



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE
CENTRO DE TECNOLOGIA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA E
DE COMPUTAÇÃO



Mapeamento com Sonar Usando Grade de Ocupação Baseado em Modelagem Probabilística

Anderson Abner de Santana Souza

Orientador: Prof. Dr. Luiz Marcos Garcia Gonçalves

Co-orientador: Prof. Dr. Adelardo Adelino Dantas de Medeiros

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e de Computação da UFRN (área de concentração: Engenharia de Computação) como parte dos requisitos para obtenção do título de Mestre em Ciências.

Natal, RN, fevereiro de 2008

Mapeamento com Sonar Usando Grade de Ocupação Baseado em Modelagem Probabilística

Anderson Abner de Santana Souza

Dissertação de Mestrado aprovada em 15 de fevereiro de 2008 pela banca examinadora composta pelos seguintes membros:

Prof. Dr. Luiz Marcos Garcia Gonçalves (orientador) DCA/UFRN

Prof. Dr. Adelardo Adelino Dantas de Medeiros (co-orientador) . DCA/UFRN

Prof. Dr. Pablo Javier Alsina DCA/UFRN

Prof. Dr. Diogo Pinheiro Fernandes Pedrosa DI/UERN

Agradecimentos

Não posso, de modo algum, deixar de agradecer à Deus, pois Ele me tem sustentado e me agraciado com grandes bençãos. Deus é, sem dúvida, fiel.

Agradeço aos meus pais Abdênego Alves de Souza e Marlene Oliveira de Santana Souza pelo apoio, pelas orações e incentivos. E aos demais familiares que de forma indireta me apoiaram durante este importante desafio.

Ao meu orientador, professor Luiz Marcos, pelo incentivo contínuo pela pesquisa. Ao meu co-orientador, professor Adelardo Adelino, que sempre de boa vontade, nos passou os preciosos conhecimentos necessários para a realização deste trabalho.

Ao professor Diogo Pinheiro Fernandes Pedrosa pelas dicas e ajuda na realização dos experimentos necessários neste trabalho.

Ao amigo e companheiro de estudo André Macedo pelas sugestões, pela ajuda e incentivo durante todo o período da pós-graduação.

À CAPES, pelo apoio financeiro.

*"Deus é a minha fortaleza.e a minha
força, e Ele perfeitamente
desembaraça o meu caminho."*

2 Sm. 22.33

Resumo

Neste trabalho, propomos um método de mapeamento probabilístico com a representação do ambiente mapeado em uma grade de ocupação modificada. A idéia principal do método proposto é deixar que um robô móvel construa de forma sistemática e incremental a geometria do seu entorno, obtendo ao final um mapa completo do ambiente. Como conseqüência, o robô poderá locomover-se no seu ambiente de modo seguro, baseando-se em um índice de confiabilidade dos dados colhidos do seu sistema perceptivo. O mapa é representado de forma coerente com os dados sensoriais, sejam esses ruidosos ou não, oriundos dos sensores exteroceptivos e proprioceptivos do robô. Os ruídos característicos incorporados nos dados de tais sensores são tratados por modelagem probabilística, de modo que seus efeitos possam ser visíveis no resultado final do processo de mapeamento. Os resultados dos experimentos realizados, mostrados no presente trabalho, indicam a viabilidade desta metodologia e sua aplicabilidade na área da robótica móvel autônoma, sendo assim uma contribuição para a área.

Palavras-chave: Probabilidade, Odometria, Mapeamento, Grade de ocupação.

Abstract

In this work, we propose a probabilistic mapping method with the mapped environment represented through a modified occupancy grid. The main idea of the proposed method is to allow a mobile robot to construct in a systematic and incremental way the geometry of the underlying space, obtaining at the end a complete environment map. As a consequence, the robot can move in the environment in a safe way, based on a confidence value of data obtained from its perceptive system. The map is represented in a coherent way, according to its sensory data, being these noisy or not, that comes from exterior and proprioceptive sensors of the robot. Characteristic noise incorporated in the data from these sensors are treated by probabilistic modeling in such a way that their effects can be visible in the final result of the mapping process. The results of performed experiments indicate the viability of the methodology and its applicability in the area of autonomous mobile robotics, thus being an contribution to the field.

Keywords: Probability, Odometry, Mapping, Occupancy Grid.

Sumário

Sumário	i
Lista de Figuras	iii
Lista de Tabelas	v
1 Introdução	1
1.1 O Problema	4
1.2 Contribuições	6
1.3 Organização do texto	6
2 Localização	9
2.1 Localização Relativa	9
2.1.1 Odometria	10
2.1.2 Navegação Inercial	10
2.2 Localização Absoluta	10
2.2.1 Faróis Ativos	11
2.2.2 Marcas	11
2.2.3 Mapas	11
2.3 Fusão de Multi-Sensores	12
2.4 Localização baseada na odometria	12
2.5 Trabalhos Relacionados	13
2.6 Medição e Correção dos Erros de Odometria	15
3 Mapeamento	21
3.1 Mapeamento Topológico	22
3.2 Mapeamento Métrico	22
3.3 Mapas Métricos vs. Mapas Topológicos	23
3.4 Mapeamento com Representação Espacial em Grade de Ocupação	24
3.4.1 Algoritmo Padrão de Mapeamento com Grade de Ocupação	25

3.4.2	Por Que Usar Técnicas Probabilísticas	27
3.4.3	Trabalhos Relacionados	27
4	Mapeamento baseado em sonares	31
4.1	Princípio de funcionamento do sonar	31
4.2	Possíveis problemas com os dados dos sonares	33
4.3	Modelo probabilístico proposto	35
4.4	Geração do mapa	39
4.5	Arquitetura do sistema de mapeamento	40
5	Implementações	43
5.1	O Sistema Robótico	43
5.1.1	Computador embarcado	44
5.1.2	Sonares	44
5.1.3	<i>Encoders</i>	45
5.2	O algoritmo de mapeamento	46
6	Experimentos e Resultados	49
6.1	Experimentos realizados em simulação	49
6.2	Experimentos práticos	50
7	Conclusões	57
	Referências bibliográficas	59

Lista de Figuras

1.1	Robô Sojourner enviado a uma missão em Marte.	2
1.2	Robô autônomo que realiza a limpeza de ambientes.	2
1.3	Diagrama da tarefa de navegação autônoma.	3
1.4	Gráfico do modelo de mapeamento com pose conhecida. As variáveis X e Z representam as poses e as medições respectivamente, e \mathbf{m} representa o mapa construído.	5
1.5	Robô Galatéia utilizado na implementação desse trabalho.	6
2.1	Deslocamento dado pela odometria.	13
2.2	Princípio de funcionamento do Encoder.	13
2.3	Elipses de incertezas ao redor da posição estimada.	14
3.1	Grade de ocupação bidimensional.	24
4.1	Funcionamento do sonar.	32
4.2	Região de operação do Sonar.	33
4.3	Possíveis problemas: a) Distância reduzida; b) Reflexão especular; c) Leituras cruzadas.	34
4.4	Regiões cobertas por um sonar.	35
4.5	Função de ocupação para um sensor ideal. Dada uma medição z pelo sensor $P(z d_{x,y})$, a célula correspondente a medição tem probabilidade de ocupação 1, células com distâncias menores têm probabilidade de ocupação 0 e células com distâncias maiores têm probabilidade 0,5.	36
4.6	Função de ocupação para um sensor modelado por uma distribuição Gaussiana. Dada uma medição $z = 3m$ pelo sensor $P(z d_{x,y})$	37
4.7	Função de ocupação para um sensor modelado por uma distribuição Gaussiana bidimensional. Tanto as incertezas no ângulo quanto na distância estão sendo representadas. O sensor está situado no pequeno ponto preto.	37
4.8	Medição degradada principalmente por erros lineares de odometria.	39
4.9	Medição degradada principalmente por erros angulares de odometria.	39

4.10	Etapas seguidas para o processamento dos dados dos sonares.	41
4.11	Diagrama de blocos do processo de mapeamento	42
5.1	Posicionamento dos sonares do <i>array</i> frontal.	45
6.1	Visão superior do ambiente a ser mapeado.	50
6.2	Momento em que o mapeamento se tornou inviável.	50
6.3	Correção da localização por um método absoluto.	51
6.4	Continuação do mapeamento com a localização do robô já corrigida. . . .	51
6.5	Mapa adquirido com modelo simplista do sensor e todos os sonares. . . .	52
6.6	Mapa adquirido com modelo simplista do sensor e com seleção de sonares.	53
6.7	Mapa sendo construído sem considerar (representar) os efeitos dos erros de odometria.	54
6.8	Mapa sendo construído com modelo proposto, considerando a represen- tação dos erros de odometria.	54
6.9	Representação do aumento dos erros de odometria.	55
6.10	Correção da localização do robô por um método de localização absoluta. .	55
6.11	Nova correção da localização.	56
6.12	Mapa mapa final adquirido com o modelo proposto.	56
6.13	Mapa construído com o modelo proposto e como correção dos erros sis- temáticos de odometria.	56

Lista de Tabelas

2.1	Experimentos para levantamento do erro no deslocamento linear. Valores dados em m e graus.	16
2.2	Experimentos para levantamento do erro no deslocamento angular. Valores dados em m e graus.	17
3.1	Vantagens e desvantagens da abordagem métrica.	23
3.2	Vantagens e desvantagens da abordagem topológica.	23

Capítulo 1

Introdução

Desde os tempos mais antigos, é comum a busca do ser humano por meios que facilitem ou que o substituam na realização de certas tarefas. O desenvolvimento tecnológico possibilitou o surgimento de equipamentos e dispositivos que afirmassem tal desejo. Os robôs são exemplos ou frutos dessa constante busca aliada ao desenvolvimento tecnológico. Concebido, inicialmente, para substituir o homem em tarefas repetitivas, com certo grau de periculosidade e/ou precisão, os robôs passaram a ser amplamente difundidos, principalmente nas linhas de montagem da indústria automobilística [Siegwart & Nourbakhsh 2004]. Nos tempo atuais eles são utilizados na realização de diferentes tarefas, por exemplo, soldagem, pintura, transporte de cargas, exploração de ambientes perigosos e/ou nocivos ao homem, limpeza, diversão, entre outras.

A robótica se apresenta como a ciência que estuda os sistemas robóticos. Thrun et al. (2005) definem, de uma maneira mais formal, a Robótica como sendo a ciência de perceber e manipular o mundo físico através de dispositivos controlados computacionalmente.

A Robótica abrange o estudo dos vários tipos de robôs existentes, os quais podem ser agrupados em duas principais categorias: robôs manipuladores e robôs móveis. Os robôs manipuladores têm forte atuação nas indústrias, principalmente nas linhas de montagens automobilísticas. A categoria dos robôs móveis é constituída por robôs que têm a capacidade de se locomover no espaço tridimensional (robôs aeroespaciais e subaquáticos) ou no espaço planar (robôs terrestres). Os robôs terrestres podem ser subdivididos em robôs com pernas (*legged robots*) e robôs com rodas (*wheeled robots*) [Santana 2007].

Segundo Siegwart & Nourbakhsh (2004), a grande vantagem dos robôs móveis em relação aos robôs manipuladores é a mobilidade. Os robôs manipuladores têm um espaço de atuação limitado, que depende do local onde são fixados. Já os robôs móveis podem se locomover dentro do ambiente de trabalho flexivelmente, aplicando suas habilidades no lugar que se deseja.

O campo de atuação dos robôs móveis vem crescendo gradualmente com os anos. O

sucesso nos resultados da aplicação desses robôs na exploração de ambientes perigosos e inóspitos tornou-os mais populares. Por exemplo, podemos citar os robôs enviados pela NASA (*National Aeronautics and Space Administration*) para exploração do planeta Marte (Figura 1.1).



Figura 1.1: Robô Sojourner enviado a uma missão em Marte.
[NASA 2007]

Fazer com que um robô móvel realize suas tarefas de forma eficiente e segura, sem a intervenção humana, ou seja, de forma autônoma, é algo desejado pela comunidade dos roboticistas. Há vários exemplos de robôs realizando tarefas de forma autônoma em diferentes ambientes, por exemplo, entregando comida e remédios em hospitais, atuando na limpeza (Figura 1.2), entre outras.



Figura 1.2: Robô autônomo que realiza a limpeza de ambientes.
[ROBOTCLEAN 2007]

Para se alcançar um bom nível de autonomia, é importante que o robô móvel seja capaz de se locomover de forma inteligente dentro do seu ambiente, essa tarefa é chamada de navegação. E para que a navegação seja possível é necessário que o robô tenha a habilidade de realizar três primitivas básicas da robótica: sentir, planejar e agir. Essas

três primitivas básicas podem ser traduzidas, em uma visão simplificada, em tarefas mais específicas, que Alsina et al. (2002)¹ e Santana (2007) definem em níveis (ver Figura 1.3).

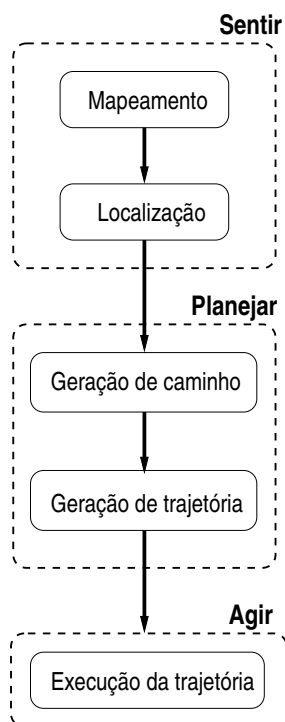


Figura 1.3: Diagrama da tarefa de navegação autônoma.

- Mapeamento - percepção e construção de um modelo do ambiente a partir de informações sensoriais;
- Localização baseada no mapa - baseando-se do modelo obtido é deduzida a posição do robô dentro do ambiente;
- Geração de caminho - o robô calcula um caminho partindo de sua posição inicial até a posição final desejada, sem colidir com obstáculos;
- Geração de trajetória - o caminho gerado é adaptado às restrições temporais e então é calculada a velocidade com que se deseja movimentar;

¹Apostila do Minicurso: Navegação e Controle de Robôs Móveis no Congresso Brasileiro de Automação em 2002.

- Execução da trajetória - os atuadores são controlados de forma a executarem a trajetória gerada o mais fielmente possível;

A realização conjunta das tarefas de Mapeamento e Localização baseada no mapa conduz a um recente problema encontrado na robótica, denominado Localização e Mapeamento Simultâneos (SLAM)² ou, como alguns autores preferem Mapeamento e Localização Concorrentes (CML)³. No SLAM, o robô adquire o mapa do seu ambiente enquanto, simultaneamente, se localiza baseando-se nesse mapa. Este problema é visto como um dos mais difíceis de tratar dentro da robótica [Thrun et al. 2005].

O presente trabalho está no contexto da tarefa de mapeamento, que pode ser definida como o processo de construção de uma descrição espacial (ou modelo) do ambiente no qual o robô está inserido. O modelo do ambiente é um importante requisito para que o robô possa interagir coerentemente com o mundo que o rodeia e, assim, poder realizar suas tarefas de maneira eficiente e agir de modo flexível frente a situações inesperadas. O produto final do processo de mapeamento é o mapa do ambiente.

Existem duas abordagens principais para a representação de ambientes mapeados, a abordagem *métrica* e a abordagem *topológica*. Na abordagem topológica o ambiente é geralmente representado por um grafo, onde os nós podem, por exemplo, representar espaços livres e as arestas podem conter informações de conectividade (por exemplo, distância) entre os nós. Essa abordagem favorece a execução de tarefas de mais alto nível, como navegação, com menor custo computacional.

A abordagem métrica define a geometria do ambiente de uma maneira mais detalhada, apresentando de uma forma relativamente precisa a posição dos objetos que compõem o ambiente tais como, paredes, armários, mesas, entre outros. Um dos métodos mais utilizados para essa abordagem é o da Grade de Ocupação⁴, onde o ambiente passa a ser representado por uma matriz e cada elemento dessa matriz corresponde a um lugar específico do ambiente que pode estar vazio ou ocupado, ou pode ainda, ser um local não explorado cuja situação é desconhecida.

1.1 O Problema

A construção de um mapa em grade de ocupação é feita a partir de dois pré-requisitos básicos: a localização (coordenadas x e y , e orientação em relação a um referencial fixo)

²Em inglês, *Simultaneous Localization and Mapping* ou em sigla SLAM.

³Em inglês, *Concurrent Mapping and Localization* ou em sigla CML

⁴Em inglês, *Occupancy Grid*

do robô, sendo conhecida dentro do ambiente, e aquisição de dados de sensores de alcance. Ao par posição (coordenadas x e y) e à orientação que define completamente um referencial para o robô em relação a uma origem do sistema, denominamos de **pose** do robô. A Figura 1.4 ilustra o princípio do mapeamento.

