



**ESTRATÉGIA DE MAPEAMENTO EM 2D DE AMBIENTES
INTERNOS BASEADA NA EXTRAÇÃO DE SEGMENTOS
DE LINHAS USANDO SENSORES DE ULTRASSOM**

CLAUDIA PATRICIA OCHOA DIAZ

**DISSERTAÇÃO DE MESTRADO EM SISTEMAS MECATRÔNICOS
DEPARTAMENTO DE ENGENHARIA MECÂNICA**

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia
Departamento de Engenharia Mecânica

**ESTRATÉGIA DE MAPEAMENTO EM 2D DE AMBIENTES
INTERNOS BASEADA NA EXTRAÇÃO DE SEGMENTOS
DE LINHAS USANDO SENSORES DE ULTRASSOM**

CLAUDIA PATRICIA OCHOA DIAZ

ORIENTADOR: ALBERTO JOSÉ ALVARES

**DISSERTAÇÃO DE MESTRADO EM SISTEMAS
MECATRÔNICOS**

Brasília-D.F, Julho de 2010

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia
Departamento de Engenharia Mecânica

**ESTRATÉGIA DE MAPEAMENTO EM 2D DE AMBIENTES
INTERNOS BASEADA NA EXTRAÇÃO DE SEGMENTOS
DE LINHAS USANDO SENSORES DE ULTRASSOM**

CLAUDIA PATRICIA OCHOA DIAZ

Dissertação de Mestrado submetida ao Departamento de Engenharia
Mecânica da faculdade de Tecnologia da Universidade de Brasília
como parte dos requisitos necessários para a obtenção
do grau de Mestre em Sistemas Mecatrônicos

Banca Examinadora

Prof. Alberto José Álvares, Dr., _____
ENM/UnB
Orientador

Prof. Geovany Araújo Borges, Dr., _____
ENE/UnB
Examinador externo

Prof. Carla Maria Chagas e Cavalcante _____
Koike, Dra. ENM/UnB
Examinador interno

Brasília, 15 de Julho de 2010

FICHA CATALOGRÁFICA

OCHOA D., CLAUDIA PATRICIA

Estratégia de Mapeamento em 2D de Ambientes Internos Baseada na Extração de Segmentos de Linhas Usando Sensores de Ultrassom. [Distrito Federal] 2010.

VIII, 75p. 210 × 297 mm (ENM/FT/UnB, Mestre, Sistemas Mecatrônicos, 2010). Dissertação de Mestrado – Universidade de Brasília. Faculdade de Tecnologia.

Departamento de Engenharia Mecânica.

- | | |
|------------------------|------------------------------------|
| 1. Robótica Móvel | 2. Mapeamento |
| 3. Sensor de Ultrassom | 4. Detecção de Segmentos de Linhas |
| I. ENM/FT/UnB | II. Título (série) |

REFERÊNCIA BIBLIOGRÁFICA

OCHOA D., C. P. (2010). Estratégia de Mapeamento em 2D de Ambientes Internos Baseada na Extração de Segmentos de Linhas Usando Sensores de Ultrassom. Dissertação de Mestrado em Sistemas Mecatrônicos, Publicação ENM.DM-34/10, Departamento de Engenharia Mecânica, Universidade de Brasília, Brasília, DF, 75p.

CESSÃO DE DIREITOS

AUTOR: Claudia Patricia Ochoa Diaz.

TÍTULO: Estratégia de Mapeamento em 2D de Ambientes Internos Baseada na Extração de Segmentos de Linhas Usando Sensores de Ultrassom.

GRAU: Mestre

ANO: 2010

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte dessa dissertação de mestrado pode ser reproduzida sem autorização por escrito do autor.

Claudia Patricia Ochoa Diaz

SCLN 408 Bloco A, Apartamento 223

70856-510 Brasília-DF-Brasil.

Para meus pais, Hernando e Isaura, seres absolutamente valiosos que me deram uma educação exemplar.

Para Margarita, minha melhor amiga.

Para Jorge, meu colega, meu amigo e meu amor.

CLAUDIA PATRICIA OCHOA DIAZ

Agradecimentos

Quero agradecer a todas aquelas pessoas que de forma direta e indireta me colaboraram na execução dessa dissertação de mestrado. A todos eles quero expressar-lhes minha sincera gratidão.

Ao meu orientador, Prof. Alberto José Álvares, agradeço pela confiança depositada em mim e pela orientação desse trabalho.

Aos colegas de Laboratório do Grupo de Automação e Controle (GRACO), pelo apoio constante e vontade de ajuda. Em especial agradeço a Liliana, Ronald, Jesús, Daniel, Janier, Jones, Ênio Prates, Edgar, Evandro, e por último, mas não menos importante, ao técnico de Laboratório Antônio Marrocos, por sua incansável disposição em todos os momentos em que eu precisei de sua ajuda.

Agradeço também a minha família, representada no Brasil por minha irmã Margarita, quem tem sido um suporte fundamental nesta etapa da minha vida fora de casa, Obrigada pela força!

Ao Jorge, por ser esse motor que me motiva a me superar cada dia. Obrigada pela paciência, pelos conselhos e pelas palavras de carinho.

Por fim, agradeço à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro para a realização deste projeto.

CLAUDIA PATRICIA OCHOA DIAZ

RESUMO

ESTRATÉGIA DE MAPEAMENTO EM 2D DE AMBIENTES INTERNOS BASEADO EM EXTRAÇÃO DE SEGMENTOS DE LINHAS USANDO SENSORES DE ULTRASSOM

Autor: Claudia Patricia Ochoa Diaz

Orientador: Prof. Alberto José Álvares, EME/UnB

Programa de Pós-graduação em Sistemas Mecatrônicos

Brasília, Julho de 2010

Entre as diferentes *features* existentes, o segmento de linha é uma das primitivas geométricas de mais fácil representação para descrever ambientes internos. Vários trabalhos recentes que propõem diferentes estratégias de mapeamento baseados na detecção de segmentos de linhas utilizam medições provenientes de sensores como lasers e sensores de visão, sendo menos comum o uso de sensores de ultrassom para estes propósitos.

Este trabalho apresenta uma estratégia de mapeamento baseado em segmentos de linhas usando a informação fornecida por um grupo de sensores de ultrassom. A estratégia consiste basicamente em dois processos, a extração das linhas e o processo de combinação ou *merging*. Na etapa de extração, é implementado um algoritmo baseado na abordagem Incremental para a segmentação de linhas. A saída deste processo está composta pelas diferentes linhas estimadas a partir do conjunto de dados fornecidos pelos sonares. Já a etapa de *merging* tem como objetivo agrupar as diferentes linhas que representam a mesma estrutura real, utilizando para isto, um teste chi-quadrado como critério de combinação. Por fim, junto com os dados de posição dados pelo módulo de odometria da plataforma robótica, é construído um mapa global a partir das linhas estimadas. A validação da estratégia é feita tanto em um ambiente simulado quanto em um ambiente real, onde foram representadas três situações com o objetivo de avaliar o desempenho do algoritmo quando o robô está diante de mudanças no entorno explorado.

ABSTRACT

A STRATEGY FOR LINE-BASED MAPPING FOR INDOOR ENVIRONMENTS USING ULTRASONIC SENSORS

Author: Claudia Patricia Ochoa Diaz

Supervisor: Prof. Alberto José Álvares, EME/UnB

Programa de Pós-graduação em Sistemas Mecatrônicos

Brasília, July of 2010

Among many features, line segment is one of the simplest features for describing indoor environments. Many recent works which propose different strategies for line-based maps use measurements that come from sensors like lasers and vision sensors, being less common the use of ultrasonic sensor for this purpose.

This work presents a line-based mapping strategy using collected information from a set of ultrasonic sensors. This approach is mainly composed by two processes, the line segments extraction and the merging process. For the line extraction phase, an algorithm based on the Incremental approach for line segmentation is used. The output from this phase consists on a set of estimated lines that were extracted from the raw data. After that, a merging process is performed in order to cluster those extracted lines that match to the same planar structure. A chi-squared test is used as grouping criterion for this process. Finally, along with the odometry data provided by the robotic platform, a map in global coordinates is constructed from the extracted line segments. This strategy is validated using a simulated environment as well as a real one, where three different situations were represented in order to evaluate the performance of the algorithm when the robot faces changes in the explored area.

SUMÁRIO

| | | |
|----------|---|-----------|
| 1 | INTRODUÇÃO | 1 |
| 1.1 | CONTEXTUALIZAÇÃO | 1 |
| 1.1.1 | INTRODUÇÃO À ROBÓTICA MÓVEL | 1 |
| 1.1.2 | ENFOQUE PROBABILÍSTICO DA ROBÓTICA MÓVEL | 2 |
| 1.1.3 | ESTIMAÇÃO DE PARÂMETROS | 3 |
| 1.2 | MOTIVAÇÃO | 3 |
| 1.2.1 | TRABALHOS RELACIONADOS FEITOS NO GRACO | 3 |
| 1.2.2 | MAPAS GEOMÉTRICOS A PARTIR DE DADOS PROVENIENTES DE SENSORES DE ULTRASSOM | 4 |
| 1.3 | DEFINIÇÃO DO PROBLEMA | 4 |
| 1.4 | OBJETIVOS | 6 |
| 1.4.1 | OBJETIVO GERAL | 6 |
| 1.4.2 | OBJETIVOS ESPECÍFICOS | 6 |
| 1.5 | APRESENTAÇÃO DO DOCUMENTO | 6 |
| 2 | REVISÃO BIBLIOGRÁFICA | 7 |
| 2.1 | PERCEPÇÃO | 7 |
| 2.2 | SISTEMA SENSORIAL DA PLATAFORMA NOMAD XR4000 | 9 |
| 2.2.1 | MÓDULO DE ODOMETRIA | 10 |
| 2.2.2 | SENSOR DE ULTRASSOM | 12 |
| 2.3 | MAPEAMENTO | 16 |
| 2.3.1 | MAPEAMENTO BASEADO EM GRADES DE OCUPAÇÃO | 17 |
| 2.3.2 | MAPEAMENTO BASEADO EM EXTRAÇÃO DE <i>features</i> | 18 |
| 3 | ESTRATÉGIA DE MAPEAMENTO | 24 |
| 3.1 | MODELO DE MEDIÇÃO | 24 |
| 3.1.1 | ANTECEDENTES | 24 |
| 3.1.2 | MODELO ESCOLHIDO | 26 |
| 3.2 | PRÉ-PROCESSAMENTO DOS DADOS | 27 |
| 3.2.1 | TRANSFORMAÇÃO DE COORDENADAS | 27 |
| 3.2.2 | PROPAGAÇÃO DO ERRO | 29 |
| 3.3 | MODELO DO SEGMENTO DE LINHA | 30 |
| 3.3.1 | ESTIMAÇÃO DOS PARÂMETROS DA LINHA RETA | 31 |
| 3.3.2 | CÁLCULO DE $C_{\hat{\theta}}$ | 33 |

| | | |
|----------|---|-----------|
| 3.3.3 | EXTRAÇÃO DOS SEGMENTOS DE LINHA | 33 |
| 3.4 | <i>MERGING</i> | 36 |
| 4 | RESULTADOS EXPERIMENTAIS | 39 |
| 4.1 | CRITÉRIOS DE AVALIAÇÃO | 39 |
| 4.1.1 | ERRO DE ESTIMAÇÃO DOS PARÂMETROS | 39 |
| 4.1.2 | ASSOCIAÇÃO..... | 40 |
| 4.2 | SIMULAÇÃO..... | 40 |
| 4.2.1 | DESCRIÇÃO DO AMBIENTE SIMULADO..... | 40 |
| 4.2.2 | IMPLEMENTAÇÃO DO ALGORITMO | 41 |
| 4.3 | VALIDAÇÃO EXPERIMENTAL NA PLATAFORMA NOMAD XR4000 | 47 |
| 4.3.1 | DESCRIÇÃO DO AMBIENTE | 47 |
| 4.3.2 | PARÂMETROS INICIAIS GERAIS DO ALGORITMO | 49 |
| 4.3.3 | EXPERIMENTO 1..... | 50 |
| 4.3.4 | EXPERIMENTO 2..... | 55 |
| 4.3.5 | EXPERIMENTO 3..... | 57 |
| 4.4 | DISCUSSÃO FINAL DOS RESULTADOS..... | 60 |
| 4.4.1 | COMPARAÇÃO DO DESEMPENHO DA ESTRATÉGIA COM OUTROS TRABA- | |
| | LHOS..... | 60 |
| 5 | CONCLUSÕES..... | 62 |
| 5.1 | PROPOSTAS PARA TRABALHOS FUTUROS..... | 64 |
| | REFERÊNCIAS BIBLIOGRÁFICAS | 65 |
| | APÊNDICES..... | 69 |
| A | CÁLCULO COMPLETO DA MATRIZ DE COVARIÂNCIA $C_{\hat{\theta}}$..... | 70 |
| B | ESTIMAÇÃO DO ERRO DOS SENSORES DE ULTRASSOM..... | 72 |
| C | ESTIMAÇÃO DO ERRO DE POSICIONAMENTO DO ROBÔ..... | 74 |

LISTA DE FIGURAS

| | | |
|------|---|----|
| 1.1 | Exemplos de aplicações de robôs móveis..... | 2 |
| 2.1 | Esquemático geral de um sistema robótico móvel (Adaptado de [17]). | 8 |
| 2.2 | Plataforma Nomad XR4000..... | 10 |
| 2.3 | Codificador Incremental (Adaptado de [23]). | 11 |
| 2.4 | Mecanismo de funcionamento do sensor de ultrassom (Adaptado de [14]). | 13 |
| 2.5 | Padrão de propagação do sinal de ultrassom [11]. | 13 |
| 2.6 | Fenômenos de reflexão especular (Adaptado de [10]). | 15 |
| 2.7 | Configuração dos sonares no robô Nomad XR4000 (vista superior). Os pontos verdes representam cada um dos sensores que compõem um anel..... | 15 |
| 2.8 | Mapa topológico de um ambiente interno [17]. Os círculos com números representam os nós que denotam áreas importantes do ambiente, neste caso, área livre do objetos. As linhas são a conexão entre nós..... | 17 |
| 2.9 | Exemplos de mapas baseados em grades de ocupação..... | 19 |
| 2.10 | Mapas baseado em extração de segmentos de linhas..... | 22 |
| 3.1 | Estratégia de mapeamento implementada. | 25 |
| 3.2 | Padrão de medida do sensor de ultrassom..... | 25 |
| 3.3 | Padrão de medida de uma estrutura plana detectada por um grupo de sonares dispostos ao redor de uma circunferência segundo [41]. d_n é a distância medida e φ_n é o ângulo de orientação de cada sensor com respeito a sua direção normal de incidência, representada pelas linhas ponteadas de cor cinza. | 26 |
| 3.4 | Transformação das medidas à referência global. | 28 |
| 3.5 | Situação ilustrativa de extração de linhas. | 34 |
| 3.6 | Casos estabelecidos para o segundo critério no processo de <i>merging</i> . As linhas são comparadas em termos dos pontos que as compõem, delimitadas por seus pontos iniciais e finais. Para fins ilustrativos, uma das linhas foi colocada embaixo da outra. | 37 |
| 4.1 | Ambiente simulado. | 41 |
| 4.2 | Sequência da exploração do ambiente executado pelo robô. | 42 |
| 4.3 | Resultado da extração das linhas. | 44 |

| | | |
|------|---|----|
| 4.4 | Resultado final do algoritmo incremental básico e a modificação implementada. As linhas azuis correspondem às linhas estimadas, o mapa do ambiente real está desenhado com linhas ponteadas, e as leituras associadas a cada linha estão representadas pelos pontos de cor carmim. | 46 |
| 4.5 | Esquema do ambiente real. A referência global é localizada no extremo inferior esquerdo, que coincide com o canto formado pelas paredes L_1 e L_6 . A posição inicial do robô é dada pelo ponto $\{x_r, y_r\}$ | 48 |
| 4.6 | Imagens do ambiente real construído. | 49 |
| 4.7 | Representação do ambiente real no algoritmo programado. Os pontos azuis representam cada uma das posições do robô durante a trajetória executada..... | 51 |
| 4.8 | Saída do algoritmo após da etapa de pré-processamento das medições. | 52 |
| 4.9 | Dados após da compensação dos dados de odometria. | 52 |
| 4.10 | Resultado final de estratégia de mapeamento para o experimento 1. | 53 |
| 4.11 | Imagens da execução do teste 2. | 55 |
| 4.12 | Resultado final da estratégia de mapeamento para o teste 2. | 56 |
| 4.13 | Ambiente modificado para a execução do teste 3. No quadrado verde ressalta-se a presença do espaço feito na parede L_2 | 58 |
| 4.14 | Resultados da estratégia de mapeamento para o teste 3..... | 59 |
| B.1 | Vista superior da configuração do teste. O robô está posicionado de frente à estrutura plana a uma distância D_n . As diferentes distâncias detectadas pelos 5 sensores usados são coletadas para cada mudança de D_n | 72 |
| B.2 | Curvas do erro sistemático para cada distância D_n medida. | 73 |
| B.3 | Identificação dos sensores segundo sua orientação com respeito aos eixos x^R e y^R | 73 |
| C.1 | Medições dos deslocamentos efetuados nos eixos x e y do robô. Os pontos azuis representam as medições, enquanto a linha pontuada vermelha é o valor médio da distância de deslocamento. | 75 |
| C.2 | Medições compensadas com $\varepsilon_{s,x}$ e $\varepsilon_{s,y}$ | 75 |

LISTA DE TABELAS

| | | |
|------|--|----|
| 1.1 | Principais Trabalhos baseados no uso do sonar | 5 |
| 2.1 | Principais Sensores Exteroceptivos (Adaptado de [17]) | 9 |
| 3.1 | Representações matemáticas para uma linha reta..... | 30 |
| 4.1 | Cálculos dos erros de estimação. As linhas agrupadas com os mesmos símbolos (*, **, †) são aquelas combinadas erroneamente..... | 47 |
| 4.2 | Indicadores de associação..... | 47 |
| 4.3 | Parâmetros iniciais do algoritmo..... | 50 |
| 4.4 | Cálculo dos erros de estimação para o experimento 1 | 54 |
| 4.5 | Indicadores de associação..... | 54 |
| 4.6 | Cálculo dos erros de estimação para o experimento 2 | 57 |
| 4.7 | Indicadores de associação..... | 57 |
| 4.8 | Cálculo dos erros de estimação para o experimento 3 | 58 |
| 4.9 | Indicadores de associação..... | 58 |
| 4.10 | Comparação do resultado da estratégia proposta com o trabalho de [38] | 61 |

LISTA DE SÍMBOLOS

Símbolos Latinos

| | | |
|------------|--|-------|
| r | Raio do robô | [mm] |
| d_n | Medição de distância proveniente do sensor | [mm] |
| D_n | Valor real da informação de distância | [mm] |
| x_R | Coordenada x do robô | [mm] |
| y_R | Coordenada y do robô | [mm] |
| θ_R | Orientação do robô | [rad] |
| d_{euc} | Distância euclidiana medida | [mm] |
| d_{pl} | Distância entre ponto e linha estimada | [mm] |

Símbolos Gregos

| | | |
|-----------------|---|---------------------|
| φ_i | Ângulo de orientação do sensor | [rad] |
| β | Feixe angular | [rad] |
| ϕ_n | Medida de orientação do obstáculo detectado | [rad] |
| ε_s | Erro sistemático da medição de distância | [mm] |
| ε_a | Erro aleatório da medição de distância | [mm] |
| θ | Vetor das estimativas dos parâmetros da linha | [mm] e [rad] |
| σ_d^2 | Incerteza na medição de distância | [mm ²] |
| σ_ϕ^2 | Incerteza na estimativa de orientação | [rad ²] |

Grupos Adimensionais

| | |
|-------------|--|
| C_{xy}^G | Matriz de incerteza das medições transformadas |
| C_{robot} | Matriz de incerteza da posição do robô |

Subscritos

| | |
|-----|---|
| n | Referente ao número de medições |
| k | Referente ao número de linhas estimadas |
| i | Referente ao número de sonares |

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

1.1.1 Introdução à robótica móvel

Os avanços mais significativos da robótica datam desde os anos 60, quando foram introduzidos os primeiros manipuladores industriais para realizar tarefas que até esse momento só eram feitas pelos seres humanos. Posteriormente, na medida em que os robôs se incorporaram em outros campos de aplicação, surgiram outros tipos de requerimentos para melhorar aspectos como flexibilidade, mobilidade e autonomia.

Robótica Móvel é uma área de estudo relativamente nova¹ que abrange os sistemas robóticos movimentados por elementos locomotivos. Suas raízes incluem várias disciplinas da engenharia, que vão desde a mecânica, passando pela elétrica e eletrônica, assim como as ciências da computação e sociais.

Esta multidisciplinaridade evidencia o amplo número de aplicações dos diferentes robôs móveis construídos. Por citar alguns temas (ver figura 1.1), por exemplo, robôs móveis exploradores usados nas missões a Marte, robôs aéreos não tripulados em missões militares de alto risco, robôs em ambientes industriais com capacidades de navegação autônoma, ou robôs de serviço em clínicas e hospitais encarregados do transporte de medicamentos e instrumentos.

Independente da aplicação final da plataforma robótica, as tarefas realizadas por estes relacionam principalmente o estudo de quatro aspectos que têm-se convertido em áreas específicas de pesquisa, estes são a percepção, o mapeamento, a localização e a navegação.

Posterior à tarefa de percepção, a tarefa de mapeamento tem como objetivo construir uma representação do ambiente a partir das informações fornecidas pelo sistema sensorial do robô. Este mapa então representa o conhecimento que o robô possui sobre as estruturas que se encontram neste ambiente, e que lhe permitirão cumprir com determinadas tarefas subsequentes, como por exemplo, estimar a localização destas estruturas segundo uma referência global, assim como sua própria posição dentro do ambiente, ou a execução de determinada trajetória desde um ponto inicial até um ponto final. Portanto, a tarefa de mapeamento converte-se em uma etapa fundamental em um esquema de controle de uma plataforma robótica que vá desde a percepção do ambiente até a navegação dentro do mesmo.

¹considera-se por alguns autores que o início formal de robótica móvel é no começo dos anos 90, quando aparecem os primeiros robôs móveis com as características básicas que prevalecem até a atualidade.



(a)



(b)



(c)



(d)

Figura 1.1: Exemplos de aplicações de robôs móveis: **(a)** Robô *Opportunity*, utilizado na missão de exploração de Marte (<http://marsrover.nasa.gov/home/>); **(b)** RQ-1/MQ-1 Predator, utilizado em missões militares de reconhecimento (<http://www.ga-asi.com/>); **(c)** Tugger AGV, robô transportador de materiais em linhas de produção (<http://www.agvrobots.com/tugger.html>); **(d)** TUG, robô de serviço hospitalar (<http://www.umm.edu/news/releases/robot.htm>).

1.1.2 Enfoque probabilístico da robótica móvel

A probabilidade na robótica móvel está relacionada principalmente com as etapas de percepção e decisão, sendo uma ferramenta útil na hora de representar informação e tomar decisões [1]. Especificamente, a probabilidade na robótica tenta representar os erros aleatórios próprios de determinadas condições do ambiente assim como dos diferentes tipos de sensores utilizados durante o processo de percepção. Este erro afeta o conhecimento que o robô tem sobre seu ambiente, assim como sobre os estados internos do mesmo.

O uso de modelos probabilísticos como alternativa para modelar estas informações faz com que o grau de incerteza presente nas tarefas de medição e estimação esteja presente em todo o processo relacionado com o esquema de controle do robô. É por isto que cada vez é maior o número de propostas em robótica móvel que possuem um enfoque probabilístico.

1.1.3 Estimação de parâmetros

Do enfoque probabilístico explicado acima, pode-se dizer que os modelos probabilísticos das medições (para citar uma aplicação específica) são distribuições de probabilidade que representam a tendência ou comportamento estatísticos destes dados. Estas distribuições estão definidas por parâmetros próprios a cada modelo que devem ser estimados. Um estimador é precisamente uma aproximação ao valor real para esses parâmetros, e deve ser escolhido para que cumpra com certas características, como o de ser eficiente, em outras palavras, um estimador de variância mínima [2].

Especificamente para as aplicações em robótica móvel, a estimação de parâmetros se vê aplicada em vários problemas, como a já mencionada estimação de parâmetros que definem o valor médio e a variância de dados reais, a estimação de parâmetros dos modelos que definem certas características do ambiente, ou a estimação dos parâmetros que definem o modelo cinemático do robô [3].

Seja qual for o caso, existem várias abordagens para enfrentar o problema de estimação de parâmetros, dependendo principalmente se o sistema pode ser representado por um modelo linear em parâmetros ou não. Para o primeiro caso, o estimador de Mínimos Quadrados é uma técnica clássica. A partir dele várias abordagens tem se derivado, como por exemplo, o estimador de Máxima Verossimilhança que representa um enfoque mais estatístico do problema de estimação. Este o caso também dos estimadores Bayesianos, onde os parâmetros são tratados como variáveis aleatórias que possuem uma distribuição conhecida *a priori*. Já para o caso não linear tem-se o estimador de Mínimos Quadrados Totais, os M-Estimadores, assim como métodos iterativos por séries de Taylor. Lennart Ljung [4] apresenta o desenvolvimento matemático do estimador de Mínimos Quadrados Linear e das abordagens derivadas. Já para o caso não linear se recomenda consultar os trabalhos de Van Huffel e Vandewalle [5] e de Huber [6].

1.2 MOTIVAÇÃO

1.2.1 Trabalhos relacionados feitos no GRACO

O Grupo de Automação de Controle (GRACO) tem dado contribuições nas áreas tanto de robótica de manipuladores como robótica móvel. Especificamente no ultimo caso, os trabalhos que foram desenvolvidos estão relacionados com navegação baseada em um sistema de visão [7], projeto de uma arquitetura para a tele-operação de robôs moveis [8], construção de mapas baseados em grades de ocupação [9] e estudo do sonar aplicado à robótica móvel [10]. Todos estes trabalhos fizeram uso da plataforma robótica Nomad XR4000 construída pela *Nomadics Technologies* em 1999 [11], a qual será apresentada em capítulos subsequentes.

1.2.2 Mapas geométricos a partir de dados provenientes de sensores de ultrassom

Com a evolução tecnológica dos diferentes tipos de sensores, as opções para projetar um sistema de sensoriamento preciso e confiável são variadas. Apesar deste crescimento tecnológico, sensores clássicos como os sensores de ultrassom ainda são uma opção muito viável na hora de resolver problemas relacionados com a construção de mapas. O sensor de ultrassom possui duas características que o fazem popular entre outros tipos de sensores, sua simplicidade na resposta (o que facilita o processamento da mesma) e seu baixo custo comercial, o que promove o projeto e construção de plataforma robóticas mais acessíveis. Apesar da fama (pouco fundamentada em muitos casos) de sensor impreciso e obsoleto, uma quantidade importante de trabalhos reafirmam ainda o uso do sonar na solução do problemas como mapeamento e desvio de obstáculos. A tabela 1.1 apresenta um pequeno resumo de alguns destes trabalhos, que vão desde os estudos do comportamento do sinal de ultrassom até aplicações diretas em problemas de robótica.

