde três vezes o robô realizava curvas apenas para esquerda, e outras três apenas para a direita.



**Figura 4.13.** Trajeto quadrado para validação do *yaw* (girando para a esquerda).



**Figura 4.14.** Trajeto quadrado para validação do *yaw* (girando para a direita).

É possível notar pelas Figuras 4.13 e 4.14 que o valor fornecido pela odometria possui uma tendência a ir em deriva para a esquerda, o que faz com que dependendo do movimento realizado diferentes gráficos sejam encontrados. Temos que o erro de odometria no experimento onde o robô apenas realiza giros para a esquerda foi de aproximadamente 50 metros, enquanto utilizando-se a metodologia esse erro ficou abaixo de 5 metros. Já para a locomoção com giros para direita o erro da odometria foi superior a 60 metros, enquanto a metodologia forneceu um erro inferior a 2 metros.

Apesar de ainda possuir um erro relativamente grande, este foi inferior a 4% do caminho total percorrido ( $\approx 145$  m), logo, o sistema se mostrou capaz de aumentar consideravelmente a precisão da localização.

É importante ressaltar que além dos erros provindos dos ruídos dos sensores, nesse experimento outro fator que contribuiu para o erro foi a própria odometria, uma vez que a partir dela obtemos a velocidade utilizada para o cálculo da posição.

Um fator que exerce influência na odometria é a calibração dos pneus. Caso os pneus de um determinado lado estejam mais cheios que os do lado contrário o robô irá sofrer um desvio lateral (da mesma forma uma má distribuição do peso também terá influência), e que não estará representado nos valores obtidos da odometria. Para solucionar esse problema foi realizada uma verificação do movimento do robô, onde ele deveria se deslocar de forma retilínea por um determinado trecho e seria verificado se algum desvio lateral ocorreria. Em caso positivo os pneus deveriam ser recalibrados.

Logo, para solucionar o erro proveniente da odometria, é acrescentado ao sistema um *GPS*, sendo o sistema completo avaliado nos experimentos realizados *outdoor*.

### 4.3.2 Experimentos Externos

Os experimentos a serem realizados de forma *outdoor* têm como principal objetivo analisar o comportamento do sistema de localização em um ambiente real.

Foram realizados quatro experimentos. O Experimento 1 possui como objetivo a avaliação inicial do sistema com o robô navegando em um ambiente de terreno pouco irregular, além de uma avaliação inicial da introdução do GPS ao sistema. O Experimento 2 tem como propósito verificar o desempenho do sistema ao navegar por uma região com a presença de uma subida/descida bem acentuados. Para o Experimento 3, temos como principal objetivo a análise do método proposto para se lidar com quedas de sinal do GPS e as baixas velocidades do robô. Já no Experimento 4 será verificado o desempenho do sistema ao navegar por diferentes tipos de terreno.

### 4.3.2.1 Experimento 1

Como experimento inicial, o robô deverá seguir uma trajetória bem simples e sem grandes variações no relevo, tendo como principal objetivo avaliar o comportamento do sistema com a introdução do GPS e em um ambiente externo.

Neste primeiro experimento o robô realiza uma locomoção composta basicamente por caminhos retos e curvas bem definidas ( $90^\circ$ ), tendo como principal objetivo verificar o grau de precisão na localização do robô no plano 2D  $(x, y, \psi)$ , ou seja, se com a inserção do GPS no sistema foi possível corrigir os erros da odometria (velocidade) e se temos também uma boa estimativa da orientação.

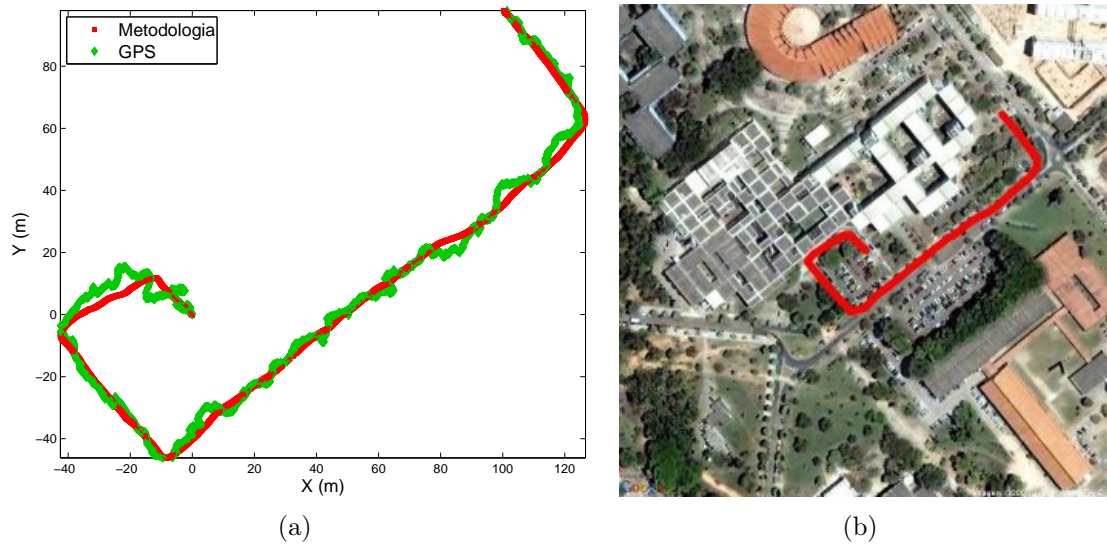
A Figura 4.15(a) apresenta um gráfico comparativo entre a trajetória realizada pelo robô calculada a partir da metodologia e os dados obtidos a partir do GPS. É possível observar que os dados do GPS são bastante ruidosos e certamente forneceriam uma precisão muito baixa se fossem utilizados diretamente. Já os valores obtidos através da metodologia forneceram um caminho mais suave e próximo ao real. O valor da posição inicial foi subtraído dos demais valores para uma melhor visualização e análise do deslocamento do robô.

É possível observar que, utilizando-se os dados do GPS, foi possível corrigir os erros no valor da velocidade fornecida pela odometria. Outro benefício na utilização do GPS está em poder-se realizar uma localização absoluta do robô. Para verificar-se quão próxima a localização global do robô está do mundo real foi utilizada uma API que permite traçar um caminho sobre determinadas coordenadas fornecidas e desenhá-lo sobre uma imagem de satélite [Google Maps API, 2008], neste caso as coordenadas a serem utilizadas são as calculadas pela metodologia. A Figura 4.15(b) mostra o caminho sobre a imagem da região onde os testes foram realizados.

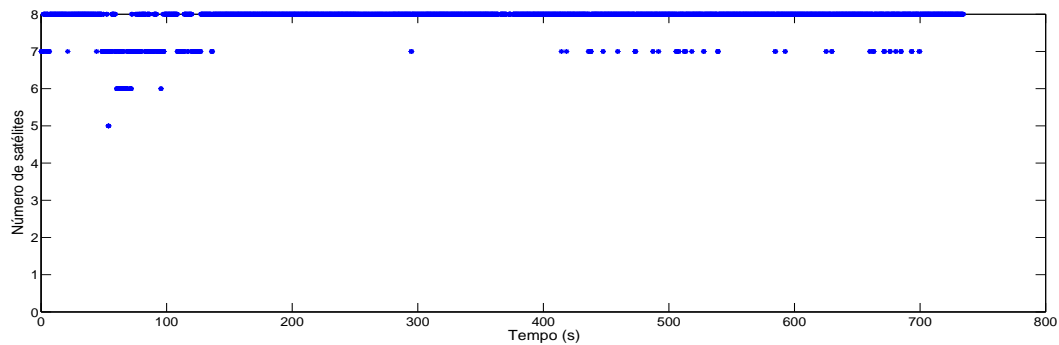
Como parte desse primeiro experimento, também desejava-se verificar a qualidade de recepção do sinal dos satélites. Para isso, durante o percurso, em determinados momentos, o robô passou por locais com uma boa cobertura de árvores, possibilitando verificar o comportamento quando o GPS encontra obstáculos que limitam sua linha de visada.

A Figura 4.16 apresenta o número de satélites vistos pelo robô ao longo da realização do experimento.

Como pode ser visto o menor valor encontrado foi de cinco satélites visíveis, sendo possível obter a informação de oito satélites durante a maior parte do percurso. Dessa forma, é possível concluir que o GPS possui uma certa robustez quanto a obstáculos que o interrompem de receber o sinal dos satélites.



**Figura 4.15.** Reconstrução da trajetória realizada pelo robô no Experimento 1.



**Figura 4.16.** Número de satélites visíveis durante a realização do Experimento 1.

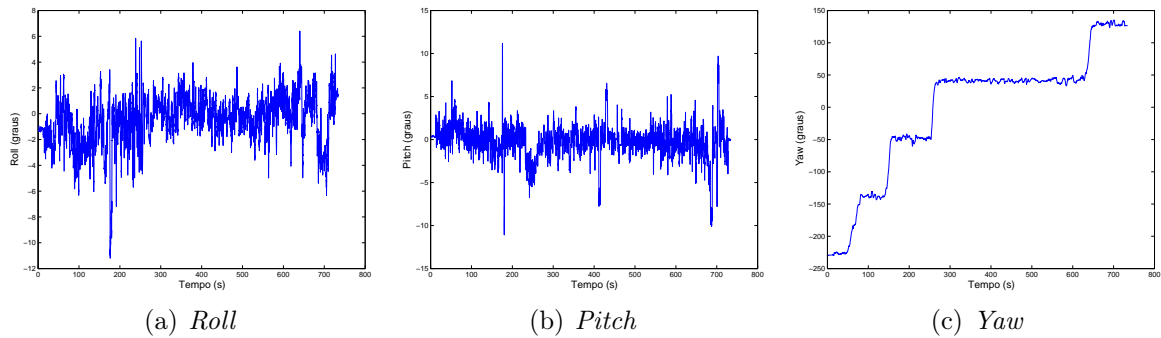
O ângulo  $\phi$  (Figura 4.17(a)) apresentou pequenas variações devido principalmente a imperfeições no solo e ruído, porém ao final do experimento é possível observar uma variação maior ( $\approx -4^\circ$ ) durante a descida de uma rampa.

O cálculo de  $\theta$  também possui grande parte de seu valor centrado em zero (terreno plano), entretanto, entre os instantes  $t = 230\text{s}$  e  $t = 270\text{s}$  (Figura 4.17(b)) a variação ( $\approx -2^\circ$ ) ocorreu durante a descida de um plano inclinado existente no local de realização do experimento, também temos três notáveis perturbações que foram decorrentes da descida e subida de pequenas rampas.

O resultado de  $\psi$  pode ser verificado diretamente, uma vez que durante o percurso o robô realizou quatro movimentos de giro bem acentuados (curvas de  $90^\circ$ ) e que podem ser observados na Figura 4.17(c) representados pelos quatro grandes degraus.



Como podemos observar nos gráficos da Figura 4.17 os valores calculados ainda possuem uma pequena quantidade de ruído, entretanto, os resultados encontrados se mostraram bem satisfatórios e representativos do movimento real realizado pelo robô. Analisando os desvios padrão temos  $\sigma_\phi = 0,0323$ ,  $\sigma_\theta = 0,0331$ . Optou-se por não calcular  $\sigma_\psi$  uma vez que a média não é uma boa representação dos valores obtidos.

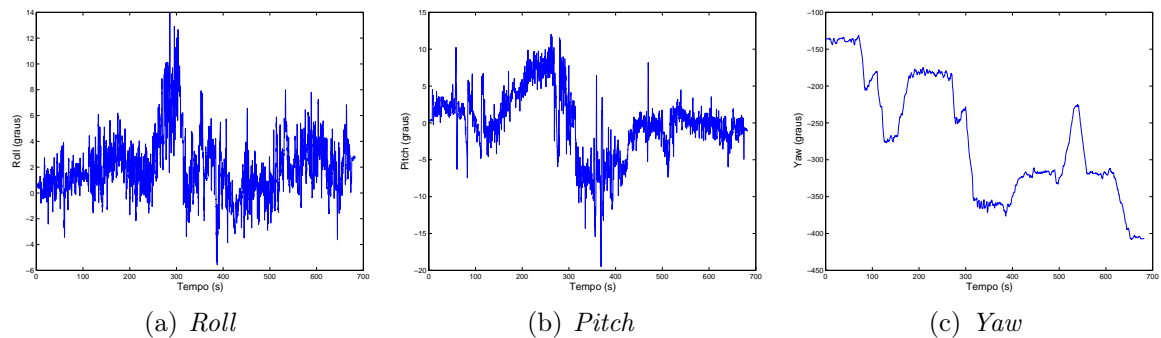


**Figura 4.17.** Ângulos de orientação calculados para o Experimento 1.

#### 4.3.2.2 Experimento 2

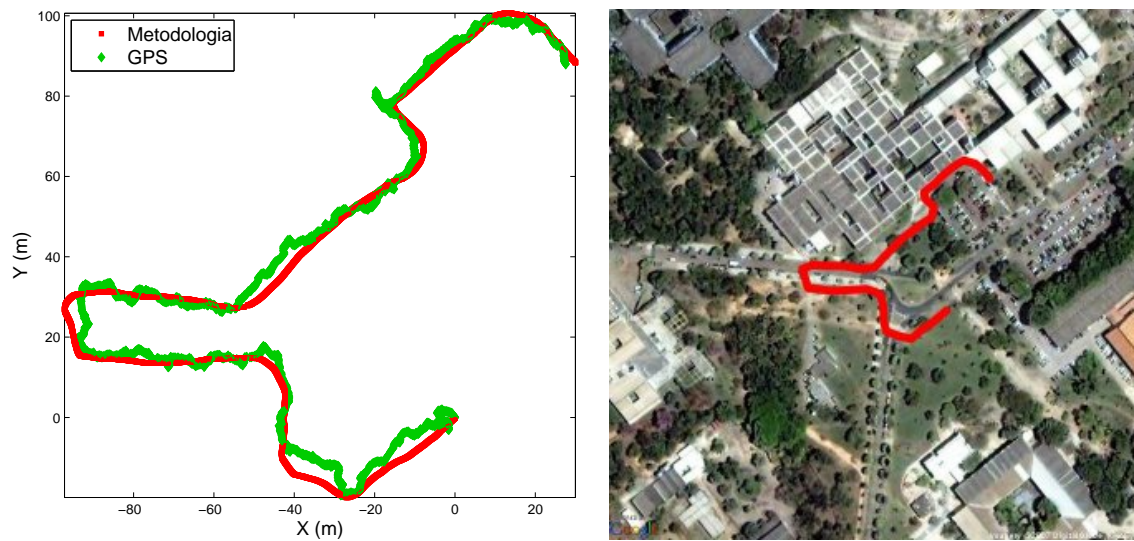
Durante a realização desse experimento o robô trafegou por um ambiente com maiores variações de relevo, sendo a principal uma grande subida que o robô deveria primeiramente subir e então em seguida descer.

