

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

KARLA FEDRIZZI MACHADO

**Módulo de Auto-Localização para um
Agente Exploratório usando Filtro de
Kalman**

Dissertação apresentada como requisito parcial
para a obtenção do grau de
Mestre em Ciência da Computação

Prof. Dr. Paulo Martins Engel
Orientador

Prof. Dr. Marco A. P Idiart
Co-orientador

Porto Alegre, junho de 2003

CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Fedrizzi Machado, Karla

Módulo de Auto-Localização para um Agente Exploratório usando Filtro de Kalman / Karla Fedrizzi Machado. – Porto Alegre: Programa de Pós-Graduação em Computação, 2003.

64 f.: il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2003. Orientador: Paulo Martins Engel; Coorientador: Marco A. P. Idiart.

1. Navegação exploratória. 2. Mapeamento de ambientes. 3. Filtro de Kalman. 4. Robótica móvel. 5. Auto-localização. I. Engel, Paulo Martins. II. Idiart, Marco A. P. III. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Prof^a. Wrana Maria Panizzi

Pró-Reitor de Ensino: Prof. José Carlos Ferraz Hennemann

Pró-Reitora Adjunta de Pós-Graduação: Prof^a. Jocélia Grazia

Diretor do Instituto de Informática: Prof. Philippe Olivier Alexandre Navaux

Coordenador do PPGC: Prof. Carlos Alberto Heuser

Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

“O que o labirinto nos ensina não é onde está a saída,
mas quais os caminhos que não levam a lugar nenhum.”

— NORBERTO BOBBIO

AGRADECIMENTOS

Várias são as pessoas que merecem ser citadas nesse agradecimento. Em primeiro lugar, meus pais e irmãos que estiveram presentes em muitos momentos difíceis e souberam dar toda a infra-estrutura para que eles passassem. Em segundo lugar, ao Dr. Edson Prestes que durante quase a totalidade deste trabalho era ainda um doutorando e esteve sempre me encorajando para que não desistisse. Em terceiro lugar, um agradecimento muito especial ao Professor Marco Idiart pelo incansável e valioso empenho em desvendar os mistérios físicos do Filtro de Kalman. Ao meu orientador pela paciência e por compreender as minhas ausências devido ao trabalho, muito obrigado. Ao meu noivo, Daniel Viero, por todo o carinho e cuidado que proporcionou na etapa final deste trabalho. E por fim, aos órgãos Capes, Fapergs e CNPq por fornecerem os robôs ao Laboratório de Robótica Inteligente e ajudarem tantos estudantes, nas mais diversas áreas, a desenvolverem seus projetos com o auxílio das bolsas.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS	7
LISTA DE SÍMBOLOS	8
LISTA DE FIGURAS	10
RESUMO	11
ABSTRACT	13
1 INTRODUÇÃO	14
1.1 Motivação	15
1.2 Objetivos	15
1.3 Estrutura da Dissertação	16
2 MAPEAMENTO E EXPLORAÇÃO DE AMBIENTES	17
2.1 Representação do Ambiente	18
2.1.1 Mapas Métricos	18
2.1.2 Mapas Topológicos	19
2.1.3 Mapas Probabilísticos	20
2.2 Os Sensores e o Ambiente	20
2.2.1 Odômetro	21
2.2.2 Câmera de Vídeo	21
2.2.3 Sonar	21
2.3 Navegação	22
2.4 Auto-Localização	23
2.5 Mapeamento e Auto-Localização Simultâneos	24
3 EXPLORAÇÃO DE AMBIENTES USANDO MÉTODO DE CAMPOS POTENCIAIS	25
3.1 Método Exploratório	25
3.1.1 Sistema Sensorial	25
3.1.2 Representação do Ambiente	26
3.1.3 Atualização do Mapa	26
3.1.4 Campos Potenciais Harmônicos	27
3.2 Arquitetura do Agente Exploratório	28

4	FILTRO DE KALMAN	31
4.1	Histórico	31
4.2	Definições	32
4.2.1	Definição de Modelo Determinístico	32
4.2.2	Definição de Ruído	32
4.2.3	Definição do Filtro de Kalman	33
4.2.4	Fases do Filtro	36
4.3	Aplicação na Robótica Móvel	36
4.3.1	Usando as Matrizes	39
4.3.2	Adaptações de implementação	41
5	MÓDULO DE AUTO-LOCALIZAÇÃO	43
5.1	O Simulador do Agente Exploratório	43
5.2	Motivação do uso de auto-localização	44
5.3	Associação com o Campo Potencial	46
5.3.1	Implementação	47
5.3.2	Marcas de Kalman	48
5.4	Experimentos	49
5.4.1	Deformação do Mapa	49
5.4.2	Exploração com Marcas de Kalman	49
5.4.3	Identidade das Marcas de Kalman	50
6	CONCLUSÃO	58
6.1	Trabalhos Futuros	59
	REFERÊNCIAS	61

LISTA DE ABREVIATURAS E SIGLAS

FK	Filtro de Kalman
FKE	Filtro de Kalman Estendido
GPS	Sistema Global de Posicionamento
HIMM	<i>Histogrammic In-Motion Mapping</i>
LRI	Laboratório de Robótica Inteligente
PVC	Problemas de Valores de Contorno
MLS	Mapeamento e Localização Simultâneos
RMA	Robótica Móvel Autônoma

LISTA DE SÍMBOLOS

a	Ação executada pelo robô
d_i	Distância entre o robô e o objeto i
m_i	Medida ao tempo i
n	Número de medidas
q^2	Variância de x
t	Tempo discreto
v_v	Erro na velocidade do robô
v_γ	Erro no ângulo de rotação do robô
w_d	Erro na medida de d_i
w_θ	Erro na medida de θ_i
x_i	Coordenada x do objeto i
x_m	Média dos valores de x
x_r	Coordenada x do robô
y_i	Coordenada y do objeto i
y_r	Coordenada y do robô
N	Número de objetos
V	Velocidade do robô
γ	Ângulo de rotação
$\delta\mathbf{x}$	Erro da estimativa do estado do sistema
$\delta\mathbf{z}$	Erro da estimativa da medida do sistema ou inovação
θ_i	Ângulo entre a orientação do robô e o objeto i
ϕ	Ângulo de orientação do robô
Δx_i	Diferença entre x_r e x_i
Δy_i	Diferença entre y_r e y_i
σ_v	Nível de ruído na velocidade
σ_γ	Nível de ruído no ângulo de rotação

f	Modelo do sistema
h	Modelo de medida do sistema
u	Ação executada pelo robô
v	Ruído associado ao modelo do sistema
w	Ruído associado ao modelo de medida do sistema
x	Vetor de estado do sistema
\hat{x}	Estimativa do estado do sistema
z	Vetor de medida do sistema
\hat{z}	Estimativa da medida do sistema
K	Matriz do ganho de Kalman
P	Matriz covariância do estado do sistema
Q	Matriz covariância do erro do sistema
R	Matriz covariância do erro da medida do sistema
S_{xz}	Matriz covariância entre o erro do estado e a inovação
S_{zz}	Matriz covariância da inovação