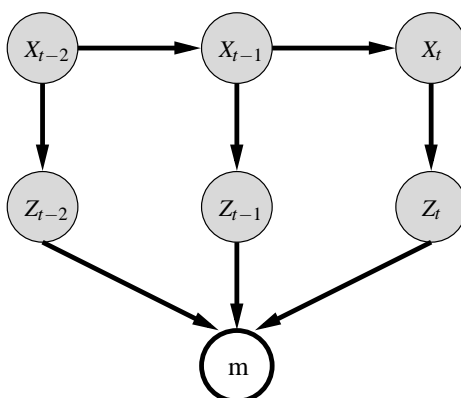


Figura 1.4: Gráfico do modelo de mapeamento com pose conhecida. As variáveis X e Z representam as poses e as medições respectivamente, e \mathbf{m} representa o mapa construído.

Sabendo que não há qualquer informação a priori do ambiente, pode ser assumido que a origem do referencial global é a pose inicial do robô. Logo, todas as informações métricas obtidas dos sensores propioceptivos e exteroceptivos estarão referenciadas a uma mesma origem, mantendo uma certeza nas relações espaciais do ambiente [Pedrosa 2006] [Frese 2004].

Entretanto, uma questão a ser discutida refere-se à existência de erros nas medidas dos sensores. Isto é, as medições sensoriais estão sujeitas a ruídos e a qualidade do mapa depende da qualidade dos dados sensoriais.

Normalmente, o cálculo da pose relativa do robô é realizado por um método matemático (odometria) que utiliza as medições providas por codificadores (*encoders*) de valores, a partir dos sensores de rotação. Tal procedimento determina a posição e orientação atual do robô baseando-se na anterior.

Pela presença de fatores sistemáticos e não-sistemáticos os valores calculados pela odometria possuem erros que vão sendo acumulados durante a movimentação do robô. Além disso, os sensores de alcance também possuem limitações internas ou externas que degradam o valor de suas leituras. Tais fatores degradam a qualidade do mapa.

1.2 Contribuições

A contribuição principal deste trabalho é o método de mapeamento probabilístico com representação espacial do ambiente em uma grade de ocupação que incorpora as incertezas presentes nas medições dos sensores proprioceptivos e exteroceptivos. Ou seja, as incertezas passam a ser representadas no mapa de forma que o mapa construído se torna mais fiel aos dados oriundos do sistema perceptivo do robô. Assim, o robô tem uma quantificação que indica os pontos em que o mapa não é mais confiável devido o crescimento das incertezas.

O mapa final pode ser utilizado para auxiliar o robô em outras tarefas como navegação, planejamento de trajetória, e outras que envolvam tomadas de decisão com base no conhecimento atual do ambiente.

Além do método de mapeamento proposto, este trabalho também apresenta um estudo do sistema de localização baseado em odometria, com a modelagem dos erros sistemáticos observados.

Este trabalho foi o primeiro a ser implementado de fato com o robô Galatéia (Figura 1.5), um modelo Pioneer 3-AT da ActivMedia Robotics, explorando suas capacidades sensoriais. No futuro, este trabalho fará parte de um sistema mais robusto de navegação e mapeamento, o qual integrará também um sistema de visão como mais uma fonte de informação.



Figura 1.5: Robô Galatéia utilizado na implementação desse trabalho.

1.3 Organização do texto

Este documento está dividido em 6 capítulos.

- O Capítulo 1 apresentou o contexto no qual este trabalho está inserido e apresentou suas contribuições;

- O Capítulo 2 trás uma fundamentação teórica sobre as abordagens existentes para localização de robôs móveis;
- O Capítulo 3 apresenta a fundamentação matemática envolvida no mapeamento com Grade de Ocupação;
- O Capítulo 4 apresenta o mapeamento realizado especificamente com sonares e apresenta o modelo probabilístico proposto;
- O Capítulo 5 dá uma visão geral das implementações realizadas com o sistema robótico utilizado neste trabalho;
- O Capítulo 6 apresenta os experimentos realizados e discute a qualidade deles;
- Por fim, o Capítulo 7 traz as conclusões e novas perspectivas sobre o presente trabalho.

Capítulo 2

Localização

O problema da localização consiste basicamente em responder a questão *Onde estou?* do ponto de vista do robô. Isto significa que o robô deve conhecer suas coordenadas x e y , e orientação, relativa a algum referencial fixo no ambiente. Alguns pesquisadores asseveram a importância de lidar com o problema da localização. Cox (1991) afirma que localizar o robô no seu ambiente é o problema mais fundamental para se ter um robô móvel com capacidades de autonomia. Thrun et al. (2001) se referem ao problema da localização como um problema chave para o sucesso de um sistema robótico autônomo.

Como informado no capítulo anterior, o conhecimento da localização do robô dentro do ambiente relativa a um sistema de coordenadas global, é um dos pré-requisitos para a construção de um mapa. A qualidade do mapa depende fortemente do sistema de localização. Os erros acumulados na estimativa da localização do robô leva a má interpretação dos dados sensoriais e conseqüentemente a construção de um mapa errado [Thrun 2002]. Se faz necessário então, que seja realizado um tratamento adequado desses erros. Neste capítulo será dada uma visão geral nos métodos de localização e em seguida será abordado método utilizado com a modelagem de suas incertezas.

Os métodos de localização existentes são classificados em três categorias: Localização Relativa, Localização Absoluta e Fusão de Multi-sensores (Localização Relativa + Absoluta).

2.1 Localização Relativa

O método de localização relativa, também conhecido como *dead reckoning*, tem sido bastante usado em vários trabalhos durante muito tempo [Cox & Nelson 1988] [Chong & Kleeman 1997] [Doh et al. 2003]. Nesse método a estimativa da localização é feita baseando-se nas localizações estimadas nos instantes anteriores, isso favorece a propagação e acúmulo de erros. A estimativa da localização pode ser feita por odometria ou

por navegação inercial.

2.1.1 Odometria

A odometria determina a localização do robô pela integração incremental do movimento das rodas. Pelo uso de *encoders* (sensores de rotação), que contam a quantidade de giros em cada roda, o robô calcula a distância linear percorrida e a orientação.

A odometria ainda é bastante usada devido ao seu baixo custo, precisão em pequenos trechos percorridos e alta taxa de amostragem [Borenstein et al. 1996]. Porém, a grande desvantagem desse método é o acúmulo de erros que aumenta proporcionalmente à distância percorrida. A seção 2.4 detalha melhor a odometria, que foi o método utilizado na estimativa da localização do robô durante o mapeamento.

2.1.2 Navegação Inercial

A navegação inercial é baseada nas informações de giroscópios e acelerômetros para medir a orientação e a aceleração, respectivamente. Os giroscópios detectam pequenas acelerações na orientação e o acelerômetro detecta pequenas acelerações nos eixos x e/ou y .

Como a odometria, a navegação inercial deduz a localização do robô a partir da integração das informações dos sensores. É um método atrativo por ser auto-contido, ou seja, não necessita de informações externas para deduzir onde o robô está. Entretanto, visto que as medidas são deduzidas por integração, pequenos erros nos sensores podem causar erros com crescimento ilimitado [Borenstein et al. 1996].

2.2 Localização Absoluta

O método de localização absoluta fornece informação sobre a localização do robô independente de localizações previamente calculadas. A localização atual não é derivada de integrações, mas derivada diretamente de uma medição [Negenborn 2003]. Diferente do método de localização relativa, os erros presentes nesse método não crescem de forma ilimitada. Na localização absoluta a dedução da posição do robô pode ser feita por três modos: a utilização de faróis ativos, marcas no ambiente e mapas.

2.2.1 Faróis Ativos

Os faróis ativos são marcas que ativamente podem enviar informações sobre localização, por exemplo, sinais de satélite, ondas de rádio e até mesmo as estrelas podem ser consideradas como faróis ativos. Várias aplicações que não são da área de robótica e usam os faróis ativos para se localizar podem ser citadas: navegação marinha, navegação aérea, localização em minas, entre outras.

Vários tipos de equipamento podem ser usados, como os baseados em sensores de ultra-som, mas, talvez, o mais popular deles seja o GPS (*Global System Positioning*). O GPS recebe sinais de satélites e por meio de triangulação estabelece as coordenadas. As desvantagens desse equipamento são: o seu uso em ambientes onde não se consegue captar os sinais de satélite e sua pouca precisão em alguns casos.

2.2.2 Marcas

As marcas são elementos facilmente perceptíveis presentes no ambiente do robô. Tais elementos podem ser nativos do ambiente ou podem, ainda, ser propositalmente inseridos para facilitar o posicionamento do robô. Elas têm a localização conhecida em relação a um referencial global, e quando as identifica o robô pode facilmente calcular a sua localização global por triangulação [Pedrosa 2006].

- **Marcas naturais:** são nativas do ambiente, por exemplo, em ambientes internos podem ser portas, janelas, quinas, luz no teto, entre outras. Em ambientes externos podem ser pedras ou árvores, por exemplo.
- **Marcas artificiais:** são postas intencionalmente em locais estratégicos. Podem ser, por exemplo, código de barras, figuras geométricas coloridas (círculos, quadrados, triângulos, etc.).

2.2.3 Mapas

Outro método de localização é o baseado em mapa. Há situações em que o mapa do ambiente está disponível ao robô. Logo, as medições sensoriais são adquiridas e comparadas (*matching*) com as informações do mapa. A partir da comparação a posição absoluta do robô é deduzida. Uma desvantagem dessa técnica é que o robô precisa colher uma grande quantidade de informações sensoriais para realizar a comparação. Além disso, geralmente as técnicas de comparação requerem um alto custo computacional [Negenborn 2003].

2.3 Fusão de Multi-Sensores

A partir de várias informações sensoriais lidas a partir de múltiplos sensores, pode-se obter uma melhor estimativa da localização de um robô dentro do seu ambiente. Quando isso é feito por sensores heterogêneos pode-se obter melhores resultados, visto que algo que não é percebido por um determinado tipo de sensor pode ser facilmente percebido por outro tipo de sensor. Além disso, a fusão de múltiplos sensores pode reduzir os efeitos dos erros nas medições.

É comum nesse método o uso de técnicas probabilísticas, onde as incertezas e confiabilidade nas informações são tratadas de maneira mais adequada. O uso de técnicas como FKE (Filtro de Kalman Extendido), Filtro de Partículas, entre outras variantes, é comum entre os pesquisadores da robótica móvel em suas pesquisas mais atuais [Ivanjko et al. 2005] [Grisetti et al. 2007] [Santana 2007].

No problema abordado neste trabalho, a fonte de informação sobre a localização do robô é a odometria. Ou seja, o mapa do ambiente do robô é construído a partir da localização derivada da odometria e das informações dos sensores de alcance (no caso, sonares). A seção seguinte detalha um pouco mais a localização baseada em odometria.

2.4 Localização baseada na odometria

Como já mencionado anteriormente, a odometria consiste na determinação das coordenadas x e y , e da orientação do robô por integração dos deslocamentos incrementais das rodas, tomados a partir de um referencial fixo. Exemplificando esse método, considere um robô deslocando-se linearmente em seu ambiente conforme certa trajetória (ver Figura 2.1). No seu ponto inicial o robô se encontra localizado em P_0 , e então inicia o deslocamento chegando, logo, ao ponto P_1 , que pode ser calculado somando-se ao ponto anterior (P_0) o deslocamento realizado D_1 . Continuando seu deslocamento conforme a trajetória indicada chega, então, a posição P_2 , a qual pode ser calculada somando-se ao ponto P_1 o novo deslocamento D_2 . Assim, a localização final é calculada com base nos deslocamentos realizados em relação à localização inicial.

A medida dos deslocamentos das rodas do robô geralmente é feita por *encoders* ópticos, cujo princípio de funcionamento se baseia na transmissão e recepção de luz através de um disco perfurado que se acopla ao eixo da roda (ver Figura. 2.2). Com base na frequência dos pulsos gerados pelo foto sensor calcula-se a velocidade de rotação da roda.

A maneira de se estimar a localização do robô pela odometria favorece o acúmulo de

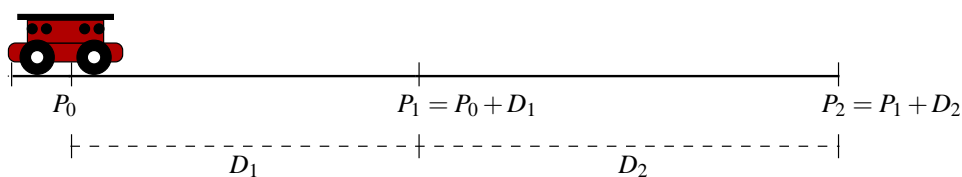


Figura 2.1: Deslocamento dado pela odometria.

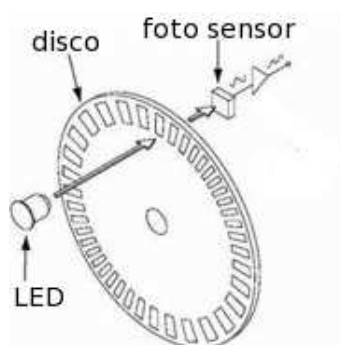


Figura 2.2: Princípio de funcionamento do Encoder.

erros, os quais aumentam proporcionalmente à distância percorrida. Um erro causado em um determinado instante se propaga para os instantes seguintes. Os erros podem ser causados por fatores sistemáticos e/ou não-sistemáticos. Os erros sistemáticos são devidos às incertezas nos parâmetros que fazem parte do modelo cinemático do robô (diâmetro das rodas diferentes, comprimento do eixo diferente do nominal, taxa de amostragem finita dos encoders, etc.). Já os erros não-sistemáticos são devidos a situações inesperadas (objetos inesperados, escorregamento das rodas, etc.) [Santana 2007].

É comum se representar os erros de odometria por elipses de incerteza (ver Figura 2.3), vários trabalhos na literatura apresentam métodos para modelar tais erros [Chenavier & Crowley 1992] [Borenstein et al. 1996] [Chong & Kleeman 1997].

2.5 Trabalhos Relacionados

Existem diferentes métodos propostos na literatura que têm por finalidade a modelagem e correção dos erros de odometria. Chenavier & Crowley (1992) implementaram em seu trabalho um sistema de posicionamento para um robô móvel baseando-se em informações de câmeras e odometria. Eles apresentaram uma forma simplificada para modelagem dos erros de odometria baseada em experimentos que foi posteriormente expandida por Irie et al. (2003). Partindo de observações no comportamento do robô ao se

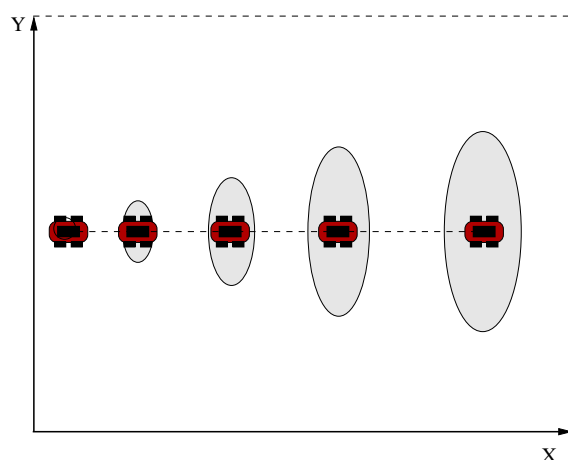


Figura 2.3: Elipses de incertezas ao redor da posição estimada.

movimentar, foram deduzidos coeficientes que, multiplicados pelos deslocamentos realizados, deram uma estimativa dos erros de odometria. Bezerra (2004) utilizou a idéia de Chenavier & Crowley (1992) para estimar as incertezas da odometria de um robô móvel, com o objetivo de calibrá-la e fundir os dados corrigidos com informações providas por um sistemas de visão para localizar o robô dentro de um ambiente. No presente trabalho foi adotado tal método devido à simplicidade apresentada. Além disso, o foco principal deste trabalho está no processo de mapeamento em grade de ocupação.

Outro trabalho, bem referenciado na literatura, que destaca a modelagem e correção dos erros de odometria foi apresentado por Borenstein et al. (1996). Os autores propõem um procedimento para medição e correção de erros sistemáticos, chamado por eles de UMBmark (*University of Michigan Benchmark*). Neste procedimento, uma trajetória quadrangular é realizada pelo robô, aferindo os valores fornecidos pela odometria e os valores reais, medidos por um método de localização absoluta. A partir de dados experimentais, o procedimento segue uma formulação matemática tendo como fim, a estimativa dos erros de odometria e sua calibração.

Outros trabalhos utilizam uma abordagem estatística na modelagem dos erros de odometria. Por exemplo, Martinelli & Siegwart (2003) usaram um Filtro de Kalman Aumentado para estimar um vetor de estados contendo a configuração do robô e parâmetros que caracterizam os erros sistemáticos de odometria. O filtro tem como entradas as leituras dos *encoders* e de sensores exteroceptivos. Um outro Filtro de Kalman, que tem como entrada valores fornecidos pelo Filtro de Kalman Aumentado, estima os erros não-sistemáticos.