É possível observar através dos gráficos presentes na Figura 4.18 os valores dos ângulos de orientação estimados para o robô durante a sua navegação.



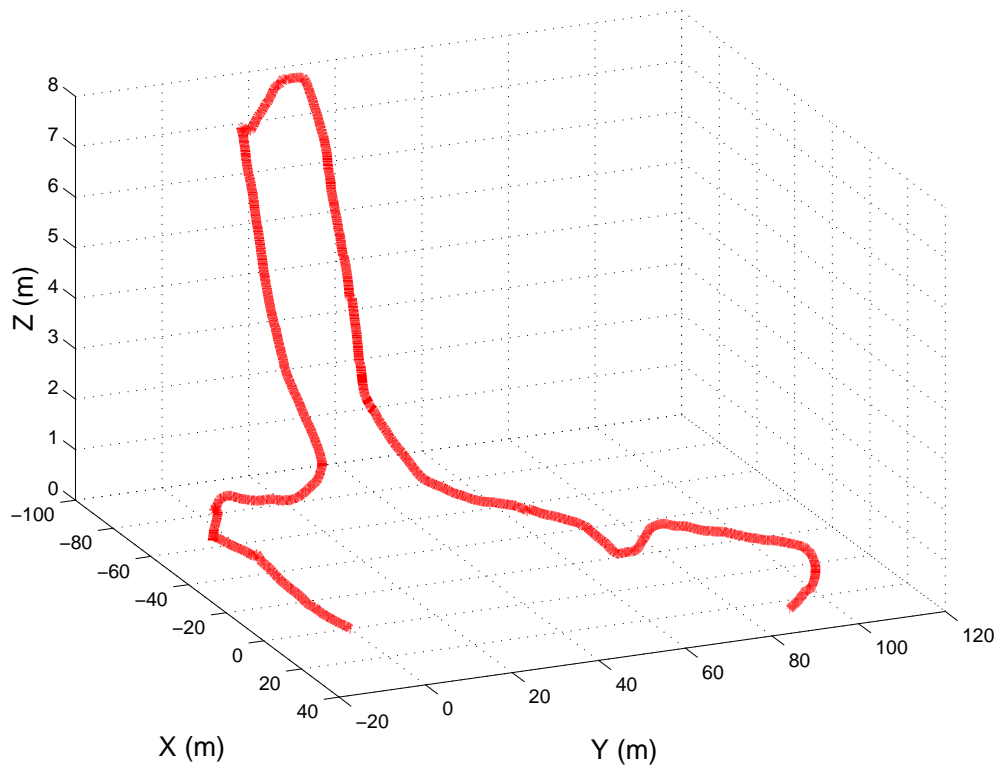
**Figura 4.18.** Ângulos de orientação calculados para o Experimento 2.

As imagens da Figura 4.19 exibem os valores de localização calculados pelo sistema e a sua disposição no mundo.



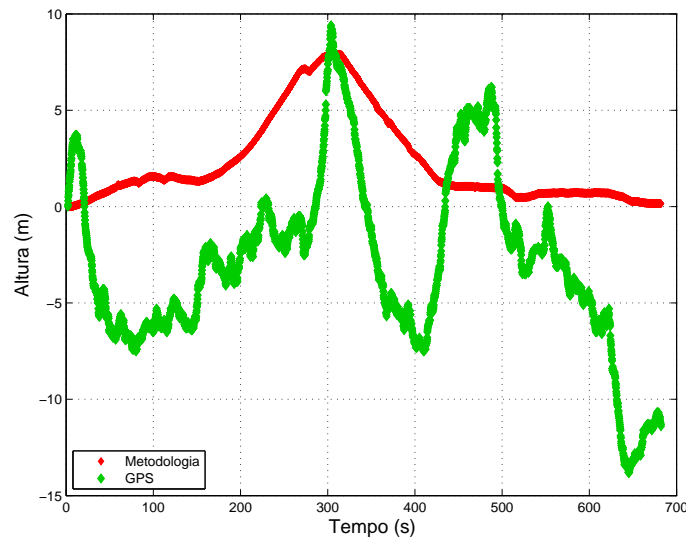
**Figura 4.19.** Reconstrução da trajetória realizada pelo robô no Experimento 2.

A Figura 4.20 exibe a variação do movimento do robô nas três dimensões, onde nota-se claramente a subida e descida realizada pelo robô durante a navegação, os eixos do gráfico encontram-se fora de escala para uma melhor visualização do deslocamento vertical realizado.



**Figura 4.20.** Variação de altura obtida durante a realização do Experimento 2.

O valor de altura também pode ser obtido a partir do GPS, entretanto, esse valor possui uma grande variação, não condizendo com o real deslocamento do robô, sendo uma das razões da sua não incorporação ao sistema. A Figura 4.21 apresenta a comparação entre a altura calculada pela metodologia e o valor retornado do GPS. É possível observar que a diferença entre o valor do ponto mais alto e mais baixo dado pelo GPS é de  $\approx 25\text{m}$ , um valor não condizente com a movimentação efetivamente realizada pelo robô.

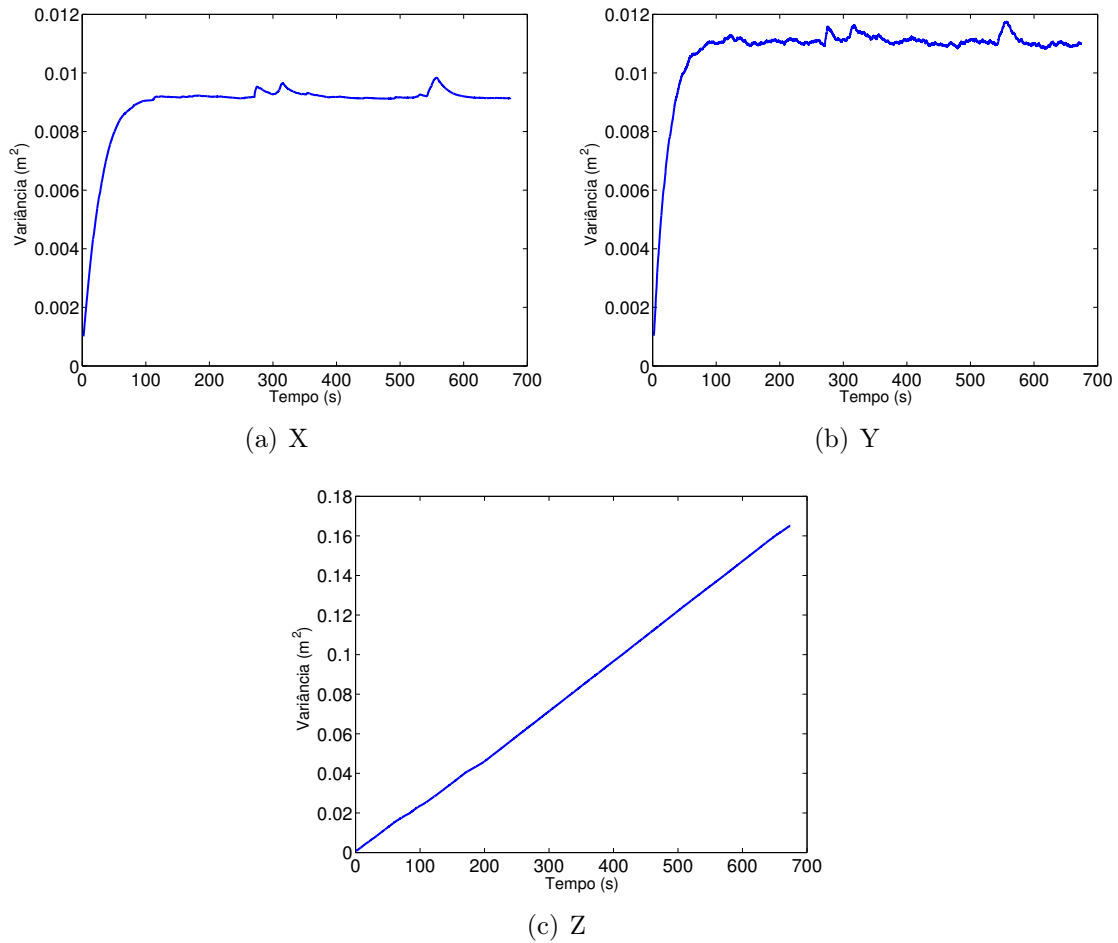


**Figura 4.21.** Comparativo entre a variação de altura estimada e o valor fornecido pelo GPS no tempo para o Experimento 2.

As Figuras 4.22 apresentam os valores de variância obtidos através da metodologia para cada variável de localização. Observa-se que a variância de  $x$  e  $y$  possuem comportamento semelhante, destacando-se três picos resultados da atribuição de uma maior variância aos dados do GPS nesses momentos da trajetória. É possível também observar que o comportamento da variância de  $y$  é menos suave que  $x$ , isso foi observado principalmente devido a uma pequena variação lateral proveniente do GPS.

É possível então verificar pela Figura 4.22(c) que o erro agregado a  $z$  tende a aumentar rapidamente, isso era esperado uma vez que o valor estimado para  $z$  não possui valor de correção. Isso tende a ser um problema, uma vez que se o robô navegar por longas distâncias o erro em  $z$  pode aumentar até se tornar inutilizável.

Com a realização desse experimento podemos concluir que foi possível estimar a variação tridimensional da posição do robô baseando-se apenas nos dados de orientação e velocidade do robô. Entretanto, apesar de produzir resultados satisfatórios, deve-se considerar que a falta de um sensor de correção que forneça um valor absoluto para a altura do robô fará com que a navegação por um longo período de tempo acabe



**Figura 4.22.** Variância das variáveis de localização durante o Experimento 2.

por acumular uma grande quantidade de erro para esse parâmetro. Outro fator que deve ser considerado é o terreno onde o robô está navegando, quanto mais acidentado maior será a variação do ângulo de  $\theta$  e maior será a imprecisão da velocidade fornecida pela odometria, logo, se o robô permanecer nessas condições por longas distâncias mais rapidamente o erro da altura irá derivar.

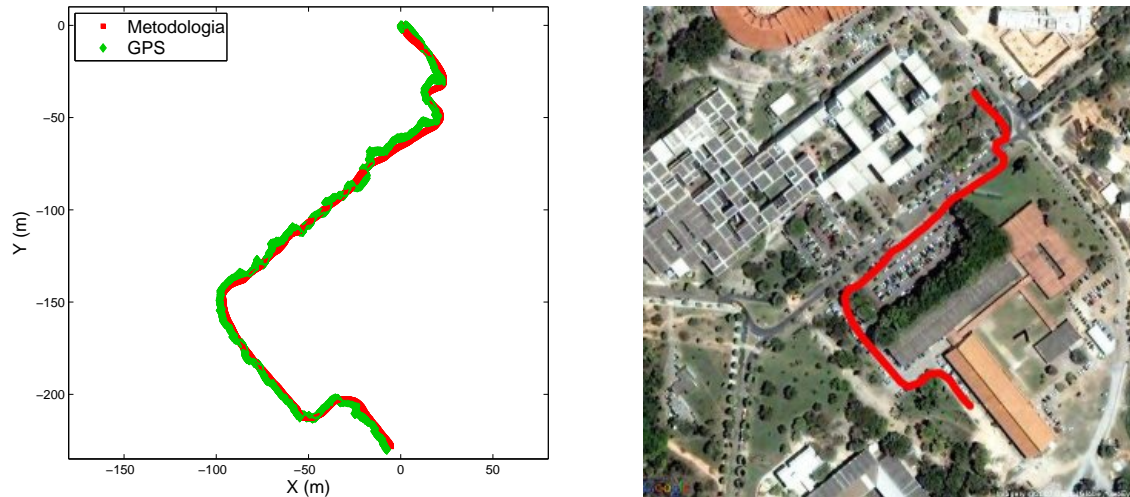
#### 4.3.2.3 Experimento 3

Este experimento possui como principal objetivo avaliar o método utilizado para o cálculo da variância das informações do GPS quando ocorre uma perda de sinal ou quando o robô está trafegando a uma velocidade muito baixa (ou se encontra parado).

Deve-se salientar que em nenhum momento durante a realização desse experimento o robô possuía um número menor que sete satélites em visada, logo, para a verificação do método será adotado que a partir de um dado momento o robô não re-

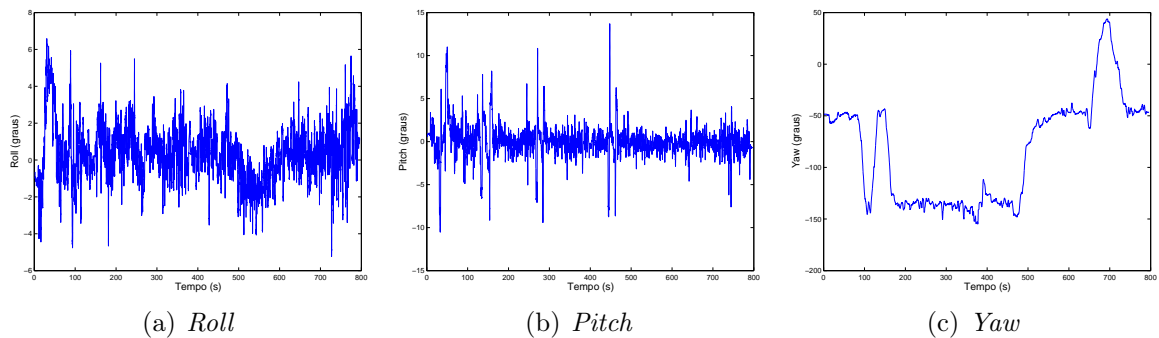
cebia mais o sinal de nenhum satélite, simulando assim uma queda completa de sinal.