LISTA DE FIGURAS

Figura 3.1:	Exploração de um ambiente usando o campo potencial.	28
Figura 3.2:	Fluxograma da Exploração	29
Figura 3.3:	Arquitetura do Agente Exploratório (PRESTES, 2003)	30
Figura 4.1:	Aplicação típica usando Filtro de Kalman	32
Figura 4.2:	Estimativa <i>a priori</i> e <i>a posteriori</i>	34
Figura 4.3:	Processo de estimativa <i>a priori</i> e <i>a posteriori</i>	36
Figura 4.4:	Fases do Filtro de Kalman	37
Figura 4.5:	Processo de observação do ambiente	38
Figura 5.1:	Tela do Simulador com um ambiente montado pelo usuário.	44
Figura 5.2:	Deformação do mapa em quatro simulações de um mesmo ambiente, com diferente nível de ruído no ângulo de orientação do robô.	46
Figura 5.3:	Fluxograma do Filtro de Kalman no simulador	47
Figura 5.4:	Níveis de mapeamento	49
Figura 5.5:	Sobreposição do ambiente simulado com o campo potencial de quatro explorações de um mesmo ambiente, com diferentes níveis de ruído no ângulo de orientação do robô.	51
Figura 5.6:	Campo potencial de quatro explorações de um mesmo ambiente, com diferentes níveis de ruído no ângulo de orientação do robô.	52
Figura 5.7:	Erro e erro médio da posição do robô durante a exploração.	53
Figura 5.8:	Sobreposição do ambiente simulado com o campo potencial utilizando marcas de Kalman	54
Figura 5.9:	Mapa gerado com as marcas de Kalman	54
Figura 5.10:	Erro e erro médio da posição do robô durante a exploração com marcas de Kalman	55
Figura 5.11:	Sobreposição do ambiente simulado com o campo potencial utilizando marcas de Kalman sem e com identidade	56
Figura 5.12:	Mapa gerado com as marcas de Kalman sem e com identidade	56
Figura 5.13:	Erro e erro médio da posição do robô durante a exploração com marcas de Kalman sem e com identidade	57

RESUMO

Construir um robô capaz de realizar tarefas sem qualquer interferência humana é um dos maiores desafios da Robótica Móvel. Dispondo apenas de sensores, um robô autônomo precisa explorar ambientes desconhecidos e, simultaneamente, construir um mapa confiável a fim de se localizar e realizar a tarefa. Na presença de erros de odometria, o robô não consegue se auto-localizar corretamente em seu mapa interno e acaba por construir um mapa deformado e não condizente com a realidade.

Este trabalho apresenta uma solução para o problema da auto-localização de robôs móveis autônomos. Esta solução faz uso de um método linear de cálculo de estimativas chamado Filtro de Kalman para corrigir a posição do robô em seu mapa interno do ambiente enquanto realiza a exploração.

A proposta leva em consideração que toda entidade que se movimenta em um ambiente conta sempre com alguns pontos de referência para se localizar. Estes pontos são implementados como objetos especiais chamados *marcas de Kalman*. Em simulação, o reconhecimento das marcas pode ser feito de duas maneiras: através de sua posição no mapa ou através de sua identidade.

Nos experimentos realizados em simulação, o método é testado para diferentes erros no ângulo de orientação do robô. Os resultados são comparados levando em consideração as deformações no mapa gerado, com e sem marcas de Kalman, e o erro médio da posição do robô durante todo o processo exploratório.

Palavras-chave: Navegação exploratória, mapeamento de ambientes, filtro de Kalman, robótica móvel, auto-localização.

Self-Localization Module for Exploratory Agent using Kalman Filter

ABSTRACT

Build a robot capable of performing tasks without any human interference is one of the biggest challenges of the Mobile Robotics. Having only sensors, an autonomous robot needs to explore unknown environments and, simultaneously, build a reliable map in order to get its own location and perform the task. In the presence of odometry errors, the robot is not capable of establish its own position on its internal map and ends up building a deformed map that does not reflect reality.

This paper presents a solution for the problem of self-localization of autonomous mobile robots. This solution uses a linear method for calculating estimatives called *Kalman Filter* to correct the robot's position on its internal mapping of the environment while exploring.

The proposal considers that any being that moves in an environment always counts on having some reference points to establish its own position. This points are implemented as special objects called *Kalman landmarks*. In simulation, the recognition of such landmarks can be done in two different ways: through its position on the map or through its identity.

In the experiments performed in simulations, the method is tested for different errors in the robot's inclination angle. The results are compared considering the deformations on the generated map, with and without the Kalman landmarks, and the average error of the robot's position during the exploratory process.

Keywords: Exploratory Navigation, Environment Mapping, Kalman Filter, Mobile Robot, Self-localization.

1 INTRODUÇÃO

A ciência e a ficção científica sempre tiveram sonhos audaciosos com relação ao futuro. Neles, as máquinas criadas pelo homem seriam capazes de realizar qualquer tarefa com a perfeição humana e, em alguns casos, poderiam até superá-la. O mundo seria completamente adaptado para facilitar a existência do homem na Terra.

A Era Tecnológica tornou realidade muitas das suposições a respeito do futuro. Trouxe os computadores, capazes de processar milhares de informações em poucos minutos; os celulares, que permitem ao homem se comunicar em qualquer lugar que esteja; as máquinas que controlam outras máquinas na automação industrial. Com certeza, se fosse possível viajar no tempo, um viajante do século passado se surpreenderia com o mundo de hoje.