Como mencionado anteriormente, neste trabalho foi adotado o método de modelagem

proposto por Chenavier & Crowley (1992) devido à simplicidade, pois o foco principal do trabalho está na construção e representação do mapa. A seção seguinte apresenta os testes realizados para se levantar o comportamento dos erros de odometria observados no robô Galatéia.

2.6 Medição e Correção dos Erros de Odometria

A partir da coleta de alguns dados experimentais com várias amostras, tornou-se possível se ter uma idéia da função que norteia o crescimento dos erros de odometria. O experimento prático se deu em duas etapas, a primeira consistiu na modelagem dos erros (linear e angular) ocorridos durante um deslocamento linear. A segunda etapa consistiu na modelagem dos erros (linear e angular) ocorridos durante um deslocamento angular. A realização dos experimentos da primeira etapa se deu da seguinte forma:

1. Obedecendo a comandos pré-definidos o robô se moveu, de um referencial fixo estabelecido, para um ponto determinado baseando-se na odometria;
2. Com o uso de uma fita métrica verificou-se, então, o valor real do deslocamento;
3. Com o auxílio de um transferidor, foi verificado o valor da orientação final do robô em relação ao referencial estabelecido;
4. Comparando o valor do deslocamento linear calculado pela odometria com o valor real medido, foi obtido o erro linear;
5. Comparando o valor da orientação estimado pela odometria, com o valor medido, foi obtido o erro angular.

A Tabela 2.1 mostra os dados e resultados do experimento realizado para um deslocamento linear e velocidade constante de $0,2m/s$. e_{lin} e e_{ang} são os erros lineares e angulares respectivamente, obtidos a partir da comparação entre os valores reais de deslocamento e os valores de odometria (Equação 2.1 e Equação 2.2).

$$e_{lin} = \Delta l_{real} - \Delta l_{odometria} \quad (2.1)$$

$$e_{ang} = \Delta \theta_{real} - \Delta \theta_{odometria} \quad (2.2)$$

Teste	Deslocamento linear									
	1 m		2 m		3 m		4 m		5 m	
	e_{lin}	e_{ang}	e_{lin}	e_{ang}	e_{lin}	e_{ang}	e_{lin}	e_{ang}	e_{lin}	e_{ang}
1	-0,085	-3	-0,175	-3	-0,27	-2	-0,36	-2	-0,45	-4
2	-0,09	-2	-0,185	-2	-0,275	-3	-0,35	-3	-0,44	-2
3	-0,095	-1	-0,17	-4	-0,275	-2	-0,365	-1	-0,44	-2
4	-0,09	-2	-0,18	-1	-0,275	-3	-0,365	-3	-0,43	-3
5	-0,09	-2	-0,19	-4	-0,27	-1	-0,355	-2	-0,45	-3
6	-0,1	-3	-0,19	-3	-0,275	-4	-0,36	-4	-0,445	-1
7	-0,105	-1	-0,19	-3	-0,275	-2	-0,365	-2	-0,435	-2
8	-0,1	-5	-0,185	-2	-0,27	-3	-0,365	-3	-0,44	-4
9	-0,105	-1	-0,19	-2	-0,27	-1	-0,355	-2	-0,45	-3
10	-0,09	-1	-0,185	-2	-0,28	-2	-0,37	-1	-0,44	-2
11	-0,1	-2	-0,19	-3	-0,27	-2	-0,36	-2	-0,445	-1
12	-0,105	-3	-0,185	-3	-0,275	-3	-0,36	-2	-0,455	-2
Media	-0,096	-2	-0,185	-2	-0,273	-2	-0,361	-2	-0,443	-2
Desvio	0,0068	1,15	0,0063	1,08	0,0031	0,91	0,0054	0,87	0,0069	1,04

Tabela 2.1: Experimentos para levantamento do erro no deslocamento linear. Valores dados em m e graus.

Posteriormente, os mesmos experimentos foram repetidos com os seguintes valores de velocidade $0,4m/s$ e $0,6m/s$. Os resultados finais apresentados foram bem semelhantes aos da Tabela 2.1.

Seguindo a mesma idéia, foram feitos os experimentos para deslocamentos angulares. Os valores apresentados na Tabela 2.2 são respectivos a uma velocidade angular de $0,7rad/s$. Novamente, este experimento foi realizado com velocidades diferentes, apresentando resultados aproximados aos da Tabela 2.2. Vale salientar que a medição da orientação real é mais difícil de ser conseguida, logo, é possível que os valores dos erros na Tabela 2.2 estejam levemente corrompidos por imprecisão nas medições reais.

A partir dos experimentos realizados é possível estabelecer uma função que descreva, de maneira aproximada, o comportamento dos erros sistemáticos presentes na odometria (Equação 2.3 e Equação 2.4).

$$E_{lin}(\Delta l) = 0.09\Delta l + \tau \quad (2.3)$$

$$E_{ang}(\Delta\theta) = 0.095\Delta\theta + \alpha \quad (2.4)$$

Onde:

- Δl é o deslocamento linear estimado pela odometria;

	Deslocamento angular					
	45°		90°		180°	
Teste	e_{lin}	e_{ang}	e_{lin}	e_{ang}	e_{lin}	e_{ang}
1	0,02	-4,5	0,03	-8	0,01	-17,5
2	0,01	-4	0,02	-9	0,03	-18,5
3	0,02	-5	0,01	-7,5	0,02	-16,5
4	0,01	-5	0,01	-8	0,01	-17
5	0,02	-4,5	0,02	-9	0,03	-18
6	0,02	-4,5	0,03	-8	0,01	-15,5
7	0,01	-4	0,01	-8,5	0,02	-18,5
8	0,01	-4	0,02	-9	0,01	-18
9	0,02	-4	0,01	-7,5	0,02	-18,5
10	0,01	-4,5	0,02	-9	0,01	-18
11	0,02	-4	0,02	-8	0,01	-17,5
12	0,01	-5	0,01	-7	0,02	-17,5
Media	0,015	-4,4	0,017	-8,1	0,016	-17,58
Desvio	0,005	0,38	0,007	0,77	0,007	0,86

Tabela 2.2: Experimentos para levantamento do erro no deslocamento angular. Valores dados em m e graus.

- $\Delta\theta$ é o deslocamento angular estimado pela odometria;
- τ é o erro linear médio devido a uma rotação;
- α é o erro angular médio devido a um deslocamento linear;

Nos experimentos realizado τ e α apresentaram valores próximos, não variando proporcionalmente com os deslocamentos linear e angular. Logo, foi adotado o maior entre os valores de erro médio, ou seja, $\tau = 0,017m$ e $\alpha = -2^\circ$ ou $\alpha = -0,035rad$.

Com esses experimentos foi possível também, estimar os erros causados por fatores externos como derrapagem das rodas, vibrações e outros [Chenavier & Crowley 1992]. Baseando-se no trabalho de Chenavier & Crowley (1992) foram estimados fatores que, multiplicados pelos deslocamentos lineares e angulares, dão uma estimativa da variância de tais erros. Neste caso, os erros são representados por distribuições normais, ou seja, distribuições gaussianas de média 0, e de variância ϵ_{lin} para o caso linear e ϵ_{ang} para o caso angular. As Equações 2.5 e 2.6 descrevem o cálculo da variância dos erros linear e angular respectivamente.

$$\epsilon_{lin} = \kappa_{ll}\Delta l + \kappa_{l\theta}\Delta\theta \quad (2.5)$$

$$\varepsilon_{ang} = \kappa_{\theta\theta}\Delta\theta + \kappa_{\theta l}\Delta l \quad (2.6)$$

Onde:

- κ_{ll} é o coeficiente do erro linear em um deslocamento linear Δl ;
- $\kappa_{l\theta}$ é o coeficiente do erro linear causado por uma rotação $\Delta\theta$;
- $\kappa_{\theta\theta}$ é o coeficiente do erro angular em uma rotação $\Delta\theta$;
- $\kappa_{\theta l}$ é o coeficiente do erro angular causado por um deslocamento linear Δl ;

O cálculo desses coeficientes é dado pelas Equações 2.7, 2.8, 2.9 e 2.10.

$$\kappa_{ll} = \frac{(\text{Desvio}(e_{lin}))^2}{\text{Media}(\Delta l)} \quad (2.7)$$

$$\kappa_{l\theta} = \frac{(\text{Desvio}(e_{lin}))^2}{\text{Media}(\Delta\theta)} \quad (2.8)$$

$$\kappa_{\theta\theta} = \frac{(\text{Desvio}(e_{ang}))^2}{\text{Media}(\Delta\theta)} \quad (2.9)$$

$$\kappa_{\theta l} = \frac{(\text{Desvio}(e_{ang}))^2}{\text{Media}(\Delta l)} \quad (2.10)$$

Tomando como base os dados apresentados nas Tabelas 2.1 e 2.2 foi possível calcular os valores dos coeficientes. Logo,

$$\begin{aligned} \kappa_{ll} &= 1 \times 10^{-5} \frac{m^2}{m} \\ \kappa_{l\theta} &= 1,53 \times 10^{-5} \frac{m^2}{rad} \\ \kappa_{\theta\theta} &= 7,2 \times 10^{-5} \frac{rad^2}{rad} \\ \kappa_{\theta l} &= 4,5 \times 10^{-5} \frac{rad^2}{m} \end{aligned}$$

Agrupando-se as duas fontes de erros modeladas chega-se às equações gerais dos erros (Equação 2.11 e 2.12).

$$\mathcal{E}_{lin} = E_{lin}(\Delta l) + \mathcal{N}(0, \varepsilon_{lin}) \quad (2.11)$$

$$\mathcal{E}_{ang} = E_{ang}(\Delta\theta) + \mathcal{N}(0, \varepsilon_{ang}) \quad (2.12)$$

Onde $\mathcal{N}(0, \varepsilon_{lin})$ representa um ruído gaussiano de média 0 e variância ε_{lin} para o caso linear, e $\mathcal{N}(0, \varepsilon_{ang})$ representa um ruído gaussiano de média 0 e variância ε_{ang} para o caso angular. A modelagem desses erros possibilita a representação deles no mapa do ambiente, resultando em um mapeamento coerente com os dados sensoriais.

Capítulo 3

Mapeamento

O processo de mapeamento se resume a adquirir um modelo espacial do ambiente de trabalho do robô a partir de informações sensoriais. Porém, os sensores estão sujeitos a erros, que são referidos como ruídos em suas medições. Além disso, os movimentos do robô também são susceptíveis a erros causados por fatores sistemáticos e não-sistemáticos, explicitados no Capítulo 2.

Thrun (2002) enumera alguns desafios encontrados no problema de mapeamento. Um deles é a questão de modelar os erros citados no parágrafo anterior. Além de imprecisões inerentes ao sistema robótico, um segundo desafio do problema de mapeamento está na dimensão do ambiente a ser mapeado. Quanto maior o ambiente mais custoso e menos preciso será seu mapa. Também surge como desafio o problema da correspondência dos dados, também conhecido como associação de dados. A correspondência dos dados visa determinar se medições tomadas em diferentes instantes de tempo correspondem ao mesmo objeto ou não. Um quarto desafio está no mapeamento de ambientes dinâmicos, como, por exemplo, ambientes onde pessoas trafegam constantemente. A grande maioria dos algoritmos de mapeamento considera o processo sendo feito em ambientes estáticos. O processo de mapeamento deve incorporar uma boa estratégia de exploração que leve em conta um modelo parcial do ambiente. Essa tarefa surge como um quinto desafio para o problema de mapeamento.

Como mencionado no Capítulo 1, existem duas abordagens principais para a representação de ambientes mapeados: *topológica* e *métrica*. Existe ainda uma terceira abordagem, a *híbrida*, que une as principais características das duas anteriores. As seções seguintes detalham as duas principais abordagens.

3.1 Mapeamento Topológico

O mapeamento topológico representa o mapa como um grafo, onde os nós correspondem a lugares significativos e as arestas contêm informações de navegação entre os nós (ou lugares). Os mapas topológicos permitem um planejamento eficiente, são mapas compactos para armazenamento e não há a necessidade de uma localização exata do robô, mas apenas o conhecimento do nó no qual ele se encontra. Por outro lado, este tipo de representação apresenta sérias dificuldades quanto ao problema da correspondência.

Um dos primeiros trabalhos nesta linha [Kuypers & Byun 1991] utiliza a abordagem topológica para representar vários tipos de ambientes 2D percorridos por um robô simulado. A partir dos resultados obtidos com o mapeamento topológico, a descrição geométrica dos ambientes foi construída.

Trabalhos mais recentes que se baseiam na presente abordagem podem ser facilmente encontrados na literatura. Por exemplo, Fabrizi & Saffioti (2000) implementaram um sistema de mapeamento baseado na topologia dos ambientes a partir de informações sobre espaços livres. O ambiente foi primeiramente discretizado em uma grade de ocupação e então sua topologia foi extraída por um método chamado de *topologia digital*, que é definido como a extração da topologia de um espaço discreto. Além disso, a extração de informações sobre a forma do ambiente também é feita através de uma ferramenta de processamento de imagens baseada em *morfologia matemática*. Com a combinação das informações obtidas da topologia digital e da morfologia matemática, um mapa topológico do ambiente foi construído.

3.2 Mapeamento Métrico

O mapeamento métrico produz uma definição geométrica do ambiente no qual o robô está inserido. Nesta abordagem o ambiente é apresentado de maneira detalhada incluindo os objetos que o compõe como paredes, obstáculos, passagens, entre outros, mantendo uma boa relação com o mundo real [Pedrosa 2006]. Além disso, os mapas métricos são fáceis de serem construídos e mantidos, e são mais adequados para tratar o problema da correspondência.

Os mapas métricos podem ser representados por *mapas de características* ou por *grades de ocupação* [Pedrosa 2006]. Os mapas de características armazenam informações geométricas de formas encontradas nos ambientes, como círculos, retângulos, retas, entre outras. Tais características são encontradas principalmente em ambientes internos e estruturados, por exemplo, casas, corredores, escritórios e laboratórios.

A representação por grade de ocupação discretiza os espaços contínuos do ambiente, de forma que o ambiente passa a ser representado sob a forma de uma matriz. Cada elemento da matriz (também chamado de célula) representa um local do ambiente que pode estar, de acordo com uma formulação probabilística, ocupado ou vazio, ou pode ainda não ter sido explorado.

Este trabalho propõe a implementação de um método de mapeamento com representação do ambiente por grade de ocupação levando em conta as incertezas presentes nas medições dos sensores exteroceptivos (que no presente trabalho são sonares) e dos sensores propioceptivos (que são instanciados pelos *encoders*). A representação por grade de ocupação será melhor discutida na seção 3.4.

3.3 Mapas Métricos vs. Mapas Topológicos

Thrun (1998) confronta as abordagens de mapeamento métrico e topológico levando em consideração, no caso métrico, principalmente mapas representados em grade de ocupação. As principais vantagens e desvantagens de cada abordagem são apresentadas nas Tabelas 3.1 e 3.2.

Abordagem métrica	
Vantagens	<ul style="list-style-type: none"> · Fácil de construir, representar e manter · Reconhecimento de lugares (baseado na geometria) · Facilita a computação de caminhos curtos
Desvantagens	<ul style="list-style-type: none"> · Planejamento ineficiente, alto custo computacional · Requer a determinação precisa da pose do robô · Linguagem simbólica pobre para solução de problemas

Tabela 3.1: Vantagens e desvantagens da abordagem métrica.

Abordagem topológica	
Vantagens	<ul style="list-style-type: none"> · Permite planejamento eficiente e baixo custo computacional · Não requer a determinação precisa da localização do robô · Representação simbólica conveniente para a solução de problemas
Desvantagens	<ul style="list-style-type: none"> · Dificuldade de construir e manter para ambientes grandes · Dificuldade no reconhecimento de lugares · Pode produzir caminhos sub-ótimos

Tabela 3.2: Vantagens e desvantagens da abordagem topológica.

3.4 Mapeamento com Representação Espacial em Grade de Ocupação

O método de mapeamento com representação do ambiente em grade de ocupação foi primeiro proposto por Elfes (1987), sendo posteriormente formalizado em sua tese de Ph.D. [Elfes 1989a]. Outro pesquisador, contemporâneo a Elfes, que também contribuiu para a formulação e formalização da grade de ocupação foi Moravec (1988), sendo este orientador de Elfes em suas pesquisas de doutorado.

O mapeamento em grade de ocupação tem como objetivo a construção de mapas consistentes a partir de dados sensoriais, sobre a hipótese de que a posição do robô é conhecida. A idéia básica desse método de mapeamento é representar o ambiente em uma grade ou matriz multi-dimensional (geralmente 2D ou 3D), com células de mesmo tamanho, onde cada célula corresponde a uma variável aleatória cujo valor representa a probabilidade de ocupação (ver Figura 3.1).