As imagens da Figura 4.23 apresentam a trajetória estimada para a navegação realizada pelo robô durante o experimento.



**Figura 4.23.** Reconstrução da trajetória realizada pelo robô no Experimento 3.

Os valores de  $\phi$ ,  $\theta$  e  $\psi$  assumidos pelo robô durante o experimento são exibidos nos gráficos da Figura 4.24.



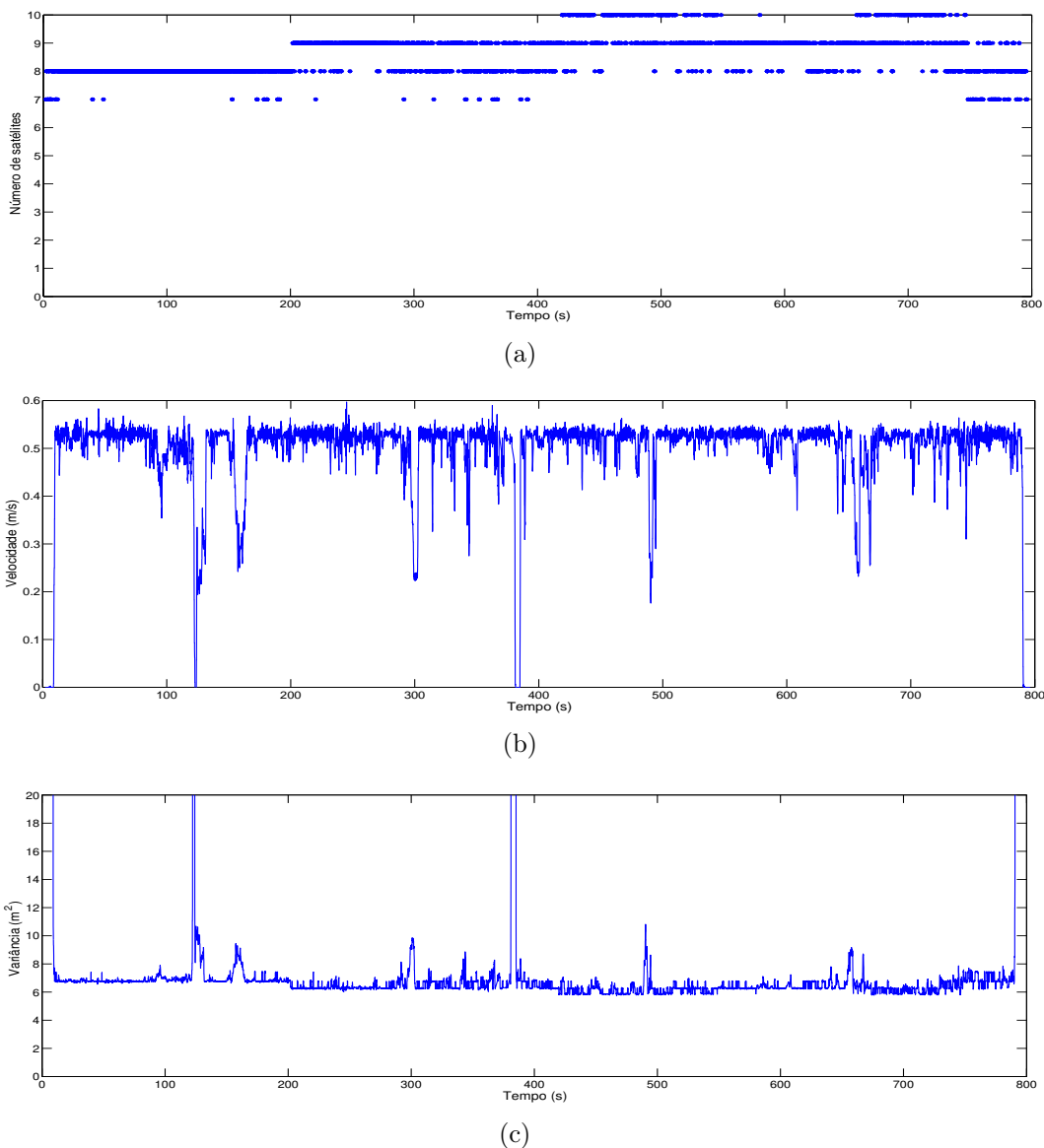
**Figura 4.24.** Ângulos de orientação calculados para o Experimento 3.

Primeiramente estaremos analisando os momentos em que o robô esteve a baixas velocidades, e conseqüentemente sujeito ao efeito de *random walk*, um fator que pode prejudicar a localização do robô caso não seja tratado.

A Figura 4.25(b) exibe como a velocidade do robô variou durante o experimento. É possível realizar uma comparação com a Figura 4.24(c) e verificar que os momentos de menores velocidades ocorrem justamente durante a realização de curvas, que em certas vezes são feitas de forma *turn-in-place* devido ao fato do robô ser do tipo *skid-steer*. Através da Figura 4.25(a) observamos o número de satélites visíveis ao longo da

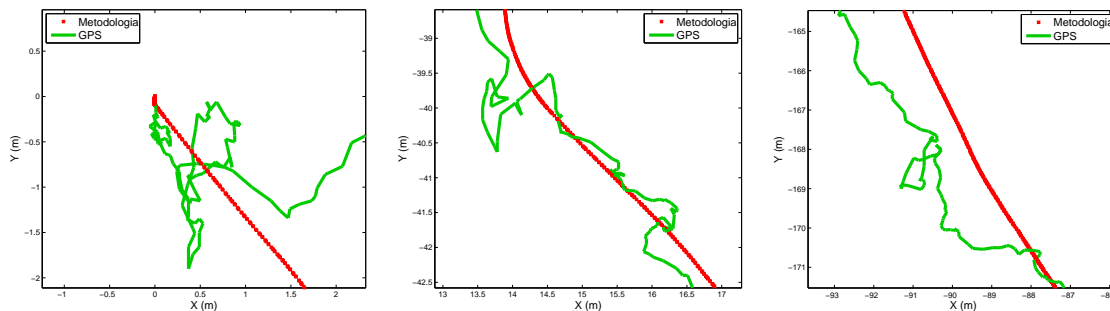
execução do experimento, e a Figura 4.25(c) exibe a variância calculada para os dados fornecidos pelo GPS baseados tanto na velocidade quanto na quantidade de satélites.

Pode-se notar quatro picos onde a variância possui valores muito altos, o quê significa que as informações do GPS devem ser descartadas nesses momentos pois possuem um alto grau de imprecisão devido à baixa velocidade do robô. Durante o restante do experimento temos que a quantidade de satélites possui valores constantes por um grande período de tempo, sendo possível então observar que a variância está sendo influenciada unicamente pela velocidade desempenhada.



**Figura 4.25.** Valores observados durante a realização do Experimento 3 para os parâmetros número de satélites visíveis, velocidade, e variância calculada a partir do modelo proposto para as informações do GPS.

A Figura 4.26 mostra em destaque os momentos em que a baixa velocidade do robô fez com que as informações obtidas do GPS sofressem do efeito de *random walk* (os dados do GPS foram ligados para uma melhor visualização). O cálculo da variância baseada na informação de velocidade produziu bons resultados, reduzindo drasticamente (e em alguns casos eliminando completamente) a influência do *random walk*, exibindo caminhos mais suaves e realistas para o trajeto realizado e mostrando-se uma forma eficaz de melhorar a estimativa de localização do robô.



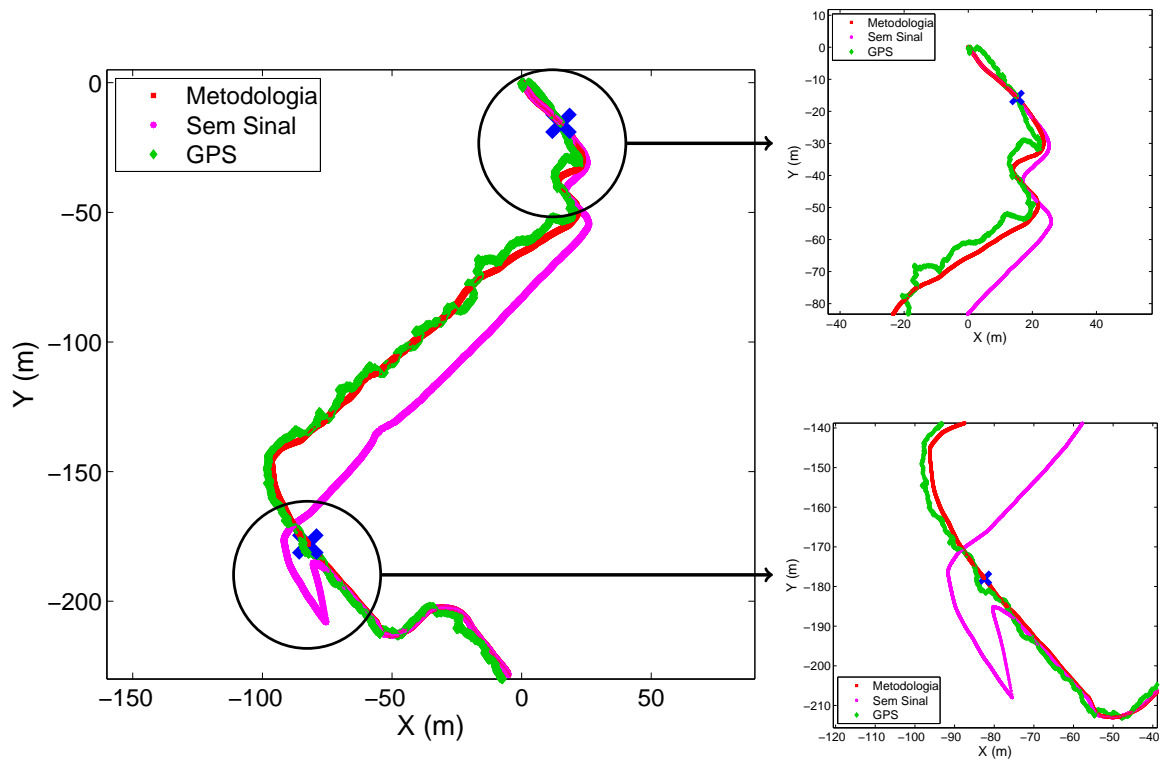
**Figura 4.26.** Momentos em que a baixa velocidade do robô levou à ocorrência do efeito de *random walk*.

A próxima etapa consiste em simular a queda do sinal durante um determinado período de tempo, verificando-se que a partir dos valores calculados para a variância os dados fornecidos pelo GPS passam a ser ignorados na etapa de correção.

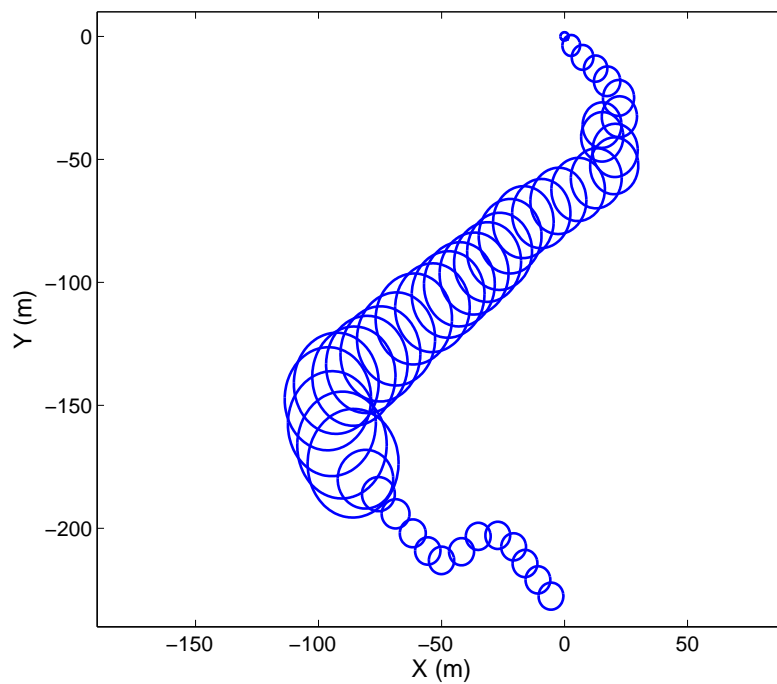
A Figura 4.27 exibe os resultados obtidos para a simulação de queda de sinal. É possível observar que um pequeno erro na orientação gera grandes erros na localização, onde grande parte do erro aqui observado está relacionado à odometria, esse fato fica bem evidente na figura que destaca o início do trajeto, onde nota-se o erro na estimativa de deslocamento do robô, erro esse devido principalmente à característica irregular do terreno. Entretanto, tão logo o sinal do GPS se torna disponível novamente a trajetória já começa a ser corrigida.

O robô trafegou por uma distância de  $\approx 200m$  sem utilizar o sinal do GPS, e no instante anterior ao sinal se tornar novamente disponível havia uma diferença entre a localização atual e a localização calculada utilizando-se o GPS durante todo o percurso de  $30,6m$ .

A Figura 4.28 apresenta o comportamento da covariância do erro para a localização durante o experimento. É importante salientar que os dados exibidos estão em escala bem maior que os valores reais calculados, permitindo assim uma melhor visualização ao longo de todo o trajeto.



**Figura 4.27.** Reconstrução da trajetória executada pelo robô no Experimento 3 com uma simulação de queda de sinal do GPS. Trajeto completo realizado e destaque do início e final da simulação de queda de sinal.



**Figura 4.28.** Covariância do erro durante a execução do Experimento 3 com simulação de queda de sinal do GPS.



É possível então observar que a partir do momento em que os dados do GPS passam a não fazer mais parte da etapa de correção, o erro na localização tende a aumentar. Entretanto, assim que os dados passam novamente a fazer do sistema o erro diminui rapidamente, voltando ao valor inicial.