Não resta dúvidas de que o homem construiu máquinas realmente capazes de realizar várias tarefas no seu lugar. Desde uma simples calculadora até um robô que chega ao solo lunar e, de lá, traz material para a pesquisa na Terra, as máquinas tornaram-se fortes aliadas do homem. Porém para a ficção científica os sonhos são bem mais fáceis de realizar do que para a ciência.

Os chips evoluíram, novos materiais foram descobertos, porém ainda há muito o que se pesquisar para chegar ao humanóide dos filmes. A Robótica, área da ciência que estuda os robôs, tem como principal desafio hoje construir máquinas capazes de realizar tarefas sem qualquer intervenção humana, ou seja, máquinas com total *autonomia*.

Estas máquinas autônomas realizariam tarefas consideradas repetitivas ou mecânicas, tarefas que exigissem rapidez na execução ou que precisariam ser realizadas em locais de grande risco para o homem, como um prédio em chamas, o interior repleto de gases venenosos de uma pirâmide ou a atmosfera desconhecida de um novo planeta.

Basicamente, um robô autônomo precisa de um sistema sensorial que lhe permita perceber o ambiente em que se encontra (detectar obstáculos, paredes, buracos, etc), um conjunto de ações que ele possa executar para realizar a tarefa e uma maneira de decidir quando executar cada uma destas ações.

Além disso, com a tecnologia atual, o robô precisaria operar em ambientes que pudessem ser definidos por uma ou mais funções matemáticas. Infelizmente, não se consegue essa definição por completo em ambientes reais. Ocorre que, em um ambiente real, o robô sofre inúmeras influências do meio. Estas influências têm diversas fontes e atuam de forma não controlada sobre o robô podendo impedir que a tarefa seja realizada com êxito. São exemplos de influências do meio: problemas mecânicos na locomoção do robô, interferência na leitura dos sensores, erros de arredondamento nos sistemas de controle.

Enfim, é preciso encontrar mecanismos para correção de erros, principalmente no que diz respeito aos erros do mapa interno do ambiente, necessário para locomoção do

robô, e aos erros na localização do robô nesse mapa.

1.1 Motivação

A motivação principal deste trabalho vem do Agente Exploratório proposto por Edson Prestes (PRESTES, 2003). Este agente foi o primeiro trabalho desenvolvido no Programa de Pós-Graduação em Computação da Universidade Federal do Rio Grande do Sul utilizando o robô Nomad 200. O trabalho define um método exploratório que é a união de diferentes estratégias - planejamento de caminhos livres de obstáculos em ambientes conhecidos e a aquisição de mapas por um processo exploratório - através de um mesmo princípio: solução de problemas de valores de contorno (PVC). Este método é uma extensão do método de planejamento de caminhos baseado em funções harmônicas (CONNOLLY; GRUPEN, 1993).

Para se localizar no mapa do ambiente, o Agente Exploratório usa somente as informações oriundas de seus odômetros. Sabe-se que erros de odometria são aditivos no tempo, ou seja, a cada interação aumenta a diferença entre a posição real do robô e a posição indicada pelas funções odométricas. Essa diferença compromete seriamente a performance do agente durante a exploração e mapeamento simultâneos do ambiente.

Em (PRESTES, 2003) o erro de odometria foi minimizado pelo uso de campos potenciais que geravam trajetórias suaves, sem mudanças abruptas de direção, e por um ajuste cuidadoso e gradual das acelerações do robô, tanto linear como angular, durante a realização do movimento. Apesar de todo o cuidado, explorações muito longas geraram erros significativos de odometria.

Diante disso, é necessário incorporar um módulo de auto-localização no Agente Exploratório a fim de que sejam minimizados os erros de odometria durante a exploração de ambientes desconhecidos.

Uma segunda motivação é aprofundar o conhecimento na área de Robótica Móvel Autônoma (RMA). Primeiramente estudando uma técnica para auto-localização muito usada pelos pesquisadores da área conhecida por Filtro de Kalman (CSORBA, 1997; GUIVANT; NEBOT, 2001; DISSANAYAKE et al., 2001) (WELCH; BISHOP, 2001; WELCH, 2001; ARAÚJO, 2000; ZANARDI; LÓTA, 2001). Posteriormente aplicando de forma prática o que foi estudado, implementando algoritmos para que possam ser utilizados em um robô real.

Outra fonte de motivação foi a existência dos robôs do Laboratório de Robótica Inteligente (LRI): o robô Khepera que possui sensores do tipo infravermelho e o robô Nomad 200 que possui vários tipos de sensores, entre sonar, câmera, infravermelho e tátil. Futuramente, serão estes os robôs que utilizarão o método, em especial o Nomad 200 por já possuir o sensor do tipo câmera.

1.2 Objetivos

O objetivo deste trabalho é propor uma técnica para implementar o módulo de auto-localização de um Agente Exploratório (PRESTES, 2003). Este módulo é o responsável por corrigir a posição do robô em seu mapa interno enquanto realiza a exploração do ambiente. Se o robô apresenta erro em sua localização, conseqüentemente, o mapa gerado não será condizente com o mapa real.

Desta forma, num primeiro momento, é preciso estudar as técnicas para auto-localização. Foram escolhidas como objeto de estudo, técnicas que utilizam um estimador

linear ótimo, conhecido como Filtro de Kalman (FK).

Num segundo momento, implementar o módulo em simulação, associando a auto-localização ao método de exploração já implementado. O simulador possibilita criar ambientes para simulação, visualizar a exploração e o mapa gerado, alterar parâmetros de erros e de sistema. Além disso, realiza a comunicação com um robô real, o minirobô Khepera.

E, finalmente, testar a eficiência do módulo de auto-localização em simulação e descobrir suas limitações para a aplicação na Robótica Móvel Autônoma.

1.3 Estrutura da Dissertação

Esta Dissertação está estruturada da seguinte maneira. O Capítulo 2 apresenta uma introdução aos principais componentes do mapeamento e da exploração de ambientes por robôs móveis. O Capítulo 3 resume o método de exploração e apresenta a arquitetura do agente exploratório. O Capítulo 4 apresenta as definições do Filtro de Kalman e mostra algumas adaptações de implementação que tiveram bons resultados de otimização. O Capítulo 5 descreve o módulo de auto-localização, como foi implementado e os experimentos realizados. O último capítulo discute os resultados, apresenta as conclusões e uma lista de trabalhos futuros.