Este trabalho propõe uma estratégia de mapeamento de ambientes internos baseado em extração de segmentos de linhas, utilizando as informações sobre o ambiente fornecidas por um sistema sensorial conformado por um conjunto de sensores de ultrassom, além de dados de odometria dados pela mesma plataforma robótica. A validação desta estratégia é feita por meio de simulações e testes experimentais no robô Nomad. É importante ressaltar que a estratégia só precisará de informação *a priori* correspondente às coordenadas iniciais da posição do robô ao início da trajetória de exploração do ambiente, e a partir disto e do conjunto de medições, será construída uma representação global do ambiente interno estudado.

1.3 DEFINIÇÃO DO PROBLEMA

O problema consiste em construir um mapa 2D de um ambiente interno com apenas as informações provenientes de um arranjo de sensores de ultrassom e do módulo de odometria do robô. O ambiente explorado é completamente desconhecido pelo robô, quer dizer, não é fornecida informação *a priori* sobre as características do ambiente, nem do número de objetos ou estruturas presentes nele.

Para coletar informação do mundo real, é executada uma trajetória de exploração onde o módulo de sensoriamento é ativado para adquirir as medições de distância dos objetos detectados. Adicionalmente, a informação de posição proveniente do módulo de odometria, responsável pela estimação da postura relativa do robô, será utilizada para ter uma estimativa da localização do robô durante a tarefa de exploração. Por fim, estas informações são pré-processadas e analisadas mediante um algoritmo de mapeamento baseado em extração de primitivas geométricas ou *features* e é construído um mapa do ambiente visto pelo robô, junto com informação sobre sua localização global dentro dele.

Tabela 1.1: Principais Trabalhos baseados no uso do sonar

| Referência | Autor/es | Descrição do Trabalho |
|------------|-------------------------|--|
| [12] | Kuc e Siegel | É simulado o comportamento de um sistema de navegação baseado em sensores de ultrassom. São desenvolvidos modelos das respostas do sensor a estruturas como paredes, cantos e bordas, as quais são funções do tamanho do transdutor, localização e orientação do mesmo. Pode ser considerado como um dos estudos mais completos do sonar feito na época e ainda é considerado como uma referência para o entendimento deste sensor. |
| [13] | Crowley | Constrói um mapa de um ambiente interno a partir de segmentos de linhas usando medidas de 24 sensores de ultrassom. O modelo utilizado para o sensor considera as incertezas próprias do transdutor assim como da posição da plataforma robótica utilizada. |
| [14] | Leonard e Durrant-Whyte | Propõe uma forma de modelar os dados reais que provêm das leituras dos sonares em ambientes internos como regiões de profundidade constante ou RCD por suas siglas em inglês (<i>Regions of Constant Depth</i>), que correspondem a arcos em coordenadas cartesianas. Estas regiões estão compostas por um conjunto de retornos ou ecos com diferenças de alcance menor do que um limiar que representa a diferença entre duas leituras consecutivas. São apresentados modelos de resposta para diferentes estruturas. |
| [15] | Lee <i>et. al</i> | A medição de cada sensor é representado como um arco. A comparação entre dados é feita em duplas, e dependendo da forma como os dois arcos são interceptados, é identificada a estrutura à qual pertencem. A partir destas associações é construído um mapa do ambiente. |
| [16] | Wang <i>et. al</i> | Apresenta um modelo chamado pelos autores de “modelo de associação de três medidas” utilizado para fusionar as medidas provenientes dos sensores de ultrassom, associando-as com linhas ou pontos. Os parâmetros destas estruturas são estimados por meio do método dos Mínimos Quadrados. O modelo da medição usa o conceito de regiões de profundidade constante. |

1.4 OBJETIVOS

1.4.1 Objetivo Geral

Projetar uma estratégia de mapeamento para ambientes internos baseada em extração de segmentos de linhas usando sensores de ultrassom e dados de odometria. Esta técnica proposta não precisa de informação *a priori* sobre a natureza das características a serem detectadas, nem do número de estruturas presentes no ambiente.

1.4.2 Objetivos específicos

1. Estudar o funcionamento do sensor de ultrassom em sua aplicação na tarefa de exploração de ambientes internos.
2. Implementar técnicas probabilísticas para modelar as incertezas das medições provenientes do módulo de sensoriamento e de odometria, como também para a propagação destes erros entre referências locais e globais.
3. Projetar uma estratégia completa de mapeamento baseada na extração de segmentos de linha com o objetivo de realizar um mapa geométrico em 2D.
4. Validar o desempenho da estratégia por meio de ambientes simulados e de ambientes reais usando a plataforma robótica Nomad XR4000.

1.5 APRESENTAÇÃO DO DOCUMENTO

No capítulo 2 é apresentada a revisão bibliográfica em torno ao problema de percepção e mapeamento. Especificamente, é explicado o funcionamento dos sensores proprioceptivos e exteroceptivos utilizados. Na sequência é definido o problema de mapeamento, apresentando as duas abordagens mais utilizadas, assim como os algoritmos específicos do mapeamento baseado em extração de *features* ou mapas geométricos. O capítulo 3 refere-se ao desenvolvimento da estratégia completa de mapeamento, onde é explicado detalhadamente o desenvolvimento matemático de todas as etapas constitutivas da abordagem proposta, sendo estas: a modelagem das medidas, o pré-processamento das mesmas, a implementação do algoritmo de extração de *features*, especificamente segmentos de linhas, e a combinação das *features* ou *merging*. Finalmente, no capítulo 4 têm-se os principais resultados obtidos na simulação e no teste experimental feito com a plataforma robótica Nomad. Para encerrar, no capítulo 5 são apresentadas as conclusões sobre este trabalho e propostas para trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

Um robô móvel pode ser considerado como um sistema constituído por diferentes etapas que permitem o uso deste tipo plataforma em diferentes tarefas. Siegwart e Nourbakhsh [17] apresentam um esquema geral deste sistema, onde é possível diferenciar claramente cada uma destas etapas (figura 2.1). É preciso ressaltar que não existe uma arquitetura estandardizada que defina um sistema robótico móvel, mas a representação apresentada por [17] é uma generalização das etapas comuns na maioria destes robôs.

A primeira etapa está constituída pela percepção do mundo real, feita através do sistema sensorial do robô. Na sequência, essas informações serão analisadas com o objetivo de detectar características importantes, construindo uma representação local do ambiente. O mapa local constitui a entrada para a seguinte etapa, a localização e construção de um mapa global, onde são estabelecidas as posições de robô e das estruturas detectadas com respeito a uma referência global. Feito o mapa final, segue o planejamento da rota que será executada pelo robô para atingir seu objetivo final. Por fim, na etapa de controle de movimento, esta rota é executada por meio de comandos passados aos atuadores, o que se traduz em movimento da plataforma robótica.

Este capítulo apresenta a revisão bibliográfica feita da teoria ao redor dos problemas de percepção e mapeamento. São apresentados conceitos básicos, que serão utilizados no resto do desenvolvimento do trabalho. Também são mostrados alguns trabalhos relacionados, ressaltando cada uma de suas contribuições.

2.1 PERCEPÇÃO

A percepção é a captura das informações do ambiente onde está inserido o robô. Esta captura é feita por meio de um sistema sensorial que pode estar conformado por um ou vários tipos de sensores que podem medir diferentes características do ambiente. Existe uma divisão clara entre os diferentes tipos de sensores segundo as características que eles possam medir [17]:

- *Sensores proprioceptivos*: medem valores internos do sistema robótico, por exemplo, a rotação de um motor, os ângulos das juntas de um braço robótico ou estado das baterias. Dentro desta classificação encontram-se os codificadores ópticos e os sensores inerciais como os acelerômetros e giroscópios.
- *Sensores exteroceptivos*: são os encarregados de adquirir as informações do entorno,

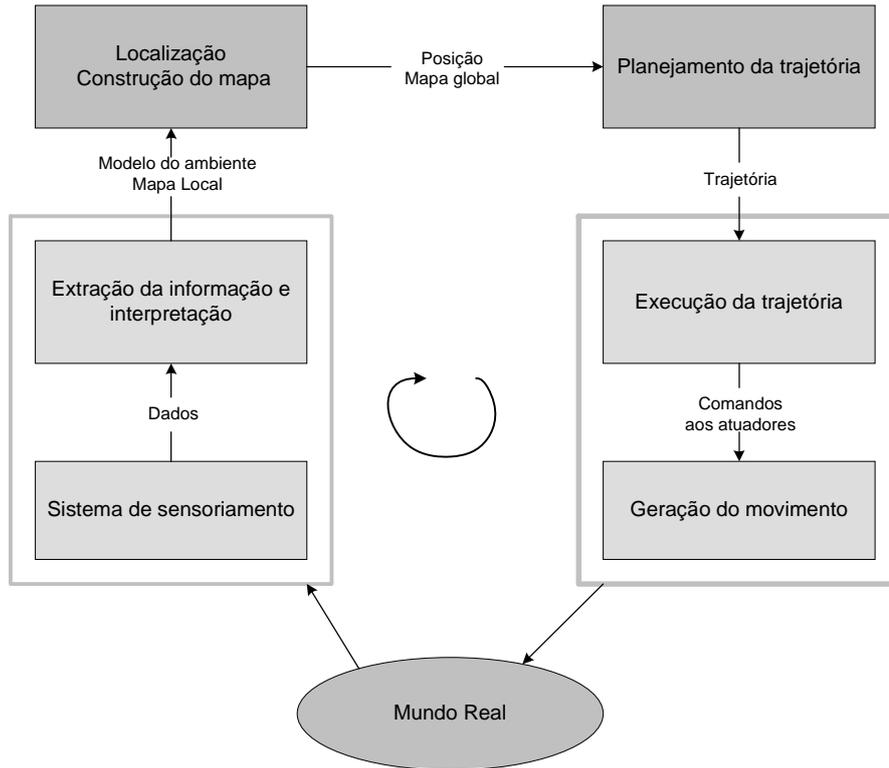


Figura 2.1: Esquemático geral de um sistema robótico móvel (Adaptado de [17]).

como medições de distância entre o robô e objetos, detecção de certos padrões ou características no ambiente. Estas leituras são usadas em diversas tarefas como seguimento de trajetórias, identificação de objetos, ou execução de algum tipo de ação predefinida por parte do robô.

Existe uma importante quantidade de sensores exteroceptivos que se diferenciam segundo seu princípio de funcionamento. A tabela 2.1 apresenta os mais utilizados classificados segundo a principal característica que os definem. Por mencionar alguns exemplos de aplicação destes sensores em robótica móvel, em [18] é desenvolvido um esquema de controle para a navegação autônoma, a partir do processamento de imagens de ambientes não-estruturados capturadas por uma câmera com um mecanismo de *pant-tilt*. Em [19] é construído um mapa baseado em grades de ocupação (*occupancy grid based-map*) usando informações fornecidas por um grupo de sensores de ultrassom. Por outro lado, alguns trabalhos fusionam dados provenientes de dois ou mais tipos de sensores. Em [20] são utilizados um sensor laser e um sistema de visão monocular para desenvolver uma estratégia de localização baseada em *features*. Já em [21] é construído um mapa local fusionando medidas de sensores infravermelhos e ultrassom por meio da implementação de uma rede neural artificial.

Tabela 2.1: Principais Sensores Exteroceptivos (Adaptado de [17])

| Classificação | Sensor |
|------------------------------|---|
| Sensores Táteis | <i>Switches</i> de contato, Barreiras ópticas, Sensores de proximidade sem contato |
| Sensores de Orientação | Bússola, Inclinômetro |
| Balizas localizadas em terra | GPS, Balizas RF, Balizas ativas de ultrassom, Balizas refletivas |
| Alcance ativo | Sensores refletivos, sensores de ultrassom, radar a laser, triangulação óptica, luz estruturada |
| Sensores de visão | Câmera CCD/CMOS |

2.2 SISTEMA SENSORIAL DA PLATAFORMA NOMAD XR4000

Como já foi mencionado no capítulo anterior, para a validação experimental da estratégia desenvolvida neste trabalho será usada a plataforma robótica Nomad XR4000, mostrada na figura 2.2. Este robô é um sistema de locomoção omnidirecional composto por quatro rodas orientáveis não centradas (conhecidas também como rodas *caster*), constituindo um sistema holonômico de três graus de liberdade (x, y, θ) de baixa complexidade mecânica e boa confiabilidade. Em relação ao sistema sensorial, este robô possui os seguintes sensores:

- Sensores Táteis. Também denominados de sensores de colisão, são sensores que retornam informações sobre o contato físico com objetos no ambiente. Os sensores táteis instalados no Nomad possuem dois níveis de sensibilidade, para contatos fracos e fortes.
- Sensores de ultrassom. Estão agrupados em dois anéis contendo 24 sonares. Estes provêm ao robô de informações sobre objetos distantes, na faixa dos 150 a 6500 mm de distância.
- Sensores infravermelhos: igual que para o caso dos sonares, a configuração destes sensores é de dois anéis de 24 sensores cada. Estes sensores são de curto alcance e portanto são usados para evitar colisões com objetos muito próximos ao robô.
- Sensor de visão. Composto por uma câmera de vídeo colorida padrão NTSC, auxiliado por uma unidade *pan-tilt* para sua movimentação.
- Módulo de odometria. Embora fisicamente não seja um sensor (mas que pode ser catalogado como tal), este módulo tem a tarefa de estimar a posição do robô com respeito a um sistema de referência global, isto com ajuda de sensores proprioceptivos instalados no sistema de locomoção do robô, especificamente codificadores.

Neste trabalho especificamente serão utilizadas as informações provenientes do módulo de odometria e dos sensores de ultrassom, portanto o funcionamento detalhado de cada um deles serão explicado a seguir.



Figura 2.2: Plataforma Nomad XR4000.

2.2.1 Módulo de Odometria

A estimação da posição do robô, isto é, o vetor $[x\ y\ \theta]^T$ é calculada por meio da medição das velocidades que atingem cada um dos eixos do robô, e sua posterior integração em um intervalo determinado de tempo. As informações de velocidades de cada eixo são obtidas por meio de uso de codificadores instalados em cada um dos motores responsáveis pelo movimento das rodas.

2.2.1.1 Codificador Incremental

Como foi mencionado no começo desta seção, os codificadores fazem parte dos sensores proprioceptivos. Especificamente, este tipo de codificadores pode dividir-se em codificadores rotatórios e codificadores lineares. Os primeiros são usados para medir posição angular e direção de um motor, enquanto os codificadores lineares medem posição linear e direção. Ambos os tipos de codificadores são capazes de calcular a velocidade de movimento [22].

Este sensor proprioceptivo é composto por um disco feito de metal ou de vidro e possui um padrão de setores opacos e transparentes conhecido como trilha de código (*code track*), e pode ser visto na figura (2.3). Além disso, uma fonte de luz é colocada em um dos dois lados do disco e um fotosensor no outro. A medida em que o disco gira, a trilha de código interrompe o passo de luz de um lado ao outro, portanto a tendência que segue o sinal que chega ao fotodetector é de um sinal digital. O número de setores opacos e transparentes corresponde a ciclos, e um número determinado destes corresponde a uma rotação do eixo de motor. O número de ciclos por rotação (em inglês CPR, por *Cycles Per Revolution*) determina a resolução do codificador.

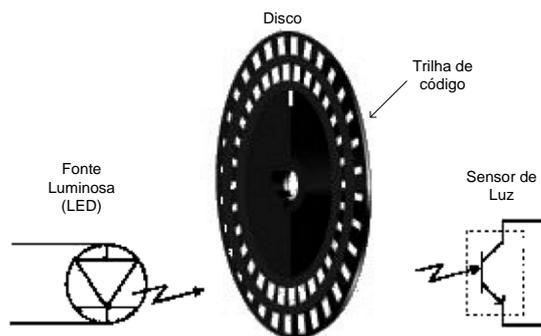


Figura 2.3: Codificador Incremental (Adaptado de [23]).

2.2.1.2 Erro de Odometria

Justamente pelo processo de integração associado com o cálculo da posição do robô, os erros provenientes deste módulo são integrados no tempo, sendo cada vez maiores e fazendo com que a estimativa posição se afaste cada vez mais de seu valor real. É possível enumerar algumas fontes de erros relacionadas com as condições do ambiente assim como do próprio processo de estimação feito pelo módulo de odometria [17]:

- Resolução limitada durante o processo de integração. Isto está associado com os intervalos de tempo usados para a integração das velocidades dos eixos. Também nesta categoria se encontra a resolução do codificador.
- Desalinhamento das rodas.
- Diferenças nas dimensões das rodas.
- Variação do ponto de contato da roda.
- Irregularidades na superfície de contato (superfície não plana e escorregadia).
- Erro de Alcance. Este erro está associado com o cálculo da distância de deslocamento do robô, a partir da integração dos deslocamentos individuais de cada roda.
- Erro de giro. É similar que o erro de alcance, mas para o caso do cálculo do giro da plataforma a partir das diferenças do giro de cada uma das rodas.
- Erro na direção. Diferenças nos erros de cada roda podem causar um erro na orientação θ do robô.

Uma alternativa para neutralizar os efeitos destes erros é mediante a utilização de informação adicional proveniente de outros tipos de sensores proprioceptivos. Um exemplo deste tipo de solução podem ser encontrado no trabalho de Von der Hardt et al. [23], onde foram

utilizados um giroscópio junto com uma bússola magnética para prover o sistema com informação redundante sobre a orientação do robô e assim obter uma melhor estimativa de sua posição.

2.2.2 Sensor de Ultrassom

Este sensor pertence à família dos sensores de alcance ativo (termo que vem do inglês *active ranging*). Suas primeiras aplicações foram na área marítima sendo utilizado como um instrumento auxiliar de navegação, de onde provém o outro nome com que também é conhecido, o de sonar, acrônimo das palavras em inglês *Sound Navigation and Ranging*. Hoje em dia é também usado no estudo e pesquisa dos oceanos (determinação de profundidades ou de depressões) e na pesca, para a localização de cardumes [24].

Já na área da robótica móvel, este sensor é muito utilizado para o desvio de obstáculos e para a construção de mapas a partir da exploração de ambientes. A razão de sua popularidade é seu baixo custo e a simplicidade da resposta que fornece, a qual consiste na distância que existe entre o sensor e o obstáculo presente no alcance deste.

Especificamente o sensor de ultrassom é um sensor de alcance ativo em tempo de vôo, quer dizer, usa a velocidade de propagação do sinal de ultrassom para calcular a distância entre o sensor e o objeto atingido. Portanto a distância d entre o sensor e a fonte de reflexão (objeto atingido) pode ser calculada usando o tempo de vôo e a velocidade de propagação do sinal de som, assim:

$$d = \frac{c \cdot t}{2} \quad (2.1)$$

Tipicamente, a uma pressão ambiental normal e a uma temperatura aproximadamente igual aos 20°C, a velocidade de propagação do som c é aproximadamente igual a 343 m/s .

2.2.2.1 Funcionamento

Na figura 2.4 apresenta-se um esquema com o funcionamento do sensor de ultrassom. Um transdutor acústico, que atua como transmissor e receptor, emite um número determinado de ciclos de sinal de ultrassom, e a partir deste momento conta-se o tempo transcorrido entre a transmissão e a posterior recepção da onda.

Depois que o pacote de ciclos é transmitido, o transdutor permanece um período de tempo sem transmitir sinal nenhum, o que é conhecido como *blanking time*. Uma vez o eco do sinal transmitido é recebido pelo sensor (que também funciona como receptor), este alimenta um amplificador operacional de ganho variável; ganho que varia na medida em que o eco demora mais ou menos tempo em ser detectado pelo sensor, isto com a finalidade de compensar possíveis atenuações ou dispersões da onda. Finalmente, a distância é calculada uma vez que o valor do limiar do circuito é superado, momento no qual a conta do tempo transcorrido entre transmissão e recepção finaliza.

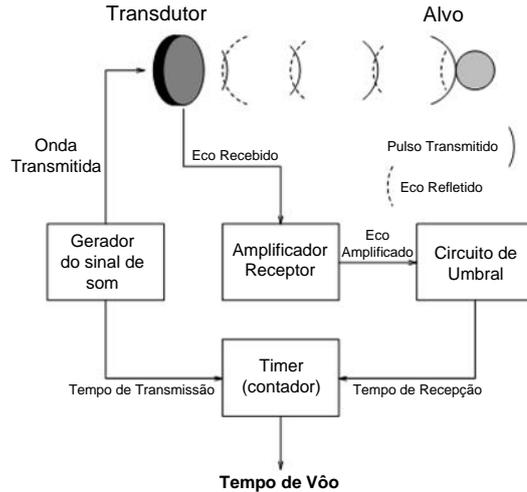


Figura 2.4: Mecanismo de funcionamento do sensor de ultrassom (Adaptado de [14]).

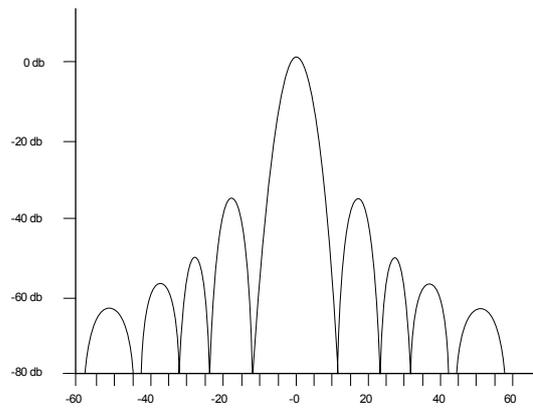


Figura 2.5: Padrão de propagação do sinal de ultrassom [11].

Uma característica importante que define o desempenho do sensor é o padrão de radiação do transdutor, apresentado na figura 2.5. Como pode ser observado, o transdutor não emite a energia de forma homogênea, em vez disso, o padrão de radiação está conformado por lóbulos que vão decrescendo em intensidade na medida em que o sinal de ultrassom transmitido se afasta do eixo principal do sensor. Este padrão de radiação define um “cone sonoro” dentro da qual se encontra a medição de distância.

2.2.2.2 Erros Típicos do Sensor

O sonar, como todos os sensores, possui algumas desvantagens que comprometem seu desempenho. Tipicamente, um sistema de sensoriamento baseado em sensores de ultrassom tende a sofrer de imprecisões nas leituras de distância devido, principalmente, a erros relacionados com o circuito do transdutor, ou devido as características dos alvos a serem atingidos [14]. A seguir, enumeram-se os dois tipos de erros.

- i. Erros causados por características do circuito do sensor
 - O cálculo do tempo de transmissão-recepção assume que o começo do pulso transmi-

tido faz parte do eco que posteriormente será recebido, e que deve superar o limiar do circuito. Se este não for o caso, o erro de distância poderia ser de até 23 cm [11].

- O amplificador de ganho variável faz uma aproximação de só 16 passos da curva exponencial ideal que compensa perdas por causa de atenuação e de múltiplas reflexões. Isto pode comprometer a magnitude do sinal amplificado que posteriormente entrará ao circuito de umbral e que definirá o tempo que transcorreu entre transmissão e recepção.
- O mecanismo de funcionamento do circuito de limiar envolve o uso de um capacitor de carga, que se vê fortemente afetado pela intensidade do eco recebido. No caso de reflexões de onda fortes, só se precisa de três ciclos do sinal para carregar o capacitor. Entretanto, para reflexões fracas, a carga de capacitor pode demorar um tempo considerável, o que pode representar, em alguns casos, um cálculo de distância errado.
- O padrão de radiação do sensor é irregular (como já foi mencionado acima), o que significa que os objetos detectados sejam atingidos pelo sinal com uma energia que depende da inclinação da onda sobre a superfície atingida.

Especificamente, para orientações perpendiculares ao alvo (isto significa um desvio de zero grau do eixo do sensor), o sinal que incide sobre o objeto provém do lóbulo principal, e portanto, a medida de distância será mais exata. Já para o caso de sinais emitidos nas regiões definidas pelos demais lóbulos, a precisão na medida distância vai se degradando conforme estes sejam mais pequenos.

ii. Erros causados por características do alvo

- *Objetos refratores*: são aqueles que possuem dimensões menores do que a longitude de onda do sinal de ultrassom. Embora estes objetos sejam pouco comuns em ambientes internos, algumas paredes com pequenas asperezas, por exemplo, poderiam se comportar como objetos refratores.
- *Objetos refletores*: possuem dimensões maiores do que a longitude de onda do sinal e são os alvos mais comuns. Tipicamente em um ambiente interno correspondem a superfícies metálicas suaves, paredes, portas, entre outros. Atuam com uma espécie de espelhos da onda de ultrassom.

Um fenômeno que prejudica a atuação do sonar ocorre quando combinado com as características dos objetos refletores, o sinal incide sobre o objeto com um ângulo diferente aos 90° . Isto em alguns casos pode ocasionar o sinal de ultrassom refletir em um ou em vários locais antes de ser detectado pelo receptor, dando a impressão falsa da localização do alvo que foi atingido inicialmente. Este fenômeno é conhecido como *reflexão especular*. Na figura 2.6 é apresentada uma representação deste fenômeno.

Outra característica desfavorável do sensor que não pode ser catalogada como um erro, é a ausência de informação sobre a orientação do objeto detectado, informação importante para definir completamente sua localização com respeito à posição do sensor.

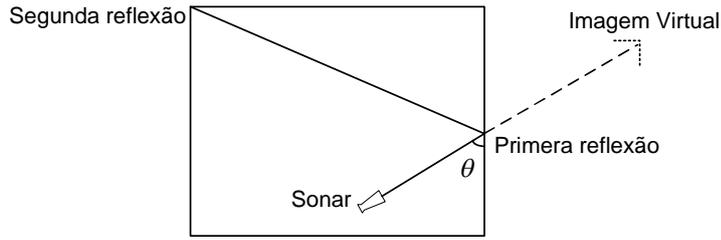


Figura 2.6: Fenômenos de reflexão especular (Adaptado de [10]).

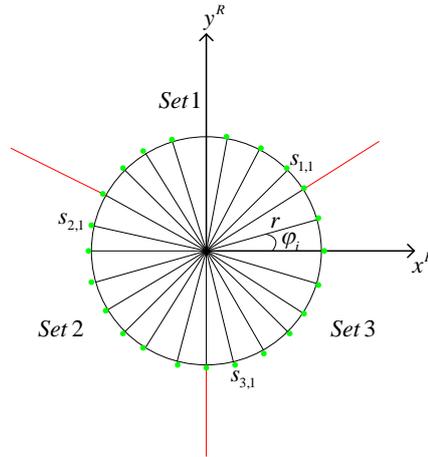


Figura 2.7: Configuração dos sonares no robô Nomad XR4000 (vista superior). Os pontos verdes representam cada um dos sensores que compõem um anel.

Mesmo se a susceptibilidade às reflexões especulares e a incerteza angular representam aspectos negativos do sensor de ultrassom, isto não representa um impedimento para que este possa ser utilizado como sensor para resolver tarefas de mapeamento.

2.2.2.3 Sensor Polaroid 6500

Neste trabalho os sonares instalados na plataforma robóticos correspondem aos transdutores de ultrassom *Polaroid 6500* [25],o qual transmite 16 ciclos de onda quadrada com frequência de 49.4 kHz. Cada transdutor possui um cone de radiação ou feixe sonoro de 25° graus e tem um alcance que vá desde os 15 até os 6500 mm.