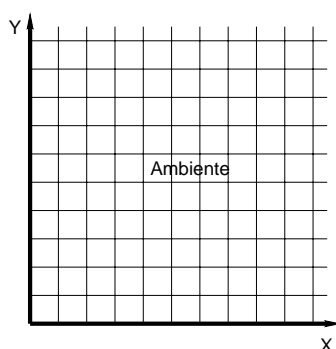


Figura 3.1: Grade de ocupação bidimensional.

Os estados ou valores das células são estimados pela interpretação dos dados provenientes dos sensores de alcance modelados de forma probabilística por uma f.d.p. (função densidade de probabilidade). Através de regras probabilísticas Bayesianas é possível atualizar o valor das células, sempre que novas leituras dos sensores forem tomadas, a partir de diferentes pontos do ambiente. O modelo espacial probabilístico resultante serve como mapa do ambiente do robô e pode ser usado diretamente para tarefas de navegação, como planejamento de caminho e desvio de obstáculo, estimativa de posição, entre outras [Elfes 1989a].

A seção seguinte descreve matematicamente os passos realizados no mapeamento por grade de ocupação na sua versão original.

3.4.1 Algoritmo Padrão de Mapeamento com Grade de Ocupação

O algoritmo padrão formalizado por Elfes (1989a) visa a aquisição de um mapa baseando-se nas medições sensoriais do robô e na sua posição. A Expressão 3.1 apresenta a forma matemática probabilística através da qual o mapeamento em grade de ocupação pode ser descrito, conforme se constata na literatura [Elfes 1987][Elfes 1989a][Thrun 2003][Thrun et al. 2005].

$$P(m|z_{1:t}) \quad (3.1)$$

Onde m representa o mapa adquirido e $z_{1:t}$ é o conjunto das medições sensoriais até o instante de tempo t . É importante deixar claro, que o algoritmo supõe que as coordenadas cartesianas e a orientação do robô são conhecidas.

O espaço contínuo do ambiente no qual o robô se encontra é discretizado em células, as quais, juntas, representam de forma aproximada o ambiente mapeado. Essa discretização pode corresponder a um corte longitudinal do ambiente 3D no plano dos sensores, no caso de ser uma grade 2D, ou pode ser uma discretização 3D do ambiente, no caso de uma grade 3D. Isso depende das características e/ou modelo dos sensores utilizados. Por exemplo, os sonares permitem uma amostra 2D do ambiente, já com câmeras é possível se obter uma representação 3D do ambiente.

Considerando a discretização do ambiente em células, pode-se definir o mapa como um conjunto finito de células $m_{x,y}$, onde cada célula possui um valor agregado que corresponde a probabilidade de ela estar ocupada. As células podem possuir valores entre o intervalo de 0 a 1. O valor 0 significa que a célula está vazia e 1 que a célula está ocupada.

Sendo o mapa um conjunto finito de células, o problema de mapeamento pode ser decomposto em um conjunto de problemas de estimativa do valor de cada célula do mapa. A Expressão 3.2 representa uma instância da estimativa do valor de uma célula $m_{x,y}$, e é traduzida como a probabilidade de a célula $m_{x,y}$ estar ocupada, quando são dadas as medições sensoriais $z_{1:t}$ até o instante t .

$$P(m_{x,y}|z_{1:t}) \quad (3.2)$$

Por razões de instabilidade numérica com probabilidades próximas de 0 ou 1, é comum calcular o *log-odds* (ou logaritmo da probabilidade) de $P(m_{x,y}|z_{1:t})$ ao invés de $P(m_{x,y}|z_{1:t})$. O *log-odds* é definido por:

$$l_{x,y}^t = \log \frac{P(m_{x,y}|z_{1:t})}{1 - P(m_{x,y}|z_{1:t})} \quad (3.3)$$

O valor da probabilidade de ocupação da célula pode ser recuperado através da Equação 3.4.

$$P(m_{x,y}|z_{1:t}) = 1 - \frac{1}{e^{\mu_{x,y}}} \quad (3.4)$$

O valor do *log-odds* pode ser estimado recursivamente em qualquer instante t pela regra de Bayes aplicada a $P(m_{x,y}|z_{1:t})$ (Equação 3.5).

$$P(m_{x,y}|z_{1:t}) = \frac{P(z_t|z_{1:t-1}, m_{x,y})P(m_{x,y}|z_{1:t-1})}{P(z_t|z_{1:t-1})} \quad (3.5)$$

$P(z_t|z_{1:t-1}, m_{x,y})$ representa o modelo probabilístico do sensor de alcance, $P(m_{x,y}|z_{1:t-1})$ é o valor de ocupação da célula $m_{x,y}$ no instante $t - 1$ e $P(z_t|z_{1:t-1})$ é o valor real medido pelo sensor. Supondo que o mapeamento é realizado em ambientes estáticos, logo, vale a assertiva de que a medição atual do sensor é independente das medições passadas, dado o mapa m , em qualquer instante, isso implica nas Expressões 3.6 e 3.7.

$$P(z_t|z_{1:t-1}, m) = P(z_t|m) \quad (3.6)$$

$$P(z_t|z_{1:t-1}) = P(z_t) \quad (3.7)$$

Visto que o mapa é decomposto em células, essa suposição pode ser estendida como mostra a Expressão 3.8.

$$P(z_t|z_{1:t-1}, m_{x,y}) = P(z_t|m_{x,y}) \quad (3.8)$$

Com base nas suposições feitas acima, a Equação 3.5 pode ser simplificada resultando na Equação 3.9.

$$P(m_{x,y}|z_{1:t}) = \frac{P(z_t|m_{x,y})P(m_{x,y}|z_{1:t-1})}{P(z_t)} \quad (3.9)$$

A aplicação da regra da Probabilidade Total à Equação 3.9 implica na Equação 3.10, a qual computa a probabilidade de a célula $m_{x,y}$ estar ocupada, tendo como base o modelo probabilístico do sensor $P(z_t|m_{x,y})$ e o valor de ocupação da célula disponível a priori $P(m_{x,y}|z_{1:t-1})$.

$$P(m_{x,y}|z_{1:t}) = \frac{P(z_t|m_{x,y})P(m_{x,y}|z_{1:t-1})}{\sum_{m_{x,y}} P(z_t|m_{x,y})P(m_{x,y}|z_{1:t-1})} \quad (3.10)$$

3.4.2 Por Que Usar Técnicas Probabilísticas

A maioria dos trabalhos encontrados na literatura relacionados a mapeamento de ambientes faz uso de técnicas probabilísticas. Em tais trabalhos os autores se direcionam a deduzir modelos probabilísticos dos robôs, dos sensores e dos ambientes mapeados. A razão dessa popularidade das técnicas probabilísticas vem do fato de que o processo de mapear ambientes sofre forte influência das incertezas presentes nas informações sensoriais e, com as técnicas probabilísticas é possível tratar esse problema modelando explicitamente as diversas fontes de ruídos e seus efeitos nas medições [Thrun 2002].

Os algoritmos probabilísticos têm se apresentado como uma boa alternativa para tratar o problema das incertezas e ruídos, produzindo resultados mais satisfatórios para o problema de mapeamento.

3.4.3 Trabalhos Relacionados

Há vários trabalhos na literatura que enfocam o mapeamento com representação em grade de ocupação. Elfes (1989b) implementou seu método proposto em dois robôs móveis: Neptune e Terragator. O método de mapeamento passou a fazer parte de um sistema mais completo, o qual integra a navegação e mapeamento baseando-se em informações providas por um *array* de sonares. Esse sistema foi posteriormente chamado de Dolphin. Assim como o trabalho de Elfes (1989b), o presente trabalho também contou com informações sensoriais oriundas de um *array* de sonares para a construção do mapa do ambiente em grade de ocupação. Porém, o modelo proposto para os sonares trata de uma maneira melhor os ruídos presentes no sistema robótico.

Moravec (1988) também enfocou em seu trabalho o uso de sonares, mas além dos sonares seu sistema se serviu de um módulo de visão estéreo. As informações dos sonares e do módulo de visão foram combinadas para a construção de um mapa em grade de ocupação.

Em um trabalho posterior, Moravec (1996) introduziu a idéia de representar ambientes em grade de ocupação tridimensional. Durante o mapeamento, os dados sensoriais foram obtidos através do processamento uma seqüência de pares de imagens estéreas. Os resultados do mapeamento foram sendo armazenados em um array 3D nomeado de Grade de Evidência¹. As células da grade foram inicialmente valoradas em zero, indicando que não havia evidência de ocupação. Depois de várias leituras sensoriais, as células foram sendo preenchidas de modo que os blocos de células negativas indicavam espaços livres,

¹Do inglês, *Evidence Grid*

enquanto que os espaços positivos definiam obstáculos. Nesse mesmo trabalho, Moravec utilizou uma abordagem baseada na Teoria da Evidência de Dempster-Shafer, diferentemente da abordagem probabilística Bayesiana inicialmente proposta por Elfes (1987).

Além das abordagens Bayesiana e Teoria da Evidência, há uma terceira abordagem, possibilística, utilizada para a construção de mapas em grade de ocupação que se baseia em conjuntos *fuzzy*, a qual foi inicialmente proposta por Oriolo et al. (1997). Nessa abordagem o mapa do ambiente é definido como um conjunto *fuzzy* de pontos incertos, cuja função membro quantifica a possibilidade de cada ponto pertencer a um obstáculo [Oriolo et al. 1997].

Ribo & Pinz (2001) fizeram um estudo comparativo entre as três abordagens citadas, apresentando diferentes modelos para os sensores de alcance objetivando encontrar uma melhor forma de manipular as incertezas presentes nas medições. A partir de vários experimentos com dados reais obtidos por dois robôs diferentes dotados de sonares, inseridos em um escritório, eles concluíram que as abordagens Bayesiana e Teoria da Evidência apresentaram melhores resultados em ambientes estruturados. Já a abordagem que se baseia em conjuntos *fuzzy* apresentou melhores resultados no tratamento do problema de múltiplas reflexões dos sonares.

Visando tratar os problemas intrínsecos aos sonares como, múltiplas reflexões e a redundância de leituras, Konolige (1997) apresentou em seu trabalho um método, que se trata de um refinamento matemático do método de mapeamento apresentado por Elfes (1987), nomeado por ele de MURIEL (*MUltiple Representation Independent Evidence Log*).

No trabalho proposto por Elfes (1987) o processo de mapeamento foi realizado sem levar em conta a dependência de um célula com suas vizinhas. Isso implica em mapas inconsistentes quando o mapeamento é feito em ambientes desordenados. Em um trabalho mais recente, Thrun (2003) apresentou um modelo avançado para os sensores que visou contornar essa deficiência apresentada pelo algoritmo padrão de Elfes (1987). Da mesma forma que o modelo proposto por Elfes (1987), o modelo de Thrun (2003) foi também baseado na teoria Bayesiana. Resultados dos experimentos e comparações com o algoritmo padrão foram apresentados.

Thrun et al. (2005) afirmaram que a principal utilidade da técnica de mapeamento em grade de ocupação está no pós-processamento, ou seja, no mapa resultante. Com o processo de mapeamento concluído a grade de ocupação pode ser útil para várias aplicações como: planejamento de trajetória, navegação, reconhecimento de marcos, desvio de obstáculos, localização, entre outras.

Por exemplo, Borenstein & Koren (1997) implementaram um método de desvio de ob-

3.4. MAPEAMENTO COM REPRESENTAÇÃO ESPACIAL EM GRADE DE OCUPAÇÃO 29

stáculos de tempo real nomeado VFF (*Vector Field Force*). Esse método se utiliza de uma grade de ocupação, obtida a partir de sonares, para definir a localização dos obstáculos presentes no ambiente.

No trabalho de Dutra et al. (2003) um robô móvel munido de um *array* de 24 sonares, construiu um mapa em grade de ocupação de sua redondeza e armazenou-o em sua memória para, posteriormente, navegar se baseando no mapa construído. Os resultados, porém, tanto para o mapeamento quanto para a posterior navegação, foram bastante influenciados pelos erros acumulados pelo sistema de odometria do robô.

O trabalho de Kong et al. (2006) traz outro exemplo de aplicação de um mapa em grade de ocupação. Eles implementaram um sistema de localização baseado no Filtro de Kalman Estendido (EKF), no qual características presentes no ambiente como cantos e superfícies planas foram detectadas, e informações a respeito dessas características foram integradas a um mapa em grade de ocupação construído a priori, produzindo a localização exata do robô.

A maioria dos algoritmos de mapeamento está direcionada aos ambientes estáticos. Visando aumentar o leque de aplicações de mapeamento, Biswas et al. (2002) apresentaram um método para mapeamento de ambientes dinâmicos nomeado de ROMA (*Robot Object Mapping Algorithm*), pelo qual é feita a combinação de vários mapas em grade de ocupação e então, os modelos dos objetos móveis foram construídos e suas localizações determinadas.

Capítulo 4

Mapeamento baseado em sonares

Este capítulo dará uma visão mais detalhada dos sensores utilizados neste trabalho: os sonares. Os sonares fazem parte da classe dos sensores de alcance, ou seja, sensores que medem o quão distantes estão de algum objeto. Junto com os sensores a *laser* e os baseados em infra-vermelho, os sonares são considerados sensores ativos, visto que emitem energia (acústica) e detectam a energia refletida para medir distância.

Os sonares são muito populares na robótica devido à alta velocidade de resposta, simplicidade em sua saída (a distância a um objeto é dada de forma direta) e devido ao seu baixo custo se comparado a outros tipos de sensores de alcance [Howard & Kitchen 1996] [Lee et al. 2006]. Geralmente são montados em grupos também chamados de *arrays* de sonares, com o fim de agilizar e melhorar a percepção do robô no seu ambiente de trabalho.

4.1 Princípio de funcionamento do sonar

Sonar se refere a qualquer sistema que usa ondas sonoras para medir distância [Murphy 2000]. O princípio de funcionamento do sonar se baseia na propagação de ondas ultrassônicas. As ondas são geradas por um emissor, refletidas por algum objeto e capturadas por um receptor. Os emissores e receptores podem ser encapsulados juntos, formando um único dispositivo, ou podem estar separados em dois dispositivos. A Figura 4.1 ilustra de forma simplificada o princípio de funcionamento do sonar.

A onda se propaga em seu meio como uma onda mecânica, cujo comprimento, λ , pode ser calculado pela Equação 4.1.

$$\lambda = \frac{v}{f} \quad (4.1)$$

Onde v é a velocidade do som no meio de propagação da onda e f é a frequência do

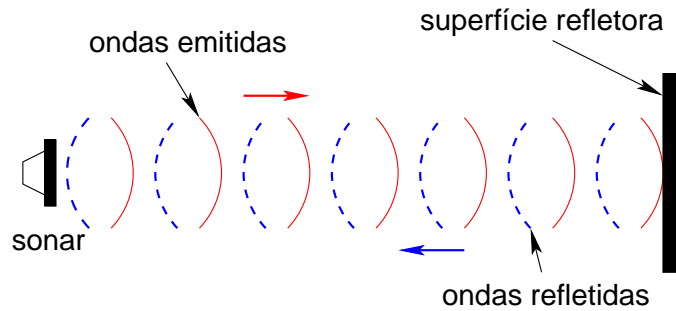


Figura 4.1: Funcionamento do sonar.

emissor ultra sônico, que geralmente tem valor entre 20 kHz a alguns poucos Gigahertz [Elfes 1989a]. Cada tipo de aplicação requer uma frequência adequada, por exemplo, um sonar para um veículo subaquático usa uma frequência apropriada para o perfeito deslocamento das ondas sonoras na água. Já em um veículo terrestre seria mais adequada uma frequência própria para o ar.

A velocidade com que a onda sonora viaja no meio depende de inúmeros fatores, incluindo a temperatura e a umidade relativa do meio, com a dependência à temperatura sendo o fator mais crítico. No ar, a variação da velocidade do som com a temperatura ambiente T (expressa em $^{\circ}\text{K}$) é dada pela Equação 4.2.

$$v = 331,31 \sqrt{\frac{T}{273,16}} \text{ m/s} \quad (4.2)$$

Sabendo a velocidade v de propagação da onda sonora no ar é possível determinar a distância entre o sonar e um objeto medindo apenas tempo t gasto pela onda para percorrer uma trajetória até o objeto e, sendo refletida por ele, retornar ao receptor (ver Equação 4.3). A onda refletida também é comumente chamada de *eco*. Essa técnica de medição da distância pelo tempo do trajeto percorrido pela onda é chamada de *time-of-flight* (TOF).

$$D = \frac{t}{2}v \quad (4.3)$$

Mas nem todos os objetos presentes em um ambiente têm o potencial de refletir as ondas sonoras. O comportamento reflexivo das superfícies depende da textura dos objetos. Se a frequência espacial (baseada no comprimento de onda da superfície) da textura é muito maior que a frequência do feixe ultra sônico, a superfície apresenta uma reflexão especular, caso contrário a reflexão é difusa [Elfes 1989a].