2 MAPEAMENTO E EXPLORAÇÃO DE AMBIENTES

Quando se encontra em um lugar completamente desconhecido, o homem realiza, sem perceber, um processo de mapeamento e exploração. A mente humana vai montando uma espécie de mapa (MAGUIRE; BURGESS; O'KEEFE, 1999) deste ambiente, localizando entidades como mesas, cadeiras, paredes e guardando a posição, a forma e a cor de cada uma delas para um posterior reconhecimento.

Por outro lado, quando percorre um ambiente já conhecido, a mente humana compara o que vê com o mapa que está armazenado. É neste momento que o homem percebe objetos novos, objetos que mudaram de lugar, de cor ou que simplesmente desapareceram. Ele faz essa comparação também de forma natural, utilizando principalmente o sentido da visão.

Mas e se o homem não puder contar com todos os seus sentidos? Um exemplo disso é percorrer uma sala completamente às escuras. A tarefa se torna bem mais complicada se a sala for completamente desconhecida, ou seja, a mente não possui aquele mapa para reconhecimento.

Assim, é possível fazer uma analogia ao que ocorre com um robô ao percorrer ambientes. Os sensores são como os sentidos humanos, captam as informações do ambiente. Porém, em comparação, os sensores possuem capacidade muito inferior. Por vezes é preciso utilizar mais de um tipo de sensor para conseguir dados mais precisos do ambiente que se está percorrendo.

O mapa mental gerado pelo homem é, para o robô, a representação interna do ambiente. Essa representação pode ser feita de diversas maneiras, como será mostrado mais adiante neste capítulo.

Mas para que tenha autonomia, o robô necessita definir um conjunto de comportamentos possíveis, também chamado de conjunto de ações, que permitirão que atue no ambiente. As ações são tomadas baseadas no conhecimento adquirido durante a exploração e o critério é, na maioria das vezes, a escolha da ação que reduz a incerteza ou a ignorância sobre o ambiente.

Pode-se visualizar, portanto, a existência de três subsistemas para um robô móvel que pretende ter autonomia:

- Subsistema de mapeamento: monta um mapa do ambiente percorrido conforme uma representação interna;
- Subsistema de auto-localização: responsável por localizar o robô dentro do mapa a fim de que ele possa se movimentar e realizar tarefas;
- Subsistema de planejamento exploratório: também chamado de planejamento de movimento, define o processo de tomada de decisão quanto à ação que será

executada.

Este capítulo é uma introdução aos conceitos da Robótica Móvel Autônoma. Na Seção 2.1 são apresentadas algumas formas de um robô representar internamente um ambiente. A Seção 2.2 descreve os sensores, especialmente os que são frequentemente utilizados nas pesquisas. Na Seção 2.3 são apresentados os tipos de navegação que o robô pode realizar no ambiente. Posteriormente são apresentados os conceitos de auto-localização e de mapeamento e auto-localização simultâneos.

2.1 Representação do Ambiente

Construir um mapa do ambiente confiável e estável é uma tarefa realizada através de um processo interativo entre o robô e o ambiente. Durante este processo, o robô conta apenas com a informação de seus sensores para estimar a posição dos objetos presentes no ambiente e atualizar sua representação interna, comumente chamada de *mapa*.

Qualquer decisão tomada pelo robô deve ser baseada em sua percepção local e em seu conhecimento global do ambiente. O conhecimento global corresponde à união da percepção local em diferentes instantes durante uma exploração.

A representação interna do ambiente é utilizada no planejamento de movimentos para guiar o robô ao seu objetivo e durante a exploração para indicar ao robô os locais ainda não visitados. A maioria das representações do ambiente pode ser classificada em: mapas métricos, mapas topológicos e mapas probabilísticos. A seguir, cada um deles é apresentado.

2.1.1 Mapas Métricos

Os mapas métricos podem ser subdivididos em:

- **grades de ocupação:** realizam decomposição espacial do ambiente (ELFES, 1987, 1989). Esse tipo de mapa representa o ambiente como uma matriz de células, onde cada célula está geometricamente relacionada com uma região quadrada do ambiente e armazena um valor que indica a probabilidade de ocupação desta área. As principais vantagens desse tipo de mapa são a facilidade de representação e a possibilidade de integrar leituras de diferentes sensores;
- **mapas baseados em características:** utilizam uma representação geométrica. Representam o ambiente através da localização das características relevantes para o mapa e suas propriedades geométricas (CROWLEY; COUTAZ, 1985; CHATILA et al., 1996; DEVY et al., 1995).

A principal diferença entre as diversas implementações das grades de ocupação é a função utilizada para a atualização das células da grade. Essa função pode ser baseada em certeza Fuzzy (ORIOLO; ULIVI; VANDITTELLI, 1997), Bayes (HOWARD; KITCHEN, 1996), heurística (ELFES, 1987), gaussiana (ELFES, 1989), frequência (BORENSTEIN; KOREN, 1991; PRESTES et al., 2001), entre outros.

As grades de ocupação possuem problemas de granularidade, escalabilidade e extensibilidade (BAILEY; NEBOT, 2001). Estes estão relacionados ao tamanho fixo do mapa e de suas células e acarretam sobrecarga computacional. Além disso, a funcionalidade do mapa é limitada, uma vez que é difícil representar entidades simbólicas como

portas, cadeiras, etc (FOX; BURGARD; THRUN, 1999). Também não oferece mecanismos para tratar obstáculos estáticos diferentemente de obstáculos dinâmicos.

Os mapas baseados em característica foram inicialmente propostos por Chatila e Laumond (CHATILA; LAUMOND, 1989) para ambientes internos. Eles representam o ambiente através de segmentos de linhas, arcos de círculos, pontos e poliedros a partir da informação oriunda dos sensores de proximidade juntamente com a incerteza associada a sua localização. Essa formulação é geralmente utilizada em conjunto com o Filtro de Kalman para minimizar o erro de estimativas dos objetos extraídos do ambiente e da posição e orientação do robô (CROWLEY; COUTAZ, 1985).