Na figura (2.7) é apresentada a configuração dos sensores na plataforma robótica. O espaçamento angular entre sensores é de 15°, o que significa que os 24 sonares de cada anel cobrem os 360° correspondentes ao campo visual do robô. A posição e a orientação de cada sensor com respeito à origem do sistema coordenado do robô estão definidas pelo raio r da circunferência que descreve o anel e o ângulo φ_i entre os eixos x dos sistemas coordenados do robô e de cada sensor, respectivamente. Para um melhor análise das informações provenientes dos sensores, estes têm sido reunidos em três grupos de acordo com sua disposição espacial.

2.3 MAPEAMENTO

Em um conceito geral, um mapa nada mais é que a representação de um conjunto de características do ambiente onde a plataforma robótica está inserida. Embora as características a serem apresentadas em cada mapa variem segundo o tipo de representação escolhida, o mapa pode ser expressado como [1]:

$$M = \{m_1, m_2, \dots, m_n | n = 1, \dots, N\} \quad (2.2)$$

onde N representa o número total de características (objetos ou estruturas) detectadas, e cada m_n é a representação específica de cada característica do ambiente.

A tarefa de mapeamento está fortemente relacionada com a capacidade do robô móvel de executar uma navegação autônoma através do ambiente previamente explorado, e é por isto que os avanços nesta área têm sido orientados a proporcionar mapas cada vez mais precisos em termos da identificação e representação correta dos objetos e estruturas presentes no ambiente, assim como sua localização dentro da representação construída.

Para garantir um bom resultado nesta tarefa, existem algumas variáveis que devem ser consideradas na hora estruturar uma estratégia de construção de mapa que se ajuste aos diferentes tipos de ambientes a serem explorados pelo robô. A seguir são apresentadas as variáveis mais destacadas por Thrun *et. al* em [1].

- *Grau de detalhe.* Na medida em que se precisa de maior detalhe no mapa a ser construído, a tarefa resulta mais complexa devido ao aumento de informação a ser analisada, aumentando também o tempo de processamento do algoritmo usado, assim como o tamanho do arquivo a ser carregado ao robô.
- *Dimensões do ambiente.* A dificuldade de adquirir informações do ambiente incrementa-se na medida em que o tamanho deste seja maior ao campo visual¹ do robô.
- *Ruído nos sensores e nos atuadores.* As leituras provenientes do sistema sensorial do robô estão contaminadas por ruído próprio do dispositivo de medição, assim como ruído do ambiente. Os atuadores também são sensíveis a ruído, afetando, por exemplo, a posição real do robô dentro do ambiente.
- *Ambiguidade Sensorial².* É muito comum a presença de locais ambíguos em um ambiente estruturado. Isto representa problemas na hora de estabelecer correspondências entre os diferentes locais percorridos pelo robô em diferentes instantes de tempo.
- *Ciclos.* Se o robô faz diferentes percursos no ambiente, o problema torna-se muito mais complexo na hora de estabelecer correspondências entre as diferentes estruturas detectadas em cada um dos percorridos.

¹O campo visual refere-se ao alcance do sistema sensorial utilizado pelo robô.

²São aquelas medições provenientes de diferentes fontes que produzem a mesma informação. Por exemplo, as medidas de distância de duas paredes equidistantes que estão localizadas a ambos os lados do robô, o que dá a falsa impressão de que se trata de mesma parede.

onde m é o mapa, $z_{1:t}$ são as medições até o instante t e $x_{1:t}$ são todas as posições do robô durante a exploração do ambiente. A expressão (2.3) pode ser vista como a probabilidade de ocupação dado o conjunto de medidas z e as posições x , o que coloca o problema dentro do contexto Bayesiano³, e a partir de sua formulação é possível calcular o valor de ocupação para cada grade.

Este método é particularmente preferido quando o robô está equipado com sensores de alcance ativo (como sonares e radar laser) porque os valores de alcance destes sensores, combinados com a informação da posição absoluta do robô, podem ser usados diretamente na atualização da probabilidade de ocupação de cada célula. Para isto, cada célula possui um contador, onde o zero significa que a célula não foi atingida por nenhum sinal. A medida em que o número de detecções aumenta, a probabilidade de ocupação aumenta, e depois de superar certo valor, a célula é considerada como parte de um obstáculo. Outra característica desejável deste tipo de representação é a capacidade de filtrar leituras espúrias provenientes de obstáculos dinâmicos que não pertencem ao ambiente, como por exemplo, pessoas movimentando-se, o que lhe reconhece um grau de robustez superior quando comparado com outras representações.

O desempenho deste tipo de representação depende principalmente do tamanho da grade. Uma célula muito pequena utilizada para representar um ambiente relativamente grande, ocasiona um uso de memória importante. O caso contrário (células com um tamanho relativamente grande), pode-se apresentar distorção da realidade, devido a que áreas de espaço livre podem ser vistas como ocupadas no mapa resultante. Quando existe um bom compromisso entre o tamanho da célula e o custo computacional associado ao mapeamento, esta técnica pode ser empregada de maneira satisfatória.

Aplicações específicas utilizando este tipo de representação podem ser encontradas no trabalho de Matthies e Elfes [31], onde é feita a fusão de leituras de sensores de ultrassom e de um sistema de visão estéreo para construir um mapa em 2D. Especificamente neste trabalho, são criados mapas separados a partir das leituras de cada sensor, os quais são depois fusionados e como resultado final é criado um mapa global do ambiente. Outros trabalhos mais recentes são, o de Kim e Lee [32], onde é construído um mapa probabilístico a partir da fusão hierárquica de dados de sensores de ultrassom e de infravermelho; e o de Nikos *et al.* [33] onde é apresentada uma técnica para construir mapas dinâmicos baseados em grades de ocupação, chamada algoritmo de grade de ocupação temporal (em inglês, *Temporal Occupancy Grid*). Os resultados destes trabalhos são apresentados na figura 2.9.

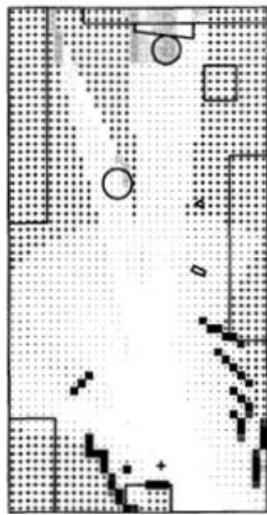
2.3.2 Mapeamento baseado em extração de *features*

Antes de aprofundar na explicação desta técnica é preciso saber o que se entende por *feature*. Segundo Siegwart [17], uma *feature* é uma estrutura reconhecível presente em um determinado ambiente que pode ser extraída a partir de medições fornecidas por um sensor,

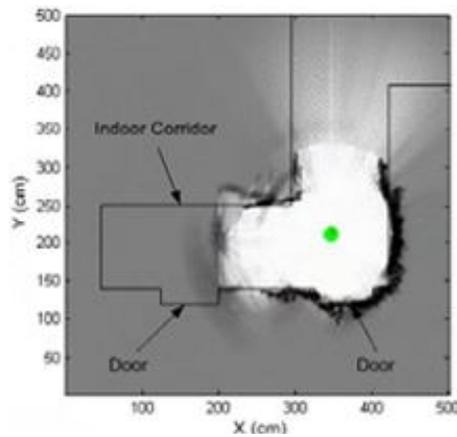
³Para maiores detalhes da formulação do filtro de Bayes para o caso de grades de ocupação, recomenda-se consultar o trabalho de Thrun apresentado em [1] e em [26].

e geralmente se ajusta a um modelo matemático. As *features* são abstrações do conjunto de dados não processados (em inglês, *raw data*), o que representa uma redução considerável no volume de informação disponível para construir uma representação do ambiente. Também são conhecidas como primitivas geométricas já que geralmente as estruturas mais extraídas em ambientes internos correspondem a pontos, segmentos de linhas, arcos, círculos, entre outros polígonos.

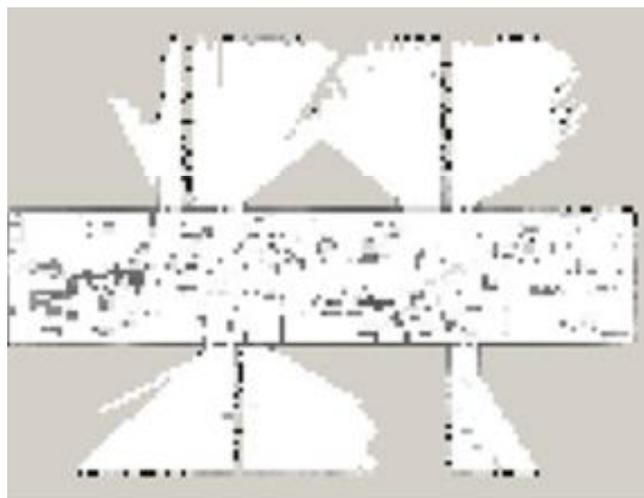
Por conseguinte, os mapas baseados em extração *features* correspondem a representações compostas por formas geométricas que descrevem estruturas presentes no ambiente, tais como paredes, portas, janelas, colunas entre outras. Este tipo de representação permite uma descrição mais compacta e robusta do ambiente explorado, por isto é muito utilizada nas tarefas



(a) Mapa gerado por Mathies e Elfes



(b) Mapa gerado por Kim e Lee



(c) Mapa gerado por Nikos *et al.*

Figura 2.9: Exemplos de mapas baseados em grades de ocupação.

de mapeamento e localização simultânea. Um dos primeiros trabalhos onde é explorada este tipo de técnica se apresenta no artigo feito por Chatila e Laumond [34], que propuseram na época a representação do um ambiente a partir de linhas.

Além da vantagem de ser uma representação mais compacta se comparada com o mapa baseado em grades de ocupação, no trabalho de Thrun [26] são especificadas outras vantagens que colocam o desempenho dos mapas baseado em *features* acima dos mapas probabilísticos, a seguir:

- Os mapas baseados em extração de *features* podem resultar mais precisos, isto condicionado na suposição em que as primitivas escolhidas sejam as adequadas para representar os objetos reais do ambiente.
- Na descrição de ambientes dinâmicos, esta representação apresenta um melhor desempenho, devido ao fato de que os objetos (cujas posições mudam no tempo) descritos mediante *features*, podem ser identificados e diferenciados durante todo o processo de mapeamento.
- As *features* se assemelham mais à percepção humana do entorno, o que facilita a interação homem-robô.

Apesar do mencionado acima, os mapas baseados em extração de *features* sofrem de uma importante desvantagem: suas representações muitas vezes se limitam a descrever ambientes que possam ser expressos mediante formas geométricas simples, o que pode dificultar seu uso em ambientes externos pouco estruturados.

2.3.2.1 Extração de *features*

A extração de *features* a partir de dados não processados está ligada à natureza destes dados, quer dizer, ao tipo de informação, assim como também ao tipo de *feature* a ser detectada. Na literatura existe um número amplo de algoritmos de extração de *features*, por exemplo, para sensores de alcance ativo, é comum identificar linhas e cantos, que correspondem a paredes e cantos reais, entre outros objetos que possam ser definidos com estas primitivas. Já os sistemas de visão computacional oferecem uma quantidade mais ampla de *features*, entre elas estão bordas, cantos e objetos de aparências distintas.

A extração de *features* é o passo seguinte no tratamento prévio dos dados provenientes dos sensores, e é a etapa que antecede a interpretação da cena, onde as primitivas detectadas tomam um significado dentro do mapa com o qual o robô poderá interagir com seu entorno.

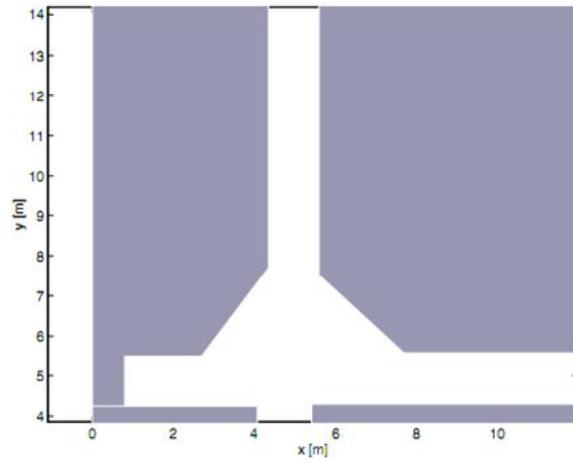
2.3.2.2 Extração de Segmentos de linhas

Em um ambiente interno, uma grande proporção das estruturas presentes pode ser representada a partir de segmentos de linhas, é por isto que esta primitiva geométrica é uma das mais utilizadas nos mapas baseados em *features*.

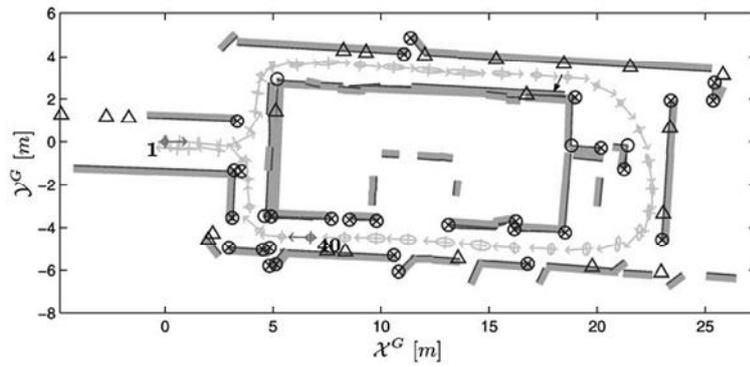
Existe uma quantidade importante de trabalhos de mapeamento onde as representações construídas estão compostas por segmentos de linhas. Por exemplo, no trabalho de Siadat *et al.* [35] é construído um mapa constituído por polígonos formados a partir de segmentos de linhas extraídos do ambiente explorado. Em Borges e Aldon [36] são apresentados resultados experimentais da implementação de três importantes algoritmos de extração de linhas usando imagens de um sensor radar laser. No trabalho de Arras e Tomatis [37] são também extraídos segmentos de linhas a partir de fusão de imagens provenientes de uma câmara CCD e de medidas de um radar laser. Na figura (2.10) são apresentados alguns resultados destes trabalhos.

Existem vários algoritmos de extração de segmentos de linhas que são utilizados para mapear ambientes, embora também tenham sido utilizados inicialmente em outras áreas, como processamento de imagens. O trabalho apresentado por Nguyen *et. al* [38] faz um estudo dos seis algoritmos mais utilizados para extrair segmentos de linhas, avaliando o desempenho de cada um por meio de indicadores quantitativos que medem rapidez, precisão, simplicidade na implementação e correspondência com as estruturas reais. Retomando alguns dos resultados deste estudo, os algoritmos que se sobressaem por sua rapidez e bom desempenho em termos gerais são aqueles baseados nos algoritmos *Split-and-Merge*, *Regressão Linear* e *Incremental*. Estes métodos aproveitam as características sequenciais do conjunto de dados, quer dizer, preservam a ordem em que os dados foram adquiridos. A seguir é apresentado brevemente o funcionamento de cada um deles.

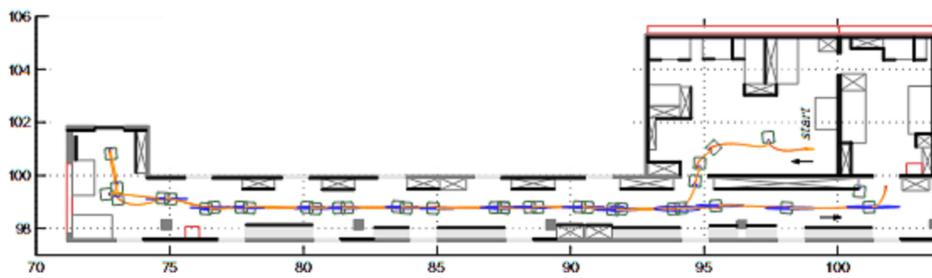
- *Split-and-Merge*. Este algoritmo teve seus origens na área de visão computacional, apresentado pela primeira vez no trabalho de Pavlidis and Horowitz [39] em 1974. O algoritmo básico funciona da seguinte maneira: um grupo inicial que abrange todos os dados é criado, e a partir deste é estimada uma linha que os representa que é armazenada numa lista de linhas \mathcal{L} . O ponto P com a maior distância linha-ponto d_P é detectado, e se este supera um determinado limiar, a linha inicial é dividida em duas linhas, onde o final de uma e o começa da outra está delimitado por P . Na sequência, a lista \mathcal{L} é atualizada. O processo inicia novamente analisando cada um dos pontos das linhas derivadas. Esta etapa, que corresponde ao *split*, finaliza quando todos os grupos são analisados. Então, a etapa de *merge* entra em funcionamento, combinando aquelas linhas que possuem características similares como, por exemplo, os parâmetros que as definem.
- *Regressão Linear*. Foi proposto por Arras e Siegwart [40]. Este algoritmo é inspirado na *transformada Hough*, convertendo o problema de extração de linhas em um problema de busca no domínio dos parâmetros da linha. O conjunto de dados é dividido em grupos de n_f pontos consecutivos, e para cada um é estimado o conjunto de parâmetros da linha, assim como a matriz de covariância. Depois disto, é utilizado o critério da distância de Mahalonobis para dividir a linhas estimadas em duas ou mais linhas, ou pelo contrário, combinar linhas adjacentes em uma só.
- *Incremental*. Este algoritmo também é conhecido como algoritmo de “seguimento de



(a) Resultado do mapa do trabalho do Siadat *et. al*



(b) Resultado do mapa do trabalho de Borges e Aldon



(c) Mapa construído no trabalho de Arras e Tomatis

Figura 2.10: Mapas baseado em extração de segmentos de linhas.

linhas ” (tradução do termo original em inglês, *Line tracking*), termo cunhado por Siadat *et. al* [35]. Como os demais algoritmos, a abordagem incremental é sequencial, quer dizer, as entradas são processadas segundo a ordem em que foram adquiridas. A primeira iteração do algoritmo inicia com os dois primeiros pontos do conjunto de dados. Assume-se que estes dois pontos pertencem ao mesmo segmento de linha, e a partir deles os parâmetros iniciais da linha são calculados. O seguinte ponto é analisado e é aceito como ponto pertencente à linha se a distância entre o ponto e a linha estimada não supera certo limiar. Se o ponto é aceito os parâmetros da linha são recalculados e o ponto seguinte é analisado. No caso de não pertencer, uma nova linha é criada a partir do ponto rejeitado. O algoritmo finaliza quando todos os pontos são analisados.

Comparando os algoritmos acima citados, o estudo de Nguyen *et. al* [38] finaliza com as seguintes conclusões: o algoritmo *Split-and-Merge* possui um bom compromisso entre complexidade na implementação e rapidez, por isso é recomendado para aplicações em tempo real assim como para resolver problemas de localização com um mapa *a priori*. Já em termos de correspondência com as estruturas reais, o algoritmo Incremental possui o melhor desempenho, pois apresenta o número menor de falsos positivos, o qual é muito importante em tarefas com as de localização e mapeamento simultâneo (ou SLAM, de *Simultaneous Localization and Mapping*).

RESUMO DO CAPÍTULO

Neste capítulo foram tratadas algumas definições importantes consideradas relevantes para o desenvolvimento deste projeto. Inicialmente, foram apresentados os sensores utilizados para extrair informações próprias do estado do robô, assim como aquelas relacionadas com percepção do ambiente. Foi detalhado o princípio de funcionamento de cada sensor, bem como as características de seus componentes mecânicos e eletrônicos. Também foram explicados os erros típicos sofridos por estes sensores e como eles podem afetar as leituras resultantes. Na sequência foi apresentado o problema de mapeamento, as principais abordagens para a construção de representações de ambientes internos, e por fim, as técnicas mais usadas para mapeamento baseado em extração de *features*.

Este capítulo foi uma apresentação geral dos conceitos ao redor das etapas de percepção e mapeamento em robótica móvel. A seguir serão aprofundadas algumas destas definições ao apresentar a estratégia de mapeamento desenvolvida para este trabalho.

3 ESTRATÉGIA DE MAPEAMENTO

O objetivo principal da tarefa de mapeamento pode ser resumido como a construção de uma representação coerente do mundo real explorado pelo robô móvel. Particularmente esta área tem uma considerável importância devido a que os resultados obtidos nesta etapa representam uma base de conhecimento inicial sobre ambiente, que será utilizada em tarefas subsequentes, como planejamento e execução de trajetórias ou navegação autônoma.

Como já foi mencionado na seção 2.3, é preciso considerar alguns aspectos importantes na hora de estruturar uma estratégia de mapeamento, aspectos que estão relacionados com o tipo de sensor usado, a precisão que se deseja no mapa resultante e algumas condições específicas do ambiente a ser representado. Neste capítulo será apresentada a estratégia do mapeamento desenvolvida neste trabalho.

Basicamente, esta está composta por três etapas (figura 3.1). A primeira etapa consiste em um pré-processamento dos dados, onde estes são transformados em uma representação que facilita sua posterior comparação. O seguinte processo é a extração das características, que é a estimação dos segmentos de linhas a partir dos dados pré-processados. As linhas detectadas, são posteriormente comparadas na última etapa, o processo de combinação ou *merging*, onde estas são agrupadas segundo um determinado critério e finalmente são estimadas as linhas definitivas que conformarão o mapa do ambiente.

A seguir será especificado o modelo de medição para o sensor de ultrassom assim com o modelo que define os segmentos de linhas. Na sequência, se apresentarão com mais detalhes cada uma das etapas que compõem a estratégia de mapeamento.

3.1 MODELO DE MEDIÇÃO

3.1.1 Antecedentes

O padrão de medida de um sensor de ultrassom é apresentado na figura 3.2. Este consiste em um cone (também conhecido como *feixe angular*) cuja abertura é dada pelo ângulo β . Dentro deste cone está situada a medida de distância dada pelo sensor, d_n , a qual possui uma orientação com respeito ao eixo principal $\beta = 0$ denominada ϕ . A medição de distância depende principalmente (entre outros aspectos já mencionados) da inclinação com que a onda sonora atinge o objeto detectado, assim como do tipo de estrutura detectada. O primeiro aspecto é amplamente explorado no trabalho de Barshan e Kuc [41] onde são diferenciadas

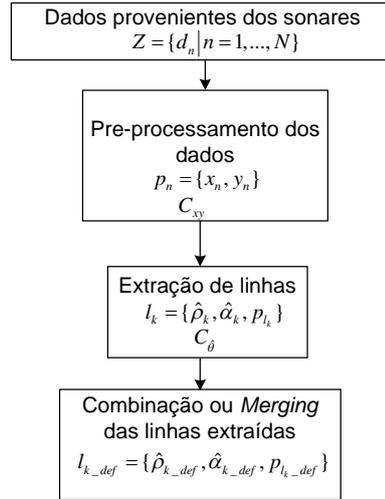


Figura 3.1: Estratégia de mapeamento implementada.

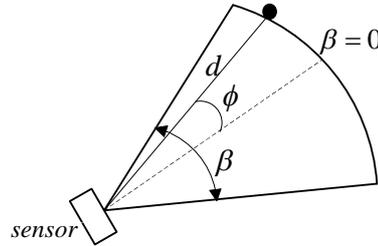


Figura 3.2: Padrão de medida do sensor de ultrassom.

as reflexões provenientes de planos e cantos por meio da análise das características físicas da onda sonora refletida. A partir dos resultados apresentados neste trabalho, os autores concluíram o seguinte:

- A amplitude do sinal refletido ou eco, assim como o valor de distância resultante, dependem do ângulo de incidência da onda, que por sua vez depende do ângulo de orientação do sensor.
- A amplitude do eco diminui na medida em que o ângulo de orientação desvia-se da direção normal de incidência do sensor.

A figura 3.3 apresenta um padrão de medida de um grupo de sonares com diferentes orientações que observam a mesma estrutura plana, extraído de [41]. A direção normal de incidência corresponde àquela com ângulo de orientação φ_n igual a zero. Aqueles sonares cujas orientações desviam-se desta normal terão uma onda refletida com menor amplitude e, portanto, com maior probabilidade de que a leitura de distância difira do valor real. Esta análise é muito útil para a modelagem da medição do sensor, explicada a seguir.

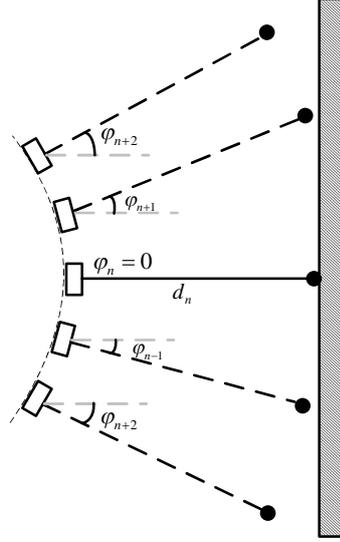


Figura 3.3: Padrão de medição de uma estrutura plana detectada por um grupo de sensores dispostos ao redor de uma circunferência segundo [41]. d_n é a distância medida e φ_n é o ângulo de orientação de cada sensor com respeito a sua direção normal de incidência, representada pelas linhas ponteadas de cor cinza.

3.1.2 Modelo escolhido

Considerando o comportamento dos sensores de ultrassom explicado acima, o modelo adotado para representar as medições provenientes deste sensor é o seguinte:

$$d_n = D_n + \varepsilon_s + \varepsilon_a, \quad \varepsilon_a \sim N(0, \sigma_d^2). \quad (3.1)$$

Neste modelo d_n representa a distância dada pelo sensor, D_n é o valor real para esta medição, ε_s é o erro sistemático da medida, que para nosso caso é o mesmo relacionado com a orientação de cada sensor com respeito à normal de incidência entre o sonar e o alvo; e o erro aleatório ε_a que se assume segue uma distribuição gaussiana de média zero variância σ_d^2 .

Os valores para os erros sistemáticos para cada sensor segundo sua localização, foram calculados a partir de testes feitos com o conjunto de sensores instalados no robô, conforme explicado no apêndice B. Já o valor para a incerteza na medição de distância (erro aleatório), σ_d^2 , este corresponde ao valor fornecido pelo *data sheet* do fabricante [25].

O modelo apresentado na equação (3.1) não inclui a orientação do objeto detectado, isto porque o modelo considera só a informação medida pelo sensor, e como já foi mencionado anteriormente, este valor não é fornecido diretamente pelo sonar. A estimação do valor de orientação, ϕ_n , pode ser calculada a partir da orientação do sensor (com referência ao sistema de coordenadas do robô onde está localizado) e da incerteza angular da onda refletida, assim:

$$\phi_n = \varphi_i + \delta_n, \quad \delta_n \sim N(0, \sigma_\phi^2), \quad (3.2)$$

em que φ_i é a orientação do i -ésimo sonar do anel e δ_n representa aquela incerteza angular da

n-ésima medida, que depende fortemente da abertura do feixe do sensor β , modelada como uma variável aleatória de distribuição normal de média zero e variância σ_φ^2 . O valor para este último parâmetro é calculado justamente usando o valor de β do sensor *Polaroid 6500*.

Desta maneira completa-se a informação (distância e orientação) do processo de medição. É preciso ressaltar que o modelo de medição corresponde àquele apresentado na equação (3.1): a expressão em (3.2) não faz parte de modelo de medição, mas representa uma informação complementar necessária para conhecer a orientação do objeto detectado pelos sensores que compõem os anéis.