É possível concluir então que apesar de o sistema conseguir lidar de forma satisfatória com as baixas velocidades do robô e quedas do sinal, a ausência da correção fornecida pelo GPS por um longo período de tempo pode prejudicar significativamente a localização do robô, entretanto, assim que os dados do GPS retornam é possível rapidamente realizar a correção da trajetória.

Além disso, conclui-se que torna-se fundamental a utilização do GPS para sistemas que se locomovem em ambientes externos, uma vez que as características irregulares do terreno prejudicam ainda mais as informações obtidas a partir da odometria, por exemplo problemas de derrapagem e erros na estimativa de altura (provindos da incerteza na orientação).

#### 4.3.2.4 Experimento 4

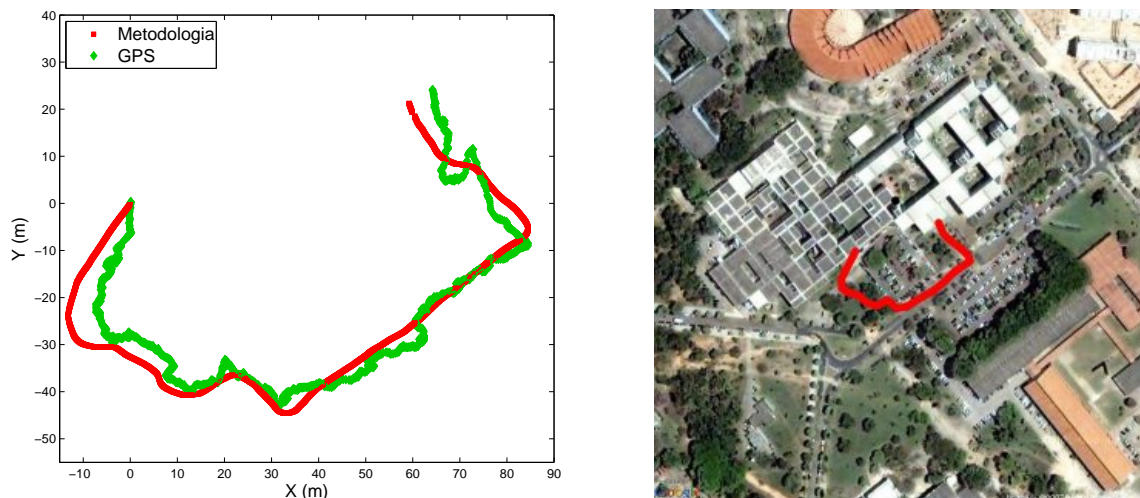
Este último experimento consistiu em fazer o robô trafegar por diferentes tipos de terrenos, tendo como principal objetivo verificar a influência que o terreno irá exercer sobre a sua localização. A Figura 4.29 apresenta os terrenos visitados, onde a Figura 4.29(a) apresenta o terreno mais acidentado e representa o limite trafegável pelo robô utilizado (uma vez que esse se move com grande dificuldade nesse tipo de terreno) e a Figura 4.29(b) o terreno mais suave, os demais terrenos possuem tanto partes mais acidentadas quanto mais planas.

A Figura 4.30 apresenta os dados absolutos coletados do GPS e a estimativa de localização calculada a partir da metodologia, assim como também a disposição no mundo do caminho realizado pelo robô.

A principal influência do terreno no sistema pode ser observada no aumento da dispersão dos parâmetros calculados, logo, para verificar-se essa influência deve ser calculado o desvio padrão das medidas para cada diferente tipo de terreno trafegado, entretanto, o cálculo do desvio padrão baseia-se na média dos valores no intervalo de interesse, porém, a média não possui uma boa representatividade dos valores reais obtido para esse caso, assim sendo, propõem-se então utilizar um filtro que deve agir como uma média variável e portanto o cálculo do desvio padrão será feito utilizando-se o valor filtrado.



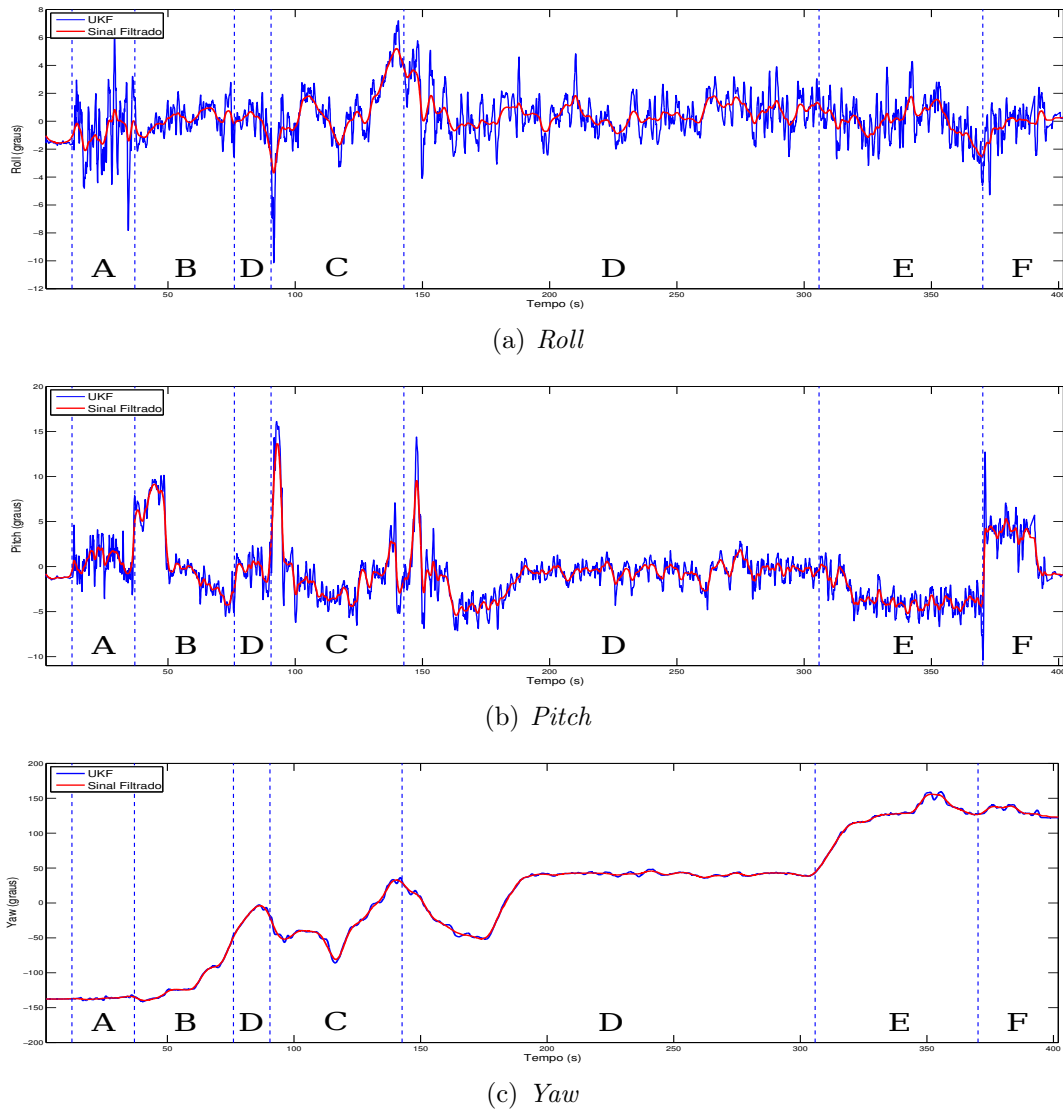
**Figura 4.29.** Diferentes tipos de terrenos trafegados durante a realização do Experimento 4.



**Figura 4.30.** Reconstrução da trajetória realizada pelo robô no Experimento 4.

A Figura 4.31 apresenta os valores de orientação calculados para o Experimento 4. Também é possível observar nas imagens o sinal filtrado utilizado no cálculo do desvio padrão e os intervalos referentes a cada tipo de terreno navegado durante o experimento.

A Tabela 4.1 apresenta os valores de desvios padrão encontrados para cada ângulo de orientação e de acordo com o tipo de terreno trafegado.



**Figura 4.31.** Ângulos de orientação calculados para o Experimento 4.

Como era esperado, o terreno **A** por ser o mais acidentado foi o que gerou o maior desvio padrão para os ângulos de *roll* e *pitch*, da mesma forma o terreno **B** produziu o menor desvio padrão, uma vez que praticamente não possuía imperfeições. Os demais terrenos produziram resultados semelhantes, entretanto, deve-se salientar que em alguns casos um desvio padrão maior foi devido à forma de cálculo utilizada, pois em determinados instantes (principalmente nas mudanças de terreno) os ângulos de *roll* e *pitch* apresentam picos, e estes valores após serem filtrados sofrem uma grande depreciação, o que acaba por gerar um alto valor ao se calcular a diferença entre o valor obtido e o valor filtrado.

**Tabela 4.1.** Desvios padrão dos ângulos de orientação calculados para cada diferente tipo de terreno trafegado durante o Experimento 4.

	Tipo de Terreno					
	A	B	C	D	E	F
<i>Roll</i>	1,92	0,68	0,94	0,97	0,98	1,02
<i>Pitch</i>	1,26	0,76	1,18	0,80	0,83	1,20
<i>Yaw</i>	0,69	0,62	1,66	0,76	0,94	0,97

Os valores apresentados de desvios padrão para o ângulo de *yaw* são baixos principalmente devido ao fato de o robô não sofrer grandes variações de angulação lateral, apenas em momentos de curva (e nesse momento possuindo uma variação de certo modo constante). Apesar disso, é possível observar que o maior valor de desvio padrão para essa variável ocorreu no terreno **C**. A explicação se deve principalmente pelo fato de ter sido o terreno onde o robô realizou mais movimentos de curva, o que influenciou no método utilizado para verificação (filtro).

É possível concluir-se então que o terreno por onde o robô trafega influencia diretamente na qualidade da estimativa, incorporando um certo grau de ruído nos valores para os ângulos de orientação e por conseqüência na localização, entretanto, os resultados obtidos pela metodologia se mostraram robustos e com uma variação inferior a 2°.

## 4.4 Mapeamento

### 4.4.1 Experimentos Internos

A etapa de mapeamento deve inicialmente ser verificada em um ambiente interno e conhecido, onde o robô deve realizar um deslocamento retilíneo por uma pequena distância e ultrapassar um obstáculo de  $\approx 5$  cm de altura. Assim como nos experimentos da etapa de localização, o mapeamento *indoor* aqui realizado baseia-se na localização calculada através da metodologia e utilizando os dados retornados pelos diversos sensores que compõem o sistema (exceto o GPS).

O principal objetivo com esse experimento é obter uma visão inicial do comportamento da filtragem realizada durante a etapa de mapeamento, que visa retirar os ruídos intrínsecos ao sinal da IMU e bússola e que acabam sendo incorporados à localização, desconsiderando-se as imperfeições do terreno, assim como também verificar a precisão do próprio *laser* em si de captar detalhes do ambiente.

Diferentes objetos foram dispostos no ambiente de experimentação para verificar-se se o formato do obstáculo exerce alguma influência na leitura, a Figura 4.32 apresenta o local por onde o robô trafegou e a disposição dos objetos, assim como uma vista superior do mapeamento realizado. O resultado geral foi satisfatório, sendo possível identificar no mapa todos os obstáculos presentes no ambiente.

Já a Figura 4.33 exibe uma visão em perspectiva do mapeamento, onde já é possível então observar maiores detalhes como a superfície esférica do obstáculo **A** e a altura de **F**.

Para uma análise qualitativa do método foram adquiridas medidas reais do ambiente e calculadas as mesmas medidas a partir da nuvem de pontos calculada. A Tabela 4.2 apresenta a comparação entre os valores.