A principal vantagem dos mapas por característica é a eficiência na representação do ambiente. Em comparação com as grades de ocupação, Dam (DAM, 1998) faz algumas considerações:

- se o ambiente é desconhecido, um mapa por características é muito mais complicado de ser construído;
- um mapa por características é difícil de ser utilizado em ambientes não-estruturados, devido a dificuldade de modelar alguns obstáculos usando primitivas geométricas;
- a posição do robô pode ser calculada de forma mais eficiente utilizando mapas por característica do que utilizando grades de ocupação;
- a representação em grade possibilita representar tanto o espaço livre quanto o espaço ocupado do ambiente.

Entretanto, os mapas por característica possuem três problemas bem definidos (ROMERO; MORALES; SUCAR, 2001): perda de estabilidade, pois a representação pode ser prejudicada por pequenas variações das leituras dos sensores; dificuldade na localização, pois devido a esta representação ser baseada em primitivas geométricas é possível que a representação não seja única; e perda de poder expressivo, decorrente da dificuldade em representar completamente todas as características do ambiente através de primitivas geométricas.

2.1.2 Mapas Topológicos

Os **mapas topológicos** correspondem a um grafo composto de vértices e arestas (KUIPERS; BYUN, 1991), onde os vértices correspondem a locais distintos e as arestas correspondem a ligações entre diferentes locais contendo informação. Esta informação corresponde ao conjunto de comportamentos que permite ao robô navegar de um vértice a outro ou à informação métrica sobre a posição relativa entre os vértices (FOX; BURGARD; THRUN, 1999).

Os vértices representam locais distinguíveis no ambiente baseados no padrão observado nos dados sensoriais (BAILEY; NEBOT, 2001). Estes padrões são representações comuns de portas, corredores, etc. Em ambientes internos, por exemplo, as arestas correspondem a corredores e os *caminhos médios* correspondem a vértices.

Segundo Dudek (DUDEK; FREEDMAN; HADJRES, 1993), a principal vantagem desta abordagem é a sua natureza qualitativa e sua relação com teorias de cognição humana. Além disso, os mapas topológicos são geralmente menos complexos que os mapas geométricos; muito mais eficientes no processo de planejamento de caminhos (SHATKAY; KAEHLING, 1997) e mais flexíveis a noção de estado.

2.1.3 Mapas Probabilísticos

Outro tipo de representação são os **mapas probabilísticos** propostos por Smith, Self e CheeseMan (SMITH et al., 1997) como uma ferramenta para tratar incertezas espaciais originadas das medidas sensoriais e dos movimentos do robô na representação espacial, ao invés de considerá-las como um problema geométrico à parte. Neste tipo de representação, tanto os obstáculos quanto o robô são representados como um conjunto de objetos que são descritos pela localização juntamente com uma matriz de covariância que descreve o relacionamento espacial entre eles (HEBERT; BETGÉ-BREZETS; CHATILA, 1995).

Este tipo de representação é frequentemente utilizado com o Filtro de Kalman Estendido (ver Capítulo 4). O principal problema é a necessidade de armazenar uma matriz de covariância necessária para a convergência do mapa. Esta matriz cresce quadraticamente com a quantidade de objetos observados resultando em alto custo computacional quando o ambiente a ser mapeado é grande.

2.2 Os Sensores e o Ambiente

A autonomia só pode ser adquirida se o robô possuir no mínimo um componente sensorial. Os sensores são sua principal forma de perceber e interagir com o ambiente. Sem sensores, a tarefa teria que ser totalmente programada.

Toda leitura ou medida de sensor tem uma precisão associada que é fundamentalmente limitada pelo ruído. A magnitude do ruído está intimamente relacionada com a combinação do tipo de ambiente com o tipo de sensor. Um sensor que possui sensibilidade à luz, terá problemas em ambientes muito escuros, por exemplo. Por isso, para escolher o melhor sensor quando não é possível utilizar vários sensores em um mesmo robô, o fator ambiente será decisivo (MACHADO, 2000).

Os ambientes sub-aquáticos (MAJUMDER et al., 2000) são especialmente difíceis de explorar porque não são estruturados e apresentam grande ocorrência de movimento (peixes, algas, correntes marítimas, etc). Além disso, a capacidade sensorial fica ainda mais limitada nesse tipo de ambiente, dificultando o processamento da informação.

Os erros que ocorrem durante a exploração podem ser de dois tipos (PRESTES, 2003): **sistemáticos** e **não-sistemáticos**. Os erros sistemáticos são dependentes inteiramente das características da plataforma móvel, enquanto que os erros não-sistemáticos são gerados a partir da interação do robô com o ambiente. Os erros sistemáticos podem ser previstos; alguns são determinísticos, outros podem ser modelados por distribuição probabilística. Os erros não-sistemáticos sempre são modelados por funções probabilísticas e são associados aos sensores do robô.

Em alguns casos, são necessários vários sensores para obter mais precisão na exploração. Há três tipos de fusão dos dados sensoriais (DAM, 1998):

- **Complementar:** quando sensores têm apenas uma visão parcial do ambiente, por exemplo, câmeras em diferentes direções. Outro exemplo são os sensores que medem aspectos diferentes como posição e cor;
- **Competitiva:** quando competem para obter uma leitura com mais precisão. Um exemplo é a fusão das leituras de um laser e de um sonar para um mesmo obstáculo. Esse tipo de fusão é geralmente utilizada para reduzir a incerteza na leitura do sensor ou para manipular dados de um mesmo sensor;

- Cooperativa: os sensores cooperam para obter dados que nenhum deles obteria sem a cooperação do outro, é o caso de imagens com duas câmeras. Neste caso, a fusão cooperativa pode ser utilizada para converter duas imagens em 2D para uma representação interna em 3D.

Entre os sensores mais utilizados em robôs móveis destacam-se: odômetro, câmera, sonar, infravermelho e laser. Esta seção apresenta alguns deles trazendo como principal enfoque os principais problemas a eles associados.

2.2.1 Odômetro

Muito utilizados nos subsistemas planejadores de movimento, os odômetros são sensores de implementação simples e de baixo custo. Nos experimentos em ambientes conhecidos, normalmente são o único sensor empregado no robô.

Os odômetros contam a quantidade de rotações realizadas por cada roda do robô. Esta informação permite calcular a distância percorrida pelo robô a qualquer instante. Este processo é chamado de *dead-reckoning* e é muito suscetível a erros por diversas razões (THRUN, 1998), entre elas: insuficiente alinhamento das rodas; erros no sinal dos sensores; variações na trajetória devido à diferença de superfícies.