Uma outra característica importante é que o feixe ultra sônico possui uma região de maior sensibilidade chamada de *feixe principal*, considerada como a região de operação

do sensor. Geralmente, o ângulo de abertura do feixe principal é modelado em torno de 30° e a distância máxima atingida 5 m. A Figura 4.2 ilustra a região alcançada pelo feixe principal de um sonar Polaroid série 6500. Na visão de alguns pesquisadores um sonar ideal teria um feixe de raio reto sem abertura, também chamado de *ray-tracer scanner* [Leonard & Durrant-Whyte 1992].

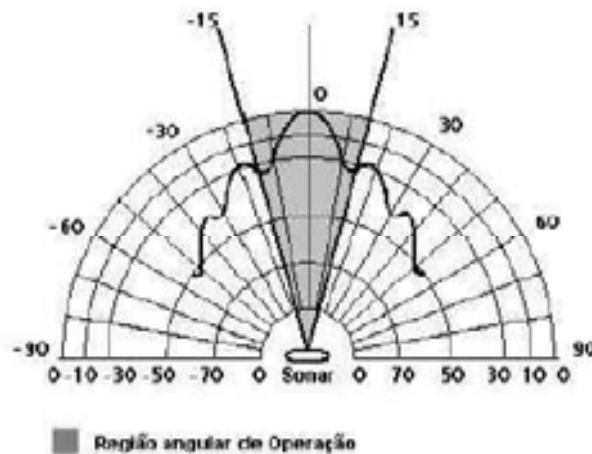


Figura 4.2: Região de operação do Sonar.

Além do limite máximo de distância os sonares também possuem um limite mínimo de distância. Distâncias menores que este limite podem ser lidas erroneamente pelo sensor. Isso se dá devido às limitações internas dos componentes físicos que geram e/ou detectam as ondas.

4.2 Possíveis problemas com os dados dos sonares

A interpretação precisa dos dados sensoriais adquiridos pelos sonares pode se tornar difícil por conta de alguns fatores que são inerentes aos dados adquiridos e que são intrínsecos aos sensores [Elfes 1989a]. São eles:

- Imprecisões no circuito temporizador limitam a precisão das medições. Os fabricantes geralmente disponibilizam *datasheets* que indicam a precisão dos sensores. O sonar Polaroid 6500, por exemplo, possui um erro na medição absoluta em torno de $\pm 1\%$.

- Supondo que a abertura do feixe principal de um sonar é de 30° e a superfície do objeto refletor não está perpendicular ao sensor. Logo, um lado do feixe alcançará primeiro a superfície do objeto retornando a distância mais próxima. Se o programa que utiliza os dados sensoriais considera que as leituras estão ao longo do eixo principal do sonar, então o dado retornado estará errado (ver Figura 4.3a).
- Os feixes sonoros podem sofrer reflexões especulares. Isso acontece quando o ângulo entre o feixe incidente e a superfície do objeto é pequena. O feixe é totalmente refletido para fora do alcance do receptor, produzindo falsas leituras (ver Figura 4.3b). As reflexões especulares também são chamadas de múltiplas reflexões.
- As reflexões especulares podem gerar também leituras cruzadas ou *cross-talk*. É muito comum o uso de vários sonares em um mesmo robô, o que pode levar a um feixe disparado por um sonar ser captado por outro gerando medidas erradas. Isso pode ser corrigido pelo sincronismo dos disparos dos feixes (ver Figura 4.3c).

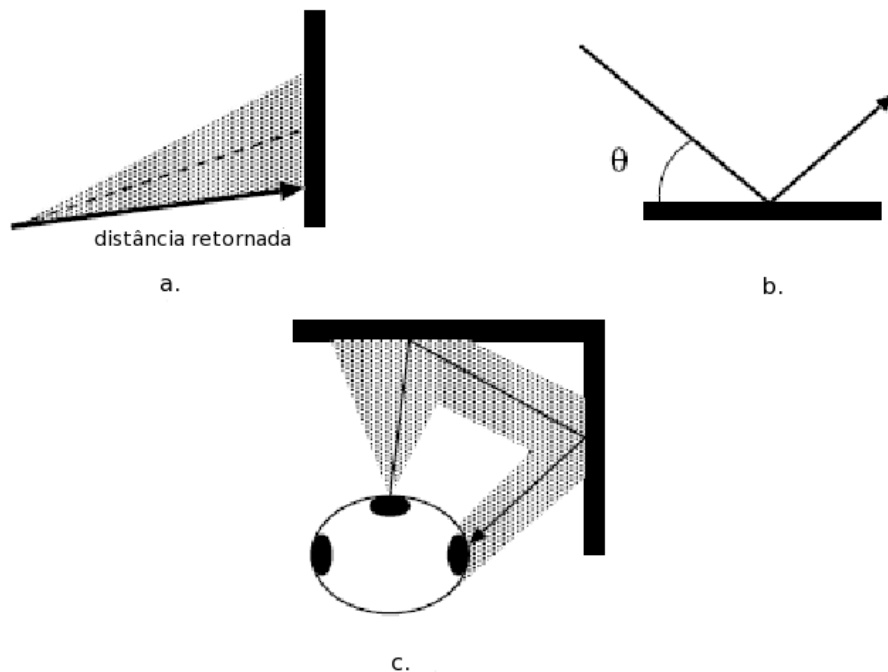


Figura 4.3: Possíveis problemas: a) Distância reduzida; b) Reflexão especular; c) Leituras cruzadas.

4.3 Modelo probabilístico proposto

A interpretação dos dados sensoriais para a construção do mapa em grade de ocupação, na maioria dos trabalhos, é feita por um modelo probabilístico. Mas alguns trabalhos, que não estão focados em construir uma grade precisa, propõem modelos simplistas para os sensores, sem considerar as incertezas do sistema. Apesar disso, na grande maioria dos casos as funções que valoram a ocupação das células caracterizam três regiões cobertas pelo feixe principal do sonar (ver Figura 4.4).

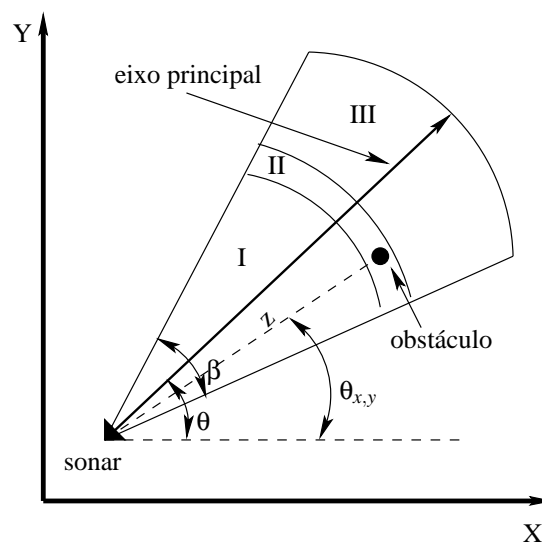


Figura 4.4: Regiões cobertas por um sonar.

A região I representa a área livre de obstáculos. A região II está associada com a medição do sensor. O objeto que refletiu a onda sonora pode estar em qualquer lugar dessa região. E por fim, a região III é a área que teoricamente está coberta pelo feixe do sonar, mas não se sabe se está ocupada ou vazia.

Um modelo bem simplista para o sensor é o chamado modelo do sensor ideal, o qual se caracteriza por não ter incertezas em suas medições e é definido pela função Delta de Kronecker (Equação 4.4). Nesse modelo z representa a medição tomada do sonar e $d_{x,y}$ a distância euclidiana do sensor à célula $m_{x,y}$, que está sendo mapeada.

$$P(z|d_{x,y}) = \delta(z - d_{x,y}) = \begin{cases} 1 & , \text{ se } z = d_{x,y} \\ 0 & , \text{ se } z \neq d_{x,y} \end{cases} \quad (4.4)$$

A função que estima a ocupação de uma célula considerando esse modelo é ilustrada na Figura 4.5. Essa função é calculada pela Equação 3.10, deduzida na seção 3.4.1, e é

dependente do modelo do sensor utilizado. Evoluindo na modelagem do sensor, pode-se considerar um modelo cujas medições de distância são corrompidas por um ruído Gaussiano de média 0 e variância σ_z^2 , o modelo correspondente é dado pela Equação 4.5. A Figura 4.6 ilustra a função que estima ocupação para esse modelo.

$$P(z|d_{x,y}) = \frac{1}{\sqrt{2\pi\sigma_z}} \exp\left(-\frac{(z-d_{x,y})^2}{2\sigma_z^2}\right) \quad (4.5)$$

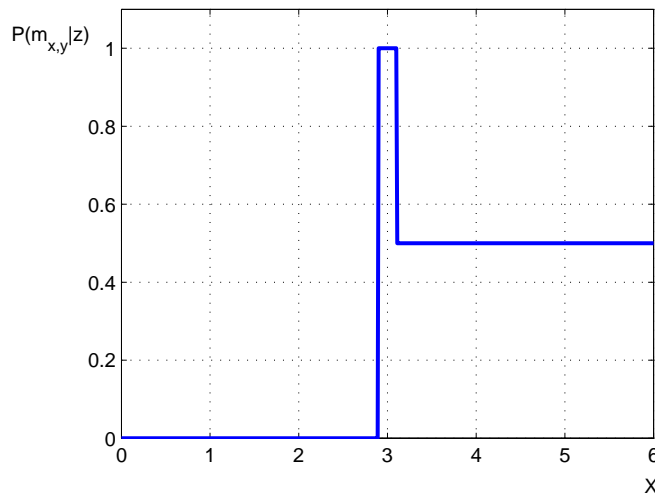


Figura 4.5: Função de ocupação para um sensor ideal. Dada uma medição z pelo sensor $P(z|d_{x,y})$, a célula correspondente a medição tem probabilidade de ocupação 1, células com distâncias menores têm probabilidade de ocupação 0 e células com distâncias maiores têm probabilidade 0,5.

O modelo proposto por Elfes (1989a) trata as incertezas, tanto na distância do objeto detectado quanto no ângulo do sensor ultra sônico. O modelo é descrito por uma distribuição Gaussiana bidimensional representada pela Equação 4.6 [Elfes 1989a].

$$P(z, \theta|d_{x,y}, \theta_{x,y}) = \frac{1}{2\pi\sigma_z\sigma_\theta} \exp\left[-\frac{1}{2}\left(\frac{(z-d_{x,y})^2}{\sigma_z^2} + \frac{(\theta-\theta_{x,y})^2}{\sigma_\theta^2}\right)\right] \quad (4.6)$$

Onde θ é o ângulo de orientação do sensor em relação ao eixo x do referencial global fixo no ambiente (ver Figura 4.4); $\theta_{x,y}$ é o ângulo formado entre um vetor, com origem no ponto em que está o sonar e passa pela célula $m_{x,y}$ (que pode estar com obstáculo ou não), e o eixo x do referencial global adotado (ver Figura 4.4); σ_z^2 e σ_θ^2 são as variâncias que computam as incertezas na distância medida z e no ângulo θ respectivamente. A Figura 4.7 ilustra a função que estima a ocupação para este modelo.

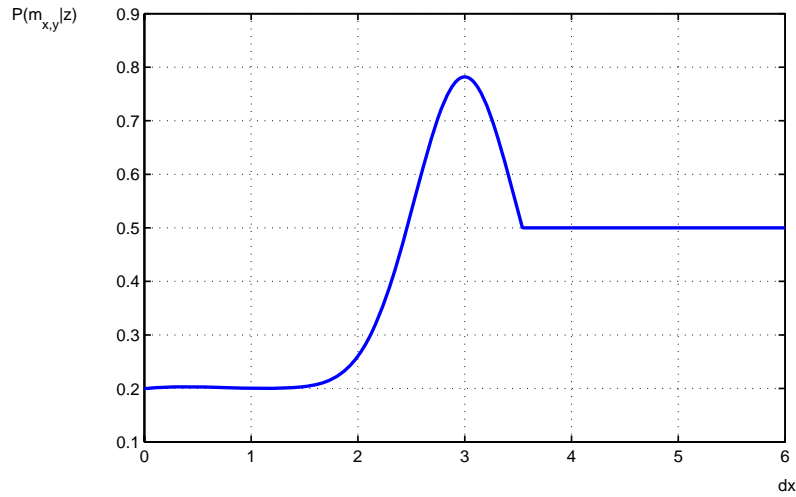


Figura 4.6: Função de ocupação para um sensor modelado por uma distribuição Gaussiana. Dada uma medição $z = 3m$ pelo sensor $P(z|d_{x,y})$.

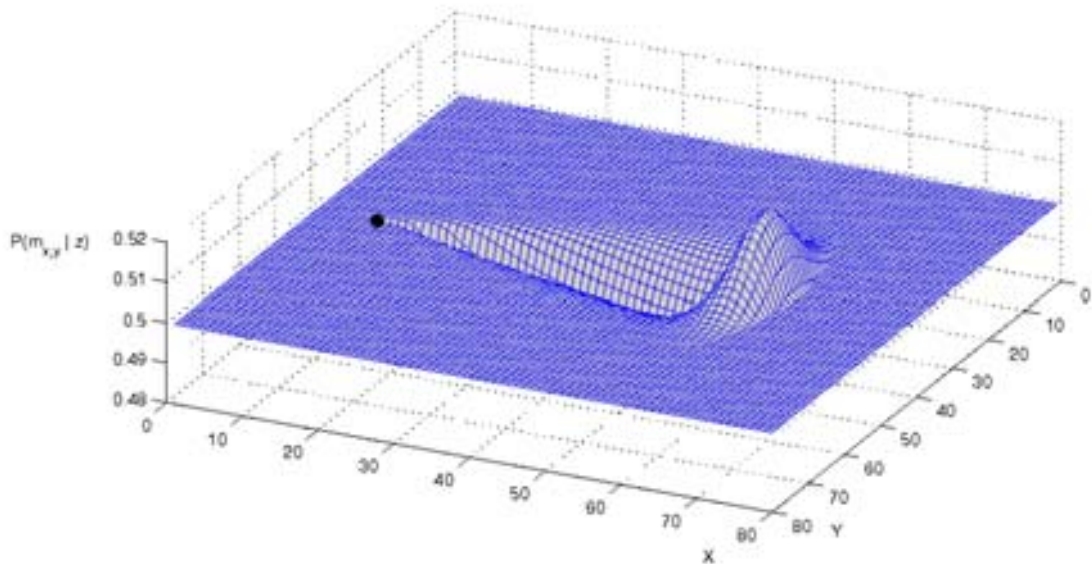


Figura 4.7: Função de ocupação para um sensor modelado por uma distribuição Gaussiana bidimensional. Tanto as incertezas no ângulo quanto na distância estão sendo representadas. O sensor está situado no pequeno ponto preto.

Tomando como base o modelo Gaussiano bidimensional proposto por Elfes (1989a), o modelo dos sonares neste trabalho leva em consideração, além das incertezas características dos sonares, também as incertezas inerentes ao sistema de odometria do robô. A partir da dedução dos erros de odometria feita na seção 2.6 e descritos pelas Equações 2.11 e 2.12, é possível estabelecer uma relação entre as variâncias σ_z^2 e σ_θ^2 , as quais descrevem os erros característicos dos sonares, com os erros de odometria:

$$\sigma_z = z \times \eta + \mathcal{E}_{lin}$$

$$\sigma_\theta = \frac{\beta}{2} + \mathcal{E}_{ang}$$

ou

$$\sigma_z = z \times \eta + E_{lin}(\Delta l) + \mathcal{N}(0, \varepsilon_{lin}) \quad (4.7)$$

$$\sigma_\theta = \frac{\beta}{2} + E_{ang}(\Delta\theta) + \mathcal{N}(0, \varepsilon_{ang}) \quad (4.8)$$

Através das Equações 4.7 e 4.8 as variâncias σ_z^2 e σ_θ^2 podem ser calculadas, passando, desta feita, a considerar as influências causadas pela odometria. A Equação 4.7 computa as incertezas para uma medida de distância z e um deslocamento linear Δl realizado pelo robô. η é o fator de erro típico do sonar em uso (nesse trabalho foi usado um erro de +/- 1%), $E_{lin}(\Delta l)$ é a função que calcula o erro causado por fatores sistemáticos na odometria (Equação 2.3) e $\mathcal{N}(0, \varepsilon_{ang})$ é a distribuição normal que computa os erros causados por fatores não-sistemáticos (Equação 2.5).