Por fim, o modelo de medição dado em (3.1) é utilizado para fazer uma compensação de todas as leituras de cada um dos sonares usados durante o processo.

3.2 PRÉ-PROCESSAMENTO DOS DADOS

3.2.1 Transformação de coordenadas

Uma vez as medições provenientes dos sonares são compensadas segundo seu modelo, o seguinte passo é a análise destas. Da mesma forma que em outros sensores de alcance ativo, é possível considerar o dado do sonar como um ponto em coordenadas polares, que dizer, a medida pode ser representada como certo ponto $p_n = \{d_n, \phi_n\}$.

Lembrando que configuração do sistema de sensoriamento está composta por 2 anéis de 24 sonares cada, é lógico pensar que durante a tarefa de exploração, vários destes sensores detectarão a mesma estrutura, mas com diferente ângulo de orientação (por causa de sua disposição circular). Para detectar semelhanças ou padrões nos diferentes valores de p_n , precisa-se colocar todas as medições provenientes dos diferentes sensores em uma mesma referência. Portanto, é levada a cabo uma transformação de coordenadas para cada uma das leituras. Além disto, o ponto p_n que inicialmente é considerado como um ponto em coordenada polar, será transformado em um ponto em coordenadas cartesianas, isto é a medição será transformada em $p_n = \{x_n, y_n\}$. A figura 3.4 representa esta transformação. Para cada sensor, as seguintes informações são necessárias para calcular a transformação da observação p_n do sistema de referência do sensor O^S ao sistema de referência global O^G :

- O raio do robô, r ;
- A orientação do i -ésimo sensor com respeito ao sistema coordenado do robô, φ_i ;
- A coordenada x da posição do robô com respeito ao referencial global, x_R ;
- A coordenada y da posição do robô com respeito ao referencial global, y_R ;
- A orientação do robô com respeito ao referencial global, θ_R ;

em que $i = 1, \dots, 24$ corresponde ao número de sensores por anel. A posição do robô é definida pelo vetor de estados $\begin{bmatrix} x_R & y_R & \theta_R \end{bmatrix}^T$ cujos dados provenientes da odometria fornecida pela

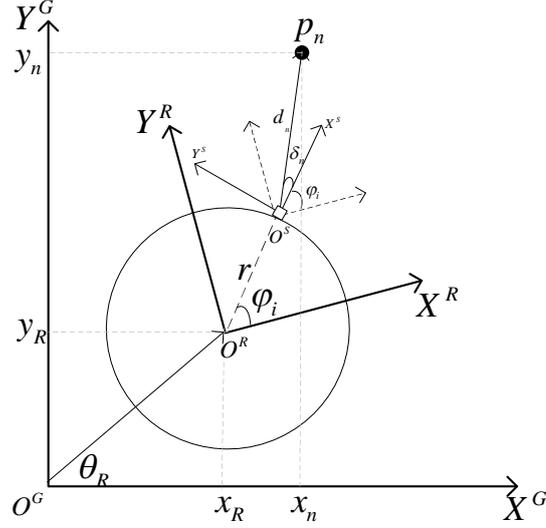


Figura 3.4: Transformação das medidas à referência global.

própria plataforma robótica. O ponto p_n em coordenadas retangulares em O^S corresponde a:

$$\{p_n\}^2 = \begin{bmatrix} d_n \cos(\delta_n) \\ d_n \sin(\delta_n) \end{bmatrix}. \quad (3.3)$$

A expressão de p_n no sistema de referência global está definida pelas seguintes transformações:

1. $O^S \rightarrow O^R$: $\{p_n\}^R = R_S^R \{p_n\}^S$
2. $O^R \rightarrow O^G$: $\{p_n\}^G = R_R^G \{p_n\}^R + t_R^G$

onde R_S^R , R_R^G correspondem a matrizes de rotação e t_R^G a uma matriz de translação, todas definidas a seguir:

$$R_S^R = \begin{bmatrix} \cos(\varphi_n) & -\sin(\varphi_n) \\ \sin(\varphi_n) & \cos(\varphi_n) \end{bmatrix}, \quad R_R^G = \begin{bmatrix} \cos(\theta_R) & -\sin(\theta_R) \\ \sin(\theta_R) & \cos(\theta_R) \end{bmatrix}, \quad t_R^G = \begin{bmatrix} x_R \\ y_R \end{bmatrix}. \quad (3.4)$$

Substituindo (3.4) nas duas transformações temos finalmente a expressão para $\{p_n\}^G$,

$$\{p_n\}^G = \begin{bmatrix} x_n \\ y_n \end{bmatrix}^G = \begin{bmatrix} (r + d_n) \cos(\phi_n + \theta_R) + x_R \\ (r + d_n) \sin(\phi_n + \theta_R) + y_R \end{bmatrix}, \quad (3.5)$$

em que $\phi_n = \varphi_n + \delta_n$.

3.2.2 Propagação do erro

Na seção anterior uma série de operações foram feitas sobre as medições com o objetivo de transformar o conjunto de dados em uma representação que facilite a extração de características. Na transformação feita, a medida d_n e a estimação da orientação ϕ_n mudam para a dupla $\{x_n, y_n\}$ em um sistema de referência global. A equação (3.1) representa a medida como sendo o valor real da magnitude medida mais um erro. Portanto, a mesma representação deve conservar-se após a transformação, quer dizer, o erro ε_d deve ser “propagado” para transformar-se em ε_x e ε_y .

Para este caso as transformações utilizadas relacionam funções não-lineares, logo, deve ser realizada uma aproximação de primeira ordem ou linearização. Especificamente, a propagação da variância na medida de distância σ_d^2 é propagada para o sistema coordenado do robô, utilizando o processo de propagação por linearização, utilizado no trabalho de Navarro *et al.* [42], assim:

$$\{C_{xy}\}^R = J_1 \{\sigma_a^2\} J_1^T, \quad (3.6)$$

em que J_1 corresponde à matriz Jacobiana:

$$J_1 = \frac{\partial \{p_n\}^R}{\partial d_n} = \begin{bmatrix} \cos(\phi_n) \\ \sin(\phi_n) \end{bmatrix}. \quad (3.7)$$

Da mesma maneira, a matriz de covariância do erro C_{xy}^G da medida transformada p_n na referência global é calculada por meio da seguinte expressão:

$$\{C_{xy}\}^G = J_2 \{C_{xy}\}^R J_2^T + J_3 C_{robot} J_3^T, \quad (3.8)$$

com J_2 e J_3 igual a:

$$J_2 = \frac{\partial \{p_n\}^G}{\partial x_n^R \partial y_n^R} \quad e \quad J_3 = \frac{\partial \{p_n\}^G}{\partial x_R \partial y_R \partial \theta_R}. \quad (3.9)$$

A expressão em (3.8) possui também a contribuição do erro na posição do robô, dado pela matriz C_{robot} , que geralmente é definida como:

$$C_{robot} = \begin{bmatrix} \sigma_{x_R}^2 & \sigma_{x_R y_R} & \sigma_{x_R \theta_R} \\ \sigma_{y_R x_R} & \sigma_{y_R}^2 & \sigma_{y_R \theta_R} \\ \sigma_{\theta_R x_R} & \sigma_{\theta_R y_R} & \sigma_{\theta_R}^2 \end{bmatrix} \quad (3.10)$$

Para este trabalho considerada-se as medidas de cada um dos eixos do robô como não correlacionadas (devido a desconhecimento da correlação entre as diferentes fontes de erro de

Tabela 3.1: Representações matemáticas para uma linha reta

| Nome da Representação | Expressão |
|--------------------------|--|
| Forma Explícita | $y = \theta_1 x + \theta_2$ |
| Forma Geral | $\theta_1 x + \theta_2 y + \theta_3 = 0$ |
| Forma Hessiana ou normal | $x \cos(\theta_1) + y \sin(\theta_1) - \theta_2 = 0$ |

cada medida x_R , y_R e θ_R), portanto a matriz C_{robot} será igual a:

$$C_{robot} = \begin{bmatrix} \sigma_{x_R}^2 & 0 & 0 \\ 0 & \sigma_{y_R}^2 & 0 \\ 0 & 0 & \sigma_{\theta_R}^2 \end{bmatrix} \quad (3.11)$$

Os valores das incertezas σ_{x_R} , σ_{y_R} foram calculados por meio de testes feitos para cada eixo. Os detalhes são explicados no apêndice C. Devido a que nos testes experimentais não foram implementadas rotações da plataforma robótica, o valor de σ_{θ_R} não foi estimado a partir de experimentos. O valor dado para este parâmetro foi um valor apenas representativo, que permitisse usar a matriz dada em (3.11).

Com os resultados extraídos nas equações (3.5) e (3.8) é possível representar o resultado das medições realizadas por qualquer um dos sonares, junto com o respectivo erro, em um sistema de referência global.

3.3 MODELO DO SEGMENTO DE LINHA

Como já ficou claro desde a apresentação inicial do trabalho, o mapa construído para o ambiente interno que será explorado baseia-se em segmentos de linhas, por serem estas as *features* mais comuns em ambientes estruturados.

Com relação ao modelo que definirá a linha reta, existem várias representações para este tipo específico de *feature*. A tabela 3.1 apresenta as representações mais comuns, extraídas do trabalho de Kapp e Gröll [43]. De todas elas a mais comum é a forma explícita, a qual possui uma desvantagem importante na hora de representar linhas verticais devido a que o parâmetro θ_1 tende ao infinito. O segundo modelo, a representação geral, também é linear em parâmetros e com a vantagem de poder descrever todo tipo de linhas, mas com custo de possuir mais um parâmetro. Já a forma Hessiana ou Normal é uma representação não linear em parâmetros que também representa qualquer tipo linha reta. Embora seja uma equação não linear, possui só dois parâmetros, o que facilita a tarefa de estimação. Justamente por esta razão, a representação Hessiana foi a escolhida para modelar os segmentos de linhas.

3.3.1 Estimação dos parâmetros da linha reta

Mudando um pouco a notação da representação Hessiana apresentada na tabela 3.1, temos que as linhas retas do mapa terão a seguinte forma:

$$\rho_k = x_n \cos(\alpha_k) + y_n \sin(\alpha_k), \quad (3.12)$$

em que o k é um contador de linhas e n o n -ésimo ponto pertencente à linha. A estimação dos parâmetros do modelo, ρ_k e α_k , é levada a cabo por meio do uso dos M-Estimadores [6]. Este tipo de técnica de estimação está dentro do grupo de métodos estatísticos robustos, os quais são projetados para ter um bom desempenho quando a forma do verdadeiro modelo subjacente desvia-se ligeiramente do modelo paramétrico, em outras palavras, quando existe presença de *outliers* nos dados ou de medições pertencentes a outras populações, o que afeta ou dificulta o processo de estimação.

Uma medida de robustez para os diferentes métodos de estimação propostos na literatura pode ser o cálculo do *breakdown point*, que corresponde à mínima porcentagem de *outliers* que pode causar que o estimador divirja arbitrariamente do valor real. Para o caso dos M-Estimadores, o valor de *breakdown point* deve ser inferior ao 20% de outliers no conjunto de dados analisados.

O problema a ser enfrentado neste trabalho se enquadra dentro das situações onde as medições estão contaminadas por ruído, devido às características já mencionadas do sensor de ultrassom, assim como a possibilidade de ter diferentes populações de medições, justamente pela detecção simultânea de varias estruturas do ambiente, o que justifica o uso deste método de estimação.

Os M-Estimadores são uma generalização dos Estimadores de Máxima Verossimilhança (EMV) e de Mínimos Quadrados (MQ) [44]. Especificamente a M-estimação de um vetor de parâmetros θ está definida como:

$$\hat{\theta} = \arg_{\theta} \min \sum_{n=1}^N \rho(r_n), \quad (3.13)$$

em que $\rho(r_n)$ é uma função robusta (ou função objetivo) que, assim como no caso do EMV, está definida em função da diferença de uma distância geométrica (entre os N pontos e o modelo a ser estimado) ou de uma função resíduo. A função robusta é uma função simétrica, positiva definida e com um único mínimo em zero. A minimização da equação (3.13) é encontrada a partir do cálculo da primeira derivada parcial de $\rho(r_n)$ com respeito a cada um dos parâmetros:

$$\sum_{n=1}^N \frac{\partial \rho(r_n)}{\partial r_n} \frac{\partial r_n}{\partial \theta} = \sum_{n=1}^N \psi(r_n) \frac{\partial r_n}{\partial \theta} = 0, \quad (3.14)$$

em que $\psi(r_n) = \frac{\partial \rho(r_n)}{\partial r_n}$ é chamada de *função de influência*. Esta função mede a influência dos dados no processo de estimação, quer dizer, a mudança no valor da estimação causada pela

inserção de um “outlier”. O passo seguinte é introduzir uma função peso $w(r_n)$ de maneira que $w(r_n)r_n = \psi(r_n)$. Desta maneira a eq (3.14) pode ser reescrita como:

$$\sum_{n=1}^N w(r_n)r_n \frac{\partial r_n}{\partial \theta} = 0. \quad (3.15)$$

A equação 3.15 assemelha-se com um problema de Mínimos Quadrados Ponderados Iterativos [4], isto é, em cada iteração do algoritmo é calculado o peso w_n usando o valor da estimativa atual e será calculado um novo valor da estimativa a partir da expressões que definem os parâmetros do modelo, até atingir o critério de parada. Um valor inicial para $\hat{\theta}$ pode ser dado como ponto de início, ou os pesos podem ser inicializados todos em 1 (o que significa que todos os dados são igualmente importantes no começo), caso não se tenha informação inicial.

As expressões que definem as estimativas dos parâmetros, ρ_k e α_k , são respectivamente:

$$\begin{aligned} \hat{\rho}_k &= \bar{x} \cos(\hat{\alpha}_k) - \bar{y} \sin(\hat{\alpha}_k) \\ \hat{\alpha}_k &= \frac{1}{2} \arctan \left(\frac{-2 \sum_n w_n (x_n - \bar{x})(y_n - \bar{y})}{\sum_n w_n (y_n - \bar{y})^2 - \sum_n w_n (x_n - \bar{x})^2} \right), \end{aligned} \quad (3.16)$$

\bar{x} e \bar{y} são as médias ponderadas das coordenadas x e y do ponto p_n , definido em (3.5). Assim, a função resíduo, $\rho(r_n)$, é definida como:

$$r_n = \hat{\rho}_k - x_n \cos(\hat{\alpha}_k) - y_n \sin(\hat{\alpha}_k). \quad (3.17)$$

Existem vários M-Estimadores propostos na literatura, que diferem entre eles pela forma da função $\rho(\cdot)$, e em conseqüência, também $\psi(\cdot)$ e $w(\cdot)$. Para este trabalho o M-Estimador que foi utilizado é o Estimador de Huber [6], o qual possui as seguintes funções:

$$\begin{aligned} \rho(r_n) &= \begin{cases} r_n^2/2, & \text{se } |r_n| \leq c \\ c(|r_n| - c/2), & \text{caso contrário} \end{cases} \\ \psi(r_n) &= \begin{cases} r_n, & \text{se } |r_n| \leq c \\ c \operatorname{sign}(r_n), & \text{caso contrário} \end{cases} \\ w(r_n) &= \min \left\{ 1, \frac{c}{|r_n|} \right\} \end{aligned} \quad (3.18)$$

O parâmetro c é o parâmetro de sintonia do estimador que esta relacionado com a eficiência¹ do mesmo e tem um valor igual a 1.345. Finalmente, com o desenvolvimento anterior é possível calcular as estimativas dos parâmetros que definem o modelo da linha reta que representa a tendência geral de um conjunto de medidas do ambiente a ser mapeado.

¹Neste contexto a eficiência do estimador está definida como razão entre a mínima variância possível do estimador e o valor real da variância do estimador (neste caso, robusto) [44].

3.3.2 Cálculo de $C_{\hat{\theta}}$

O cálculo da matriz de covariância dos parâmetros estimados, $C_{\hat{\theta}}$ reflete o grau de precisão na estimação dos de $\hat{\rho}_k$ e $\hat{\alpha}_k$. Adicionalmente a isto, este cálculo também mede influência das perturbações provenientes do conjunto de dados. Portanto, é importante seu conhecimento com a finalidade de relacionar estas incertezas com os parâmetros que definem as condições dos processos de extração de linhas e de *merging*.

O método utilizado para achar a matriz de covariância é o desenvolvido por Haralick [45] e já foi usado em outros trabalhos de extração de *features*, como o trabalho de Borges e Aldon [36]. Este método propaga as incertezas próprias das entradas (as medições) até o processo de estimação de parâmetros, o qual pode envolver expressões não lineares. A relação entre as entradas e as estimativas é através da minimização de uma função de custo ou função objetivo.

Especificamente, segundo o método de Haralick, a matriz de covariância dos dados de entrada, C_{xy} e a matriz de covariância das estimativas, $C_{\hat{\theta}}$, estão relacionadas mediante a seguinte expressão [46]:

$$C_{\hat{\theta}} = \left(\frac{\partial g(\mathbf{P}_{l_k}, \hat{\theta})}{\partial \hat{\theta}} \right)^{-1} \left(\frac{\partial g(\mathbf{P}_{l_k}, \hat{\theta})}{\partial \mathbf{P}_{l_k}} \right)^T C_{XY} \left(\frac{\partial g(\mathbf{P}_{l_k}, \hat{\theta})}{\partial \mathbf{P}_{l_k}} \right) \left[\left(\frac{\partial g(\mathbf{P}_{l_k}, \hat{\theta})}{\partial \hat{\theta}} \right)^{-1} \right]^T \quad (3.19)$$

Onde \mathbf{P}_{l_k} corresponde aos dados $[X \ Y]^T$ que pertencem à k-ésima linha estimada, cujo vetor de parâmetros é $\hat{\theta}$, ambos definidos assim:

$$\mathbf{P}_{l_k} = \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} \{x_1, x_2, \dots, x_N\} \\ \{y_1, y_2, \dots, y_N\} \end{bmatrix} \quad \hat{\theta} = \begin{bmatrix} \hat{\rho}_k & \hat{\alpha}_k \end{bmatrix}^T \quad (3.20)$$

Assim mesmo, a função g corresponde ao jacobiano da função de custo que relaciona \mathbf{P}_{l_k} com o vetor de estimativas $\hat{\theta}$, isto é:

$$g = \frac{\partial J}{\partial \hat{\theta}}, \quad J(\hat{\theta}, p_{l_k}) = \sum_{n=1}^N (\hat{\rho}_k - x_n \cos(\hat{\alpha}_k) - y_n \sin(\hat{\alpha}_k)) \quad (3.21)$$

Finalmente, com o desenvolvimento da expressão em (3.19) é possível encontrar a matriz de covariância das estimativas para a linha estimada. Isto será útil nos seguintes processos da estratégia de mapeamento. No apêndice A encontra-se o desenvolvimento matemático completo do cálculo de $C_{\hat{\theta}}$.

3.3.3 Extração dos segmentos de linha

Da seção 2.3 é possível concluir que algoritmos baseados em *Split-and-Merge* e no método Incremental apresentam bons resultados em extração de segmentos de linha para a construção de mapas 2D. A partir disto, na tarefa de extração de linhas, a estratégia de mapeamento

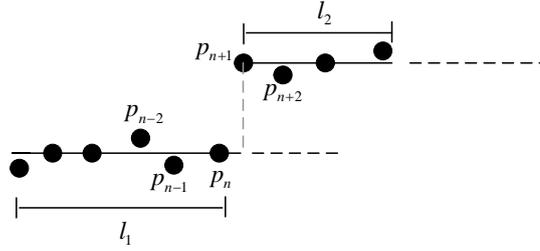


Figura 3.5: Situação ilustrativa de extração de linhas.

deste trabalho faz uso do algoritmo Incremental, aproveitando seu bom desempenho além de sua simplicidade em termos de programação. A figura 3.5 apresenta uma situação ilustrativa do uso deste algoritmo, onde se apresenta um conjunto de dados que representam duas linhas. O funcionamento do algoritmo é explicado a seguir.

| Algoritmo Incremental clássico |
|---|
| <ol style="list-style-type: none"> 1. Entrada do primeiro ponto, $p_{n-1} = (x_{n-1}, y_{n-1})$ 2. pontos iniciais = $\{p_{n-1}, p_n\}$ 3. Cálculo de $\hat{\rho}_k$ e $\hat{\alpha}_k$ 4. Inclusão do seguinte ponto p_{n+1} 5. $d_{pl} = \hat{\rho}_k - x_{n+1} \cos(\hat{\alpha}_k) - y_{n+1} \sin(\hat{\alpha}_k)$ 6. Se a condição $d_{pl} \leq D_{max}$ é satisfeita recalcula $\hat{\rho}_k$ e $\hat{\alpha}_k$ com p_{n+1} e retorne a 4. 7. Caso contrário, p_{n+1} faz parte de uma nova linha. $p_{n+1} = p_{n-1}$, $k++$ e retorne a 2. |

O conjunto de medições, que compõem a entrada do algoritmo, é analisado de forma sequencial, quer dizer, é conservada a ordem em que as leituras foram tomadas. Cada ponto p_n é analisado, calculando a distância entre este e a linha (distância d_{pl}) que esta sendo estimada, usando a expressão dada na equação 3.17. Se a distância calculada é menor que determinado limiar D_{max} , o ponto é considerado como pertencente à linha e é adicionado a esta, caso contrário este será considerado como o primeiro ponto de uma nova linha. O processo continua até que a totalidade de pontos tem sido avaliada. Na saída do algoritmo é obtido um número determinado de linhas extraídas.

Apesar das vantagens mencionadas na seção 2.3, este algoritmo possui uma notável sensibilidade ao limiar definido por D_{max} . Além disto, a abordagem Incremental faz uso de uma suposição a iniciar a extração de uma nova linha: os dois primeiros pontos analisados são supostos pertencentes a essa linha, e a partir deles são estimados os primeiros valores dos parâmetros ρ_k e α_k . Isto quer dizer que nenhum critério é implementado para estabelecer se esses dois pontos são realmente colineares, o que representa a possível inclusão de um ponto que pode não pertencer à linha, causando uma diferença entre a *feature* estimada e a estrutura real que esta representa.

Na implementação deste algoritmo, algumas melhoras foram aplicadas para evitar este tipo de situações. Primeiro, aquela suposição inicial em que os dois primeiros pontos de uma linha nova são colineares, não é mais utilizada, em vez disto, no início de cada iteração

do algoritmo é avaliada a distância euclideana entre a dupla de pontos analisada, e se estes correspondem aos dois primeiros pontos considerados depois da estimação de uma linha (o que quer dizer que uma nova linha está sendo estimada), este será o único critério avaliado para esta iteração. A distância euclideana está definida como:

$$d_{euc} = \sqrt{(x_{n+1} - x_n)^2 + (y_{n+1} - y_n)^2} \quad (3.22)$$

Se d_{euc} não supera o limiar D_{max1} , os dois pontos são considerados pertencentes a mesma linha, já no caso contrário, um dos pontos é rejeitado e o seguinte ponto do grupo de dados é tomado para ser analisado. Depois desta primeira análise, é estimada a linha com os pontos escolhidos e o processo normal do algoritmo Incremental continua.

O cálculo do parâmetro D_{max1} está relacionado com a forma que em que são analisadas as medições provenientes do conjunto de sonares. No caso em que as medições são analisadas sonar por sonar, o D_{max1} está relacionado com a distância mínima percorrida pelo robô durante a tomada de leituras consecutivas de um mesmo sonar. Agora se a análise das medições é feito por *scans*², D_{max1} corresponde com a distância de separação entre sensores adjacentes, que para o caso da configuração do Nomad corresponde ao arco descrito entre os eixos principais de cada sonar. Para este trabalho, as medições foram analisadas segundo o primeiro esquema, isto é, as leituras de cada sonar de forma individual.

Em relação ao valor para o segundo parâmetro, D_{max2} , que define o valor máximo de distância entre o ponto e linha, este é ajustado segundo um valor que está relacionado com a variância dos parâmetros estimados, calculada na seção anterior. As modificações feitas no algoritmo Incremental aparecem representadas no pseudo código apresentado a seguir.

| Algoritmo Incremental implementado |
|---|
| <ol style="list-style-type: none"> 1. Entrada do primeiro ponto, $p_{n-1} = (x_{n-1}, y_{n-1})$ 2. Entrada das matrizes C_{xy} para cada ponto analisado. 3. pontos iniciais = $\{p_{n-1}, p_n\}$ 4. $d_{euc} = \sqrt{(x_{n-1} - x_n)^2 + (y_{n-1} - y_n)^2}$ 5. Se a condição $d_{euc} \leq D_{max1}$ é satisfeita, continue 6. Caso contrário, p_{n-1} é rejeitado, $p_n = p_{n-1}$ retorne a 3. 7. Cálculo de $\hat{\rho}_k$ e $\hat{\alpha}_k$ 8. Cálculo de $C_{\hat{\theta}}$ 9. Inclusão do seguinte ponto p_{n+1} 10. $d_{pl} = \hat{\rho}_k - x_{n+1} \cos(\hat{\alpha}_k) - y_{n+1} \sin(\hat{\alpha}_k)$ 11. Se a condição $d_{pl} \leq D_{max2}$ é satisfeita recalcula $\hat{\rho}_k$, $\hat{\alpha}_k$, $C_{\hat{\theta}}$ com p_{n+1}, e retorne a 9. 12. Caso contrário, p_{n+1} faz parte de uma nova linha. $p_{n+1} = p_{n-1}$, $k++$ e retorne a 3. |

Observa-se que é introduzida a etapa de cálculo da distância de euclidiana como o primeiro critério da avaliação dos pontos, assim como o cálculo da matriz de covariância $C_{\hat{\theta}}$. O

²Um *scan* corresponde à totalidade das leituras dos 24 sonares que compõem um anel.

cálculo do segundo critério, a distância entre ponto e linha, permanece da mesma forma que o algoritmo original apresentado acima.

3.4 MERGING

Uma vez que a totalidade dos pontos é analisada e os respectivos segmentos de linhas são extraídos, estes podem superar em número as estruturas reais do ambiente, isto pelo mesmo funcionamento do algoritmo de extração e também pelo modo de análise dos dados provenientes do grupo de sensores. Uma análise de leituras por *scans* ou por sonares de modo individual, faz com que a implementação do algoritmo Incremental produza uma diferença maior ou menor entre o número de *features* extraídas e o número de estruturas planas do ambiente real.

O processo de *merging* é utilizado justamente para resolver este problema. Mediante um critério apropriado de combinação, as linhas que possuem as mesmas características são agrupadas e uma linha definitiva é estimada, reduzindo o número total de linhas do mapa, assemelhando-o mais com o ambiente real.