Em particular, dois erros sistemáticos são frequentemente identificados quando se trabalha com odômetros. O primeiro ocorre quando o diâmetro das rodas é menor do que aquele modelado no sistema. Neste caso, o sistema irá sempre superestimar a distância percorrida. O segundo ocorre como consequência da resolução finita do odômetro. Haverá sempre um erro de distribuição normal associado à leitura dos odômetros.

2.2.2 Câmera de Vídeo

Determinados ambientes tornam propenso o uso das câmeras de vídeo como dispositivo de reconhecimento. Normalmente são ambientes fechados, construídos pelo homem, com propriedades bem definidas e que podem ser tratados através de imagens.

Neste caso, o uso de câmeras é condicionado à existência de informações estruturais visuais como, por exemplo, uma cadeira, uma mesa, uma porta, um canto da sala.

Infelizmente, estes dispositivos são geralmente descartadas por sua baixa velocidade e enorme exigência computacional, enquanto que os demais sensores são conhecidos por apresentarem baixa resolução (caso do sonar) ou pelo alto custo dos equipamentos (caso do laser).

2.2.3 Sonar

O sonar é um tipo de sensor de proximidade muito empregado quando a percepção dinâmica do ambiente é um requisito fundamental. Os sensores desse tipo são mais rápidos que as câmeras por não exigirem processamento de imagens. No entanto, apresentam baixa resolução e seu uso requer uma avaliação espaço-temporal entre as medidas adquiridas e o mapa do ambiente.

Os sensores de proximidade (sonar, laser e infravermelho) são problemáticos devido à limitação da capacidade receptiva (ROY et al., 1999). Em regiões muito abertas, não conseguem identificar nenhum ponto de referência devido ao seu alcance finito.

O sonar é amplamente utilizado como dispositivo sensorial em robôs móveis. As principais razões para isso são seu baixo custo e sua facilidade de operação. Entretanto, possui sérios problemas que limitam sua utilização direta em aplicações de mapeamento

do ambiente (RASCHKE; BORENSTEIN, 1990), entre eles (BORENSTEIN; KOREN, 1989):

- **pobre direcionalidade:** embora, estes sensores forneçam bom alcance, eles são incapazes de informar a direção do objeto identificado;
- **passíveis de ruídos:** a leitura do sensor sofre influências de ruídos externos e de reflexões perdidas dos sensores vizinhos (*crosstalk*);
- **reflexão especular:** a leitura dos sensores sofre influência do ângulo de incidência do sinal sonoro nos objetos identificados, da estrutura da superfície de incidência, da refletividade e da distância do objeto em relação ao sensor. Isso acarreta uma falsa identificação, e em algumas ocasiões, total ausência dela.

Um fator importante a considerar é que a velocidade de disparo do sonar é dependente da velocidade do robô. Se o robô se move a uma velocidade alta os sensores devem ser disparados em intervalos curtos, a fim de que ele consiga reagir rapidamente a eventos inesperados. No caso de baixas velocidades, o intervalo de disparo dos sensores deve ser aumentado, para que seja minimizado o efeito colateral produzido pela reflexão especular.

Além disso, a aquisição dos sinais sonoros é feita normalmente com o robô em movimento. Isto gera um mapeamento atrasado em *ms* devido à diferença da posição do robô no momento do disparo e no momento da recepção do sinal.

2.3 Navegação

Navegação, para a Robótica Móvel, consiste em planejar um caminho viável entre duas posições do espaço. Esse planejamento, normalmente é obtido com base apenas em dados sensoriais, sem conhecimento prévio do ambiente.

Os métodos de navegação podem ser de 3 tipos (DAM, 1998):

- **métodos globais** são utilizados quando se tem o conhecimento total sobre a estrutura do ambiente onde o robô está inserido, inclusive sua localização dentro dele. Há três principais abordagens para esse tipo de método (LATOMBE, 1993): mapas de caminhos, decomposição celular e campo potencial;
- **métodos locais** são utilizados quando o robô não possui informação alguma sobre o ambiente e sua localização nele. A única informação disponível é um mapa local construído com as informações locais oriundas dos seus sensores.
- **métodos híbridos** são métodos que combinam informações locais e globais. Ou seja, possibilita adquirir a estrutura do ambiente, através da integração de diferentes mapas locais em um mapa global; corrigir erros de posicionamento do robô; corrigir informações inconsistentes do mapa do ambiente; identificar rotas alternativas; evitar colisões, entre outros.

O método híbrido é geralmente escolhido para a navegação na RMA pois elimina ou minimiza a necessidade de qualquer conhecimento prévio sobre a estrutura da área de atuação do robô (PRESTES, 2003).

Duas estratégias são comumente utilizadas para navegação exploratória: abordagem *reativa* (MEKREZ, 1997; MATARIC, 1990; CROWLEY; COUTAZ, 1985) e abordagem *baseada em modelo* (YAMAUCHI, 1997; CHOSSET; BURDICK, 2000; THRUN, 1998). A abordagem reativa consiste em guiar o robô paralelamente às paredes enquanto extrai o mapa topológico ou geométrico (ver Seção 2.1). Essa abordagem facilmente cai em mínimos locais e não funciona bem quando o ambiente é esparso pois a região estiver além das capacidades perceptíveis do robô nunca será explorada.

A outra abordagem, baseada em modelo, usa uma função que descreve quais são as ações preferenciais a serem executadas dado o estado corrente da exploração e o conhecimento obtido do ambiente. Esta função é chamada *função de custo*, *função de recompensa* ou *função heurística*.

Existem diversas funções propostas na literatura, porém, a que mais se relaciona com este trabalho é a informação de *Fischer*. A informação de Fischer sobre as incertezas nas medidas sensoriais e navegacionais pode ser utilizada como uma função heurística (FEDER; LEONARD; SMITH, 1999) durante a construção do mapa do ambiente. Neste caso, o robô tende a escolher ações que maximizem seu ganho de informação no sentido definido por esta função enquanto minimizam a incerteza da localização dos objetos e do robô no ambiente.