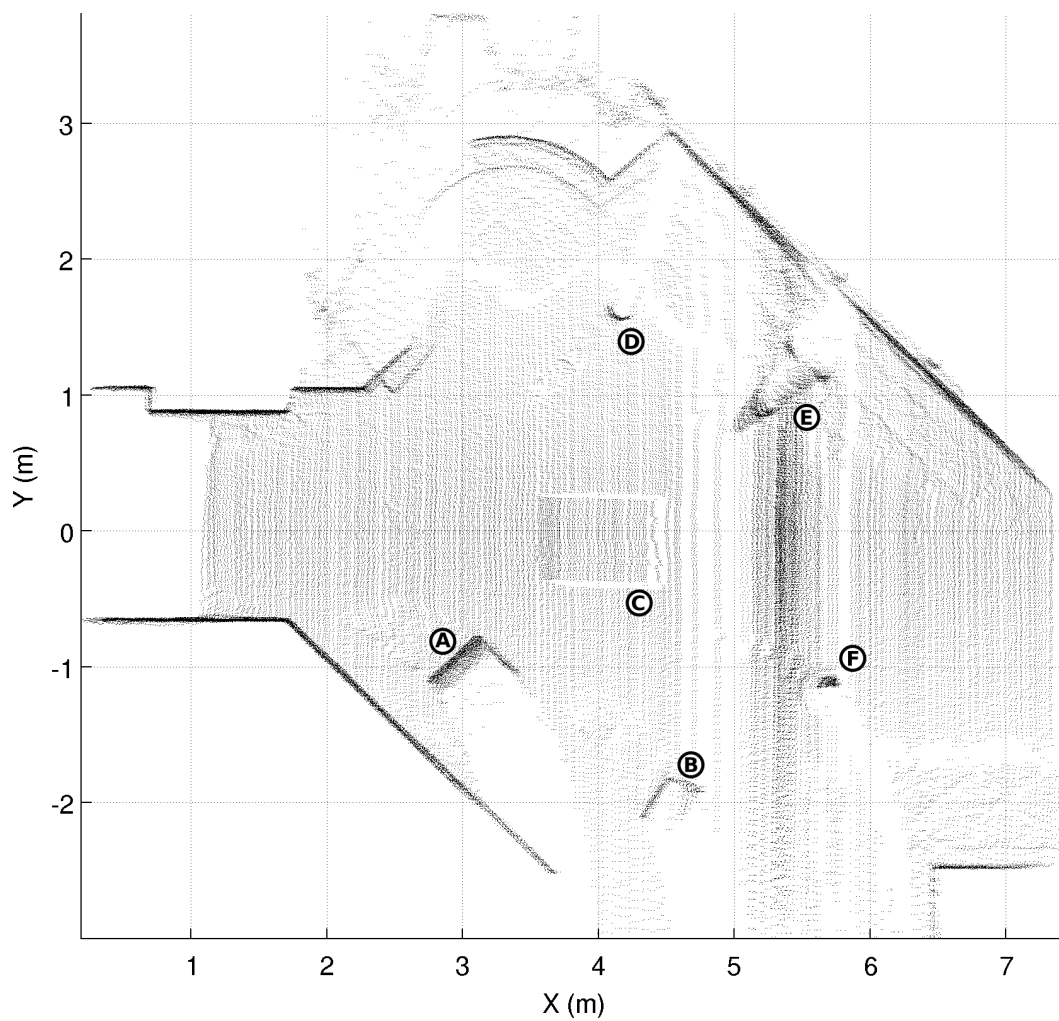
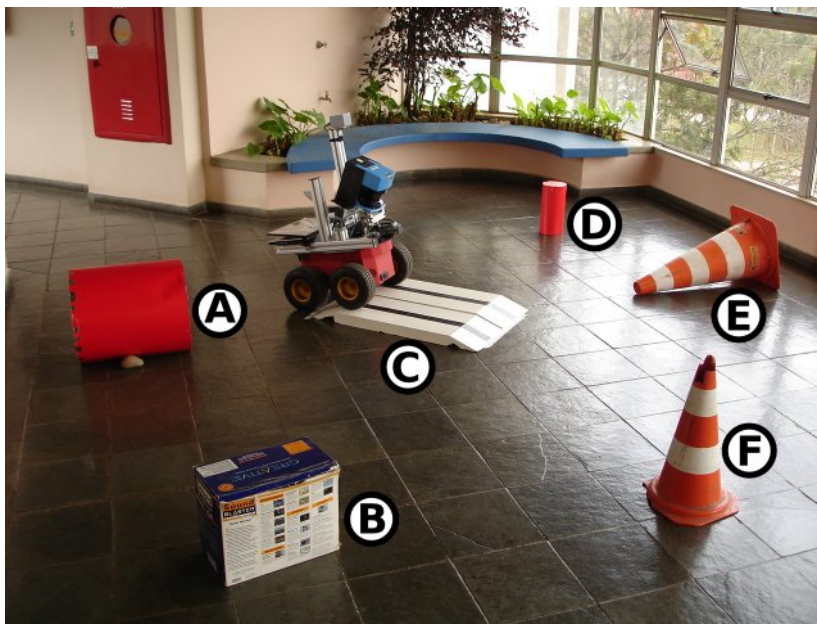
**Tabela 4.2.** Tabela comparativa entre distância reais e os valores calculados à partir da nuvem de pontos obtida em um mapeamento interno.

	Valor Real (m)	Valor Calculado (m)
Medida 1	1,68	1,69
Medida 2	1,51	1,53
Medida 3	0,59	0,57
Medida 4	0,47	0,48
Medida 5	0,51	0,54

Os valores utilizados como comparação foram escolhidos de forma a representar superfícies com diferentes ângulos de incidência do *laser*. Além disso, medições de partes dos obstáculos escolhidos foram adquiridas antes, durante e após a travessia do robô pela rampa.

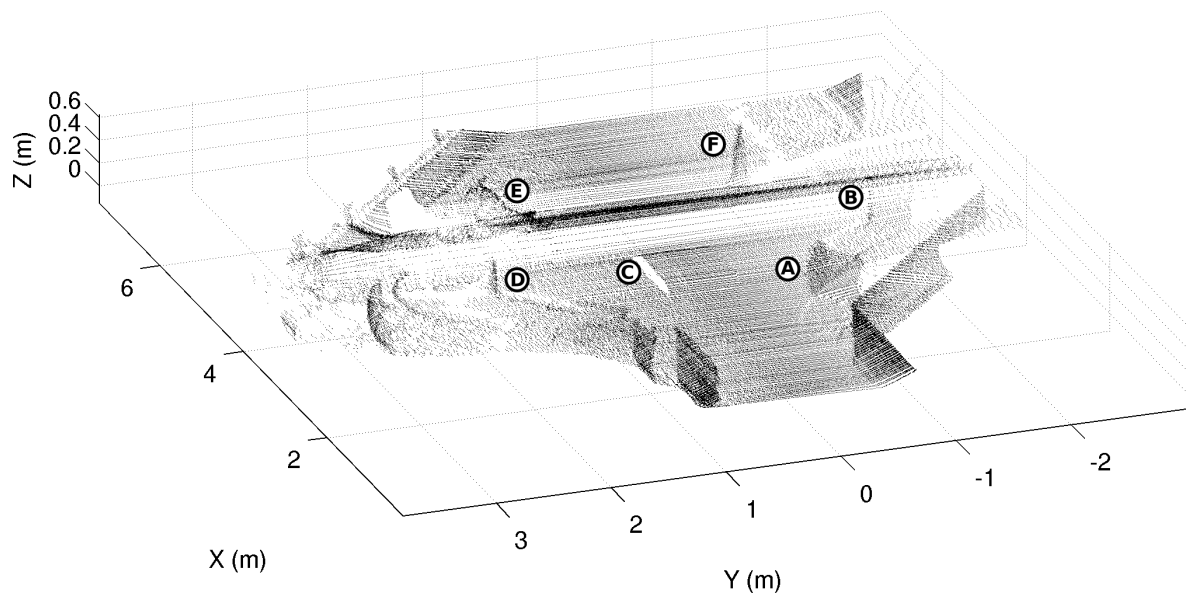
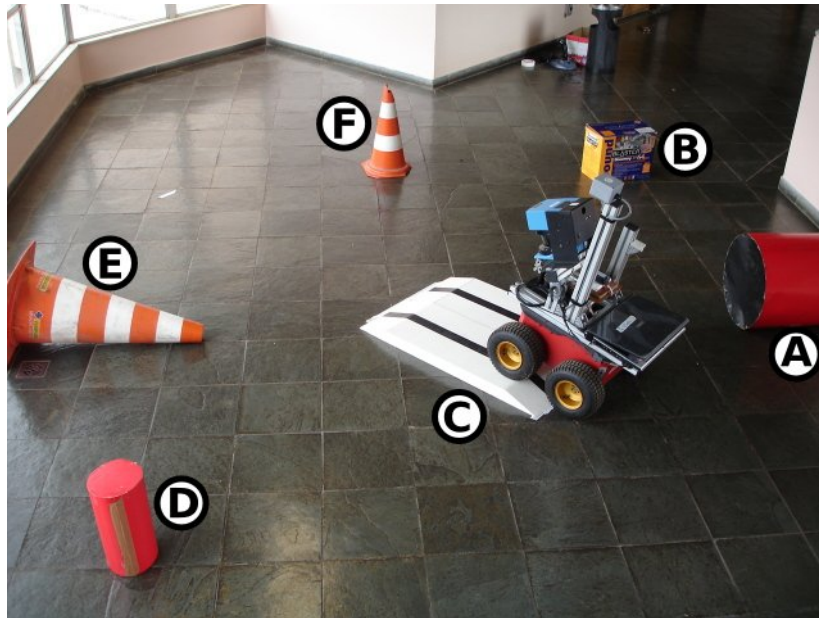
Os valores calculados obtiveram uma variação média inferior a 2 cm, um valor considerado suficientemente bom em relação ao grande nível de ruído presente nos sinais dos sensores utilizados.

Como passo seguinte deve-se então realizar os experimentos externos e verificar se mesmo com a adição do ruído agora presente devido à navegação em um ambiente com variação no relevo e da integração dos dados do GPS ao sistema é possível manter essa qualidade no mapeamento.



**Figura 4.32.** Local e visão de topo do mapeamento obtido em um ambiente interno.





**Figura 4.33.** Local e visão em perspectiva do mapeamento obtido em um ambiente interno.

### 4.4.2 Experimentos Externos

A qualidade do mapeamento está diretamente relacionada à precisão do cálculo da localização, logo, os resultados apresentados nesta seção podem ser analisados como representativos para o sistema como um todo.

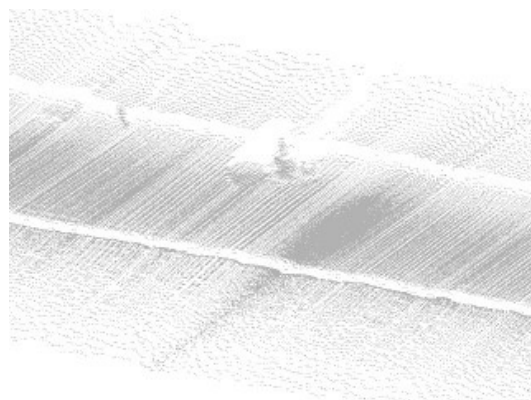
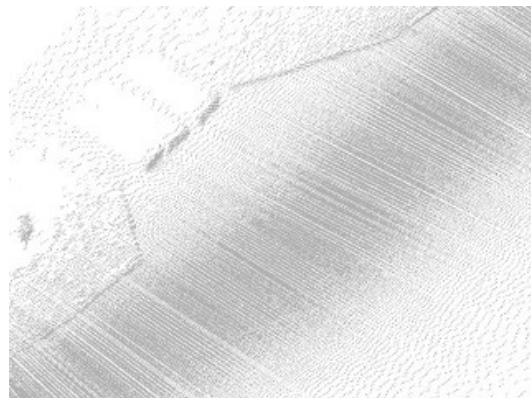
Os mapas aqui apresentados consistem na reconstrução do relevo trafegado pelo robô em um ambiente externo, e são relativos aos experimentos apresentados na Seção 4.3.2.

As imagens das Figuras 4.34 e 4.35 apresentam em destaque alguns trechos dos mapeamentos realizados durante os experimentos. Através do mapa é possível a identificação de várias características presentes no ambiente (especificamente no relevo), dessa forma evidenciando a boa qualidade o mapeamento realizado.

As Figuras 4.36, 4.37, 4.38 e 4.39 representam todo o caminho percorrido durante cada experimento, onde a coloração destaca a variação de altura dos pontos. O Experimento 2 apresenta um caminho em que a variação da localização do robô no eixo Z foi superior a 9 metros, e isso está claramente exibido no mapeamento realizado. Já o Experimento 3 ocorreu em um trajeto em que não houve grande variação de altura, logo, é possível observar que as principais diferenças são referentes a obstáculos presentes na cena, sendo possível identificar por exemplo muros presentes às margens laterais da trajetória, assim como o ponto mais baixo observado sendo um terreno em um plano inferior ao da navegação visto no início do caminho.

Conforme apresentado na seção anterior (Experimento 2), a variância do parâmetro  $y$  da localização possui um certo ruído. Isso acabou por introduzir um pequeno erro lateral na formação do mapa, e que se evidencia principalmente em trajeto longos e retilíneos (por exemplo Figuras 4.36 e 4.39).





**Figura 4.34.** Detalhes do terreno identificáveis através do mapeamento.

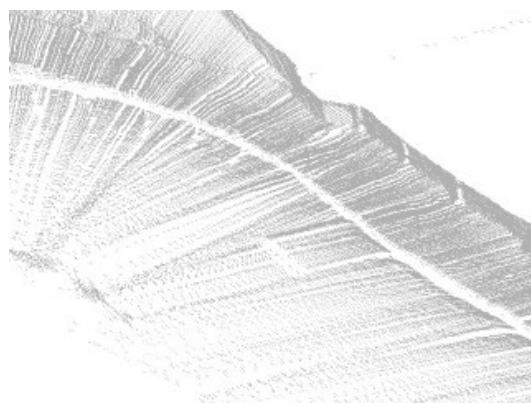
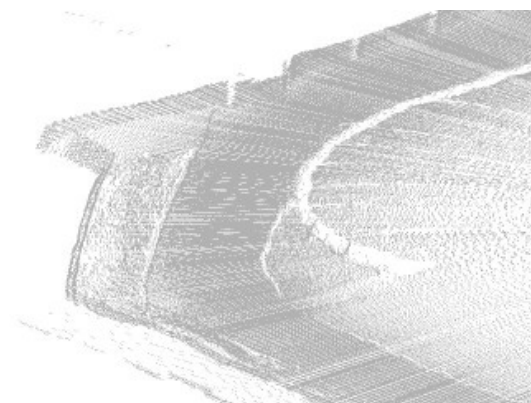
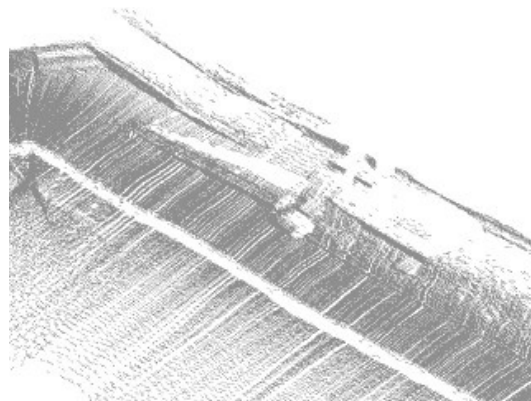


Figura 4.35. Detalhes do terreno identificáveis através do mapeamento.