Já a Equação 4.8 computa as incertezas inerentes a um ângulo de orientação do sonar θ e um deslocamento angular $\Delta\theta$ realizado pelo robô. β é o ângulo de abertura do feixe principal do sonar (ver Figura 4.4), $E_{ang}(\Delta\theta)$ (Equação 2.4) é a função que descreve o erro sistemático referente um deslocamento angular e $\mathcal{N}(0, \varepsilon_{ang})$ (Equação 2.6) é a distribuição normal que estima os erros não-sistemáticos para o mesmo deslocamento.

Através das modificações proposta ao modelo proposto por Elfes (1989a) é possível representar a degradação que os erros de odometria acarretam ao processo de mapeamento. Pois, a probabilidade de que a medição do sonar esteja correta, calculada pela Equação 4.6, passa a ser ponderada pelos erros do sistema de odometria do robô. Dessa forma, o mapa final será mais coerente com a qualidade dos dados dos sonares e da odometria. A Figura 4.8 ilustra uma medição degradada principalmente por erros lineares de odometria. Já a Figura 4.9 ilustra uma medição degradada principalmente por

erros angulares de odometria.

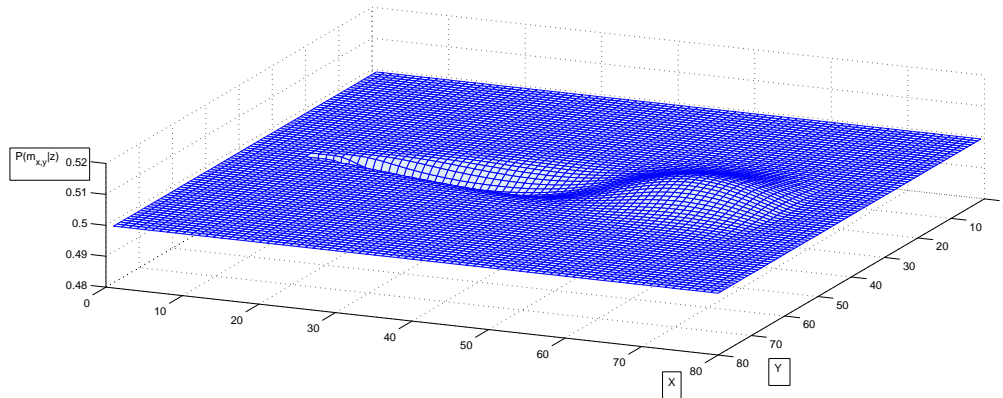


Figura 4.8: Medição degradada principalmente por erros lineares de odometria.

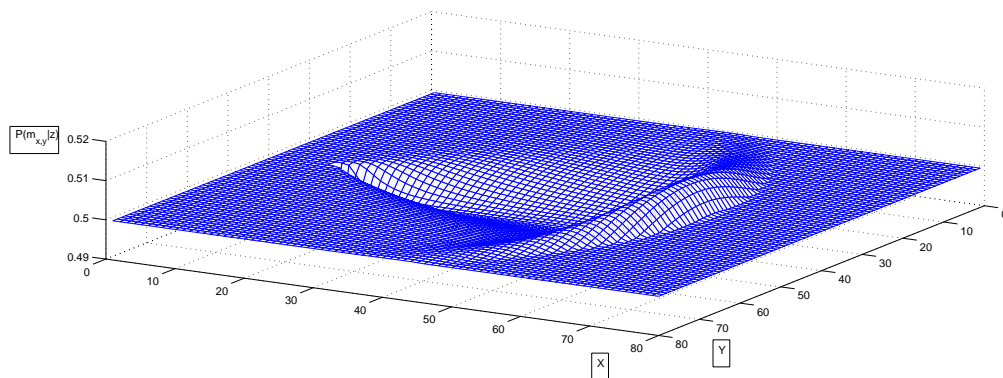


Figura 4.9: Medição degradada principalmente por erros angulares de odometria.

4.4 Geração do mapa

Para gerar o mapa em Grade de Ocupação é necessário algumas etapas de processamento dos dados sensoriais providos pelos sonares.

- **Pré-processamento:** Os dados dos sonares são examinados por um filtro, a fim de que leituras falsas sejam descartadas na construção do mapa. Medições de distâncias abaixo de um limite mínimo e acima de um limite máximo são eliminadas, visto

que valores dentro desse intervalo são mais susceptíveis à erros. Para os sonares utilizados nesse trabalho os limites são de 15 cm para o mínimo e 4m para o máximo.

- **Posição dos sensores:** Deve ser observado o cálculo correto da posição e orientação dos sensores em relação ao robô, e também a em relação ao referencial fixo adotado.
- **Interpretação pelo modelo probabilístico:** Os dados dos sonares são interpretados pelo modelo probabilístico proposto para gerar a *vista do sonar*.
- **Mapa do sonar:** Cada sonar gera seu mapa local a partir de sua *vista*. O mapa local é então incrementado ao mapa global.

A Figura 4.10 ilustra a seqüência em que essas etapas são realizadas, considerando um sistema com apenas dois sonares.

4.5 Arquitetura do sistema de mapeamento

Diante de toda teoria necessária, exposta até aqui, pode-se contar com o auxílio de um diagrama de blocos para expor de uma forma mais clara e completa as partes que compõem o sistema de mapeamento. Uma representação da arquitetura do sistema pode ser visualizada na Figura 4.11.

No instante inicial t_0 é assumido que o robô está localizado no ponto: (x_0, y_0, θ_0) dentro do ambiente, estabelecendo assim, o referencial global a ser adotado. Além disso, também é assumido que o mapa é formado, em sua totalidade, por células cuja a ocupação é desconhecida, ou seja, não se sabe se estão ocupadas ou vazias. A partir daí as informações dos sonares são adquiridas e tratadas seguindo as etapas de processamento descritas na seção anterior.

O mapa produzido pelo algoritmo de mapeamento é integrado ao mapa global, ou seja, o mapa global é atualizado a partir do mapa construído com o robô no ponto (x_0, y_0, θ_0) .

Baseando-se no mapa global atualizado o robô passa a planejar o melhor caminho a ser percorrido para alcançar o maior número de células que ainda não foram exploradas. Esse caminho deverá ser composto, exclusivamente, por células não ocupadas, evitando assim a colisão do robô contra possíveis obstáculos.

Após planejar o caminho de exploração o robô se move obedecendo as restrições impostas pelo planejamento, chegando ao novo ponto (x_1, y_1, θ_1) , no qual é executado novamente o algoritmo de mapeamento. E assim o ciclo se segue. A fase de planejamento

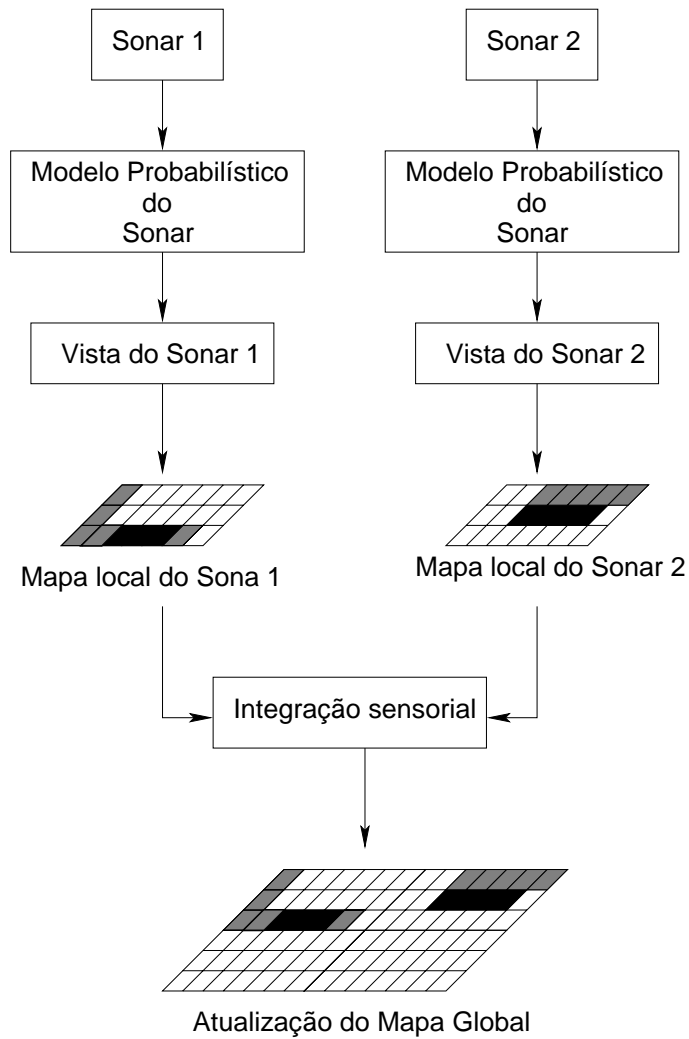


Figura 4.10: Etapas seguidas para o processamento dos dados dos sonares.

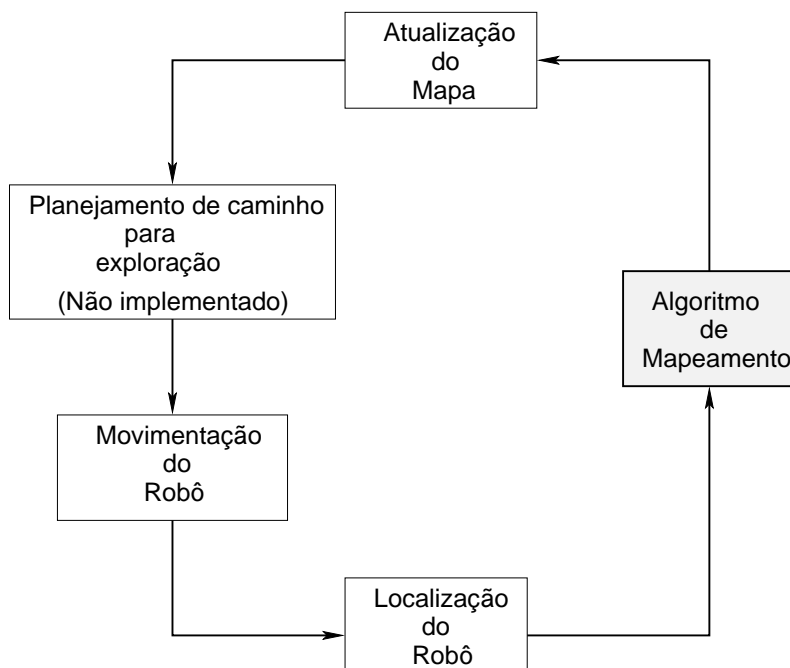


Figura 4.11: Diagrama de blocos do processo de mapeamento

de caminho para exploração não foi implementada nesse trabalho, sendo então executada por teleoperação.

Como já exposto, os erros do sistema de odometria se acumulam durante os movimentos do robô. E com isso, degradam a qualidade do mapa que está sendo construído. Chegará a um ponto em que os valores de ocupação atribuídos às células não influenciam tanto de forma a definir se estão ocupadas, vazias ou não foram ainda mapeadas. Nesse instante, um método de localização absoluta é utilizado (no nosso caso, será simulado) para corrigir os erros de localização da odometria e conseqüentemente indicar ao robô que pode continuar o processo de mapeamento sem considerar os erros acumulados no passado, melhorando assim, a qualidade do mapa que será construído das demais partes do ambiente que ainda não foram mapeadas.

Com isso, o processo de mapeamento continua até que o ambiente seja mapeado por completo.

Capítulo 5

Implementações

Este capítulo trará uma visão geral do sistema robótico utilizado neste trabalho, como também, descreverá com detalhes o algoritmo de mapeamento tomado como base para a realização das modificações propostas.

5.1 O Sistema Robótico

O robô utilizado na implementação desse trabalho é o modelo Pioneer-3AT, nomeado de Galatéia, projetado para se locomover em vários tipos de terrenos, daí a sigla AT (*All Terrain*), traduzido: "Todos os Terrenos". Os robôs da família Pioneer foram desenvolvidos pela *ActivMedia Robotics* para serem plataformas capazes de comportar vários tipos de dispositivos de percepção que facilitassem a realização de tarefas no mundo real [Act 2004].

A *ActivMedia Robotics* projetou uma arquitetura básica comum a todos os robôs da família Pioneer, onde cada robô possui um corpo construído em alumínio, equipado com sistema de freios, motores DC reversíveis, circuitos controladores para os motores, sensores e baterias, todos gerenciados por um controlador embarcado e pelos *softwares* disponibilizados. O pacote de *softwares* possibilita a comunicação do robô com sensores, atuadores e demais dispositivos, bem como dá suporte ao desenvolvimento de aplicações através de sua interface nativa chamada ARIA (*ActivMedia Robotics Interface*). A ARIA possui bibliotecas nas linguagens de programação C++, Java e Python que implementam primitivas de movimentação do robô, de leitura de sensores, bem como permite que alguns dados sobre o estado do robô sejam lidos em tempo de execução. Além disso, a ARIA vem com uma documentação razoavelmente completa.

Ainda dentro do pacote de *softwares*, é possível trabalhar com um simulador chamado MobileSim, com a vantagem de testar as aplicações feitas sem necessitar a presença física do robô. Com o simulador é possível construir ambientes de diferentes formas para servir

como base de testes e experimentos.

Dentre os dispositivos de maior relevância para este trabalho, o robô Galateia é provido de computador embarcado, dois *arrays* de sonares e *encoders* formando o sistema de odometria. As seções seguintes detalham as características desses dispositivos.

5.1.1 Computador embarcado

Montado na parte interna frontal do robô, o computador embarcado é composto pelos seguintes dispositivos:

- Placa mãe VSBC-8 com barramento PC104+;
- Processador Pentium III de 800 MHz;
- 256 MB de memória RAM;
- HD de 20 GB;
- Interface serial RS-232;
- Interface de vídeo para monitor;
- Interface para *mouse* e teclado;
- Placa Ethernet 10/100Base-T;
- Placa de rede sem fio.

O Sistema Operacional instalado pelo fabricante no momento da aquisição do robô, foi o Linux RedHat 7.3, e a comunicação desse com os controladores internos se dá através das interfaces seriais do PC.

5.1.2 Sonares

Os sonares, dispostos em dois grupos de oito sensores, estão distribuídos na parte frontal e traseira do robô, sendo um grupo (também chamado de *array*) em cada parte. A Figura 5.1 ilustra a disposição dos sonares dentro do *array* frontal.

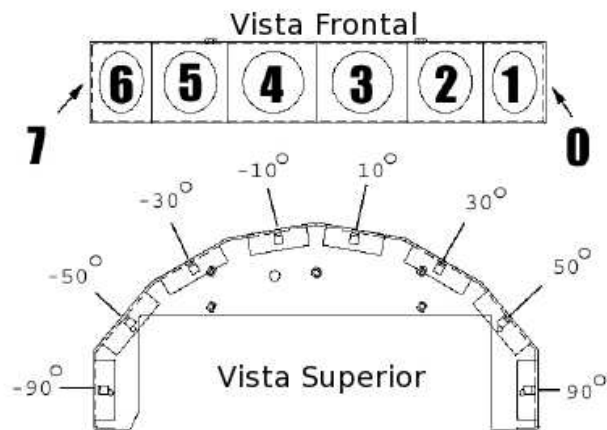


Figura 5.1: Posicionamento dos sonares do *array* frontal.

Cada *array* é constituído de dois sonares laterais, dispostos um em cada lado do robô, e seis sonares separados em intervalos de 20° formando um semi-círculo. Juntos, os dois *arrays* cobrem uma área aproximada a área de círculo com raio de 4m. Os sonares são da família Polaroid 6500, série bastante usada no meio dos roboticistas.

No momento de funcionamento um sonar é disparado por vez em cada *array*, ou seja, haverá dois sonares sendo disparados ao mesmo tempo, um em cada *array*. A taxa de aquisição dos sonares é ajustada normalmente à 25Hz (40 milisegundos por sonar por *array*). A configuração adotada permite que os sonares consigam detectar objetos entre uma distância de 10cm e 4m. Esse intervalo de alcance pode ser ajustado por um fator de sensibilidade que pode ser manipulado por *software*.

O valor da sensibilidade pode ser ajustado em função do ambiente de trabalho do robô. Uma sensibilidade baixa reduz a habilidade de o robô detectar objetos pequenos no ambiente, o que as vezes pode ser desejável. Também é recomendado para ambientes ruidosos e com piso altamente reflexivo, pois com alta sensibilidade corre-se o risco de o piso ser detectado como obstáculo. O aumento da sensibilidade possibilita a detecção de objetos pequenos à distâncias maiores, e é recomendado para ambientes pouco ruidosos e pisos que dificultem a reflexão sonora [Act 2004].

5.1.3 Encoders

O sistema de odometria é composto por dois *encoders* ópticos de alta resolução, um acoplado ao motor direito e o outro ao esquerdo. A resolução do *encoder*, de acordo com o fabricante, é de 22.500 pulsos por rotação resultando em uma relação de 49 pulsos por

milímetro deslocado pela roda.