2.4 Auto-Localização

A construção de um modelo confiável do ambiente deve ser realizada considerando os erros gerados pelo deslocamento do robô. Esses erros correspondem a erros de odometria, que são observados através de comparações entre a percepção local e o conhecimento global sobre o ambiente, e devem ser reduzidos através de um processo de auto-localização.

A auto-localização é a capacidade que um robô tem de se localizar no seu mapa interno usando somente as informações captadas por seus sensores. Um robô autônomo genuíno é capaz de se localizar, ou auto-localizar, em ambientes genéricos, ou seja, ambientes que não foram artificialmente modificados para este fim.

As técnicas para auto-localização podem ser divididas em dois grupos: técnicas de posicionamento relativo e absoluto. Posicionamento relativo é baseado nos odômetros e na percepção local do robô (YAMAUCHI, 1998; DEVY et al., 1995; BETKE; GURVITS, 1994). O posicionamento absoluto é baseado em sensores como GPS, ou em *landmarks* (BETKE; GURVITS, 1994, 1997), cuja localização é conhecida no ambiente. Dependendo do deslocamento do robô, o GPS é mais preciso que os odômetros. Entretanto, em determinados ambientes é impraticável seu uso, um exemplo é o ambiente sub-aquático (FEDER; LEONARD; SMITH, 1999). Nesse caso, o robô deve vir à superfície para se localizar, o que é indesejável.

Roy (ROY et al., 1999) propõe uma técnica chamada *navegação costeira* que sugere uma navegação próxima às regiões que contêm alto conteúdo informativo, ou seja, regiões com maior probabilidade de ocupação. É inspirado na navegação costeira realizada por barcos quando não possuem GPS. Esta técnica utiliza *localização markoviana* (THRUN, 1998) sobre uma representação baseada em grades.

A localização markoviana é também utilizada no *problema do robô raptado*. Este problema foi proposto por Engelson (ENGELSON, 1994) e consiste em determinar a posição do robô em um sistema de referência global com total incerteza sobre sua posição.

Golfarelli (GOLFARELLI; MAIO; RIZZI, 2001) propõe uma técnica chamada

correção elástica para corrigir o erro posicional gerado durante a exploração do ambiente. Essa técnica assume que o robô é capaz de identificar *landmarks* e que o conhecimento adquirido é representado em um mapa topológico. A correção elástica é baseada na analogia entre o modelo do ambiente e a estrutura mecânica, onde cada rota é uma barra elástica e cada *landmark* um nó. Os erros são corrigidos como um resultado das deformações induzidas a partir de forças que surgem dentro da estrutura quando medidas inconsistentes são tomadas. Os parâmetros de elasticidade que caracterizam a estrutura são usados para modelar a incerteza na odometria.

2.5 Mapeamento e Auto-Localização Simultâneos

O processo de mapeamento de ambientes é extremamente importante para todas as aplicações onde o ambiente é desconhecido. Ele consiste em determinar a localização de entidades de interesse dado um referencial global (FOX; BURGARD; THRUN, 1999).

Este problema está relacionado ao problema da auto-localização descrito na Seção 2.4. Por um lado o robô necessita saber sua posição para construir o mapa do ambiente, por outro necessita desse mapa para se localizar. Juntos estes problemas levam a um terceiro chamado *mapeamento e auto-localização simultâneos* (MLS) (FEDER; LEONARD; SMITH, 1999).

O mapeamento e a auto-localização simultâneos elimina a necessidade de uma infra-estrutura artificial para a localização do robô e de conhecimento *a priori* topológico ou geométrico sobre o ambiente, além de reduzir ou eliminar erros inseridos pelo *dead-reckoning*.

Um robô capaz de realizar o mapeamento de um ambiente desconhecido e se auto-localizar a partir de uma versão parcial do mapa é capaz de construir um mapa representativo e confiável de qualquer ambiente somente a partir de observações relativas deste ambiente (PRESTES, 2003).

De acordo com Dissanayake (DISSANAYAKE et al., 2001), os trabalhos adotam uma das 3 filosofias a seguir:

- a mais popular é a utilização do Filtro de Kalman. Sua popularidade deriva do fato de ele fornecer tanto uma solução recursiva ao problema de navegação como um mecanismo para calcular incerteza sobre a localização do robô e dos *landmarks*;
- a segunda filosofia é a utilização de conhecimento qualitativo dos locais relativos dos *landmarks* e do robô para construir o mapa e guiar o movimento;
- A terceira utiliza grades de ocupação probabilísticas juntamente com um método de relaxação para a navegação e o MLS (YAMAUCHI, 1998; THRUN, 1998).

3 EXPLORAÇÃO DE AMBIENTES USANDO MÉTODO DE CAMPOS POTENCIAIS

Este capítulo apresenta o método de navegação exploratória proposto por Edson Prestes (PRESTES, 2003) que define uma teoria unificada para o planejamento e a exploração de ambientes. Esta teoria é uma extensão da solução de Problemas de Valores de Contorno (PVC) envolvendo a equação de Laplace, primeiramente proposta por Connolly (CONNOLLY; GRUPEN, 1993).

Adicionalmente, Edson propõe um agente exploratório que possui uma arquitetura dividida em sub-funções ou módulos. Nesta arquitetura é possível identificar o objetivo deste trabalho: o módulo de auto-localização.

Neste capítulo, é descrito o método exploratório desenvolvido para o agente e posteriormente a arquitetura onde está inserido o módulo de auto-localização.

3.1 Método Exploratório

Esta seção apresenta o método exploratório (PRESTES, 2003) e suas principais características, desde o sistema sensorial até a maneira como os Campos Potenciais Harmônicos atuam no robô durante a exploração.

3.1.1 Sistema Sensorial

O sistema sensorial que é utilizado nos experimentos do método exploratório é composto por dois tipos de sensores: odômetros e sonares. A informação fornecida pelos odômetros é utilizada para atualizar a posição corrente do robô no espaço bidimensional contínuo.

A informação fornecida pelos sonares é utilizada para atualizar um mapa local que será posteriormente incorporado ao mapa global do ambiente. O robô Nomad 200 possui um anel de sensores do tipo sonar, em número de 16, que são disparados a cada 500ms.

Em simulação, o sistema sensorial é representado por dois componentes: uma janela de visão para o robô e um conjunto de objetos montados no ambiente simulado. Durante o processo de exploração, o robô percebe apenas o objeto que se encontra completamente ou parcialmente dentro de sua janela de visão. Este processo tenta simular os sensores do tipo sonar. Já os odômetros são facilmente simulados pela informação da posição do robô.