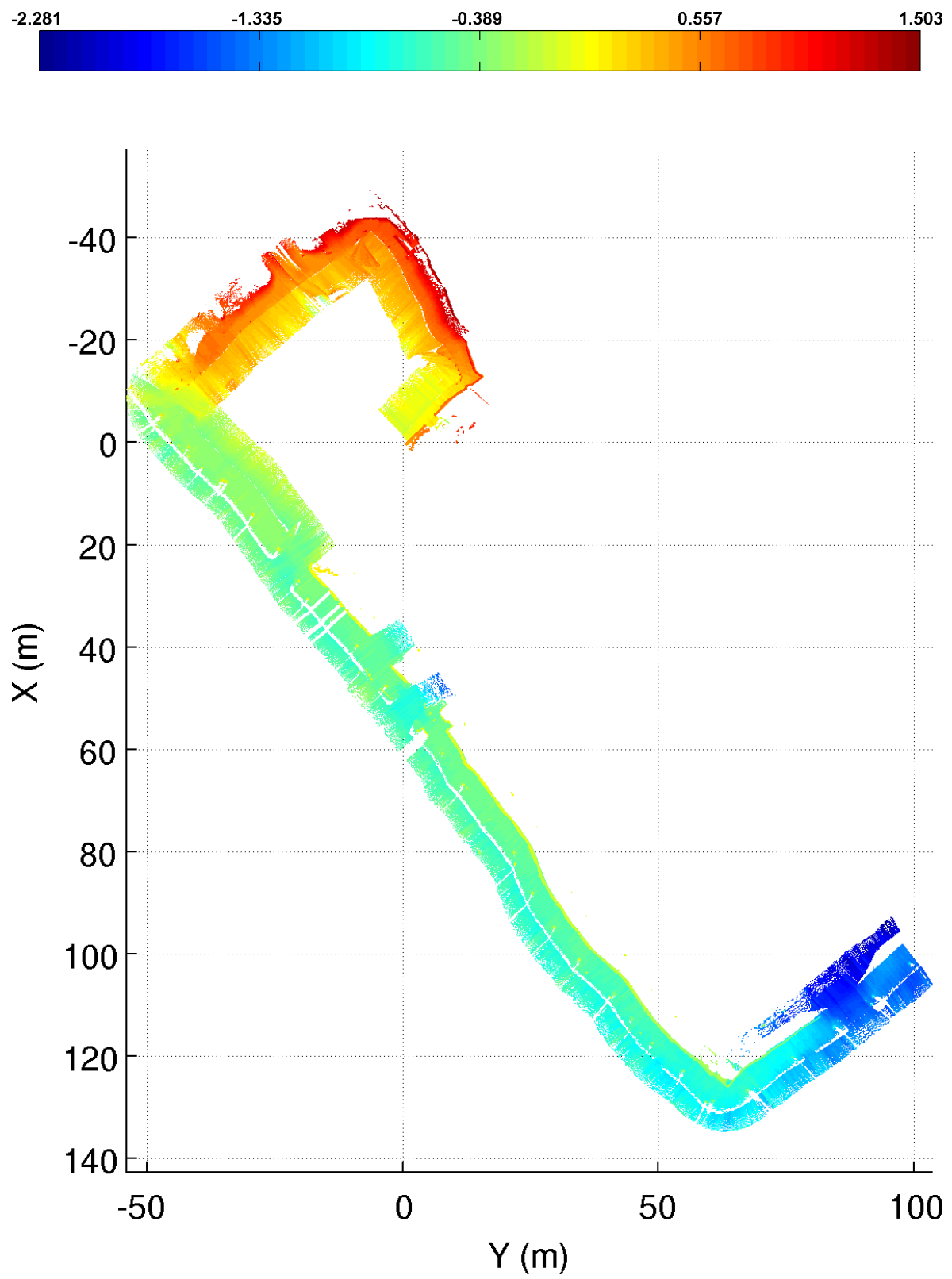


Figura 4.36. Mapeamento completo obtido para o Experimento 1.



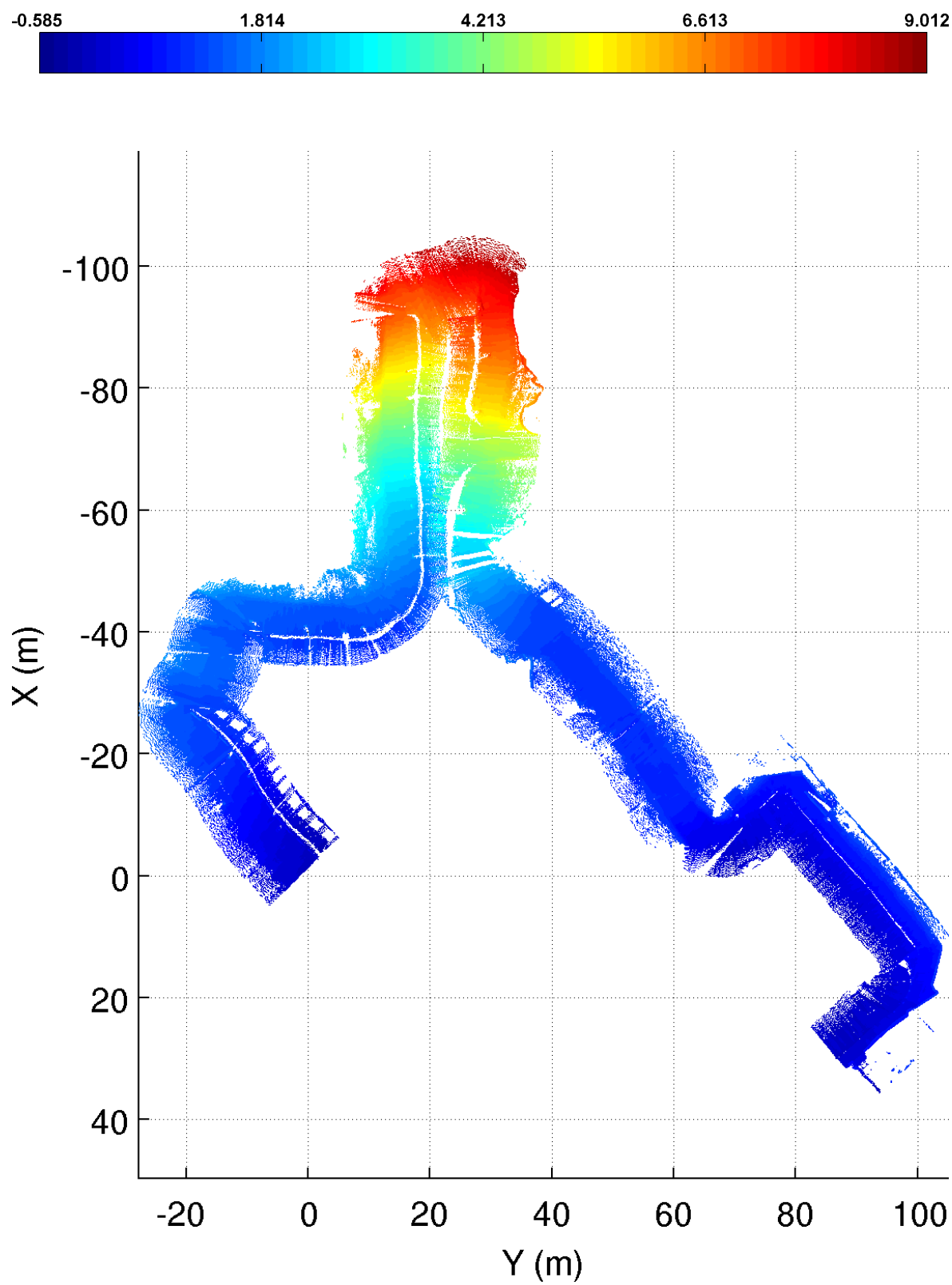


Figura 4.37. Mapeamento completo obtido para o Experimento 2.

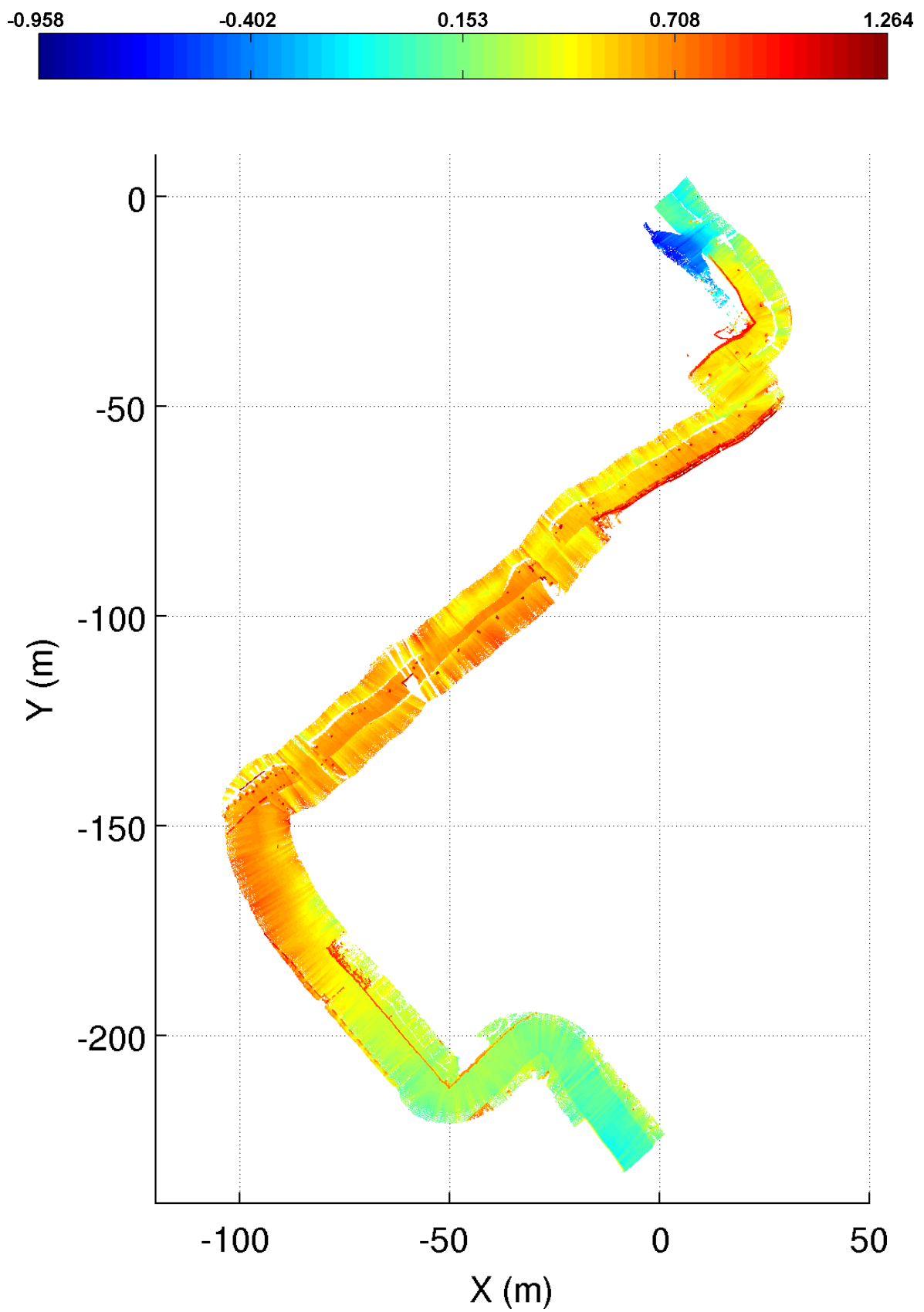


Figura 4.38. Mapeamento completo obtido para o Experimento 3.

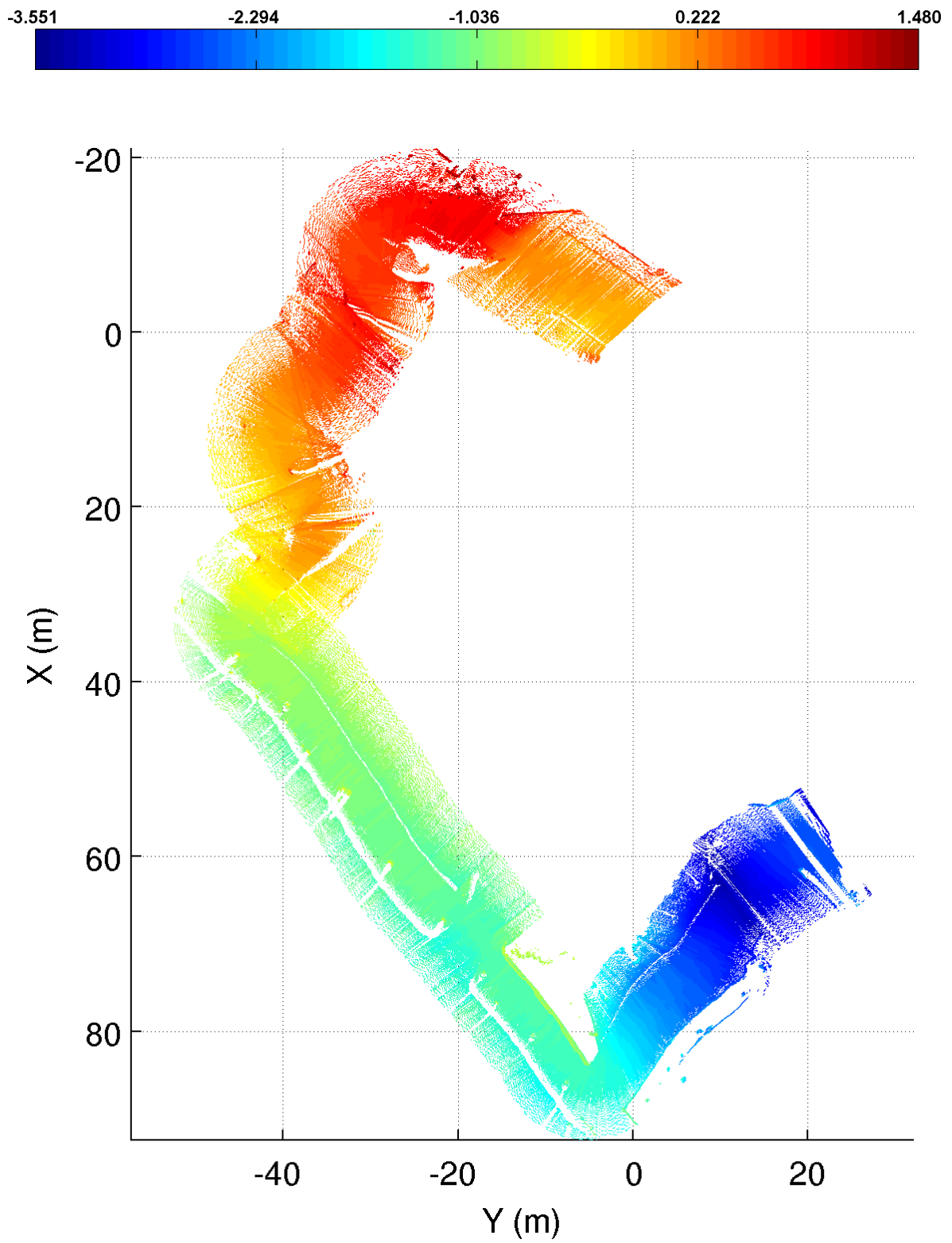


Figura 4.39. Mapeamento completo obtido para o Experimento 4.