No processo de *merging* implementado, o critério usado para combinar duas ou mais linhas consta de duas condições que devem ser cumpridas. A primeira baseia-se num teste Chi-Quadrado apresentado no trabalho de Pfister et. al [47], que determina se a diferença entre os parâmetros de duas linhas está dentro do intervalo de confiança de aproximadamente um 95%, definido pela combinação das matrizes de covariância das *features* analisadas. Expressando isto matematicamente temos:

$$\chi^2 = (\delta\hat{\theta})^T (C_{\hat{\theta}_k} + C_{\hat{\theta}_{k+1}})^{-1} (\delta\hat{\theta}) < 5.99, \quad (3.23)$$

onde

$$\delta\hat{\theta} = \begin{bmatrix} \hat{\rho}_{k+1} - \hat{\rho}_k \\ \hat{\alpha}_{k+1} - \hat{\alpha}_k \end{bmatrix}.$$

Já na segunda condição que compõe o critério, avalia-se os pontos pertencentes às linhas que cumpriram com a primeira condição. A razão desta análise é que em alguns casos, as linhas que possuem os mesmos parâmetros ρ_k e α_k podem não representar as mesmas linhas, o que pode ocasionar o agrupamento errado de linhas que não representam a mesmas estruturas reais. Este pode ser o caso de linhas que estão à mesma profundidade se vistas desde o mesmo ponto pelo robô, quer dizer, linhas que estão à mesma distância e com a mesma orientação, mas não conformam a mesma estrutura, por exemplo, duas paredes separadas por um espaço considerável.

Esta condição especificamente detecta semelhanças entre pontos que compõem as duas linhas que passaram a primeira condição. Para isto, são identificados os pontos iniciais e finais de cada uma das linhas, assim como se ambas as linhas são horizontais (coordenada

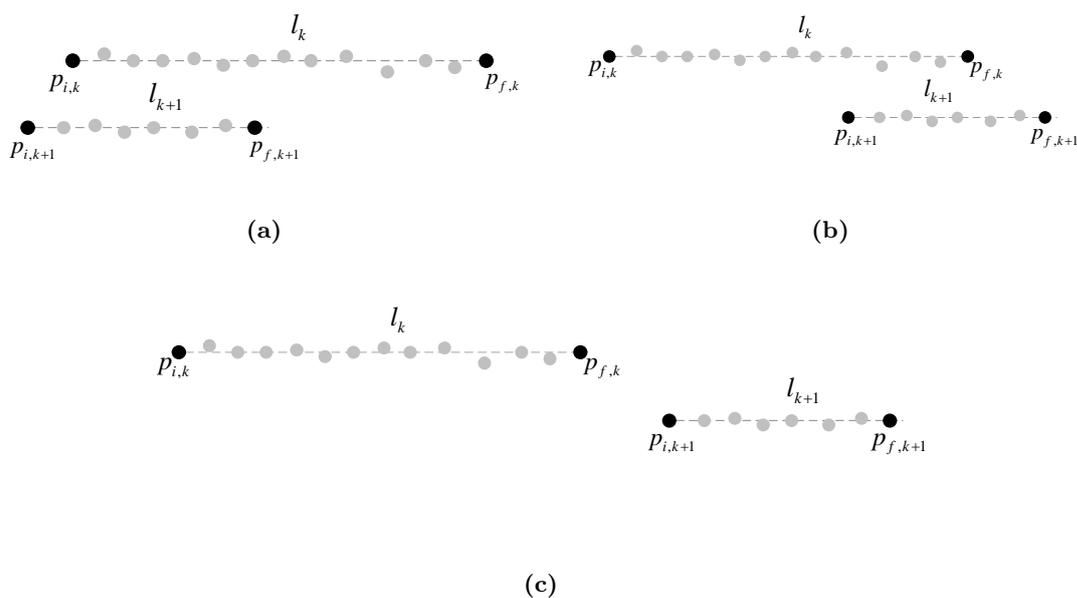


Figura 3.6: Casos estabelecidos para o segundo critério no processo de *merging*. As linhas são comparadas em termos dos pontos que as compõem, delimitadas por seus pontos iniciais e finais. Para fins ilustrativos, uma das linhas foi colocada embaixo da outra.

y é quase constante para cada um dos pontos, e a variação em x é mais pronunciada) ou verticais. Então, são estabelecidos três casos gerais (figura 3.6) que representam as situações que podem ocorrer em relação a essas linhas. Se estas caírem no caso I ou II, as linhas serão imediatamente combinadas numa única linha. No caso caso III, a distância entre o ponto final de uma linha e ponto inicial da outra será calculada, e se esta supera a distância máxima que pode existir entre duas paredes reais da mesma profundidade (chamada de D_{max3}), dada pelas mesmas condições do ambiente, as linhas não serão combinadas.

Com este processo é encerrada a estratégia de mapeamento proposta neste trabalho. O desempenho da mesma será avaliado com os resultados dos testes em ambientes simulados assim como na exploração de um ambiente interno real. Tudo isto será apresentado no capítulo de resultados.

RESUMO DE CAPÍTULO

Neste capítulo é explicada detalhadamente a estratégia de mapeamento em 2D baseada em segmentos de linhas. O processo inicia com o modelo de medição, onde a incerteza própria do sensor é incluída. Na sequência, uma série de transformações são feitas para levar todas as medições até uma referência global, e desta maneira facilitar a tarefa de extração de *features*. O seguinte processo corresponde precisamente à extração de linhas do conjunto de dados, o qual faz uso do algoritmo Incremental. Na saída desta etapa, um número considerável de

linhas são extraídas. O último processo corresponde à combinação de linhas que possuem as mesmas características e que representam a mesma estrutura real. Por fim, um conjunto de linhas, definidas pelos parâmetros e pelos pontos pertencentes a cada uma delas é entregue como saída final desta estratégia. Os resultados visualizados no capítulo seguinte validarão a efetividade da abordagem aqui apresentada.

4 RESULTADOS EXPERIMENTAIS

A estratégia proposta neste trabalho foi avaliada em duas etapas. Inicialmente, foi desenvolvido um ambiente simulado em *Matlab* com o objetivo de verificar o desempenho geral da estratégia. A segunda etapa consta de três testes experimentais em um ambiente real, utilizando a plataforma robótica Nomad XR4000. Neste capítulo são apresentados os resultados obtidos tanto na simulação quanto nos testes experimentais. Também são calculados os erros dos parâmetros estimados para cada linha extraída, assim como valores de indicadores relacionados com a associação destas linhas com as estruturas reais.

4.1 CRITÉRIOS DE AVALIAÇÃO

4.1.1 Erro de estimação dos parâmetros

O Erro Quadrático Médio de um estimador $\hat{\Theta}$, ou MSE por suas siglas em inglês (*Mean Square Error*), está definido como [2]:

$$\text{MSE}(\hat{\Theta}) = E \left[\left(\hat{\Theta} - \theta \right)^2 \right], \quad (4.1)$$

em que θ é o valor real do parâmetro. Se o vetor de parâmetros $\hat{\Theta}$ está conformado pelas estimativas $\theta_1, \theta_2, \dots, \theta_N$, o $\text{MSE}(\hat{\Theta})$ pode ser reescrito como:

$$\text{MSE}(\hat{\Theta}) \approx \frac{1}{N} \sum_{i=1}^N \left(\hat{\theta}_i - \theta \right)^2, \quad (4.2)$$

Especificamente para este trabalho o cálculo da precisão dos estimadores para cada linha extraída é representado por meio do RMSD (*Root Mean Square Deviation*), que nada mais é que a raiz quadrada do MSE. Isto é:

$$\begin{aligned} \text{RMSD}(\hat{\alpha}_k) &= \sqrt{\frac{1}{M} \sum_{i=1}^M (\hat{\alpha}_{i,k} - \alpha_k)^2} \quad e \\ \text{RMSD}(\hat{\rho}_k) &= \sqrt{\frac{1}{M} \sum_{i=1}^M (\hat{\rho}_{i,k} - \rho_k)^2}, \end{aligned} \quad (4.3)$$

em que M são as linhas extraídas que correspondem à k -ésima estrutura real representada pelos parâmetros ρ_k e α_k .

4.1.2 Associação

A associação¹ abrange o cálculo do número de linhas que representam as diferentes estruturas do ambiente, assim como aquelas que não representam nenhuma delas, denominadas “verdadeiros positivos” (*true positive*) e “falsos positivos” (*false positive*), respectivamente. O cálculo destas características foi definido no trabalho de Nguyen *et al.* [38] como:

$$VerdPos = \frac{N.Corresp}{N.LinhasExtAlg} \quad e \quad (4.4)$$

$$FalsoPos = \frac{N.LinhasExtAlg - N.Corresp}{N.LinhasExtAlg},$$

em que $N.Corresp$ é o número de linhas extraídas que correspondem às linhas verdadeiras, e $N.LinhasExtAlg$ é o número total de linhas extraídas por o algoritmo.

Além destas duas características de associação, é adicionada outra que estabelece o número médio de linhas correspondentes por linha verdadeira ($N.LinhasVerd$) ou linha real, definida como:

$$VerdPos/linha = \frac{N.LinhasExtAlg}{N.LinhasVerd}. \quad (4.5)$$

Estes indicadores serão utilizados na avaliação dos resultados tanto do exercício do mapeamento em ambiente simulado, como no ambiente real com a plataforma robótica.

4.2 SIMULAÇÃO

4.2.1 Descrição do ambiente simulado

O ambiente simulado que vai ser usado tem dimensões de 1.2×2 m e está composto por 13 paredes e um espaço considerável entre duas delas (L11 e L12); com isto, deseja-se verificar o desempenho do algoritmo ante espaços livres entre paredes de igual profundidade. Na figura 4.1 é apresentado o gráfico deste ambiente, assim como a posição inicial do robô no início da simulação. O percurso do robô segue a sequência desde L1 até L13.

O sistema de sensoriamento simulado está conformado por oito sonares com um espaçamento entre cada um de 45° (identificados com os pontos verdes na circunferência que representa o robô). Ao igual que o sistema real, o campo de visão do robô simulado abrange os 360° . Para manter uma proporção entre o ambiente e as dimensões de robô, o raio do mesmo foi reduzido a uma terceira parte do valor real.

¹o termo usado em inglês para esta característica é *correctness*, e pela definição dada em [38], considera-se que esta é a tradução mais adequada.

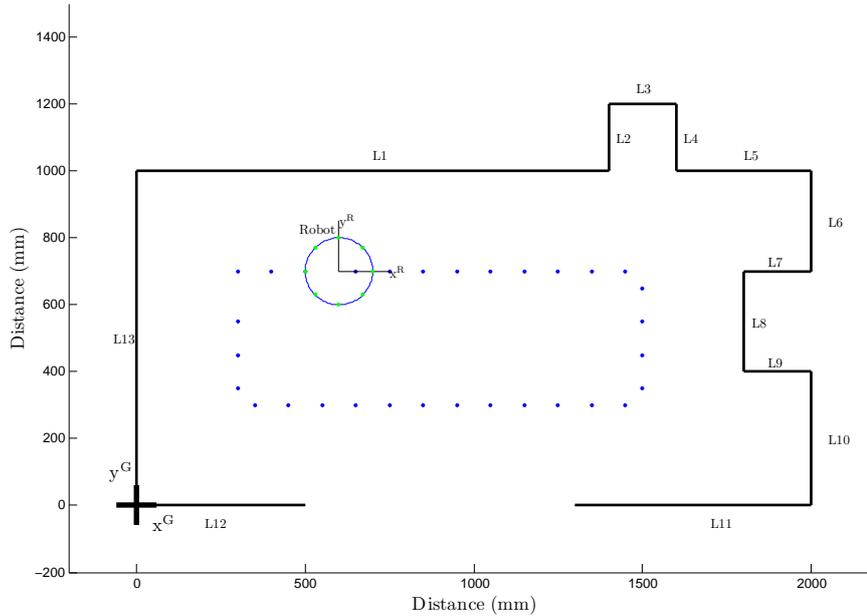


Figura 4.1: Ambiente simulado.

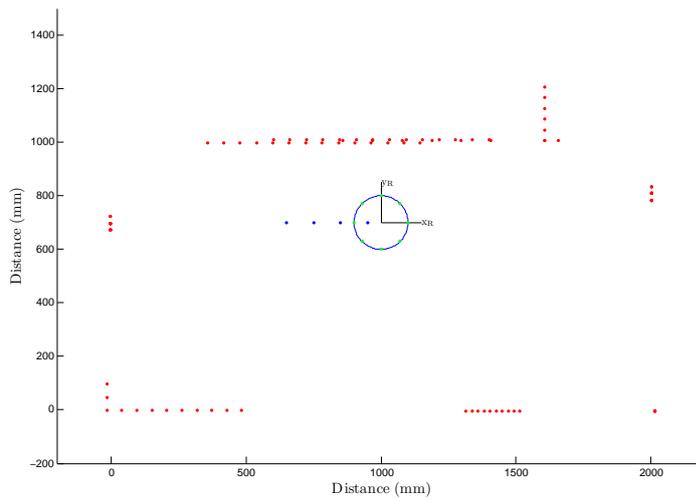
A posição do robô é medida cada 50mm, o que corresponde também à finalização de um *scan* do sistema sensorial completo, portanto a exploração completa do ambiente é conseguida com um total de 63 *scans*, o que corresponde a um conjunto de medições de 504 dados. É preciso ressaltar que os dados de posição foram simulados sem erros, portanto não existe influência do erro de odometria. Aos dados dos sonares foi adicionado um ruído gaussiano aleatório de média zero e variância igual a 40mm.

Uma sequência da trajetória completa de exploração do ambiente junto com o conjunto de medições simulados é apresentada na figura 4.2. Os dados mostrados nesta figura já possuem a transformação inicial de coordenadas que faz parte do pré-processamento dos dados, de maneira que estes estão em coordenadas globais correspondentes à referência global do mapa que coincide com a origem do mesmo, localizada no extremo inferior esquerdo do gráfico. Pode ser observado na figura 4.2 que, igual a um caso real, existe uma forte possibilidade que nem todas as estruturas sejam detectadas pelo sistema sensorial, portanto, das 13 paredes presentes no ambiente, só duas não conseguiram ser detectadas: as paredes L7 e L9.

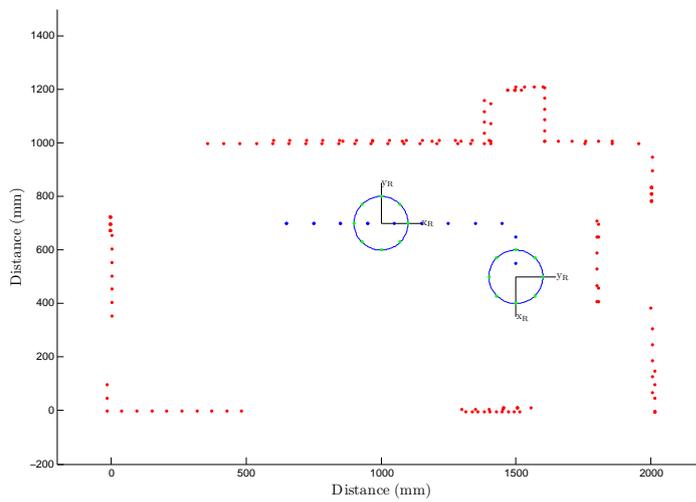
4.2.2 Implementação do algoritmo

Como já foi mencionado antes, a única informação *a priori* (além do ajuste dos parâmetros de cada processo) que o algoritmo precisa para ser executado é a posição inicial do robô com respeito à referência global que tem sido definida previamente, e que corresponde justamente à origem do mapa.

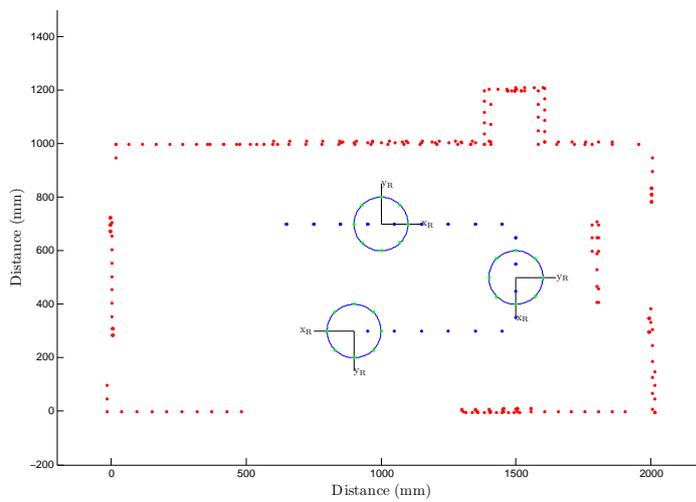
No relacionado ao ajuste dos parâmetros para o algoritmo de extração de features, estes dependem das condições do experimento, assim como dos erros dos sensores. Especificamente, para os parâmetros correspondentes aos limiares estabelecidos pelas distâncias D_{max_1} e D_{max_2}



(a)

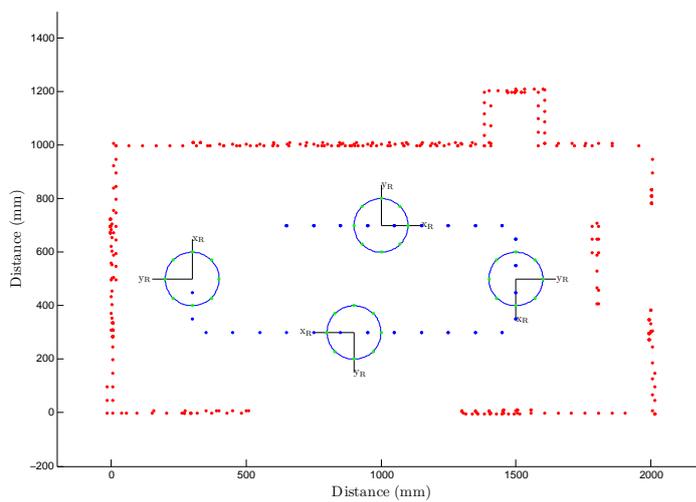


(b)

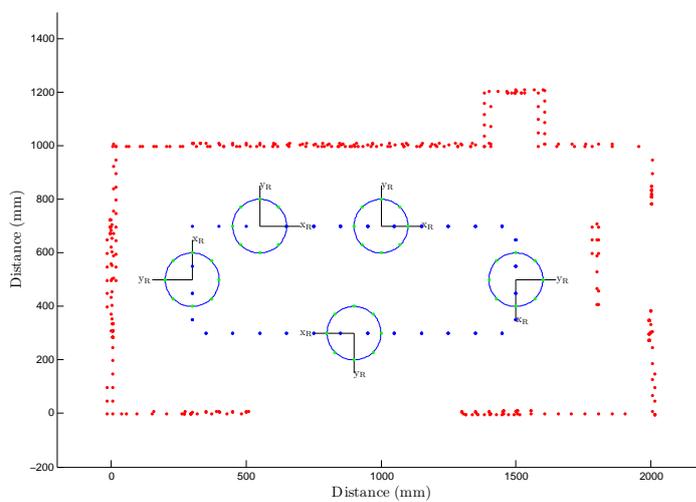


(c)

Figura 4.2: Sequência da exploração do ambiente executado pelo robô.



(d)



(e)

Figura 4.2: Sequência da exploração do ambiente executado pelo robô.

usados pelo algoritmo Incremental (explicado na seção 3.4.3), o primeiro, que avalia a distância entre medições consecutivas, foi ajustado em 50mm, que é justamente a distância percorrida pelo robô entre duas medições consecutivas. Já para o caso do parâmetro D_{max2} , o qual condiciona a distância entre a medição atual e última linha estimada, além de depender da distância entre leituras consecutivas (50 mm), este está relacionado com a incerteza das estimativas dos parâmetros da reta, dada por $C_{\hat{\theta}}$, portanto é um valor que varia com cada linha estimada.

O algoritmo Incremental é executado por cada sonar presente no sistema de sensoriamento (neste caso, 8), extraindo as linhas detectadas por cada sensor. O algoritmo tem uma etapa intermédia de *merging*, a qual é executada toda vez que algoritmo finaliza com a análise das medições de um sonar, desta maneira se combinam as linhas que representam a mesma estrutura real detectadas por este sensor. Com isto se pretende diminuir o volume de dados a serem analisados na etapa final de *merging*, e assim melhorar o tempo de execução final da estratégia de mapeamento completa.

Além do ajuste inicial dos parâmetros, é estabelecido um número mínimo de pontos por linha estimada, isto com a finalidade de não incorporar linhas muito pequenas provavelmente formadas por dados espúrios que não representam nenhuma estrutura real do ambiente. Este valor foi ajustado em 5, pois foi observado nos resultados que este é o número mínimo de pontos que se precisam para formar uma linha.

Na figura 4.3 é mostrado o resultado da etapa de extração de segmentos de linhas, onde podem ser observados as diferentes linhas que são extraídas das leituras de cada sonar. Alguns pontos são desconsiderados, pois não atendem com os critérios estabelecidos de mínima distância entre pontos consecutivos e mínima distância entre ponto e linha estimada, mas em

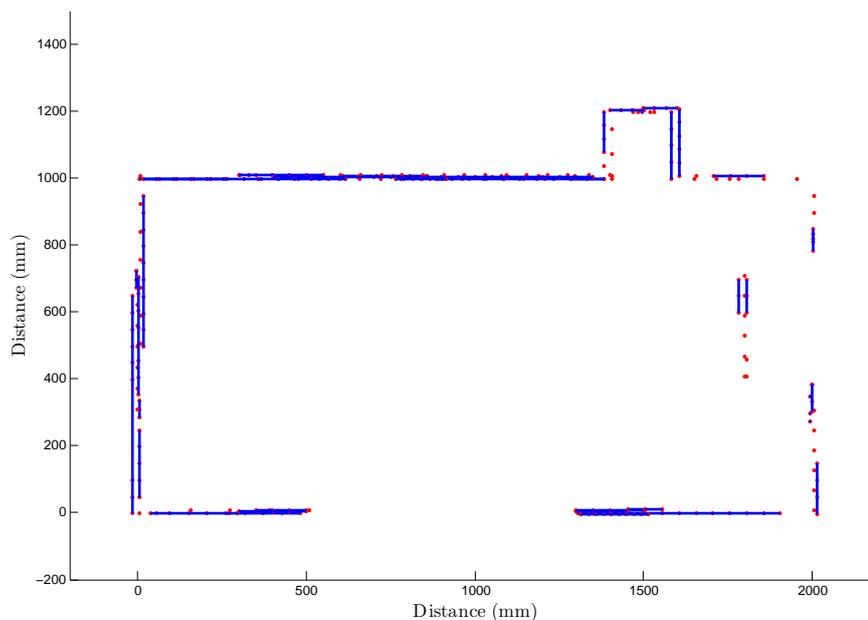


Figura 4.3: Resultado da extração das linhas.

termos gerais a maioria dos pontos são aproveitados para extrair as características do entorno.

Na etapa final de *merging* de linhas ou *merging*, todas as linhas extraídas por cada sonar são comparadas entre si para combinar aquelas com parâmetros similares que cumpram com as condições de avaliação explicadas na seção 3.4. Dentro dos parâmetros relacionados com os critérios de combinação desta etapa, está a distância mínima que pode existir entre paredes de igual profundidade, como é o caso para este ambiente simulado. Esta distância é ajustada em um valor igual ao espaço que existe entre as paredes L1 e L5, o qual é de 200mm.

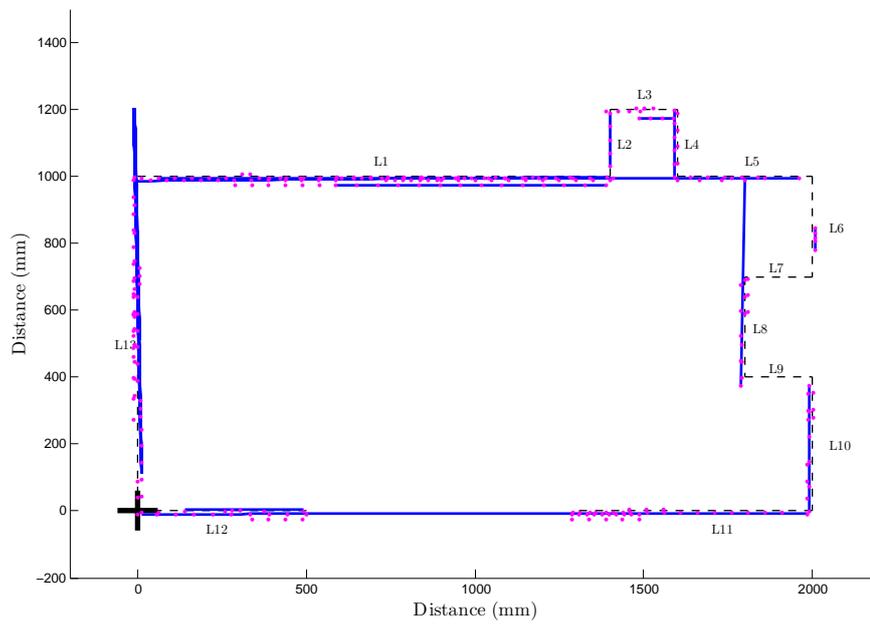
Com o objetivo de comparar o desempenho do algoritmo Incremental implementado, que possui algumas modificações do original, também foi testado este último no ambiente simulado, ajustando seus parâmetros com os mesmos valores usados para o algoritmo modificado. O resultado de ambas as abordagens é mostrado no gráfico 4.4.

Observa-se que no mapa resultante a partir da implementação do algoritmo Incremental básico (figura 4.4a), algumas linhas não são bem identificadas e são combinadas erroneamente, como é caso dos pares de linhas L1/L5 e L11/L12. Como já foi explicado na seção 3.4.3, onde é descrito o algoritmo de extração de *features*, este fenômeno acontece porque o critério de inclusão de pontos só avalia a distância entre a linha estimada atual e o ponto em análise, e se este último possui a mesma tendência da linha, mas não pertence à mesma, de qualquer modo será identificado como ponto colinear.

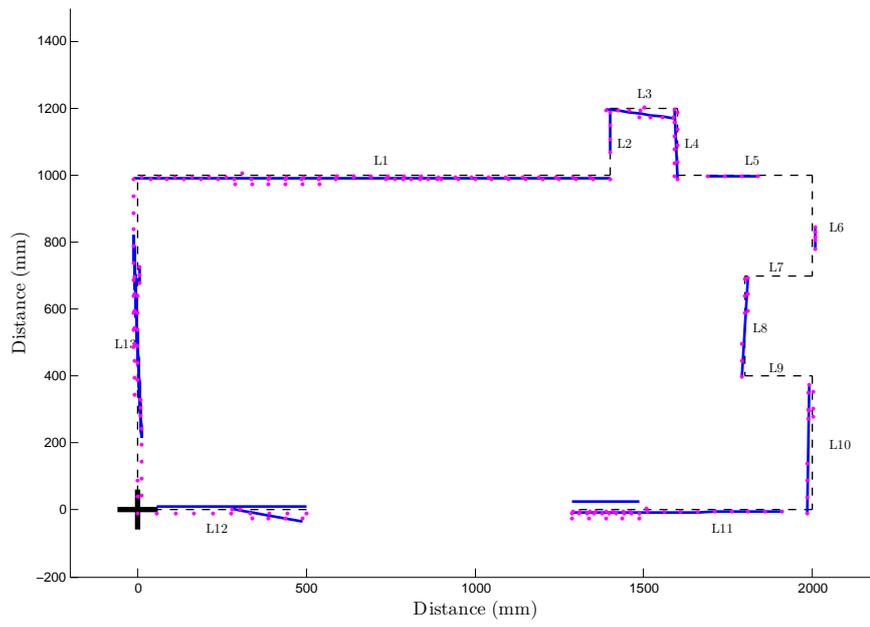
Este também é o caso de linha L8, a qual inclui um ponto que realmente pertence à linha L5 e que fica numa distância linha-ponto que cumpre com o critério de inclusão, mas não pertence à linha. Estas situações não se apresentam no mapa resultante com o algoritmo modificado, onde a condição adicional de distância entre pontos consecutivos, além da distância entre ponto e linha, permite que os pontos colineares sejam bem diferenciados, e desta maneira, as linhas sejam estimadas corretamente.