De acordo com o manual Act (2004) é possível ajustar alguns parâmetros internos (por exemplo, fator de derrapagem) para se conseguir um melhor desempenho do sistema na medição de velocidade e posicionamento.

5.2 O algoritmo de mapeamento

Para validar o método de mapeamento proposto, foi implementado, como estudo inicial o algoritmo de mapeamento em grade apresentado em [Thrun et al. 2005] (ver Algoritmo 1).

Algoritmo 1 $occupancy_grid_mapping(\{l_{t-1,(x,y)}\}, x_t, z_t)$

```

1: for all cells  $m_{x,y}$  do
2:   if  $m_{x,y}$  in perceptual field of  $z_t$  then
3:      $l_{t,(x,y)} = l_{t-1,x,y} + inverse\_sensor\_model(m_{x,y}, x_t, z_t) - l_0$ 
4:   else
5:      $l_{t,(x,y)} = l_{t-1,(x,y)}$ 
6:   end if
7: end for
8: return  $\{l_{t,(x,y)}\}$ 

```

O algoritmo tem como variáveis de entrada uma matriz com todos os valores de ocupação $\{l_{t-1,(x,y)}\}$ atribuídos à grade de ocupação construída até o instante $t - 1$, o vetor de localização do robô $x_t = (x, y, \theta)'$ no instante t e os valores de leitura dos sensores z_t no instante t . Para cada célula $m_{x,y}$ da grade construída é verificado se ela está dentro do campo de visão formado por todos os sonares (linha 2). Caso, esteja, o valor de ocupação da célula será atualizado levando-se em conta o valor de ocupação prévio da célula $l_{t-1,(x,y)}$, o modelo do sensor $inverse_sensor_model(m_{x,y}, x_t, z_t)$ e a constante l_0 que é atribuída a todas as células no início do mapeamento, indicando que não foram mapeadas (linha 3). Caso a célula em questão, $m_{x,y}$, não esteja no campo de visão dos sonares ela permanecerá com o mesmo valor de ocupação (linha 5).

Neste algoritmo, é usado um modelo simplista para o sensor o $inverse_sensor_model$. Esse modelo, apesar de usar técnicas probabilísticas, se comporta de uma forma um tanto determinística. Além disso, não são levados em conta os erros sistemáticos dos sonares e de odometria. O Algoritmo 2 descreve o $inverse_sensor_model$.

O $inverse_sensor_model$ tem como entradas a célula $m_{x,y}$ a ser analisada, o vetor de localização do robô x_t e as medições sensoriais z_t . Primeiro são calculadas a distância e a

Algoritmo 2 *inverse_range_sensor_model*($m_{x,y}, x_t, z_t$)

```

1: Let  $X, Y$  be the center-of-mass of  $m_{x,y}$ 
2:  $r = \sqrt{(X - x)^2 + (Y - y)^2}$ 
3:  $\phi = \text{atan2}(Y - y, X - x) - \theta$ 
4:  $k = \text{argmin}_j |\phi - \theta_{j,sens}|$ 
5: if  $r > \min(z_{max}, z_t^k + \alpha/2)$  or  $|\phi - \theta_{k,sens}| > \beta/2$  then
6:   return  $l_0$ 
7: end if
8: if  $z_t^k < z_{max}$  and  $|r - z_t^k| < \alpha/2$  then
9:   return  $l_{occ}$ 
10: end if
11: if  $r \leq z_t^k$  then
12:   return  $l_{free}$ 
13: end if

```

orientação entre o robô e a célula (linhas 2 e 3 respectivamente), para depois se verificar qual o sensor que possui orientação mais próxima à da célula (linha 4).

Sabendo-se, então, que o sensor k está mais próximo à célula, a linha 5 passa a testar se a célula está fora do cone do sonar k baseando-se na distância z_t^k medida por este sensor e no seu ângulo de orientação $\theta_{k,sens}$ em relação ao robô. Aqui, z_{max} é a distância máxima possível de ser medida pelo sonar, β é a abertura do feixe do sonar e α é um valor que indica a espessura média dos obstáculos no ambiente (por exemplo, paredes de 15cm de espessura $\alpha = 15cm$). Caso o teste seja verdadeiro a célula recebe o valor de ocupação l_0 , indicando que ainda não foi possível mapeá-la. Para simplificar o entendimento, l_0 pode ser avaliado em 0.

Caso o teste feito na linha 5 tenha resultado falso, a linha 8 testa se a distância medida pelo sensor z_t^k é menor que distância máxima z_{max} e se a célula está compreendida na região do obstáculo detectado. Caso verdadeiro, o valor l_{occ} , indicando ocupação é atribuído à célula. A constante l_{occ} pode ser avaliada em 1.

E por fim, se a expressão da linha 8 for falsa, é feito um teste para verificar se a distância entre o robô e a célula é menor que a medição z_t^k do sonar (linha 11). Se verdadeiro, a célula recebe o valor l_{free} indicando que está vazia. A constante l_{free} pode ser avaliada em -1 .

De forma geral, o algoritmo do *inverse_sensor_model* atribui valores constantes para as células que estão dentro do cone do sonar, na extremidade frontal do cone e na região fora no cone.

É importante frisar, que os valores de ocupação das células no Algoritmo 1 são calculados através do *log-odd* ou logaritmo da probabilidade para evitar instabilidades numéri-

cas, como já mencionado no Capítulo 3 seção 3.4.1. Para recuperar os valores de probabilidades pode ser utilizada a Equação 3.4.

Com base no algoritmo descrito acima implementamos o método de mapeamento proposto neste trabalho, cuja principal mudança se encontra na modelagem probabilística dos sensores. O modelo proposto substituiu o *inverse_sensor_model* utilizado no Algoritmo 1.

Capítulo 6

Experimentos e Resultados

Visando testar a metodologia proposta, foram realizados vários experimentos, tanto em simulação quanto usando o robô real. Os experimentos iniciais foram realizados usando a plataforma de simulação MobileSim. Os experimentos práticos foram realizados com dados reais de sonares e odometria tomados e processados em tempo de execução, ou seja, para um mapeamento *on-line*.

6.1 Experimentos realizados em simulação

Com o simulador MobileSim foi possível realizar testes preliminares do algoritmo proposto e avaliar seus resultados. Foi utilizado, como ambiente a ser mapeado, um modelo CAD ¹ que já vem com o simulador, o qual descreve uma das dependências da Universidade de Columbia nos Estados Unidos. A Figura 6.1 mostra a geometria do ambiente no simulador. O robô mapeou uma parte do ambiente seguindo o caminho pontilhado.

Utilizando o algoritmo de mapeamento com suas modificações propostas, fizemos o robô mapear o ambiente até chegar a um ponto em que os erros de odometria degradaram bastante a qualidade do mapa e os valores de ocupação atribuídos às células já não definem se estão ocupadas, vazias ou não foram ainda mapeadas. A Figura 6.2 ilustra esse momento. O robô é representado pelo ponto em vermelho, as regiões em branco são as células vazias, as regiões em preto são as células ocupadas e as regiões em cinza são as células não mapeadas.

Nesse instante, um método de localização absoluta é utilizado para corrigir os erros de localização da odometria e conseqüentemente indicar ao robô que pode continuar o processo de mapeamento sem considerar os erros acumulados no passado, melhorando assim, a qualidade do mapa que será construído das demais partes do ambiente que ainda

¹Do inglês, *Computer-Aided Design*

não foram mapeadas. A Figura 6.3 ilustra o momento no qual a localização do robô é corrigida, e a Figura 6.4 ilustra a continuação do mapeamento após a correção.



Figura 6.1: Visão superior do ambiente a ser mapeado.

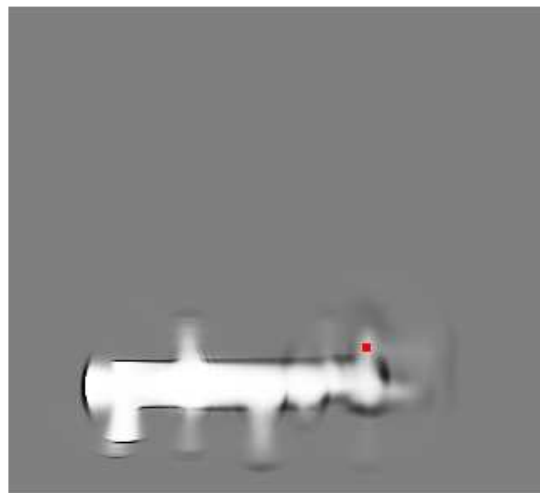


Figura 6.2: Momento em que o mapeamento se tornou inviável.

6.2 Experimentos práticos

Realizamos vários experimentos nas dependências do Departamento de Engenharia de Computação e Automação - DCA - UFRN, cuja geometria é formada basicamente por

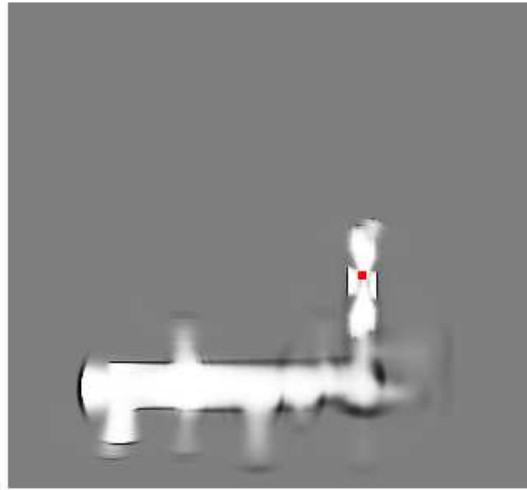


Figura 6.3: Correção da localização por um método absoluto.

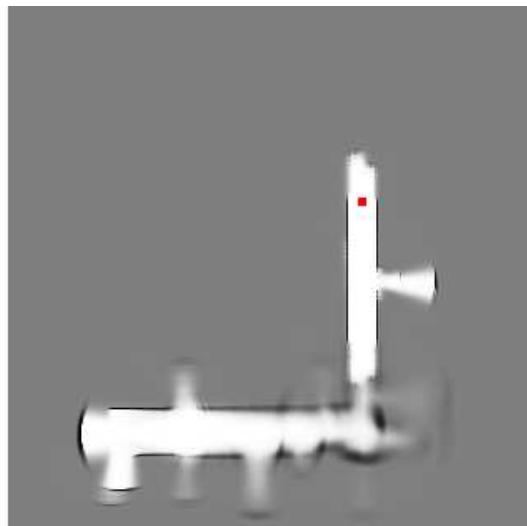


Figura 6.4: Continuação do mapeamento com a localização do robô já corrigida.

corredores e salas retangulares. Apesar do modelo simplista adotado no Algoritmo 1, apresentado no Capítulo 5, foi importante implementá-lo, pois a partir de seus resultados foi possível avaliar a influência que alguns problemas típicos dos sonares acarretam na qualidade do mapa, o que não foi possível nas simulações. Dentre os problemas o mais acentuado foi o das múltiplas reflexões. A Figura 6.5 ilustra o mapa adquirido de um dos corredores do DCA. A linha pontilhada indica a localização real das paredes. É fácil observar que várias medições indicam a existência de obstáculos ou áreas vazias por trás do plano das paredes, ou seja, em áreas que não poderiam ser mapeadas. Essas medições falsas são causadas tipicamente pelas múltiplas reflexões sofridas pelas ondas sonoras dentro do ambiente.

Robôs com formato estrutural mais complexo, como no caso do robô utilizado neste trabalho, são mais susceptíveis às medições falsas. Isso acontece pelo fato de que seus sensores são distribuídos, normalmente, de forma irregular, o que pode facilitar o aparecimento de falsas leituras. Já os robôs em formato de cilindros, geralmente possuem os sensores distribuídos uniformemente pelo seu corpo, o que pode evitar o aparecimento dessas medições falsas [Ivanjko & Petrovic 2005].

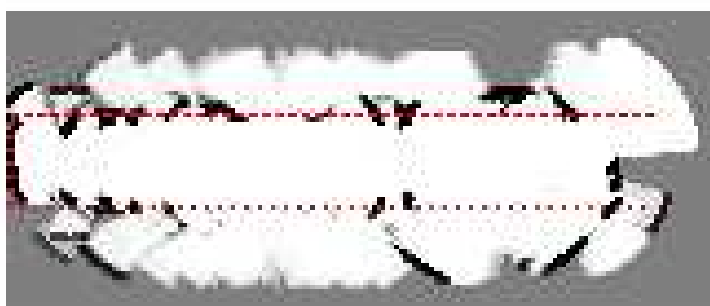


Figura 6.5: Mapa adquirido com modelo simplista do sensor e todos os sonares.

Para tentar solucionar esse problema foi implementada uma estratégia encontrada na literatura para filtragem de medições falsas apresentada por Lee & Chung (2006). Essa estratégia representa as medições dos sonares por círculos, de forma que se uma medição invade a região circular definida por outra medição, a medida da região invadida é eliminada. Os autores nomearam a técnica de *Bubble Circle (BC) Threshold*. Os resultados verificados para este trabalho não foram tão bons quanto esperado, de forma que outras técnicas foram buscadas.

Thrun (2003) aplicou uma técnica baseada em um modelo probabilístico avançado para o sensor, a fim de obter mapas consistentes em seu trabalho. Na técnica utilizada

por ele o problema do mapeamento é visto como um problema de otimização. A grande desvantagem apresentada por ela, é o seu elevado custo computacional, não sendo, portanto, apropriada para mapeamentos realizados em tempo de execução ou *on-line*.

Depois de vários experimentos e observações, foi possível concluir que, em ambientes tipicamente retangulares, formados por corredores estreitos e salas não tão grandes, os mapas mais consistentes eram obtidos pelos sensores laterais que formam um ângulo de 90° com as paredes; e pelos sensores frontais e traseiros com menor ângulo em relação ao eixo do robô. Os demais sensores foram desprezados, visto que, as suas medições eram as que mais resultavam em falsas leituras, devido a disposição angular deles em relação às paredes. Ivanjko & Petrovic (2005) também fizeram considerações parecidas com as adotadas aqui, para melhorar a construção do mapa em grade por um robô Pioneer-2DX.

A "filtragem" adotada reduziu de maneira significativa os efeitos causados pelas falsas leituras no mapa, mesmo com o modelo simplista apresentado no Algoritmo 2 do Capítulo 5. Ainda assim, o mapa adquirido pelos sonares sofre alguma degradação pelas características não desejadas inerentes a esses sensores. A Figura 6.6 ilustra um mapa adquirido depois da seleção de sonares.

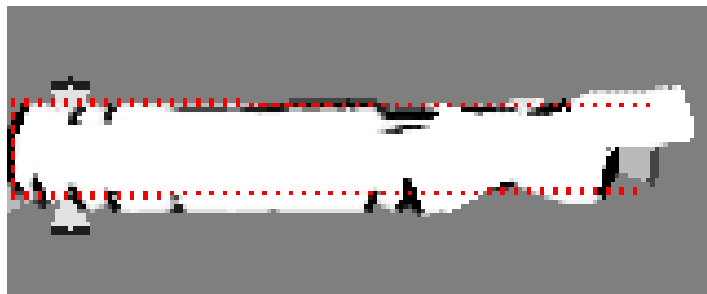


Figura 6.6: Mapa adquirido com modelo simplista do sensor e com seleção de sonares.

Tendo obtido resultados razoáveis na resolução dos problemas mencionados acima e tomando como base o Algoritmo 1 do Capítulo 5, apresentado por Thrun et al. (2005), foram realizados outros experimentos para testar a implementação do algoritmo de mapeamento proposto neste trabalho. O principal diferencial do algoritmo proposto aqui é o modelo probabilístico do sensor, o qual representa no mapa em grade de ocupação as incertezas inerentes ao sistema perceptivo do robô.

O experimento a seguir foi feito em um dos corredores do DCA, com o objetivo de mostrar os efeitos causados pelos erros do sistema de odometria no mapeamento. O mapa adquirido e ilustrado na Figura 6.7 foi totalmente construído baseando-se nos dados de

sonares e odometria sem correção ou tratamento. O contorno pontilhado em vermelho é uma aproximação da posição real das paredes e obstáculos do corredor. O ponto em vermelho na região branca indica a localização final do robô no mapa. É fácil verificar que as dimensões reais diferem bastante da forma real do ambiente, isso se dá devido os erros de odometria. Logo, esse não seria um mapa confiável para ser utilizado na realização de outras tarefas.



Figura 6.7: Mapa sendo construído sem considerar (representar) os efeitos dos erros de odometria.