# Capítulo 5

## Conclusões e Trabalhos Futuros

### 5.1 Conclusões

Nesse trabalho é apresentado uma metodologia para mapeamento de relevo a partir de robôs móveis navegando em ambientes externos e de terreno irregular. Inicialmente foi apresentado o sistema para o cálculo da posição e orientação do robô, em seguida, baseando-se nessa informação de localização, utiliza-se um sensor *laser* para geração de um mapa tridimensional (nuvem de pontos) do local por onde o robô trafegou.

Os resultados obtidos para a localização do robô foram satisfatórios, uma vez que através das imagens de satélites é possível observar que o caminho estimado através da fusão sensorial realizada com a utilização da metodologia proposta possui grande fidelidade ao caminho real realizado pelo robô. A variável com menor precisão é o valor estimado para a variação em  $z$  (altura). Como visto através do comportamento da covariância do erro, a imprecisão apenas tende a aumentar com o tempo de navegação, uma vez que não existe medida de correção. Uma melhor avaliação do sistema consistiria em realizar experimentos em um veículo que se locomovesse em maiores distâncias e também maiores velocidades.

Já os valores obtidos para os ângulos de orientação  $(\phi, \theta, \psi)$  apesar de apresentarem uma boa estimativa ainda possuem incorporado a si um certo grau de ruído. Isso está relacionado principalmente ao fato de se ter utilizado uma IMU de baixo custo, e que acaba por introduzir uma quantidade razoável de ruído nas medições. Um sistema que demandasse uma grande precisão no cálculo da orientação teria a necessidade de utilizar também uma IMU mais precisa, e conseqüente de custo mais elevado.

O valor que se mostrou mais preciso e confiável foi  $\psi$ , isso principalmente devido à introdução da bússola digital no sistema, uma vez que nos primeiros experimentos realizados (ainda sem a presença da bússola) o valor sofria uma grande deriva, pois seu

cálculo baseava-se exclusivamente em informações da odometria e velocidade angular ao redor do eixo Z (fornecida pela IMU). Também pode-se concluir que a qualidade na estimativa da orientação do robô está diretamente ligada ao terreno onde ele está trafegando, ou seja, a dinâmica de movimento do robô, pois quanto mais acidentado o terreno maior será o ruído incorporado à estimativa.

Apesar de os experimentos terem ocorrido de forma *offline* (processamento pós navegação), foi possível observar que a metodologia é eficiente. Isso leva a crer então que seria possível a realização do cálculo da posição do robô de forma *online*, ou seja, enquanto a navegação está ocorrendo.

A adaptação dinâmica da variância (podendo-se entender também como a confiança) relativa aos dados do GPS baseando-se no número de satélites visíveis e na velocidade do robô se mostrou uma técnica eficaz, permitindo lidar com alguns problemas fundamentais desse sistema (*random walk*, queda de sinal) e melhorar a estimativa de localização. Devido a isso, entende-se que seria interessante estender esse método aos demais sensores do sistema, por exemplo, a confiança nos dados da IMU poderia diminuir caso fosse notado que se está trafegando por um terreno muito acidentado, reduzindo assim o ruído incorporado no cálculo da orientação (e conseqüentemente na localização). É possível encontrar na literatura trabalhos que já utilizam a abordagem adaptativa para o KF [Hide et al., 2004; Hu et al., 2003; Gao et al., 2007].

Durante a parte experimental foi utilizado um sistema real, logo, foi possível também tirar-se determinadas conclusões sobre os aspectos operacionais envolvidos no sistema. Concluiu-se que apesar de o robô utilizado nos experimentos ser classificado como um robô *all terrain* isso realmente não pôde ser comprovado na prática, uma vez que seu sistema de direção *skid-steer* mostrou-se altamente ineficiente para aplicações em ambientes externos. Os principais problemas encontrados foram a dificuldade na realização de curvas fechadas e o fato de o robô não conseguir realizar o movimento de *turn-in-place* mesmo em solos que não possuem grande rugosidade (deve-se considerar também que o robô possui uma certa quantidade de peso sobre si devido ao equipamento utilizado, o quê influencia diretamente).

Outra dificuldade encontrada foi na utilização da bússola, que se mostrou extremamente sensível a variações magnéticas ao seu redor, o que levou à necessidade então de montá-la o mais distante possível de outros equipamentos que pudessem interferir em suas medições. Essa sensibilidade também dificultou sua utilização nos experimentos realizados no ambiente interno, esse fato deve ser considerado, uma vez que caso o robô passe próximo a algum local com um certo grau de atividade magnética o valor para a orientação será prejudicada.



O mapeamento realizado mostrou-se com uma qualidade satisfatória, permitindo sua utilização em sistemas de navegação do próprio robô (desvio local de obstáculos) ou de outros robôs que ainda não passaram pelas regiões já mapeadas (cálculo de trajetórias). É possível identificar pequenos obstáculos e detalhes do ambiente, tanto com variações positivas de altura quanto negativas. O principal problema do mapeamento encontrado foi o erro lateral, devido basicamente à variação existente no sinal do GPS, e que destaca-se principalmente em trajetos longos e retilíneos (*e.g.* Figura 4.36).

A montagem utilizada (*laser* fixo) também apresentou alguns problemas. Ao realizar um movimento de arfagem, o *laser* passa a realizar suas leituras em uma região mais à frente do que a que ele está no momento, deixando então espaços no mapa. Isso pode ser um problema durante o mapeamento, pois, caso o terreno onde o robô trafega produza muita variação em  $\phi$ , várias partes do terreno podem deixar de ser mapeadas, reduzindo a qualidade geral do mapa.

Por estar diretamente ligado à precisão da localização acredita-se que a utilização de sensores mais precisos e com um menor nível de ruídos produzirá ainda melhores resultados na etapa do mapeamento.

## 5.2 Trabalhos Futuros

O trabalho aqui apresentado possui aplicação direta em sistemas de navegação de veículos terrestres autônomos, e a seguir são apresentados os próximos passos a serem tomados para extensão do sistema.

Conforme visto anteriormente, o cálculo da variação de altura do robô foi a informação que mostrou-se mais problemática, podendo agregar uma grande quantidade de erro, uma vez que não possui um valor de correção durante a execução do UKF. Portanto, para solucionar esse problema é proposto a integração de um novo sensor ao sistema que forneça um valor absoluto para a altura do robô. Propõem-se então a utilização de um altímetro.

Outra extensão direta do sistema, consiste também na adição de uma câmera ao robô, tendo como finalidade a obtenção da informação de textura do local trafegado (terreno) e adição dessa informação de textura sobre o modelo geométrico levantado. Com o uso da câmera, também poderiam ser identificadas determinadas características no mundo (*e.g.* retas) e dessa forma, utilizar essas características para a realização de correções no mapa caso necessário (por exemplo o problema do desvio lateral).

Após feito isso, deseja-se realizar experimentos com o sistema acoplado a um veículo de maior porte, no caso um carro, onde seria possível verificar o comportamento do sistema para localização e mapeamento em um trajeto de maior escala do que o visto durante os experimentos aqui realizados.

O sistema é responsável apenas por criar o modelo tridimensional do ambiente (mais especificamente do terreno navegado), logo, podemos citar como trabalhos futuros secundários a implementação de aplicações que utilizariam o modelo gerado, dentre as possíveis podemos citar:

- Inicialmente, realizar um estudo sobre possíveis estruturas de dados que permitam armazenar e manipular as informações do mapa de forma eficaz e eficiente;
- Utilizando o dado de posição deseja-se extrair informações como nível de trepidação e orientações perigosas (*e.g.* ângulos de *roll* e *pitch* maiores que os aceitáveis para o veículo) e baseado no mapeamento realizado espera-se calcular o custo relacionado a cada parte do terreno por onde o robô navegou;
- A partir do custo inferido ao relevo é possível então implementar algoritmos que baseados nos lugares já previamente mapeados e classificados sejam analisados para a geração de novas trajetórias do robô [Pereira et al., 2006]. Considerando-se as possíveis características do ambiente e da aplicação a ser executada, por exemplo, pode-se desejar que o robô faça uma navegação mais suave (com menos trepidação) se a missão dada a ele consistir em filmar uma área, ou faça o caminho mais curto caso se queira ter um menor gasto energético;
- De acordo com a qualidade do mapa poderiam ser identificados determinados marcos e características presentes no ambiente, logo, seria possível para um robô baseado apenas com um *laser* se localizar globalmente no ambiente após identificar um marco cuja posição é conhecida. Da mesma forma, o robô também poderá ter uma localização absoluta mesmo sem utilizar o GPS.

# Referências Bibliográficas

- ActiveMedia Robotics (2007). MobileRobots Research & University Robots. Disponível em: <http://www.activrobots.com/>. Acesso em: 21 de agosto de 2007.
- Baevelde, A.-J. & Klammer, R. (1997). A low-cost and low-weight attitude estimation system for an autonomous helicopter. In *Proceedings of IEEE International Conference on Intelligent Engineering Systems (INES)*, pp. 391–395.
- Bailey, T. (2002). *Mobile Robot Localisation and Mapping in Extensive Outdoor Environments*. PhD thesis, University of Sydney.
- Bailey, T. & Durrant-Whyte, H. (2006). Simultaneous Localisation and Mapping (SLAM): Part II State of the Art. *IEEE Robotics and Automation Magazine*, 13(3):108–117.
- Besl, P. J. & McKay, N. D. (1992). A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256.
- Biesiadecki, J. J. & Maimone, M. W. (2006). The mars exploration rover surface mobility flight software driving ambition. In *Proceedings of the Aerospace Conference*.
- Borenstein, J. & Feng, L. (1996). Gyrodometry: A New Method for Combining Data from Gyros and Odometry in Mobile Robots. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pp. 423–428.
- Buehler, M.; Iagnemma, K. & Singh, S., editores (2007). *The 2005 Darpa Grand Challenge : The Great Robot Race*. Springer-Verlag New York, LLC.
- Choset, H.; Lynch, K. M.; Hutchinson, S.; Kantor, G. A.; Burgard, W.; Kavraki, L. E. & Thrun, S. (2005). *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press.

- Cloudcap (2007). Cloudcap Technology - Integrated autopilots, camera gimbals and inertial sensors for Unmanned Systems. Disponível em: <http://www.cloudcaptech.com/>. Acesso em: 4 de setembro de 2007.
- Craig, J. J. (1989). *Introduction to Robotics: Mechanics and Control*. Prentice Hall.
- Davison, A. J.; Cid, Y. G. & Kita, N. (2004). Real-Time 3D SLAM with Wide-Angle Vision. In *Proceedings of 5th IFAC/EURON Symposium on Intelligent Autonomous Vehicles*.
- Diosi, A. & Kleeman, L. (2004). Advanced Sonar and Laser Range Finder Fusion for Simultaneous Localization and Mapping. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.
- Dudek, G. & Jenkin, M. (2000). *Computational Principles of Mobile Robotics*. Cambridge University Press.
- Durrant-Whyte, H. & Bailey, T. (2006). Simultaneous Localisation and Mapping (SLAM): Part I The Essential Algorithms. *IEEE Robotics and Automation Magazine*, 13(2):99–110.
- Everett, H. R. (1995). *Sensors for mobile robots: theory and application*. A. K. Peters, Ltd.
- Gao, W.; Yang, Y.; Cui, X. & Zhang, S. (2007). Application of adaptive Kalman filtering algorithm in IMU/GPS integrated navigation system. *Geo-Spatial Information Science*, 10:22–26.
- Garcia, R. F.; Shiroma, P. M.; Chaimowicz, L. & Campos, M. F. M. (2007). Um arcabouço para a localização de enxames de robôs. In *Simpósio Brasileiro de Automação Inteligente*, Florianópolis, SC - Brazil.
- Garmin (2008). Follow the leader. Garmin. Disponível em: <http://www.garmin.com/>. Acesso em: 23 de abril de 2008.
- Gerkey, B. P.; Støy, K. & Vaughan, R. T. (2000). Player Robot Server. Technical Report IRIS-00-392, Institute for Robotics and Intelligent Systems, School of Engineering, University of Southern California.
- Goldberg, S. B.; Maimone, M. W. & Matthies, L. (2002). Stereo Vision and Rover Navigation Software for Planetary Exploration. *IEEE Aerospace Conference*, 5:2025–2036.

- Google Maps API (2008). Google Maps API - Google Code. Disponível em: <http://code.google.com/apis/maps/>. Acesso em: 29 de julho de 2008.
- Grewal, M. S. & Andrews, A. P. (2001). *Kalman Filtering: Theory and Practice Using MATLAB*. Wiley-Interscience, 2<sup>a</sup> edição.
- Grewal, M. S.; Weill, L. R. & Andrews, A. P. (2007). *Global Positioning Systems, Inertial Navigation, and Integration*. Wiley-Interscience, 2<sup>a</sup> edição.
- Hide, C.; Moore, T. & Smith, M. (2004). Adaptive Kalman filtering algorithms for integrating GPS and low cost INS. pp. 227–233.
- Howard, A.; Wolf, D. F. & Sukhatme, G. S. (2004). Towards 3D Mapping in Large Urban Environments. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 419–424, Sendai, Japan.
- Hu, C.; Chen, W.; Chen, Y. & Liu, D. (2003). Adaptive Kalman filtering for Vehicle Navigation. *Journal of Global Positioning Systems*, 2:42–47.
- Iscold, P.; de Oliveira, G. R. C.; Neto, A. A.; Pereira, G. A. S. & Torres, L. A. B. (2007). Desenvolvimento de Horizonte Artificial para Aviação Geral baseado em Sensores MEMS. In *V Congresso Brasileiro de Engenharia Inercial*, Rio de Janeiro.
- Julier, S. & Uhlmann, J. K. (1996). A General Method for Approximating Nonlinear Transformations of Probability Distributions. Technical report, University of Oxford.
- Julier, S. & Uhlmann, J. K. (1997). A new extension of the kalman filter to nonlinear systems. In *Proceedings of The 11th Int. Symp. on Aerospace/Defense Sensing, Simulation and Controls, Multi Sensor Fusion, Tracking and Resource Management*.
- Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering*, 82 (Series D):35–45.
- Katz, R.; Melkumyan, N.; Guivant, J.; Bailey, T.; Nieto, J. & Nebot, E. (2006). Integrated sensing framework for 3D mapping in outdoor navigation. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Beijing, China.
- Kurusu, M.; Yokokohji, Y. & Oosato, Y. (2005). Development of a Laser Range Finder for 3D Map-Building in Rubble. In *Proceedings of the IEEE International Conference on Mechatronics & Automation*, Niagara Falls, Canada.