Na tabela 4.1 encontra-se o cálculo dos erros de estimação para cada parâmetro de cada linha extraída tanto com o algoritmo básico quanto com o modificado. Especificamente para as linhas extraídas com o algoritmo básico, os desvios de cada parâmetro são bastante pequenos, o que reflete o bom desempenho deste tipo de algoritmo, confirmado também no trabalho de *Nguyen et al.* [38], embora algumas linhas não tenham sido bem diferenciadas (as quais são marcadas na tabela com um símbolo especial). Da mesma maneira, nos resultados do algoritmo modificado é possível observar que a maioria dos parâmetros possuem valores pequenos, com exceção das linhas L3 e L8 as quais são as que mais se afastam dos parâmetros das linhas reais.

Por fim, os resultados dos indicadores relacionados com a associação de cada linha estão agrupados na tabela 4.2. Verifica-se que as modificações feitas no algoritmo se refletem no resultado dos *verdadeiros* e dos *falsos positivos*, onde o número do primeiro indicador, comparado com o mesmo do algoritmo clássico, aumentou em aproximadamente 30%, reduzindo nessa mesma porcentagem o número de falsos positivos. A efetividade do processo de *merging* ou *merging* se reflete no último indicador, o número de verdadeiros positivos por linha real, que para ambos os casos é quase um, o que significa que da totalidade de linhas extraídas



(a)



(b)

Figura 4.4: Resultado final do algoritmo incremental básico e a modificação implementada. As linhas azuis correspondem às linhas estimadas, o mapa do ambiente real está desenhado com linhas ponteadas, e as leituras associadas a cada linha estão representadas pelos pontos de cor carmim.

Tabela 4.1: Cálculos dos erros de estimação. As linhas agrupadas com os mesmos símbolos (*, **, †) são aquelas combinadas erroneamente.

| Linha | Parâmetros Reais | | Incremental básico | | Incremental modificado | |
|-------|------------------|--------------------|----------------------------------|-----------------------------------|----------------------------------|-----------------------------------|
| | ρ_k (mm) | α_k (rad\°) | RMSD($\tilde{\rho}_k$) (mm) | RMSD($\tilde{\alpha}_k$) (°) | RMSD($\tilde{\rho}_k$) (mm) | RMSD($\tilde{\alpha}_k$) (°) |
| 1 | 1000 | 1.571\90° | 17* | 0.3* | 10 | 0 |
| 2 | 1400 | 0 \ 0° | 0 | 0 | 0 | 0 |
| 3 | 1200 | 1.571\90° | 26 | 0 | 172 | 7.6 |
| 4 | 1600 | 0 \ 0° | 10 | 0.1 | 41 | 1.8 |
| 5 | 1000 | 1.571\90° | 17* | 0.3* | 3 | 0 |
| 6 | 2000 | 0 \ 0° | 9 | 0 | 9 | 0 |
| 8 | 1800 | 0 \ 0° | 22† | 1.6† | 41 | 3.8 |
| 10 | 2000 | 0 \ 0° | 9 | 0 | 13 | 0.5 |
| 11 | 0 | 1.571\90° | 8** | 0** | 26 | 0.7 |
| 12 | 0 | 1.571\90° | 8** | 0** | 11 | 0 |
| 13 | 0 | 0 \ 0° | 13 | 1 | 14 | 1.6 |

Tabela 4.2: Indicadores de associação

| Indicador | Incremental básico | Incremental modificado |
|----------------------|--------------------|------------------------|
| <i>VerdPos</i> [%] | 64 | 91 |
| <i>FalsoPos</i> [%] | 36 | 9 |
| <i>VerdPos/linha</i> | 1.27 | 1.09 |

por sensor, o processo de *merging* conseguiu agrupar satisfatoriamente todas as linhas que representavam as mesmas estruturas reais.

Dos resultados da simulação e dos indicadores de qualidade, verificou-se a efetividade da estratégia proposta para condições favoráveis de operação devido à ausência de fontes de erros ocasionadas por condições ambientais, tanto da própria plataforma robótica, como o erro na informação da posição do robô.

4.3 VALIDAÇÃO EXPERIMENTAL NA PLATAFORMA NOMAD XR4000

4.3.1 Descrição do ambiente

Como parte final da validação da estratégia de mapeamento proposta, foi executado um teste de exploração em um ambiente real com ajuda da plataforma robótica Nomad. O

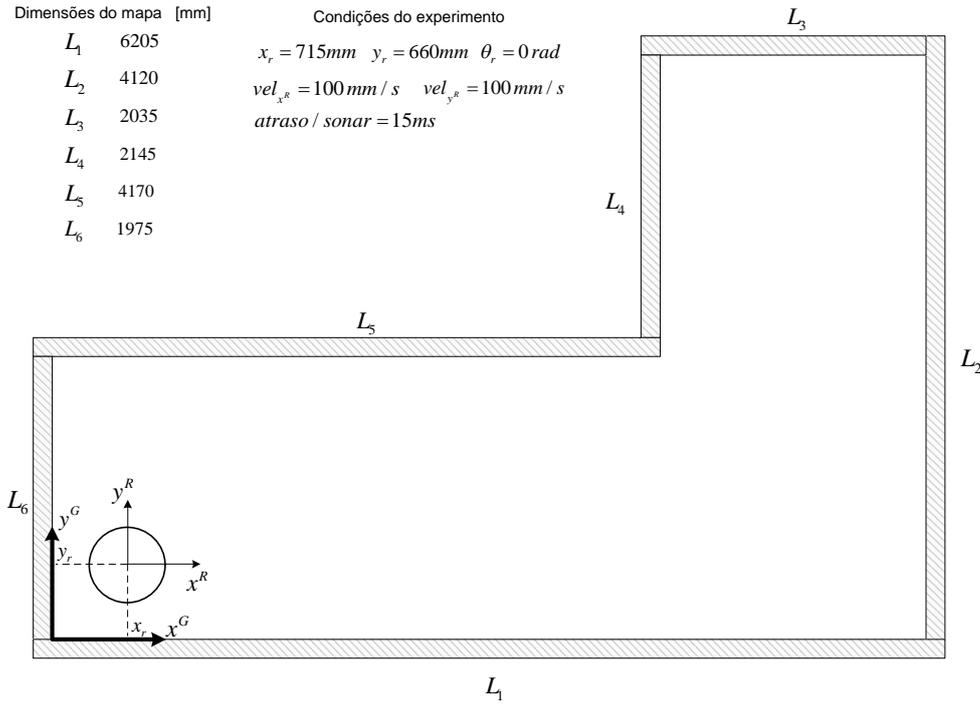


Figura 4.5: Esquema do ambiente real. A referência global é localizada no extremo inferior esquerdo, que coincide com o canto formado pelas paredes L_1 e L_6 . A posição inicial do robô é dada pelo ponto $\{x_r, y_r\}$.

ambiente interno explorado é composto de um grupo de estruturas planas unidas uma a outra formando uma espécie de corredor. As estruturas estão dispostas de forma perpendicular ao chão e estão unidas de maneira tal que não existe risco de inclinarem ou caírem. A figura 4.5 apresenta um esquema que descreve o ambiente real com as dimensões de suas estruturas constitutivas, assim como as condições do experimento. Já na figura 4.6 amostram-se imagens do ambiente real construído.

Em relação ao robô, este foi programado para deslocar-se segundo uma trajetória predefinida. A posição inicial do robô $[x_r \ y_r \ \theta_r]^T$ também é definida antes de iniciar a exploração do ambiente. As velocidades de translação e a velocidade de rotação foram mantidas constantes em todo o percurso, para garantir uma boa tarefa de exploração, assim como diminuir a contribuição do erro de odometria. Por motivos de segurança estabeleceu-se uma distância mínima entre o robô e as estruturas do ambiente, ajustada em 200mm. Os anéis de sensores foram programados para uma sequência de disparo da totalidade dos sonares. Com o objetivo de evitar medições cruzadas² entre sensores adjacentes, foi programado um atraso na sequência de disparo, de maneira que o tempo transcorrido entre disparos consecutivos fosse de 15ms.

²este fenômeno, cujo termo em inglês é *crosstalk* acontece quando um sensor recebe o eco proveniente do disparo do sensor adjacente, fazendo com que o primeiro forneça uma leitura errada [48].

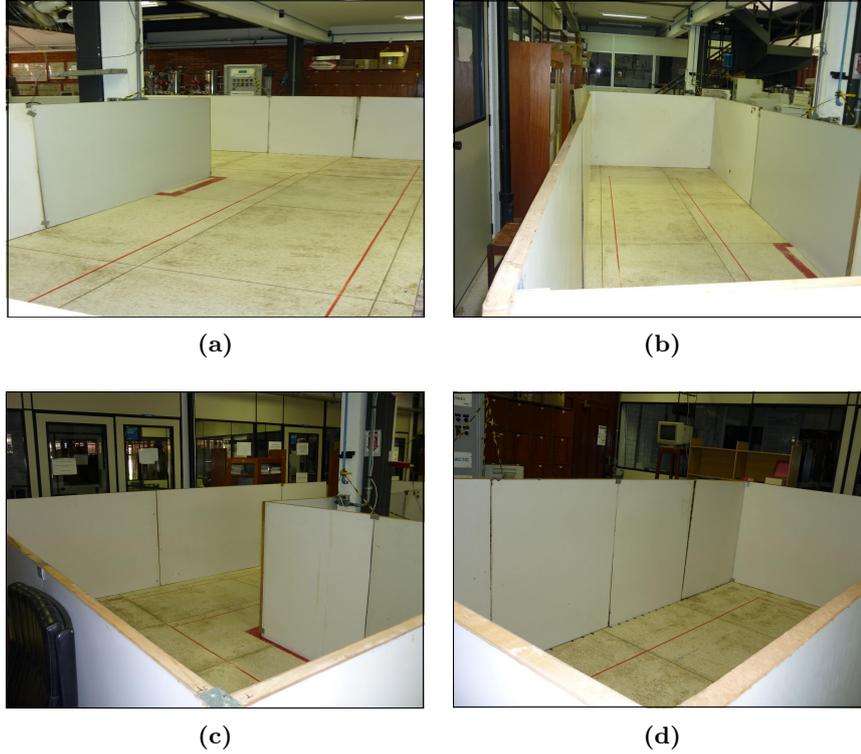


Figura 4.6: Imagens do ambiente real construído.

A parte experimental tem como objetivo avaliar a estratégia de mapeamento em três situações: execução de um trajeto normal, trajeto com presença de objetos dinâmicos, e detecção de discontinuidades entre estruturas a igual profundidade. A seguir serão explicados e apresentados os testes para cada um destes cenários.

4.3.2 Parâmetros iniciais gerais do algoritmo

Para as diferentes situações avaliadas nos testes feitos, os parâmetros iniciais do algoritmo conservam-se, pois as condições gerais do teste são as mesmas, e a configuração da plataforma robótica permanece sem variação de um teste a outro. A tabela 4.3 apresenta os valores para estes parâmetros. Nesta tabela são especificados os valores dos parâmetros relacionados com as etapas de pré-processamento dos dados, extração de segmentos de linhas e *merging*, além de outros valores relacionados com o número mínimo de pontos por linha e a distância de segurança entre o robô e as estruturas do ambiente.

Especificamente, os valores das variâncias $\sigma_{x_R}^2$ $\sigma_{y_R}^2$ correspondentes às medições da posição do robô que foram estimadas a partir de testes de caracterização do erro, são descritos no anexo C. O valor da variância na orientação do robô, $\sigma_{\theta_R}^2$ não é estimado de maneira experimental, pois na trajetória de exploração os deslocamentos só são efetuados nos eixos x_R e y_R . No entanto, foi ajustado um valor pequeno para $\sigma_{\theta_R}^2$ para representar a variância deste eixo na matriz C_{robot} definida na equação 3.10 de seção 3.2.2.

Os erros relacionados com o sensor de ultrassom são também estimados, levando em

Tabela 4.3: Parâmetros iniciais do algoritmo

| Parâmetro | Definição | Valor |
|---|---|----------------------------------|
| $\sigma_{x_R}^2$ [mm ²] | incerteza na medida da posição x do robô | 161.6 |
| $\sigma_{y_R}^2$ [mm ²] | incerteza na medida da posição y do robô | 184.6 |
| $\sigma_{\theta_R}^2$ [rad ²] | incerteza na medida da posição θ do robô | 0.01 (aprox. (5°) ²) |
| σ_d^2 [mm ²] | incerteza na medida da distância do sensor de ultrassom | 1600 |
| σ_ϕ^2 [mm ²] | incerteza na estimação da orientação do obstáculo com respeito à localização do sensor de ultrassom | $\beta/6$ |
| D_{max_1} [mm] | Distância euclideana entre pontos consecutivos | 300 |
| D_{max_2} [mm] | Distância entre ponto e linha estimada | $D(C_{\hat{\theta}})$ |
| D_{max_3} [mm] | Distância entre ponto inicial e final de linhas comparadas na etapa de <i>merging</i> | 1000 |
| - [mm] | Distância mínima de segurança entre robô e obstáculos | 200 |
| - | Número mínimo de pontos que compõem uma linha estimada | 15 |

consideração a disposição dos sensores no plataforma robótica. Isto está explicado em detalhe no anexo B. De aqui é ajustado um valor para σ_d , e em consequência para σ_d^2 .

O limiar D_{max_1} relacionado com a distância entre medições consecutivas usado na etapa de extração de segmentos de linhas, está relacionado com a distância percorrida pelo robô entre medições consecutivas de um mesmo sensor, a qual é extraída da informação proveniente do módulo de odometria, além de levar em consideração as incertezas do sensor de ultrassom e da medição de posição dada por este módulo. Já para o limiar D_{max_2} que define a máxima distância entre linha estimada e ponto analisado, este depende da incerteza nos parâmetros da linha em questão, é por isso que seu valor não é fixo e depende de cada linha estimada, tal como se mostra na tabela 4.3.

4.3.3 Experimento 1

Neste primeiro experimento o robô percorre uma trajetória predefinida, cuja sequência é $L1L2L3L4L5$, como se mostra na figura 4.7. A posição inicial do robô corresponde aquela especificada na figura 4.5. O tempo total de duração do teste foi de 4.02 s, onde coletaram-se 237 medições por sensor, para um total de 1896 leituras por anel de sonares. Este conjunto de medições foi pré-processado de acordo com o explicado na seção 3.3.

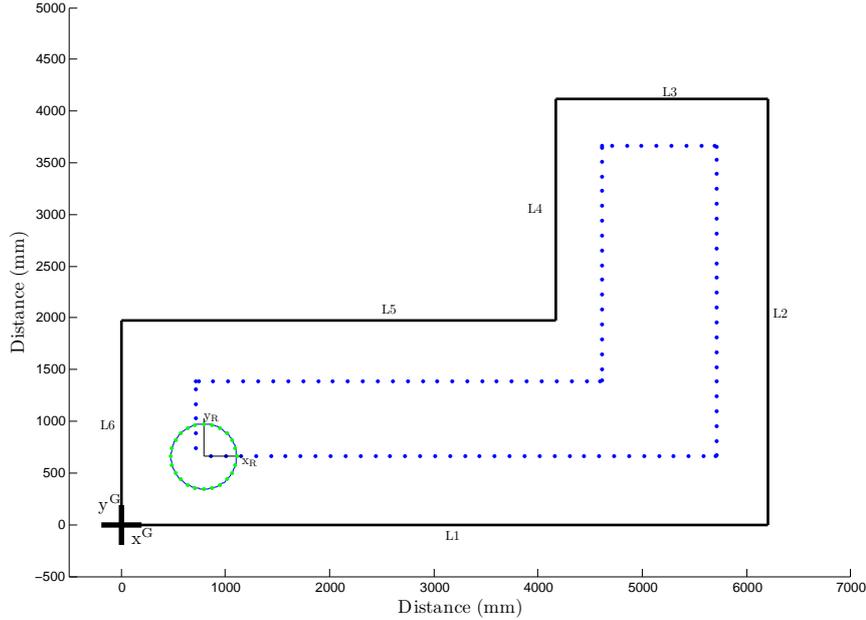


Figura 4.7: Representação do ambiente real no algoritmo programado. Os pontos azuis representam cada uma das posições do robô durante a trajetória executada.

Previamente à transformação em coordenadas cartesianas e à compensação dos erros dados pelo próprio sensor e sua configuração, na etapa de pré-processamento é realizada uma filtragem das leituras segundo o máximo valor de distância possível detectado pelo sensor, que corresponde a 7000 mm. Se qualquer medição d_n supera este valor, esta será rejeitada. A figura 4.8 mostra os dados depois da etapa pré-processamento.

Nota-se que apesar da compensação feita a partir do modelo de medição, explicado na seção 3.1.2, e do filtro para excluir leituras que superassem o alcance do sonar usado, ainda existe uma presença importante de leituras espúrias. Além disto, são bastante notáveis os efeitos do erro de odometria, que faz com que a posição do robô não siga a trajetória programada, apreciando-se uma orientação no sentido horário de todas as leituras com respeito ao sistema coordenadas global. Contudo, é possível ver que todas as estruturas foram observadas pelo sistema sensorial do robô, e a que a transformação de todas as medições em coordenadas cartesianas foi bem sucedida.

Como uma tentativa de melhorar a orientação geral destas medições, foi efetuada uma compensação dos dados de odometria usando os valores calculados para os erros sistemáticos que aparecem descritos no anexo C. Estes valores correspondem a $\varepsilon_{s,x} = 13$ mm e $\varepsilon_{s,y} = 16$ mm. A partir disto, foram compensado os valores de cada trajeto do percurso total para o experimento. O resultado desta operação é mostrada na figura 4.9, onde é possível ver uma diminuição importante da influência do erro de odometria. A partir desta melhora, é executado o resto do algoritmo. A compensação é realizada também para os demais experimentos.

Na sequência, a etapa de extração de segmentos de linhas é executada sonar por sonar, extraindo as linhas que cada sensor tem detectado. O número total de linhas nesta etapa é de 150, o que corresponde a aproximadamente 7 linhas por sonar, considerando as leituras dos

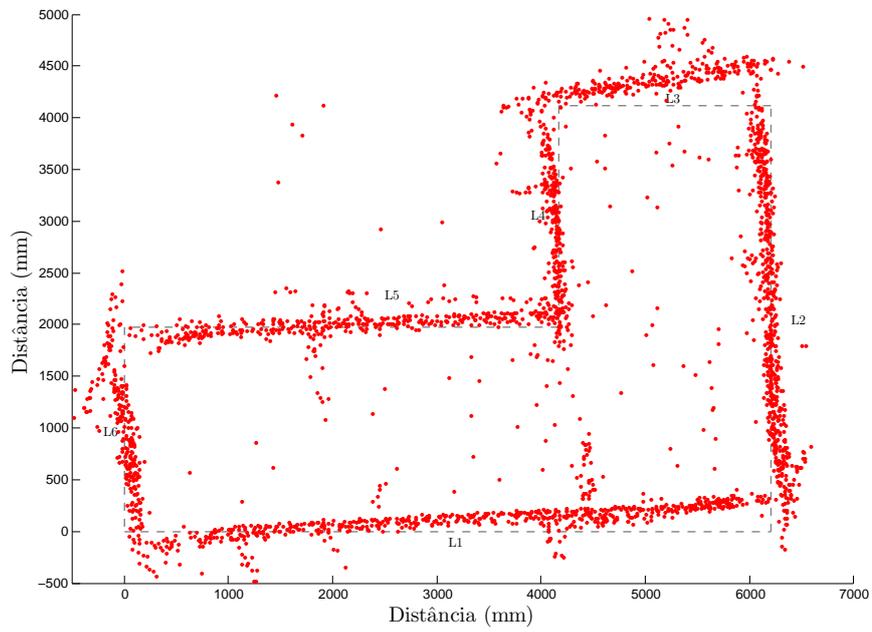


Figura 4.8: Saída do algoritmo após da etapa de pré-processamento das medições.

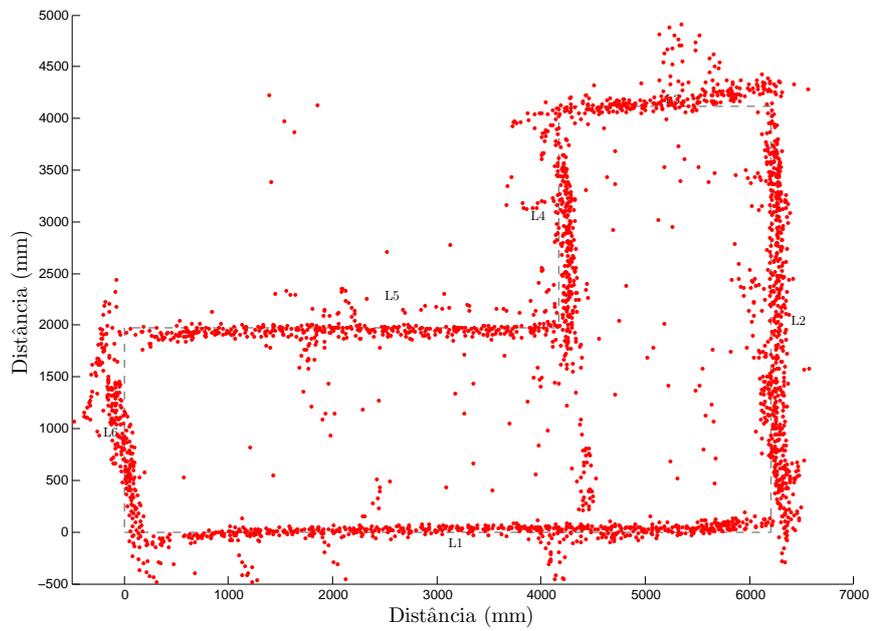


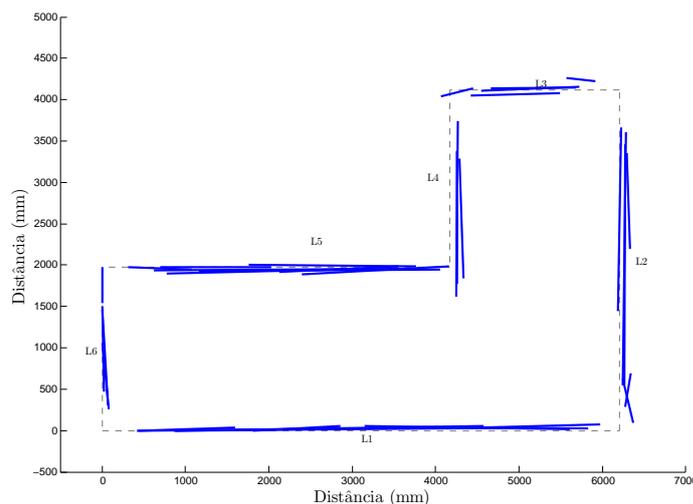
Figura 4.9: Dados após da compensação dos dados de odometria.

24 sonares correspondentes a um anel. Isto é um bom resultado levando em consideração que o número total de linhas reais a serem detectadas é 6.

Por fim, o processo de *merging* é levado a cabo. Para esta etapa foi definido um número mínimo de pontos por linha, de maneira que aquelas que possuem um número inferior a este valor no final da execução desta etapa serão rejeitadas definitivamente. O número final de linhas resultantes é de 35. O resultado final encontra-se na figura 4.10.

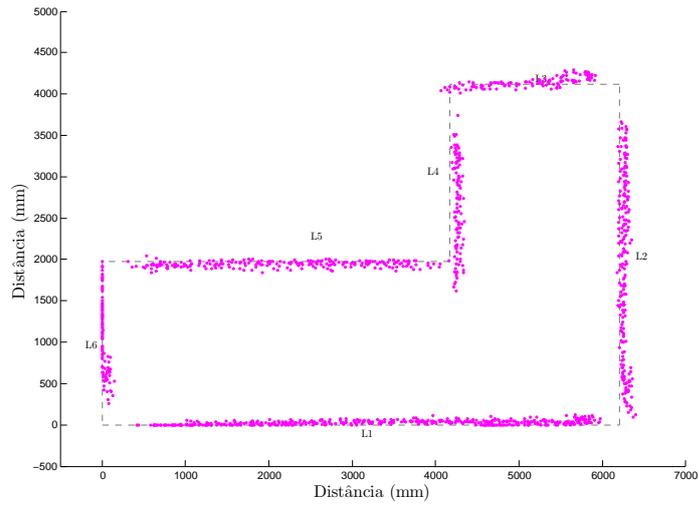
É possível observar que apesar da presença de dados espúrios no conjunto de leituras dos sensores, o algoritmo conseguiu diferenciar bem todas as estruturas reais do ambiente. Por outro lado, a dispersão dos dados que pertencem a cada uma das linhas reais, faz com que sejam identificadas várias linhas por parede, algumas destas bastante diferentes da linha real. Com o objetivo de avaliar quantitativamente o desempenho do algoritmo para este experimento, antes do cálculo do erro de estimação e dos indicadores de associação, serão definidas primeiro as linhas que serão consideradas como representativas a cada parede. Uma linha pode ser considerada como correspondente a uma linha real se a diferença absoluta entre seus parâmetros e os parâmetros reais não supera $\Delta_\rho = 300$ mm e $\Delta_\alpha = 0.17$ rad. Desta maneira aquelas que cumpram com isto serão usadas para os cálculos do erro de estimação. Já as outras linhas serão levadas em consideração nos indicadores de associação.

Nas tabelas 4.4 e 4.5, encontram-se os valores dos erros de estimação e dos indicadores de associação, respectivamente. Da tabela 4.4 é possível observar que as estimações com maior erro foram as correspondentes às paredes L1 e L6, já as demais linhas possuem erros que estão na faixa dos 76 mm para ρ e de 0.9° para α . No caso dos indicadores de associação, estes mostram um número de verdadeiros positivos que supera amplamente os falso positivos detectados, o que reflete o bom desempenho do algoritmo frente à presença moderada de dados espúrios.



(a) Linhas finais extraídas

Figura 4.10: Resultado final de estratégia de mapeamento para o experimento 1.



(b) Pontos pertencentes às linhas finais

Figura 4.10: Resultado final da estratégia de mapeamento para o teste 1.