Com o método proposto é possível modificar a representação do mapa de forma que, os erros de odometria sejam incorporados dando um efeito negativo a qualidade do mapa. A Figura 6.8 ilustra o processo de mapeamento do mesmo corredor, ainda na sua fase inicial, mas já com alguma degradação causada pelos erros da odometria. O contorno pontilhado em vermelho indica a posição real das paredes do corredor. O ponto em vermelho na região branca indica a localização do robô no mapa e o ponto na extremidade direita da figura indica onde o mapeamento deve parar.

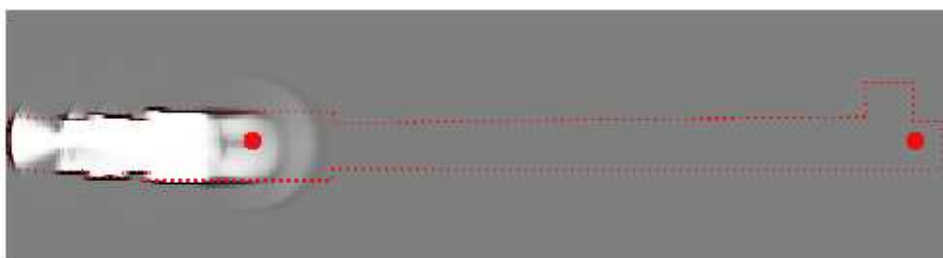


Figura 6.8: Mapa sendo construído com modelo proposto, considerando a representação dos erros de odometria.

A Figura 6.9 mostra a evolução do mapeamento. Observe que a qualidade do mapa

diminui com o aumento do deslocamento que, por sua vez, aumenta os erros de odometria.

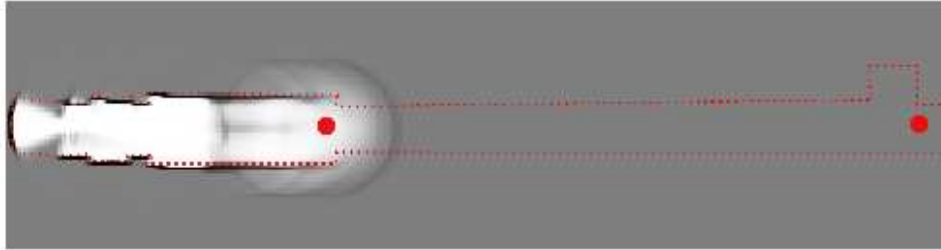


Figura 6.9: Representação do aumento dos erros de odometria.

No mapa ilustrado pela Figura 6.10 começa a haver uma melhoria na qualidade do mapa. Nesse ponto foi simulado um método de localização absoluta, devido o mapa estar fortemente degradado. A localização absoluta corrige a localização do robô com precisão e como consequência, zera os erros de odometria.

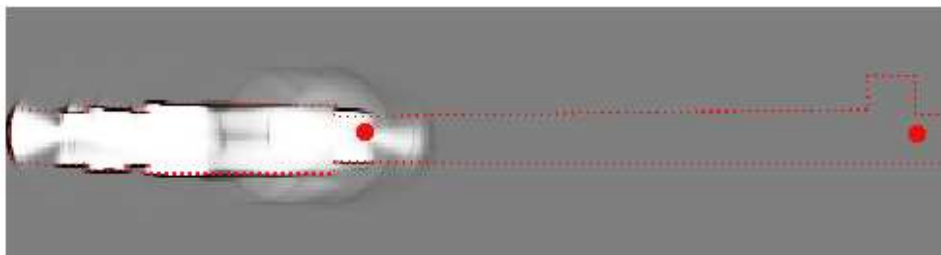


Figura 6.10: Correção da localização do robô por um método de localização absoluta.

O mapeamento segue até chegar ao ponto ilustrado pela Figura 6.11, onde será necessária uma nova correção pelo método de localização absoluta. Seguindo depois até chegar ao final do mapeamento ilustrado pela Figura 6.12

O mesmo experimento foi realizado, mas desta feita com a correção dos erros de odometria modelados no Capítulo 2. A Figura 6.13 ilustra o mapa final adquirido. É difícil notar alguma degradação, pois os erros, após a correção, se tornaram muito pequenos e o ambiente mapeado não era grande o bastante para que se fosse possível perceber um aumento considerável dos efeitos dos erros não-sistemáticos. Mas ainda assim, é possível verificar um desvio no mapeamento do corredor. Uma modelagem da odometria mais completa pode solucionar esse problema.

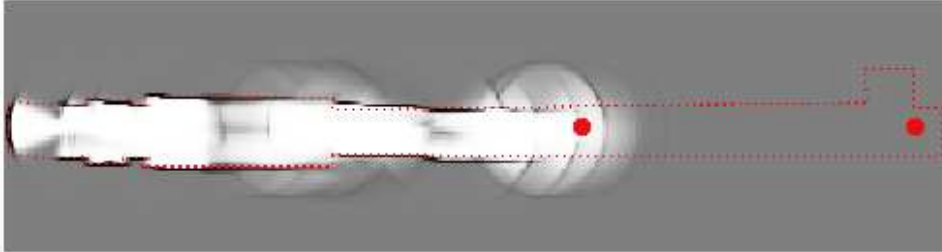


Figura 6.11: Nova correção da localização.

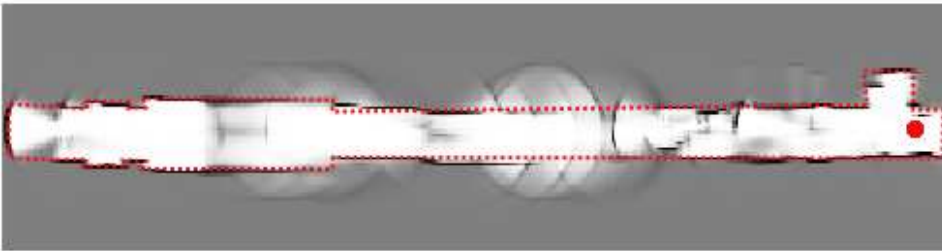


Figura 6.12: Mapa mapa final adquirido com o modelo proposto.



Figura 6.13: Mapa construído com o modelo proposto e como correção dos erros sistemáticos de odometria.

Capítulo 7

Conclusões

Neste trabalho foi proposto um método de mapeamento com representação espacial do ambiente em grade de ocupação que incorpora um modelo probabilístico para sonares que leva em consideração as incertezas inerentes a esses sensores, bem como os erros acumulativos do sistema de odometria do robô. Com isso, a qualidade do mapa adquirido pelo robô, através do processo de mapeamento em grade de ocupação, passa a ser influenciado pelas incertezas presentes no sistema de odometria, podendo assim indicar até que ponto o robô pode confiar na veracidade das informações colhidas no mapa.

Com o mapa adquirido o robô móvel poderá utilizá-lo como fonte preliminar de informações na execução de outras tarefas como navegação, planejamento, de trajetória, tomada de decisão, entre outras.

Este trabalho foi o primeiro implementado de fato no robô Galatéia explorando suas capacidades sensoriais que até então não tinham sido totalmente desbravadas. O sistema de mapeamento implementado aqui, fará parte de um sistema mais robusto de navegação e mapeamento, o qual integrará um sistema de visão estéreo como mais uma fonte de informações. Além disso, essa pesquisa também contribuirá em futuras pesquisas na área de mapeamento com time de robôs.

A partir dos experimentos realizados conclui-se que o sistema de mapeamento proposto neste trabalho fornece uma maneira mais realística de representar um ambiente mapeado usando a técnica de grade de ocupação, conhecendo-se que as informações originadas tanto dos sonares quanto do sistema de odometria possuem erros. Mesmo com as dificuldades encontradas nas limitações dos sonares o sistema apresentou resultados satisfatórios. Outros tipos de sensores de alcance poderiam ser utilizados no sistema e com o modelo proposto, como por exemplo, os *lasers* com grande probabilidade de apresentar mapas ainda mais consistentes.

Como atividades futuras, pretende-se fazer um estudo das técnicas e heurísticas de exploração para tornar o processo de mapeamento autônomo. Além disso, serão estudadas

formas de melhorar a localização do robô dentro do ambiente com a incorporação de outros sensores, que juntos, poderão melhorar qualitativamente os mapas. Atividades futuras com ênfase em Localização e Mapeamentos Simultâneos (SLAM) também serão realizadas tomando como base o trabalho exposto nesse documento.

Referências Bibliográficas

Act (2004), *Pioneer 3 - Operations Manual*.

Alsina, Pablo Javier, Luiz Marcos Garcia Gonçalves, Frederico Carvalho Vieira, Diogo Pinheiro Fernandes Pedrosa & Adelardo Adelino Dantas de Medeiros (2002), 'Navegação e controle de robôs móveis', CBA - Congresso Brasileiro de Automática. Natal, RN.

Bezerra, Clauber Gomes (2004), Localização de um robô móvel usando odometria e marcos naturais, Dissertação de mestrado, Universidade Federal do Rio Grande do Norte, Natal, RN.

Biswas, R., B. Limketkai & S. Thrun (2002), Towards object mapping in dynamic environments with mobile robots, *em* 'Proceedings of the Conference on Intelligent Robots and Systems', Lausanne, Switzerland.

Borenstein, J. & Y. Koren (1997), 'The vector field histogram - fast obstacle avoidance for mobile robots', *IEEE Journal of Robotics and Automation* **7**(3), 278–288.

Borenstein, Johan, R. H. Everett & Liqiang Feng (1996), *Where am I? Sensors and Methods for Mobile Robot Positioning*, 1ª edição, University of Michigan, Michigan, EUA.

Chenavier, Frederic & James L. Crowley (1992), Position estimation for a mobile robot using vision and odometry, *em* 'Proceeding of the 1992 IEEE International Conference on Robotics and Automation', Nice, France.

Chong, Kok Seng & Lindsay Kleeman (1997), Accurate odometry and error modeling for a mobile robot, *em* 'IEEE International Conference on Robotics & Automation', Albuquerque, New Mexico.

Cox, Ingemar J. (1991), 'Blanche - an experiment in guidance and navigation of an autonomous robot vehicle', *IEEE Transactions on Robotics and Automation* **7**(2).

Cox, Ingemar J. & Winston L. Nelson (1988), Path control for an autonomous vehicle, *em* 'IEEE International Conference on Robotics & Automation'.

- Doh, Nakju, Howie Choset & Wan Kyun Chung (2003), Accurate relative localization using odometry, *em* 'IEEE International Conference on Robotics & Automation', Taipei, Taiwan, pp. 14–19.
- Dutra, Paulo R. C., Márcio M. de Souza, Gabriel F. Andriolli, Alberto J. Álvares & João C. E. Ferreira (2003), Navmap: Um sistema para navegação por mapeamento do robô móvel nomad xr4000, *em* 'VI Simpósio brasileiro de Automação Inteligente (SBAI)', Bauru, SP.
- Elfes, Alberto (1987), 'Sonar-based real-world mapping and navigation', *IEEE Journal of Robotics and Automation* **3**(3), 249–265.
- Elfes, Alberto (1989a), Occupancy Grind: A Probabilistic Framework for Robot Perception and Navigation, Tese de doutorado, Carnegie Mellon University, Pittsburg, Pennsylvania, USA.
- Elfes, Alberto (1989b), 'Using occupancy grids for mobile robot perception and navigation', *Computer* **6**(22), 46–57.
- Fabrizi, Elisabetta & Alessandro Saffioti (2000), Extracting topology-based maps from gridmaps, *em* 'IEEE International Conference on Robotic and Automation', San Francisco, CA, USA, pp. 2972–2978.
- Frese, Udo (2004), An $O(\log n)$ Algorithm for Simultaneous Localization and Mapping of Mobile Robots in Indoor Environment, Tese de doutorado, University of Erlangen-Nürnberg.
- Grisetti, Giorgio, Cyrill Stachniss & Wolfram Burgard (2007), 'Improved techniques for grid mapping with rao-blackwellized particle filter', *IEEE Transactions on Robotics and Automation* **23**(1).
- Howard, A. & L. Kitchen (1996), 'Sonar mapping for mobile robots'.
URL: citeseer.ist.psu.edu/howard96sonar.html
- Irie, Masahiro, Keiji Nagatani & Akio Gofuku (2003), Path evaluation for a mobile robot based on a risk of collision, *em* 'IEEE International Symposium on Computational Intelligence in Robotics and Automation', Kobe, Japan.
- Ivanjko, Edouard & Ivan Petrovic (2005), Experimental evaluation of occupancy grids maps improvement by sonar data correction, *em* '13th Mediterranean Conference on Control and Automation', Limassol, Cyprus.

- Ivanjko, Edouard, Mario Vasak & Ivan Petrovic (2005), Kalman filter theory based mobile robot pose tracking using occupancy grid maps, *em* 'International Conference on Control and Automation', Budapest, Hungria.
- Kong, Fatian, Youping Chen, Jingming Xie, Gang Zhang & Zude Zhou (2006), Mobile robot localization based on extended kalman filter, *em* '6th Congress on Intelligent Control and Automation', Dalian, China.
- Konolige, Kurt (1997), 'Improved occupancy grids for map building', *Autonomous Robots* (4), 351–367.
- Kuypers, B. J. & Y. T. Byun (1991), 'A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations', *Journal of Robotics and Autonomous Systems* 8(8), 47–63.
- Lee, Kyoungmin & Wan Kyun Chung (2006), Filtering out specular reflections of sonar sensor readings, *em* 'The 3rd International Conference on Ubiquitous Robots and Ambient Intelligent (URAI)'.
- Lee, Yu-Cheol, Sang-Ik Nah, Hyo-Sung Ahn & Wonpil Yu (2006), Sonar map construction for a mobile robot using a tethered-robot guiding system, *em* 'The 6rd International Conference on Ubiquitous Robots and Ambient Intelligent (URAI)'.
- Leonard, Jonh J. & Hugh F. Durrant-Whyte (1992), *Direct Sonar Sensing for Mobile Robot Navigation*, 1ª edição, Kluwer Academic Plublishers, Boston, EUA.
- Martinelli, Agostino & Roland Siegwart (2003), Estimating the osometry error of a mobile robot during navigation, *em* 'European Conference on Mobile Robots, ECMR'.
- Moravec, Hans P. (1988), 'Sensor fusion in certainty grids for mobile robots', *Computer* 9(2), 61–74.
- Moravec, Hans P. (1996), Robot spacial perception by stereoscopic vision on 3d evidence grid, Relatório Técnico CMU-RI-TR-96-34, CMU Robotics Institute, Pittsburgh, Pennsylvania.
- Murphy, Robin R. (2000), *Introduction to AI Robotics*, 1ª edição, The MIT Press, Cambridge, Massachusetts.
- NASA (2007), Página na internet, National Aeronautics and Space Administration, EUA.
URL: www2.jpl.nasa.gov/basics/sj.html

- Negenborn, Rudy (2003), Robot localization and kalman filters: On finding your positions in a noisy world, Dissertação de mestrado, Utrecht University.
- Oriolo, G., G. Ulivi & M. Vendittelli (1997), ‘Fuzzy maps: a new tool for mobile robot perception and planning’, *Robotics Systems* **14**(3), 179–197.
- Pedrosa, Diogo P. F. (2006), Mapeamento de Ambientes Estruturados com Extração de Informações Geométricas Atavés de Dados Sensoriais, Tese de doutorado, UFRN, Natal, RN.
- Ribo, Miguel & Axel Pinz (2001), ‘A comparison of three uncertainty calculi for building sonar-based occupancy grids’, *Robotics and Autonomous Systems* **31**, 201–209.
- ROBOTCLEAN* (2007), Página na internet, Alemanha.
URL: <http://www.robocleaner.de/>
- Santana, André Macedo (2007), Localização e planejamento de caminhos para um robô humanoíde e um robô escravo com rodas, Dissertação de mestrado, Universidade Federal do Rio Grande do Norte, UFRN, Natal, RN.
- Siegwart, Roland & Illah R. Nourbakhsh (2004), *Introduction to Autonomous Mobile Robot*, 1ª edição, MIT Press, Cambridge, Massachusetts, EUA.
- Thrun, Sebastian (1998), ‘Learning metric-topological maps for indoor mobile robot navigation’, *Artificial Intelligence* **99**(1), 21–71.
- Thrun, Sebastian (2002), Robotic mapping: A survey, *em* G.Lakemeyer & B.Nebel, eds., ‘Exploring Artificial Intelligence in the New Millenium’, Morgan Kaufmann, pp. 163–175.
- Thrun, Sebastian (2003), ‘Learning occupancy grig maps with forward sensor models’, *Autonomous Robots* **15**, 111–127.
- Thrun, Sebastian, Dieter Fox, Wolfran Burgard & Frank Dellaert (2001), ‘Robust monte carlo localization for mobile robots’, *Artificial Intelligence* .
- Thrun, Sebastian, Wolfram Burgard & Dieter Fox (2005), *Probabilistic Robotics*, 1ª edição, MIT Press, Cambridge, Massachusetts, EUA.