- KVH (2008). KVH Industries, Inc.: Mobile Internet Satellite, Roaming, GPS Vehicle Tracking System. Disponível em: <http://www.kvh.com/>. Acesso em: 14 de março de 2008.
- Kwok, C.; Fox, D. & Meila, M. (2004). Real-time particle filters. *Proceedings of the IEEE*, 92:469–484.
- Li, Y.; Wang, J.; Rizos, C.; Mumford, P. & Ding, W. (2006). Low-cost Tightly Coupled GPS/INS Integration Based on a Nonlinear Kalman Filtering Design. In *National Technical Meeting of the Institute of Navigation*, pp. 958–966.
- Liu, B.; Adams, M. & Ibañez-Guzmán, J. (2005). Multi-aided Inertial Navigation for Ground Vehicles in Outdoor Uneven Environments. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, Barcelona, Espanha.
- Mahon, I. & Williams, S. (2003). Three-Dimensional Robotic Mapping. In *Proceedings of Australasian Conference on Robotics and Automation*, Brisbane, Australia.
- Maimone, M. W.; Biesiadecki, J. J.; Tunstel, E.; Cheng, Y. & Leger, C. (2006). Surface navigation and mobility intelligence on the Mars Exploration Rovers. In Howard, A. M. & Tunstel, E. W., editores, *Intelligence for Space Robotics*, pp. 45–69. TSI Press.
- Maimone, M. W.; Johnson, A.; Cheng, Y.; Willson, R. & Matthies, L. (2004). Autonomous navigation results from the mars exploration rover (mer) mission. In *Proceedings of the International Symposium on Experimental Robotics (ISER)*.
- Murphy, R. R. (2000). *An Introduction to AI Robotics*. The MIT Press.
- Mázl, R. & Preucil, L. (2000). Building a 2D Environment Map from Laser Range-Finder Data. In *Proceedings of IEEE Intelligent Vehicles Symposium*, pp. 290–295, Dearborn, USA.
- Ng, T. C. (2005). Development of a 3D LADAR system for autonomous vehicle guidance. Technical Report vol 6, no. 1, SIMTech Technical Reports. pp. 13-18.
- NOAA Satellite and Information Service (2008). USDOC/NOAA/NESDIS/National Geophysical Data Center (NGDC). Disponível em: <http://www.ngdc.noaa.gov/>. Acesso em: 17 de abril de 2008.
- Nüchter, A.; Lingemann, K.; Hertzberg, J. & Surmann, H. (2007). 6D SLAM – 3D Mapping Outdoor Environments. *Journal of Field Robotics (JFR), Special Issue*

- on Quantitative Performance Evaluation of Robotic and Intelligent Systems*, 24(8-9):699–722.
- Ojeda, L. & Borenstein, J. (2002). FLEXnav: Fuzzy Logic Expert Rule-based Position Estimation for Mobile Robots on Rugged Terrain. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pp. 317–322.
- Orderud, F. (2005). Comparison of Kalman Filter Estimation Approaches for State Space Models with Nonlinear Measurements. In *Proceedings of Scandinavian Conference on Simulation and Modeling (SIMS)*.
- Pereira, G. A. S.; de Araújo Pimenta, L. C.; Chaimowicz, L.; Fonseca, A. R.; de Almeida, D. S. C.; de Queiroz Corrêa, L.; Mesquita, R. C. & Campos, M. F. M. (2006). Robot navigation in multi-terrain outdoor environments. In *Proceedings of the International Symposium on Experimental Robotics (ISER)*.
- Player (2007). The Player Project. Disponível em: <http://playerstage.sourceforge.net/>. Acesso em: 13 de agosto de 2007.
- Ryde, J. & Hu, H. (2007). Mobile Robot 3D Perception and Mapping without Odometry Using Multi-Resolution Occupancy Lists. In *IEEE International Conference on Mechatronics and Automation*, pp. 331–336.
- Sick (2007). SICK. Disponível em: <http://www.sick.com/>. Acesso em: 5 de setembro de 2007.
- Siegwart, R. & Nourbakhsh, I. R. (2004). *Introduction to Autonomous Mobile Robots*. The MIT Press.
- Stavens, D. & Thrun, S. (2006). A Self-Supervised Terrain Roughness Estimator for Off-Road Autonomous Driving. In *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI)*.
- Sukkarieh, S.; Nebot, E. M. & Durrant-Whyte, H. F. (1999). A high integrity IMU/GPS navigation loop for autonomous land vehicle applications. In *Proceedings of IEEE Transactions on Robotics and Automation*, pp. 572–578.
- Surmann, H.; Nüchter, A. & Hertzberg, J. (2003). An autonomous mobile robot with a 3D laser range finder for 3D exploration and digitalization of indoor environments. *Robotics and Autonomous Systems*, 45(3-4):18–198.

- Swerling, P. (1958). A proposed stagewise differential correction procedure for satellite tracking and predictions. Technical Report P-1292, RAND Corporation.
- Thompson, S. & Kagami, S. (2004). Stereo vision terrain modeling for non-planar mobile robot mapping and navigation. *IEEE International Conference on Systems, Man and Cybernetics*, 6:5392–5397.
- Thrun, S. (2002). Robotic Mapping: A Survey. In Lakemeyer, G. & Nebel, B., editores, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann.
- Thrun, S.; Burgard, W. & Fox, D. (2005). *Probabilistic Robotics*. The MIT Press.
- Thrun, S.; Diel, M. & Hähnel, D. (2003). Scan Alignment and 3D Surface Modeling with a Helicopter Platform. In *Proceedings of International Conference on Field and Service Robotics*, Lake Yamanaka, Japan.
- Thrun, S.; Montemerlo, M.; Dahlkamp, H.; Stavens, D.; Aron, A.; Diebel, J.; Fong, P.; Gale, J.; Halpenny, M.; Hoffmann, G.; Lau, K.; Oakley, C.; Palatucci, M.; Pratt, V.; Stang, P.; Strohband, S.; Dupont, C.; Jendrossek, L.-E.; Koelen, C.; Markey, C.; Rummel, C.; van Niekerk, J.; Jensen, E.; Alessandrini, P.; Bradski, G.; Davies, B.; Ettinger, S.; Kaehler, A.; Nefian, A. & Mahoney, P. (2006). Stanley: The robot that won the DARPA Grand Challenge: Research Articles. *Journal of Robotic Systems*, 23(9):661--692.
- Titterton, D. H. & Weston, J. L. (2004). *Strapdown Inertial Navigation Technology*. Institution of Engineering and Technology, 2ª edição.
- Velodyne (2008). HD Lidar by Velodyne. Disponível em: <http://www.ghostriiderrobot.com/hdlidar/>. Acesso em: 6 de junho de 2008.
- Wagner, D. & Schmalstieg, D. (2007). ARToolKitPlus for Pose Tracking on Mobile Devices. In *12th Computer Vision Winter Workshop (CVWW)*.
- Walchko, K. J. & Mason, P. A. C. (2002). Inertial navigation. In *Florida Conference on Recent Advances in Robotics*.
- Wan, E. A. & van der Merwe, R. (2000). The Unscented Kalman Filter for Nonlinear Estimation. In *Proceedings of Symposium 2000 on Adaptive Systems for Signal Processing, Communication and Control (AS-SPCC)*, pp. 153–158, Alberta, Canada.
- Welch, G. & Bishop, G. (1995). An Introduction to the Kalman Filter. Technical Report TR 95-041, University of North Carolina at Chapel Hill.



- Wulf, O.; Nüchter, A.; Hertzberg, J. & Wagner, B. (2007). Ground Truth Evaluation of Large Urban 6D SLAM. In *Proceedings of International Conference on Intelligent Robots and Systems (IROS)*, San Diego, USA.
- Zhang, P.; Gu, J.; Milios, E. E. & Huynh, P. (2005). Navigation with IMU/GPS/Digital Compass with Unscented Kalman Filter. In *Proceedings of IEEE International Conference on Mechatronics & Automation*, Niagara Falls, Canada.

# Apêndice A

## Recursos Utilizados

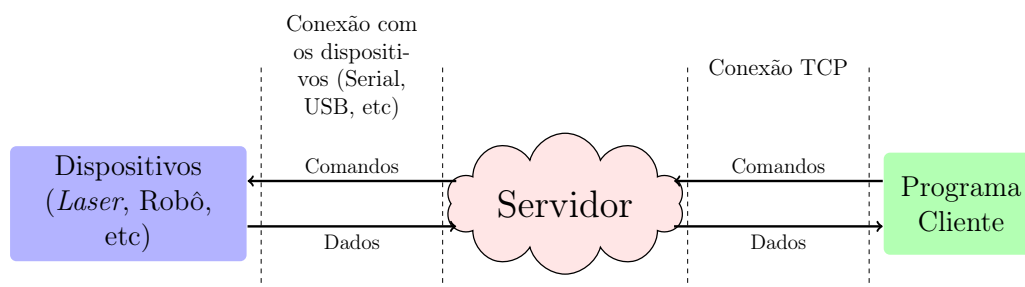
Nas próximas seções são descritos todos os recursos utilizados ao longo do projeto, abordando as partes de *software* e *hardware*.

### A.1 Plataforma de Programação

Para a implementação do sistema utilizou-se a plataforma *Player* [Player, 2007], sendo essa uma das interfaces de programação para sistemas robóticos mais utilizadas atualmente. Pode-se entender o *Player* como um sistema operacional para robôs, uma vez que ele age como uma camada de abstração de *hardware* para dispositivos robóticos. Essa é uma plataforma que desempenha basicamente dois papéis, sendo eles:

- Interface entre os equipamentos, permitindo uma programação em mais alto nível;
- Servidor de controle para o robô.

A plataforma *Player* possui duas divisões básicas, o servidor e o cliente [Gerkey et al., 2000]. O servidor é executado no próprio robô, e fornece uma interface de comunicação via rede IP. O programa cliente se comunica com o servidor através de *socket* TCP, Figura A.1.



**Figura A.1.** Esquema da plataforma *Player*.

A plataforma pode ser executada em sistemas *Posix*, incluindo Linux, Mac OS X, Solaris e BSD. Além de permitir uma programação independente do sistema robótico a ser utilizado possui suporte à diferentes linguagens, entre elas C, C++, Java e Python, sendo que para o desenvolvimento desse trabalho utilizou-se a linguagem de programação C++.

## A.2 Robô

A plataforma robótica móvel utilizada consiste no *Pioneer P3-AT*, desenvolvido pela empresa *ActiveMedia Robotics* [ActiveMedia Robotics, 2007].

Principais características:

### Dados gerais

Dimensões:	50 cm x 49 cm x 26 cm
Peso:	20 kg (com todas as baterias)
Velocidade máxima de translação:	0,7 m/s
Sistema de direção:	Skid-steer
Altura máxima transponível:	10 cm
Vão máximo transponível:	15,2 cm
Inclinação máxima transponível:	20°
Carga máxima:	até 30 kg

**Tabela A.1.** Especificações técnicas - *Pioneer P3-AT*.

### A.3 Sensor *Laser*

Para a coleta de dados do mapeamento utilizou um sensor *laser* da empresa *Sick* [Sick, 2007], modelo LMS 200.

Principais características:

#### Dados gerais

Campo de visão:	180°
Resolução:	1 mm / 10 mm
Resolução angular:	0,25° / 0,5° / 1°
<i>Range</i> máximo:	8,191 m / 81,91 m
Peso:	4,5 kg
Dimensões:	15,6 cm x 15,5 cm x 21,0 cm
Interface:	RS-232, 38400 bps

**Tabela A.2.** Especificações técnicas - *laser SICK* LMS 200.

### A.4 Unidade de Medição Inercial

A unidade de medição inercial utilizada no sistema consiste em um sensor *Crista IMU* da *Cloud Cap Technology* [Cloudcap, 2007].

Principais características:

#### Giroscópios

Sensores:	3 sensores ortogonais
Faixa de atuação:	$\pm 300^\circ/s$
Resolução:	$0,009^\circ/s$

#### Acelerômetros

Sensores:	3 sensores ortogonais
Faixa de atuação:	$\pm 10 g$
Resolução:	$0,3mg$

#### Demais informações

Peso:	36,8 g
Dimensões:	5,20 cm x 3,93 cm x 2,54 cm
Interface:	RS-232, 115200 bps
Aceleração máxima:	500 g

**Tabela A.3.** Especificações técnicas - *Crista IMU*.

A unidade  $g$  aqui utilizada representa o valor da aceleração gravitacional, sendo  $g \approx 9,78 m/s^2$ .

## A.5 Bússola Digital

O equipamento utilizado consiste em uma bússola digital modelo C-100 da *KVH Industries, Inc.* [KVH, 2008].

Principais características:

### Dados gerais

Precisão:	$\pm 0,5^\circ$
Resolução:	$0,1^\circ$
Peso:	400 g
Dimensões:	8,0 cm x 7,5 cm x 5,7 cm
Interface:	RS-232, 9600 bps

**Tabela A.4.** Especificações técnicas - bússola digital KVH C-100.

## A.6 GPS

Para a coleta de dados da posição absoluta do robô no referencial terrestre foi utilizado um aparelho de GPS modelo 18x-5Hz da empresa *Garmin* [Garmin, 2008].

Principais características:

### Dados gerais

Precisão:	< 15 metros (95% dos casos)
Peso:	165 g (com cabeamento incluído)
Dimensões:	6,1 cm (diâmetro) x 1,95 cm (altura)
Interface:	RS-232, 19200 bps

**Tabela A.5.** Especificações técnicas - GPS Garmin 18x-5Hz.