Tabela 4.4: Cálculo dos erros de estimação para o experimento 1

| Linha | Parâmetros Reais | | RMSD ($\bar{\rho}_k$) (mm) | RMSD($\bar{\alpha}_k$) (°) |
|-------|------------------|----------------------------|---------------------------------|---------------------------------|
| | ρ_k | α_k | | |
| 1 | 0 | $1.571 \setminus 90^\circ$ | 100 | 2 |
| 2 | 6205 | $0 \setminus 0^\circ$ | 26 | 0.7 |
| 3 | 4120 | $1.571 \setminus 90^\circ$ | 73 | 0.9 |
| 4 | 4170 | $0 \setminus 0^\circ$ | 76 | 0.1 |
| 5 | 1975 | $1.571 \setminus 90^\circ$ | 57 | 0.6 |
| 6 | 0 | $0 \setminus 0^\circ$ | 105 | 8 |

Tabela 4.5: Indicadores de associação

| Indicador | Valor |
|----------------------|-------|
| <i>VerdPos</i> [%] | 70 |
| <i>FaloPos</i> [%] | 30 |
| <i>VerdPos/linha</i> | 4 |

4.3.4 Experimento 2

Neste teste o robô percorre a mesma trajetória feita no teste 1, mas com a presença de um obstáculo dinâmico no ambiente, neste caso, uma pessoa. O indivíduo percorre o mesmo ambiente aproximando-se em várias ocasiões ao robô (figura 4.11), muitas vezes bloqueando o espaço entre o robô e as diferentes paredes. Os dados pré-processados, assim como os dados finais, junto com sua representação em forma de linhas são mostrados na figura 4.12.

Nota-se que a presença de um obstáculo dinâmico no ambiente faz com que o conjunto de dados possua uma quantidade importante de leituras espúrias, as quais influenciam em todo o processo de extração e posterior combinação de linhas. O algoritmo só consegue identificar 3 estruturas, com muito poucos pontos cada uma, o que pôde ser ocasionado pela alta dispersão dos dados, fazendo com que o algoritmo rejeite a maioria deles, conservando só aqueles com menor variância. Outra razão deste resultado pode ser pela diferença de velocidades entre a pessoa que esta-se movimentando e o disparo dos sensores, que faz como que estes não detectem o obstáculo dinâmico.

O baixo desempenho do algoritmo para este experimento se reflete nos resultados das tabelas 4.6 e 4.7, onde os erros de ambos parâmetros são relativamente altos. Já para o caso dos indicadores de associação parecesse que estes tivessem uma melhora com respeito aos mesmos resultados do experimento anterior, mas este comportamento enganoso deve-se ao número reduzido de linhas detectadas (7), as quais correspondem à metade das estruturas reais. Conclui-se então, que a presença de um obstáculo dinâmico afeita consideravelmente o desempenho da estratégia de mapeamento implementada.

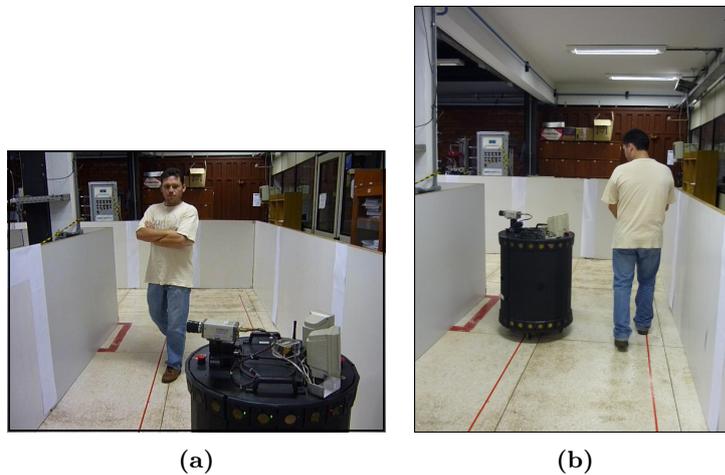
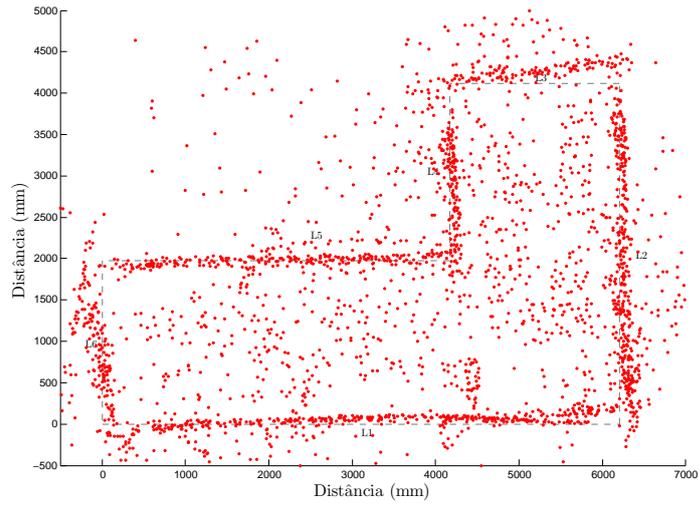
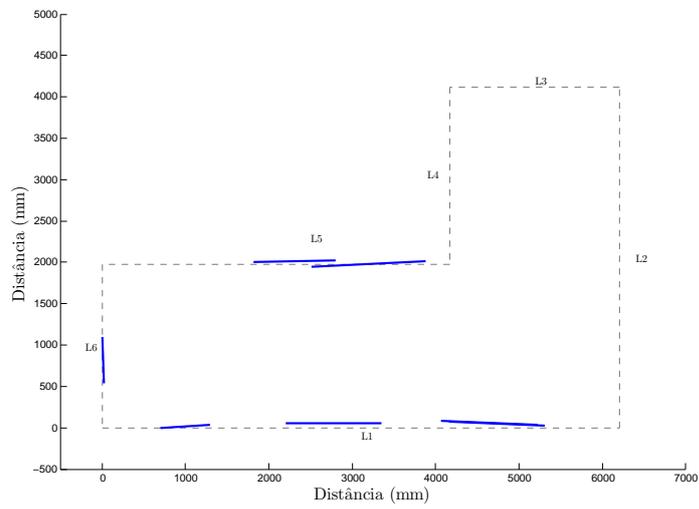


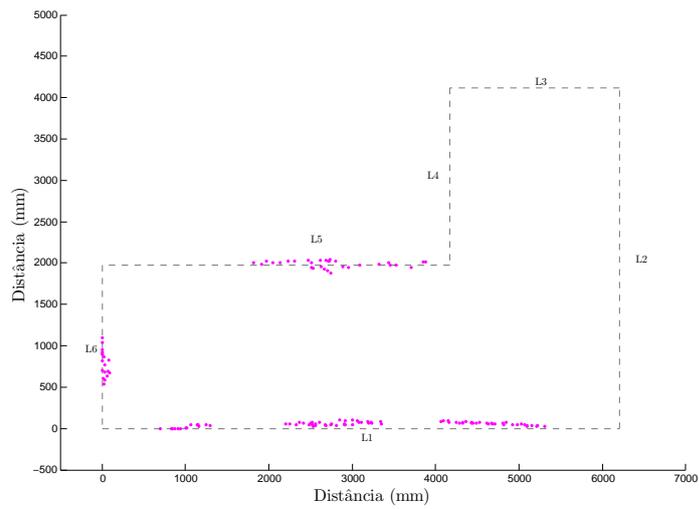
Figura 4.11: Imagens da execução do teste 2.



(a) Conjunto de dados resultantes da execução do teste



(b) Representação das linhas extraídas



(c) Pontos correspondentes às linhas detectadas

Figura 4.12: Resultado final da estratégia de mapeamento para o teste 2.

Tabela 4.6: Cálculo dos erros de estimação para o experimento 2

| Linha | Parâmetros Reais | | RMSD ($\bar{\rho}_k$) (mm) | RMSD($\bar{\alpha}_k$) (°) |
|-------|------------------|------------|---------------------------------|---------------------------------|
| | ρ_k | α_k | | |
| 1 | 0 | 1.571\90° | 220 | 2.2 |
| 2 | 6205 | 0 \ 0° | - | - |
| 3 | 4120 | 1.571\90° | - | - |
| 4 | 4170 | 0 \ 0° | - | - |
| 5 | 1975 | 1.571\90° | 130 | 2.5 |
| 6 | 0 | 0 \ 0° | 136 | 9.5 |

Tabela 4.7: Indicadores de associação

| Indicador | Valor |
|----------------------|-------|
| <i>VerdPos</i> [%] | 86 |
| <i>FaloPos</i> [%] | 14 |
| <i>VerdPos/linha</i> | 2 |

4.3.5 Experimento 3

Este último experimento é realizado com o objetivo de avaliar a capacidade do algoritmo implementado de detectar a presença de uma descontinuidade importante entre paredes de igual profundidade, isto é, paredes que estão a mesma distância do robô e, portanto possuem os mesmos parâmetros. Este resultado pode ser fundamental no caso em que este espaço represente uma porta aberta pela qual o robô deve passar numa determinada tarefa de navegação.

Para simular este espaço, foi retirada uma das estruturas que faziam parte de parede L_2 deixando um espaço de aproximadamente 1000 mm de comprimento, tal como aparece na figura 4.13. O resto do ambiente ficou igual, assim como a trajetória executada pelo robô.

Os resultados gráficos amostram-se na figura 4.14, onde é possível verificar que efetivamente o algoritmo conseguiu detectar satisfatoriamente o espaço gerado na antiga parede L_2 , assim como as demais paredes. No caso dos indicadores de desempenho apresentados nas tabelas 4.8 e 4.9, estes mostram que os parâmetros das linhas extraídas possuem um comportamento similar a aqueles estimados no primeiro experimento. O número de medições provenientes do sistema sensorial é de 5640, das quais 1251 (22% da totalidade) correspondem aos pontos pertencentes às 34 linhas finalmente extraídas.

De forma similar que no experimento 1, o número de verdadeiros positivos supera amplamente um número de linhas constituídas por dados espúrios, mantendo-se também um número baixo de verdadeiros positivos por linha real, o que reflete o desempenho satisfatório do processo de *merging*.

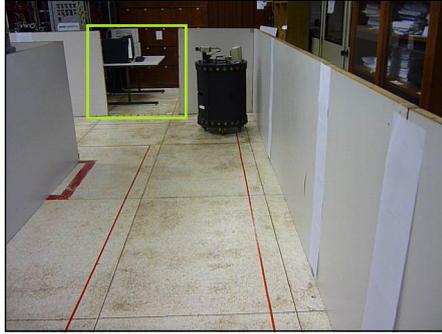


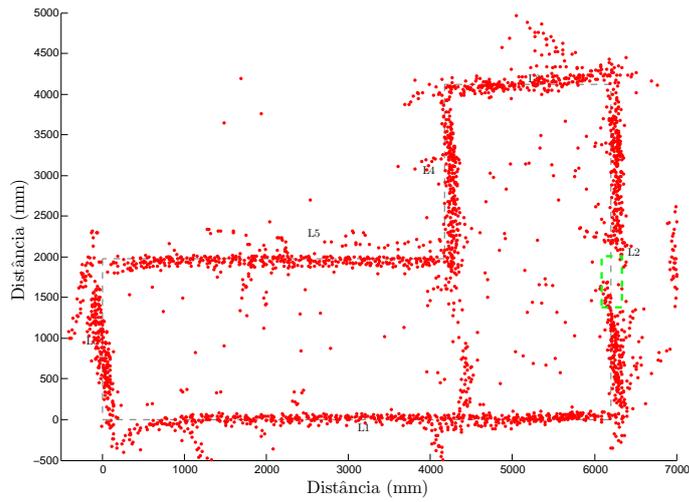
Figura 4.13: Ambiente modificado para a execução do teste 3. No quadrado verde ressaltase a presença do espaço feito na parede L_2 .

Tabela 4.8: Cálculo dos erros de estimação para o experimento 3

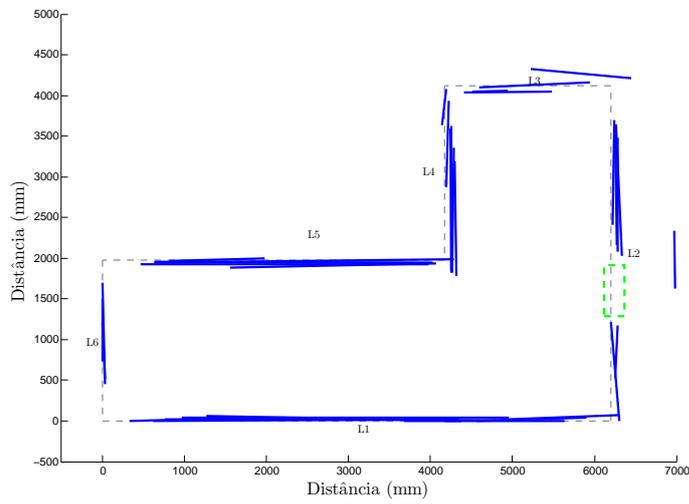
| Linha | Parâmetros Reais | | RMSD ($\bar{\rho}_k$) (mm) | RMSD($\bar{\alpha}_k$) (°) |
|-------|------------------|----------------------------|---------------------------------|---------------------------------|
| | ρ_k | α_k | | |
| 1 | 0 | $1.571 \setminus 90^\circ$ | 125 | 2.3 |
| 2 | 6205 | $0 \setminus 0^\circ$ | 60 | 4.8 |
| 3 | 4120 | $1.571 \setminus 90^\circ$ | 46 | 1.3 |
| 4 | 4170 | $0 \setminus 0^\circ$ | 121 | 1 |
| 5 | 1975 | $1.571 \setminus 90^\circ$ | 42 | 0.5 |
| 6 | 0 | $0 \setminus 0^\circ$ | 120 | 6.8 |

Tabela 4.9: Indicadores de associação

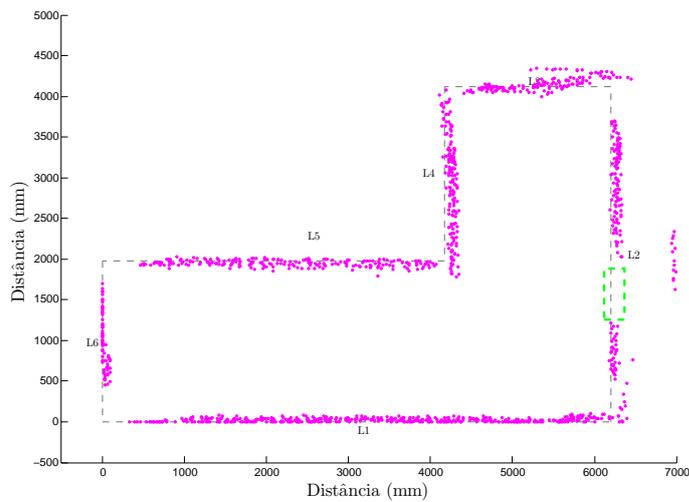
| Indicador | Valor |
|----------------------|-------|
| <i>VerdPos</i> [%] | 74 |
| <i>FaloPos</i> [%] | 26 |
| <i>VerdPos/linha</i> | 4 |



(a) Conjunto de dados resultantes da execução do teste.



(b) Representação das linhas extraídas.



(c) Pontos correspondentes às linhas detectadas.

Figura 4.14: Resultados da estratégia de mapeamento para o teste 3.

4.4 DISCUSSÃO FINAL DOS RESULTADOS

Tanto nos testes no ambiente simulado como no ambiente real, foi possível constatar a efetividade da estratégia proposta para extração de segmentos de linhas e a posterior construção de uma representação global do ambiente explorado.

Especificamente para o caso da validação experimental com a plataforma robótica, observou-se por meio dos resultados gráficos que a informação dada pelo conjunto de sensores e pelo módulo de odometria foi analisada satisfatoriamente, conseguindo extrair a informação relevante do conjunto de medições, assim como uma rejeição da boa parte dos dados espúrios provenientes das diferentes fontes de erro do ambiente e da próprio funcionamento de ambos os tipos de sensores.

Das três situações propostas no ambiente real, o algoritmo teve um bom comportamento nos experimentos 1 e 3, com desempenhos muito similares em termos de número final de linhas extraídas, valores dos erros de estimação dos parâmetros e indicadores de associação. Já no experimento 2, comprovou-se que a estratégia tem um comportamento muito deficiente em ambientes internos com objetos dinâmicos.

4.4.1 Comparação do desempenho da estratégia com outros trabalhos

Da revisão dos trabalhos relacionados com extração de segmentos de linhas para a construção de mapas, é possível observar que a maioria destes possuem diferentes indicadores (muitos deles propostos pelos mesmos autores) para avaliar o desempenho de suas estratégias, as vezes estreitamente relacionados com as características específicas de cada ambiente estudado, o que dificulta um pouco a comparação entre trabalhos.

Apesar disto, é possível extrair algumas características gerais de alguns trabalhos. Por exemplo, dos resultados expostos no trabalho de *Nguyen et. al* [38], onde é feita a comparação de seis algoritmos para extração de linhas, entre estes o algoritmo incremental usado aqui, é possível confirmar a porcentagem de verdadeiros positivos para este algoritmo, a qual está na ordem de 78%, muito perto do 70% e 74% conseguidos nos resultados apresentados neste trabalho.

Já no caso dos erros de estimação dos parâmetros, os autores apresentam um indicador único para cada parâmetro, o qual é o desvio padrão de todas as linhas extraídas com respeito aos parâmetros reais. Mesmo que neste trabalho o erro foi calculado individualmente para cada linha, pode ser feita uma comparação entre os valores dos erros apresentados em [38] ($\sigma_{\Delta\rho}$ e $\sigma_{\Delta\alpha}$), com a média dos erros calculados para cada parâmetro de cada linha detectada, quer dizer, o valor médio para $RMSD(\rho)$ e para $RMSD(\alpha)$ apresentados nas seções anteriores. Ambos os resultados são mostrados na tabela 4.10, junto com o número de verdadeiros positivos de cada abordagem, os quais foram discutidos inicialmente.

Comparando os erros de cada parâmetro é possível ver que os valores correspondentes à estratégia proposta são maiores que os erros calculados em [38]. Estas diferenças estão muito

relacionadas com as características do sensor utilizado, que para o trabalho em referência trata-se de um sensor laser com características, como resolução de alcance e precisão, superiores a um sensor de ultrassom. Apesar disto, os valores dos erros de estimação dos parâmetros das linhas extraídas não são tão grandes se comparados com as dimensões de estruturas reais usadas nos testes, as quais superam os 2 m de comprimento.

Tabela 4.10: Comparação do resultado da estratégia proposta com o trabalho de [38]

| Indicador | Estratégia proposta | Trabalho de <i>Nguyen et. al</i> |
|-------------------------------|----------------------------|---|
| <i>VerdPos</i> [%] | 72 | 78 |
| Erro em ρ (mm) | 70 | 20 |
| Erro em α ($^\circ$) | 2 | 0.7 |

5 CONCLUSÕES

Este trabalho apresenta uma estratégia de mapeamento para robôs móveis baseada na extração de segmento de linhas, aplicada à exploração e navegação de ambientes internos. O sistema sensorial utilizado compõe-se de um conjunto de sensores de ultrassom em configuração de anel, assim como de um módulo de odometria. A partir destes dados é construída uma representação em uma referência global do ambiente explorado pela plataforma robótica.

Foi feito um estudo preliminar da resposta do sensor de ultrassom para a detecção de estruturas planas a partir da análise realizado no trabalho de Barshan e Kuc [41]. Como resultado, conclui-se que os erros correspondentes a cada sonar variam segundo a posição em que estes estão com respeito ao alvo atingido, levando em consideração a configuração circular do sistema sensorial. De maneira específica, este resultado ajudou na construção do um modelo de medição para o sensor de ultrassom, em que o erro de cada medição foi compensado de maneira específica segundo a localização do sensor do qual provém a leitura.

Na etapa de pré-processamento, a transformação das medições de distância de cada sensor em pontos em coordenadas cartesianas dentro de uma referência global, facilita substancialmente a comparação das medições entre diferentes sensores. Além disto, o cálculo das matrizes de covariâncias correspondentes a cada dado, a partir da propagação do erro, permite obter uma medida de dispersão dos dados.

O algoritmo de extração de linhas, baseado no algoritmo Incremental, avalia a distância euclidiana entre pontos consecutivos como condição adicional para a inclusão de pontos dentro da estimação de um segmento de linha. Sem esta avaliação inicial, corre-se o risco de incluir dados espúrios dentro do conjunto de pontos que definem determinado segmento, ou combinar pontos de linhas adjacentes. A avaliação da distância entre pontos analisados e linha estimada, presente no algoritmo clássico, é conservada, relacionando esta com o cálculo da covariância própria dos parâmetros que definem cada linha, representada por meio do modelo *Hessiano*. Desta maneira, é considerada no critério de inclusão de pontos a incerteza própria da estimação dos parâmetros de cada linha.

A etapa final correspondente ao processo de *merging* ou combinação, utiliza dois critérios para agrupação de linhas. O primeiro é um teste Chi-Quadrado, o qual define um intervalo de confiança de aproximadamente um 95% para as linhas analisadas a partir da combinação de suas matrizes de covariância e da diferença de seus parâmetros. Este critério mede a similitude das linhas em termos dos parâmetros que as definem. O segundo critério tem como objetivo a análise dos pontos que compõem cada linha, estabelecendo a presença de relações entre

ambos os conjuntos. Como resultado final, as linhas que possuam semelhança em termos de parâmetros e de pontos constitutivos são agrupadas em uma única linha.

O algoritmo proposto foi avaliado tanto em um ambiente simulado, como em um ambiente real usando uma plataforma robótica. Nos experimentos e estudos de casos conduzidos, a única informação prévia fornecida ao robô é sua posição inicial no começo da estratégia de exploração. A forma das estruturas que constituem o ambiente, assim como suas posições são completamente desconhecidas. Os resultados obtidos validam o método e o algoritmo proposto, onde a partir dos dados dos sensores de ultrassom e da informação fornecida pelo módulo de odometria do robô, é possível obter uma representação aproximada do ambiente explorado.

Os três experimentos com a plataforma Nomad em um ambiente real têm como objetivo avaliar o comportamento da estratégia diante de diferentes situações. Em todos os estudos de caso foi possível perceber a influência do erro de odometria nas medições, assim como da incerteza angular dos sensores de ultrassom, fazendo com que os dados apresentassem em todos os casos uma notória dispersão. Ainda assim, as representações obtidas são coerentes com a realidade do ambiente explorado.

O primeiro experimento avalia o comportamento do algoritmo numa situação normal, onde não existe presença de agentes perturbadores no ambiente. Aqui, o resultado confirma que o algoritmo é capaz de identificar todas as estruturas de maneira satisfatória, com erros nos parâmetros que estão na faixa de 70 mm para $\hat{\rho}$ e 2° para $\hat{\alpha}$.

Os resultados do segundo experimento, onde o robô explora o ambiente junto com a presença de um obstáculo dinâmico, mostram que o desempenho da estratégia se vê comprometido. Se bem que o algoritmo conseguiu identificar algumas das estruturas, os pontos que as definem são insuficientes para representá-las em um mapa final. Isto é ocasionado justamente pela quantidade de dados espúrios originados pela presença do obstáculo dinâmico.

No último experimento da avaliação experimental, o ambiente possui uma descontinuidade entre paredes de igual profundidade, isto é, os parâmetros que as definem são as mesmas, mais os pontos que as constituem são diferentes. Nesta situação o algoritmo implementado conseguiu diferenciar as duas estruturas, identificando o espaço entre elas.

Por fim, por meio de uma pequena comparação de resultados com outro trabalho que aborda o problema de extração de segmentos de linhas usando o algoritmo incremental, verificou-se que o desempenho do algoritmo aqui proposto é adequado. A partir desta comparação, e considerando que o trabalho tomado como referência usa outro tipo de sensor mais preciso, foi possível demonstrar que o uso de sensores de ultrassom na tarefa de mapeamento de um ambiente interno, condicionado a uma adequada modelagem dos dados produzidos por estes sensores, pode gerar um mapa confiável que pode ser utilizado em tarefas subsequentes de navegação.

5.1 PROPOSTAS PARA TRABALHOS FUTUROS

As propostas de trabalhos futuros visam melhorar o nível de robustez da estratégia proposta diante de diferentes condições de ambiente, assim como dos erros que afetam os sensores utilizados.

- **Estimar a posição do robô durante a trajetória de exploração.** O problema de localização é fundamental para obter uma adequada informação sobre a posição real do robô dentro do ambiente. A estimação da posição do robô na medida em que este explora o ambiente, e a correção da trajetória a partir desta informação, melhora a qualidade dos dados obtidos pelos sensores de ultrassom. Vários trabalhos têm proposto estratégias para resolver o problema de localização fusionando os dados provenientes do módulo de odometria com a informação fornecida por outros sensores proprioceptivos, como giroscópios ou compasses magnéticos [49].
- **Mapeamento Dinâmico.** Para melhorar os resultados da estratégia proposta diante da presença de obstáculos dinâmicos, propõe-se estudar o desempenho dos sensores de ultrassom na detecção deste tipo de objetos. Especificamente deverão ser estudados aspectos como a velocidade de disparo do sensor em relação à velocidade em que se movimenta o obstáculo, assim como o retardo que deverão ser programado entre sensores adjacentes com o objetivo de evitar o fenômeno de *crosstalk*.
- **Localização baseada em mapas.** Com o objetivo de melhorar as estimativas das linhas extraídas no processo de mapeamento, pode-se implementar uma etapa posterior de localização, onde a informação *a priori* seja o mapa construído por meio da estratégia aqui implementada, e a partir das estruturas já identificadas, das informações de odometria e dos sonares, possa-se estimar as medições de posição onde o robô detectou essas estruturas e assim obter a localização do robô dentro do mapa. Esta estimação pode ser calculada por meio de técnicas de filtragem estocástica como o *Filtro de Kalman Estendido*.
- **Ampliar os tipos de *features* a serem detectadas.** Os ambientes internos possuem outros tipos de estruturas como cantos, bordas, e objetos curvos que não podem ser representados a partir de segmentos de linhas. A modelagem e posterior detecção destas estruturas enriquecem enormemente o mapa resultante, assemelhando-o ainda mais com o ambiente real.
- **Fusão sensorial.** O uso de outro tipo de sensores, como por exemplo a câmera instalada na plataforma robótica utilizada para este trabalho, pode fornecer informação complementar adicional sobre o ambiente a ser explorado. A fusão destas informações poderia criar um mapa mais completo e mais exato.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, 2006.
- [2] D. C. Montgomery and G. C. Runger, *Applied Statistics and Probability for Engineers*. John Wiley & Sons, Inc., 2003.
- [3] G. Campion, G. Bastin, and B. D’Andréa-Novel, “Structural properties and classification of kinematics and dynamic models of wheeled mobile robots,” *IEEE Transactions on Robotics and Automation*, vol. 12, no. 1, pp. 47–62, february 1996.
- [4] L. Ljung, *System Identification*. Prentice Hall, 1999.
- [5] S. V. Huffel and J. Vandewalle, *The total least squares problem, computational aspects and analysis*. SIAM, 1991.
- [6] P. Huber, *Robust Statistics*. John Wiley, 1981.
- [7] C. E. V. Cano, “Técnica de navegação de um robô móvel baseado em um sistema de visão para integrá-lo a uma célula flexível de manufatura,” Dissertação de Mestrado em Sistemas Mecatrônicos, Departamento de Engenharia Mecânica, Universidade de Brasília, 2006.
- [8] S. R. G. Tourino, “Guiagem do robô móvel XR4000 para inspeção via internet de tubulações industriais soldadas,” Relatório de projeto de graduação II, Departamento de Engenharia Mecânica, Universidade de Brasília, 2000.
- [9] M. M. de Sousa and P. R. C. Dutra, “Navegação do robô móvel nomad XR4000 baseada em mapeamento dinâmico e estático,” Relatório de projeto de graduação II, Departamento de Engenharia Mecânica, Universidade de Brasília, 2003.
- [10] G. G. D. Souza, “Estudo e aplicação do sonar na robótica móvel,” Dissertação de Mestrado, Departamento de Engenharia Mecânica, Universidade de Brasília, 2003.
- [11] N. Technologies, *Nomad XR4000 Hardware Manual Release 1.0*, 1999.
- [12] R. Kuc and M. W. Siegel, “Physically based simulation model for acoustic sensor robot navigation,” *Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, no. 6, pp. 766–778, 1987.

- [13] J. Crowley, “World modeling and position estimation for a mobile robot using ultrasonic ranging,” *IEEE Conference on Robotics and Automation, ICRA 89*, vol. 3, pp. 1574–1579, may 1989.
- [14] J. J. Leonard and H. F. Durrant-Whyte, *Direct Sonar Sensing for Mobile Robot Navigation*. Kluwer Academic Publishers, 1992.
- [15] S. J. Lee, D. W. Cho, W. K. Chung, J. H. Lim, and C. U. Kang, “Feature based map building using sparse sonar data,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 492–496.
- [16] H.-M. Wang, Z.-G. Hou, J. Ma, Y.-C. Zhang, Y.-Q. Zhang, and M. Tan, “Sonar feature map building for a mobile robot,” in *IEEE International Conference on Robotics and Automation*, april 2007, pp. 4152–4157.
- [17] R. Siegwart and I. Nourbakhsh, *Introduction to Autonomous Mobile Robots*. The MIT Press, 2004.
- [18] T. Nierobisch, W. Fischer, and F. Hoffman, “Large view visual servoing of a mobile robot with a pant-tilt camera,” in *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 3307–3312.
- [19] H. P. Moravec and A. Elfes, “High resolution maps from wide angle sonar,” in *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, 1985, pp. 116–121.
- [20] J. A. Castellanos, J. Neira, O. Strauss, and J. D. Tardós, “Detecting high level features for mobile robot localization,” in *Proceedings of the 1996 IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 1996, pp. 611–618.
- [21] V. Koval, O. Adamiv, and V. Kapura, “The local area map building for mobile robot navigation using ultrasound and infrared sensors,” in *IEEE International Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications*, 2007, pp. 454–459.
- [22] T. R. Kurfess, Ed., *Robotics and Automation Handbook*. CRC Press, 2005.
- [23] H.-J. von der Hardt, D. Wolf, and R. Husson, “The dead reckoning localization system of the wheeled mobile robot romane,” in *Proceedings of the 1996 IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 1996, pp. 603–610.
- [24] “Sonar,” Wikipédia, a enciclopédia livre, Tech. Rep. [Online]. Available: <http://pt.wikipedia.org/wiki/Sonar>
- [25] “Polaroid 6500 ranging system,” SensComp, Inc, Tech. Rep. [Online]. Available: <http://www.senscomp.com>

- [26] S. Thrun, “Robotic mapping: A survey,” in *Exploring Artificial Intelligence in the New Millennium*, G. Lakemeyer and B. Nebel, Eds. Morgan Kaufmann, 2002.
- [27] M. J. Mataric, “A distributed model for mobile robot environment-learning and navigation,” Dissertação de Mestrado, MIT, Cambridge MA, 1990.
- [28] B. Kuipers and Y.-T. Byun, “A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations,” *Journal of Robotics and Autonomous Systems*, vol. 8, pp. 47–63, 1991.
- [29] A. Elfes, “Occupancy grids: A probabilistic framework for robot perception and navigation,” Tese de Doutorado, Department of Electrical and Computer Engineering, Carnegie Mellon University, 1989.
- [30] H. P. Moravec, “Sensor fusion in certainty grids for mobile robots,” *AI Magazine*, p. 61–74, 1988.
- [31] L. Matthies and A. Elfes, “Integration of sonar and stereo range data using a grid-based representation,” 1988, pp. 727–733.
- [32] G. W. Kim and B. H. Lee, “Hierarchical sensor fusion for building an occupancy grid map using active sensor modules,” in *SICE-ICASE International Joint Conference*, October 2006, pp. 2600–2605.
- [33] N. C. Mitsou and C. S. Tzafestas, “Temporal occupancy grid for mobile robot dynamic environment mapping,” in *Mediterranean Conference on Control and Automation*, July 2007, pp. 1–8.
- [34] R. Chatila and J.-P. Laumond., “Position referencing and consistent world modeling for mobile robots,” in *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, 1985, pp. 138–145.
- [35] A. Siadat, A. K. S. Klausmann, M. Dufaut, and R. Husson, “An optimized segmentation method for a 2d laser-scanner applied to mobile robot and navigation,” in *Proceedings of the 3rd IFAC Symposium on Intelligent Components and Instruments for Control Applications*, 1997, pp. 153–158.
- [36] G. A. Borges and M.-J. Aldon, “Line extraction in 2D range images for mobile robotics,” *Journal of Intelligent and Robotic Systems*, no. 40, pp. 267–297, 2004.
- [37] K. O. Arras and N. Tomatis, “Improving robustness and precision in mobile robot localization by using laser range finding and monocular vision,” in *Proceedings of the Third European Workshop on Advanced Mobile Robots, EUROBOT*, 1999.
- [38] V. Nguyen, A. Martinelli, N. Tomatis, and R. Siegwart, “A comparison of line extraction algorithms using 2d laser rangefinder for indoor mobile robotics,” *Conference on Intelligent Robots and Systems, IROS 2005*, pp. 1929–1934, 2005.

- [39] T. Pavlidis and S. Horowitz, "Segmentation of plane curves," *Computers, IEEE Transactions on*, vol. C-23, no. 8, pp. 860 – 870, aug. 1974.
- [40] K. O. Arras and R. Siegwart, "Feature extraction and scene interpretation for map-based navigation and map building," *Proceedings of Symposium on Intelligent Systems and Advanced Manufacturing*, vol. 3210, pp. 42–53, 1997.
- [41] B. Barshan and R. Kuc, "Differentiating sonar reflections from corners and planes by employing an intelligent sensor," *Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, p. 560–569, 1990.
- [42] D. Navarro, G. Benet, and F. Blanes, "Line-based incremental map building using infrared sensor ring," *IEEE International Conference on Emerging Technologies and Factory Automation. ETFA 2008.*, pp. 833–838, Sept. 2008.
- [43] A. Kapp and L. Gröll, "Optimal estimation of line segments in noisy lidar data," *Signal processing*, vol. 86, pp. 2304–2317, 2006.
- [44] C. V. Stewart, "Robust parameter estimation in computer vision," *Society for Industrial and Applied Mathematics*, vol. 41, pp. 513–537, 1999.
- [45] R. M. Haralick, "Propagating covariance in computer vision," in *Proceedings of 12th ICPR*, 1994, pp. 493–498.
- [46] ———, "Breakpoint detection using covariance propagation," *Pattern analysis and machine intelligence*, vol. 20, pp. 845–851, 1998.
- [47] S. T. Pfister, S. I. Roumeliotis, and J. W. Burdick, "Weighted line fitting algorithms for mobile robot map building and efficient data representation," *IEEE International Conference on Robotics and Automation. ICRA 2003.*, pp. 1304–1311, 2003.
- [48] J. Borenstein and Y. Koren, "Noise rejection for ultrasonic sensors in mobile robot applications," in *Proceedings of 1992 IEEE International Conference on Robotics and Automation*, 1992, pp. 1727–1732.
- [49] H.-J. von der Hardt, D. Wolf, and R. Husson, "The dead reckoning localization system of the wheeled mobile robot romane," in *Proceedings of 1996 IEEE/SICE/RSJ International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 1996, pp. 603–610.
- [50] A. A. Gonçalves, *Metrologia*, L. de metrologia e automatização Universidade de Santa Catarina, Ed., 2001.

APÊNDICES

A. CÁLCULO COMPLETO DA MATRIZ DE COVARIÂNCIA $C_{\hat{\theta}}$

A expressão geral para o cálculo da matriz de covariância das estimativas, $C_{\hat{\theta}}$, segundo o método de Haralick [45] é a seguinte:

$$C_{\hat{\theta}} = \left(\frac{\partial g(\mathbf{P}_{l_k}, \hat{\theta})}{\partial \hat{\theta}} \right)^{-1} \left(\frac{\partial g(\mathbf{P}_{l_k}, \hat{\theta})}{\partial \mathbf{P}_{l_k}} \right)^T C_{XY} \left(\frac{\partial g(\mathbf{P}_{l_k}, \hat{\theta})}{\partial \mathbf{P}_{l_k}} \right) \left[\left(\frac{\partial g(\mathbf{P}_{l_k}, \hat{\theta})}{\partial \hat{\theta}} \right)^{-1} \right]^T \quad (\text{A.1})$$

\mathbf{P}_{l_k} corresponde aos dados $[X \ Y]^T$ da k -ésima linha estimada, e a função g corresponde ao jacobiano da função de custo que relaciona \mathbf{P}_{l_k} com o vetor de estimativas $\hat{\theta}$, isto é:

$$g = \frac{\partial J}{\partial \hat{\theta}}, \quad \text{com} \quad J(\hat{\theta}, p_{l_k}) = \sum_m (\hat{\rho}_k - x_m \cos(\hat{\alpha}_k) - y_m \sin(\hat{\alpha}_k)) \quad (\text{A.2})$$

Primeiro serão definidos os seguintes somatórios:

$$\begin{aligned} \mu_x &= \sum_{m=1}^M x_m, & \mu_y &= \sum_{m=1}^M y_m \\ S_{xx} &= \sum_{m=1}^M x_m^2, & S_{yy} &= \sum_{m=1}^M y_m^2, & S_{xy} &= \sum_{m=1}^M x_m y_m \end{aligned} \quad (\text{A.3})$$

Expressando o jacobiano de g definido em A.2 como

$$g = \left[\frac{\partial J}{\partial \hat{\rho}} \quad \frac{\partial J}{\partial \hat{\alpha}} \right]^T \quad (\text{A.4})$$

temos que cada termo de g é dada por

$$\frac{\partial J}{\partial \hat{\rho}} = 2M\hat{\rho} - 2M\mu_x \cos(\hat{\alpha}) - 2N \sin(\hat{\alpha})$$

$$\frac{\partial J}{\partial \hat{\alpha}} = 2M\hat{\rho} (\sin(\hat{\alpha})\mu_x - \cos(\hat{\alpha})\mu_y) + \sin(2\hat{\alpha})(S_{yy} - S_{xx}) + 2 \cos(2\hat{\alpha})S_{xy}$$

a função g fica então igual a

$$g = \left[\begin{array}{c} 2M\hat{\rho} - 2M\mu_x \cos(\hat{\alpha}) - 2N \sin(\hat{\alpha}) \\ 2M\hat{\rho} (\sin(\hat{\alpha})\mu_x - \cos(\hat{\alpha})\mu_y) + \sin(2\hat{\alpha})(S_{yy} - S_{xx}) + 2 \cos(2\hat{\alpha})S_{xy} \end{array} \right] \quad (\text{A.5})$$

Com o resultado de acima pode-se resolver $\left(\frac{\partial g(\mathbf{P}_k, \hat{\theta})}{\partial \hat{\theta}}\right)$, assim:

$$\frac{\partial g}{\partial \hat{\theta}} = \begin{bmatrix} \frac{\partial g(1)}{\partial \hat{\rho}} & \frac{\partial g(1)}{\partial \hat{\alpha}} \\ \frac{\partial g(2)}{\partial \hat{\rho}} & \frac{\partial g(2)}{\partial \hat{\alpha}} \end{bmatrix} \quad (\text{A.6})$$

onde

$$\frac{\partial g(1)}{\partial \hat{\rho}} = 2N; \quad \frac{\partial g(1)}{\partial \hat{\alpha}} = \frac{\partial g(2)}{\partial \hat{\rho}} = 2N (\mu_x \sin(\hat{\alpha}) - \mu_y \cos(\hat{\alpha}))$$

$$\frac{\partial g(2)}{\partial \hat{\alpha}} = 2N \hat{\rho} (\mu_x \cos(\hat{\alpha}) + \mu_y \sin(\hat{\alpha})) + 2 \cos(2\hat{\alpha})(S_{yy} - S_{xx}) - 4 \sin(2\hat{\alpha})S_{xy}$$

Agora, desenvolvendo $\left(\frac{\partial g(\mathbf{P}_k, \hat{\theta})}{\partial \mathbf{P}_k}\right)$, sabendo que:

$$\mathbf{P}_k = \begin{bmatrix} \mathbf{X} \\ \mathbf{Y} \end{bmatrix} = \begin{bmatrix} \{x_1, x_2, \dots, x_m\} \\ \{y_1, y_2, \dots, y_m\} \end{bmatrix}$$

temos que

$$\frac{\partial g}{\partial \mathbf{P}_k} = \left[\begin{array}{cccccc} \frac{\partial g(1)}{\partial x_1} & \dots & \frac{\partial g(1)}{\partial x_m} & \frac{\partial g(1)}{\partial y_1} & \dots & \frac{\partial g(1)}{\partial y_m} \\ \frac{\partial g(2)}{\partial x_1} & \dots & \frac{\partial g(2)}{\partial x_m} & \frac{\partial g(2)}{\partial y_1} & \dots & \frac{\partial g(2)}{\partial y_m} \end{array} \right]^T \quad (\text{A.7})$$

em que

$$\begin{aligned} \frac{\partial g(2)}{\partial x_m} &= 2 [y_m \cos(2\hat{\alpha}) - x_m \sin(2\hat{\alpha}) + \hat{\rho} \sin(\hat{\alpha})]; & \frac{\partial g(1)}{\partial x_m} &= -2 \cos(\hat{\alpha}) \\ \frac{\partial g(2)}{\partial y_m} &= 2 [x_m \cos(2\hat{\alpha}) - y_m \sin(2\hat{\alpha}) - \hat{\rho} \cos(\hat{\alpha})]; & \frac{\partial g(1)}{\partial y_m} &= -2 \sin(\hat{\alpha}) \end{aligned}$$

Por último, define-se a matriz de covariância C_{XY} associada a cada ponto $\{x_m, y_m\}$ de cada linha extraída,

$$C_{XY} = \begin{bmatrix} c_{xy} & \dots & 0 & 0 \\ 0 & c_{xy} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & c_{xy} \end{bmatrix}_{N \times N}, \quad \text{onde } c_{xy} = \begin{bmatrix} \sigma_x^2 & \sigma_x \sigma_y \\ \sigma_y \sigma_x & \sigma_y^2 \end{bmatrix} \quad (\text{A.8})$$

cada matriz c_{xy} corresponde à matriz de covariância que foi calculada na seção 3.3.2 a qual é propagação do erro dos dados d_n transformados em coordenadas retangulares como pontos p_n . Por fim, substituindo os resultados das equações A.6, A.7 e A.8 na equação A.1 é possível calcular a matriz de covariância dos parâmetros estimados que definem um segmento de linha extraído, a qual expressa-se como:

$$C_{\hat{\theta}} = \begin{bmatrix} \sigma_{\hat{\rho}}^2 & \sigma_{\hat{\rho}\hat{\alpha}} \\ \sigma_{\hat{\alpha}\hat{\rho}} & \sigma_{\hat{\alpha}}^2 \end{bmatrix} \quad (\text{A.9})$$

B. ESTIMAÇÃO DO ERRO DOS SENSORES DE ULTRASSOM

Para a caracterização do erro sistemático do sistema sensorial conformado pelos sensores de ultrassom, foi executado o seguinte teste experimental. Uma estrutura plana de 3.10m de comprimento com 1.20m de altura foi colocada perpendicular ao chão e de frente à plataforma robótica, onde a distância D_n entre o robô e a estrutura foi variando desde 0.5m até 3m, coletando as leituras dos sonares em cada mudança. Considerando as dimensões do robô e da estrutura que representa o alvo, assim como o valor máxima para a distância D_n , foi estudado os efeitos das orientações dos sensores que estão a $\pm 15^\circ$ e $\pm 30^\circ$ com respeito a qualquer dos dois eixos que definem o sistema de referência do robô. A figura B.1 apresenta uma ilustração que explica a configuração do teste realizado.

Para cada uma das distância D_n , foi coletado um total de 60 medições por sensor s_i . Além das leituras dadas pelos sonares, as distâncias entre estes e o alvo foram também medidas com ajuda de uma trena, com o objetivo de ter um grupo de medidas auxiliares para comparar-las com as medidas provenientes do grupo de sensores. Especificamente, a comparação é feita por meio do cálculo do erro quadrático médio (EQM),

$$EQM(D_n)_i = \sqrt{\frac{\sum_{k=1}^z (d(s_i)_k - d(aux)_k)^2}{z}} \quad (B.1)$$

A equação B.1 expressa a diferença em termos do erro quadrático médio entre as z medições do sensor s_i e a leitura tomada pelo sistema auxiliar a uma determinada distância D_n . Este cálculo se repete para cada sensor, e os valores dos diferentes $EQM(D_n)_i$ são plotados em Matlab com o objetivo de observar a tendência do erro para cada sonar ao variar as distâncias D_n . O gráfico das curvas apresenta-se na figura B.2. Nota-se que, embora o número de medições talvez não seja o suficiente como para descrever uma curva suave, a tendência do erro é clara, este se incrementa na medida em que o ângulo de incidência do sensor é maior, o

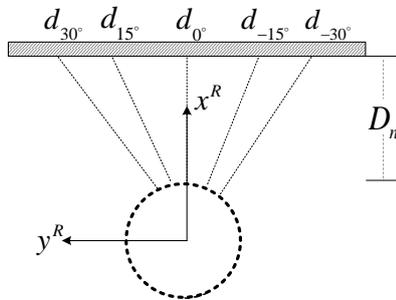


Figura B.1: Vista superior da configuração do teste. O robô está posicionado de frente à estrutura plana a uma distância D_n . As diferentes distâncias detectadas pelos 5 sensores usados são coletadas para cada mudança de D_n .

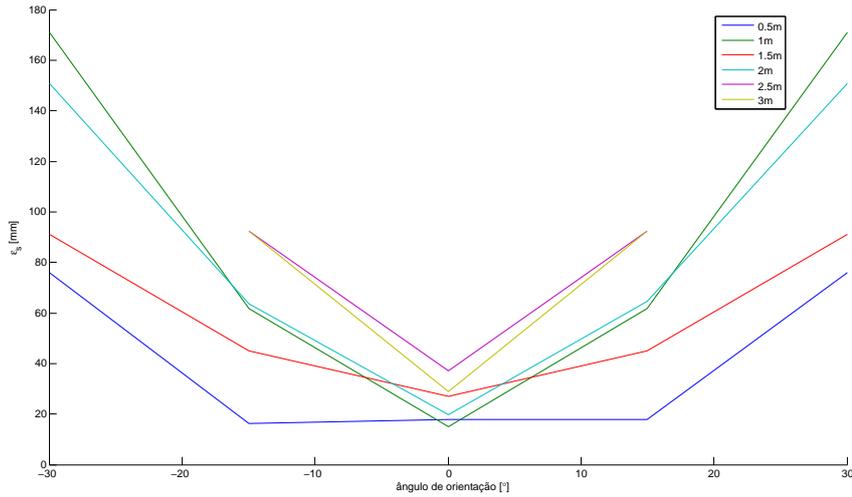


Figura B.2: Curvas do erro sistemático para cada distância D_n medida.

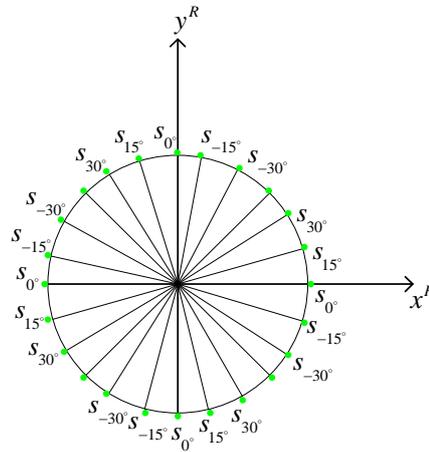


Figura B.3: Identificação dos sensores segundo sua orientação com respeito aos eixos x^R e y^R

que confirma uma das conclusões do trabalho de Barshan e Kuc [41] com respeito à influencia do ângulo de incidência do sonar na medição de distância do mesmo.

Com este resultado presente, o erro sistemático é interpolado para gerar um número maior de medições intermédias no intervalo definido pelos valores mínimo e máximo de D_n . Com esta informação disponível foi construída uma espécie de base de dados de erro sistemático que será utilizada para compensar as leituras reais. Desta maneira os diferentes sensores do sistema de sensoriamento são identificados segundo sua orientação com respeito aos eixos x^R e y^R , como pode ser observado na figura B.3. A medição d_{s_i} será compensada com um valor de erro específico para uma determinada orientação do sensor e de medida de distância real, que varia de 0.5m a 3m.

C. ESTIMAÇÃO DO ERRO DE POSICIONAMENTO DO ROBÔ

Para a estimação do erro de posicionamento da plataforma robótica, esta é programada para que, partindo de uma posição inicial conhecida, percorresse certa distância pré-definida na direção do eixo cuja erro deseja-se calcular. Especificamente foram feitos dois experimentos: no primeiro, o Nomad percorreu uma distância igual a 1020mm na direção x , enquanto no segundo experimento, o percurso foi feito na direção y . Em ambos os testes as quatro rodas estão alinhadas na direção do movimento. Foram realizadas 30 corridas em cada direção.

Para a determinação da parcela sistemática do erro é utilizada a seguinte expressão [50]:

$$\varepsilon_s = \bar{v}_{medido} - v_{real}. \quad (C.1)$$

A figura C.1 apresenta os dados coletados para os dois percursos. Usando a expressão dada em C.1, calculou-se os erros para ambas direções, o quais correspondem a $\varepsilon_{s,x} = 13$ mm e $\varepsilon_{s,y} = 16$ mm. Na figura C.2 mostra-se os dados compensados com os erros sistemáticos para cada experimento. A partir disto é possível determinar o erro aleatório por meio do cálculo do desvio padrão amostral, definido como [2]:

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (C.2)$$

Por fim, os valores de desvio amostral são $s_{x_R} = 13$ mm e $s_{y_R} = 14$ mm, portanto a variância na posição x_R e y_R foram estimadas em $\sigma_{x_R}^2 = 169$ e $\sigma_{y_R}^2 = 196$, respectivamente. Estes valores de variância serão usados na matriz de covariância do erro do robô apresentada na seção 3.3.2.

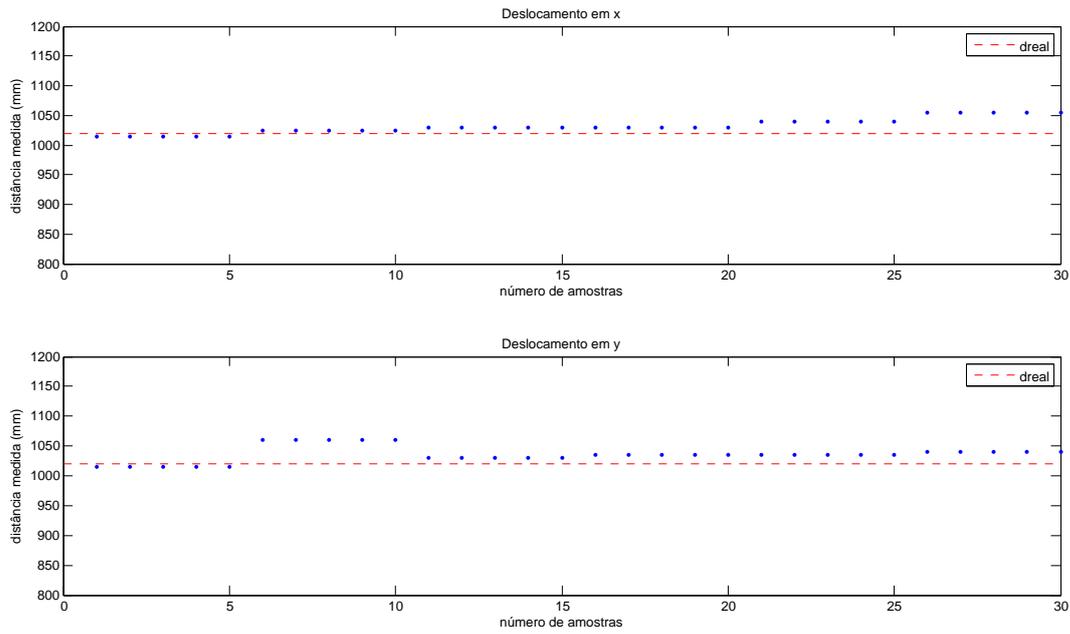


Figura C.1: Medições dos deslocamentos efetuados nos eixos x e y do robô. Os pontos azuis representam as medições, enquanto a linha pontuada vermelha é o valor médio da distância de deslocamento.

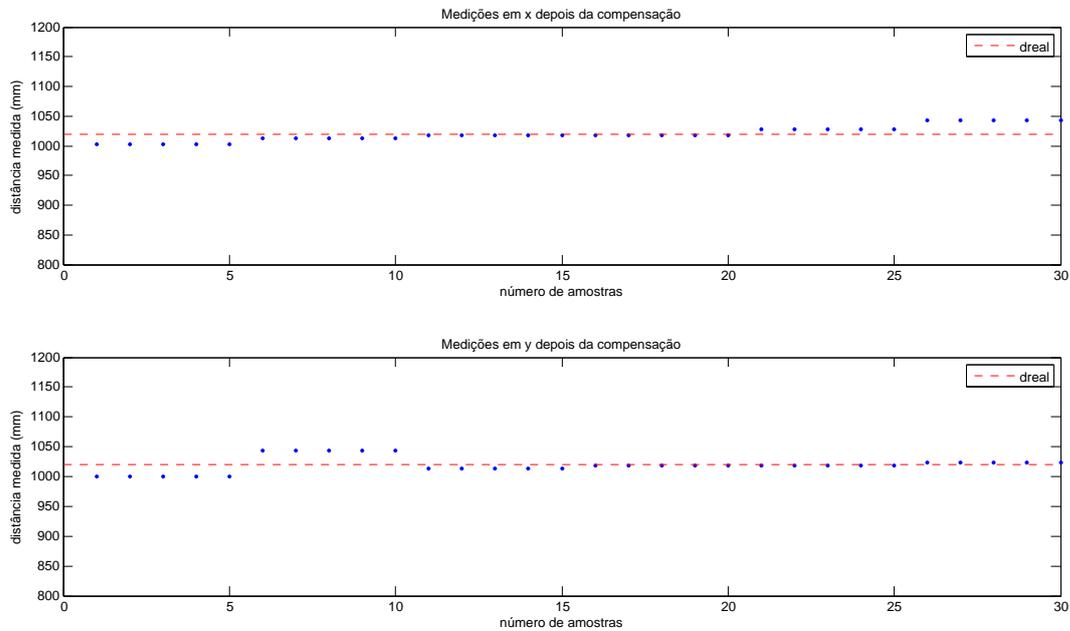


Figura C.2: Medições compensadas com $\varepsilon_{s,x}$ e $\varepsilon_{s,y}$.