Como em simulação o ambiente e as variáveis são conhecidas e perfeitamente controladas pelo sistema, foi necessário adicionar erros para a implementação deste trabalho. Estes erros simulam os erros de odometria do robô e erros de leitura sensorial.

3.1.2 Representação do Ambiente

A representação do ambiente é um mapa métrico do tipo grade de ocupação (ver Seção 2.1) implementado na forma de uma matriz bidimensional $M_{L_x \times L_y}$. Esta representação foi escolhida por ser uma representação discretizada, pronta para ser utilizada em conjunto com os métodos de relaxação para o cálculo da função potencial. Além disso, a representação por grade de ocupação possibilita fundir diferentes tipos de sensores, além de representar facilmente o espaço livre do ambiente.

Cada célula da grade está centrada nas coordenadas do mundo real $\mathbf{r} = (r_i, r_j)$ e corresponde a uma região quadrada do mundo real que armazena os atributos:

- **estado:** indica o estado corrente da célula. Este atributo pode assumir um dos seguintes valores: *não explorado*, *espaço livre* e *ocupado*;
- **certeza:** é uma medida de probabilidade de encontrar um obstáculo naquela célula (limitada ao intervalo $[0, 15]$);
- **potencial:** se refere à função heurística, no espaço de estados discreto, que quando convertido ao espaço contínuo indica caminhos de navegação;
- **região:** armazena um número inteiro e é utilizado apenas para identificar quando se deve parar o processo exploratório.

O estado *não explorado* indica que a posição correspondente à célula de índice ij ainda não foi visitada e seu potencial é igual a zero. O estado *espaço livre* indica que o potencial da célula pode ser atualizado. O estado *ocupado* indica que a célula está ocupada por um objeto do mundo real e seu valor de potencial é igual a 1 no caso de um obstáculo e igual a 0 no caso de um objetivo.

3.1.3 Atualização do Mapa

Como visto anteriormente, a atualização do mapa ocorre durante o processo de exploração. Inicialmente, os atributos de todas as células são inicializados da seguinte maneira:

- o estado recebe o valor *não explorado*;
- a certeza e o potencial recebem ambos o valor 0 (zero)
- a região recebe o valor 5.

A cada passo, as células observadas serão as correspondentes à janela de visão do robô (seu mapa local). Após a atualização do mapa, as células são casadas com as células correspondentes do mapa global.

A atualização é feita utilizando um método baseado na frequência de observações chamado HIMM (*Histogramic In-Motion Mapping*) proposto por Borenstein e Koren (BORENSTEIN; KOREN, 1991). Ele usa uma função linear que conta o número de observações sensoriais feitas para cada objeto no ambiente. A partir daí, a propriedade certeza das células é atualizada. Se a taxa de observações positivas por unidade de tempo exceder um limiar pré-definido, a célula tem sua propriedade estado alterada para *ocupado*, caso contrário, para *espaço livre*. Esse processo permite o tratamento de obstáculos tanto dinâmicos quanto estáticos.

Um caso particular da função heurística é o cálculo do potencial harmônico. Em seu trabalho, Connolly (CONNOLLY; GRUPEN, 1993) mostrou que no caso específico onde os obstáculos têm potencial fixo igual a 1 e o objetivo têm potencial igual a 0, o potencial harmônico em uma célula fornece a probabilidade de um agente, partindo de qualquer célula, colidir com um obstáculo antes de atingir o alvo (*hitting probability*).

3.1.4 Campos Potenciais Harmônicos

O planejamento de caminhos livres de obstáculos se baseia no método de campos potenciais harmônicos. Esse método utiliza a equação de Laplace para cálculo do potencial de cada célula.

A equação de Laplace é calculada através de um método de relaxação similar à *iteração valorada* (SUTTON; BARTO, 1998) que interpola o valor do potencial entre os obstáculos e o alvo, a partir de um ponto inicial.

Inicialmente, é atualizado o valor de potencial das células da grade de ocupação. Quando uma célula representa a superfície de um obstáculo, o seu potencial é igual a um (1); quando representa a superfície de um objetivo a ser atingido (alvo), seu potencial é zero (0). Em seguida, o potencial das células que não correspondem nem a obstáculo, nem a objetivo é calculado até a sua convergência. Após a convergência, retorna a curva definida pelo gradiente descendente iniciando no ponto inicial e finalizando na posição objetivo, formando assim um caminho para o robô.

Esse caminho gerado é análogo a uma linha de força da Teoria Eletromagnética que apresenta as seguintes propriedades (FEYNMAN; LEIGHTON; SANDS, 1972):

- corresponde a uma curva aberta, ou seja, não apresenta ciclos;
- conecta pontos na superfície dos obstáculos (alto potencial) aos pontos na fronteira (baixo potencial);
- duas linhas de força nunca se cruzam. Isto é consequência do campo de força ser derivado de um campo potencial. O potencial é uma função analítica e tem gradiente univocamente definido em todos os seus pontos.

Além desta, as funções harmônicas podem ter diversas interpretações: princípio da energia mínima ou máxima, probabilidade de colisão ou derivadas de processos de aprendizado.

A Figura 3.1 mostra o campo potencial, representado pelas setas vermelhas, guiando o robô para as regiões não exploradas do ambiente.

No método de navegação exploratória usando PVC (PRESTES, 2003), o campo potencial indica o caminho em direção às regiões não exploradas do espaço. Isto é feito considerando as fronteiras da região não explorada como objetivos temporários, isto é, regiões com potencial igual a 0.

Seguir o gradiente descendente sobre o potencial faz o robô evitar obstáculos e se aproximar destes sub-objetivos. As propriedades das linhas de força garantem que o robô sempre se movimentará em direção a uma fronteira com regiões não exploradas se ela existir. Uma vez que o robô ultrapassa a fronteira, a região explorada cresce e novas fronteiras surgem. Esse comportamento identifica os chamados *contornos dinâmicos* das regiões que estão sendo exploradas.

A Figura 3.2 mostra ao fluxo de execução do algoritmo durante a exploração. A condição de parada é o que determina o fim do processo exploratório e será explicada na Seção 3.2 pois faz parte do módulo de controle e atuação do robô.

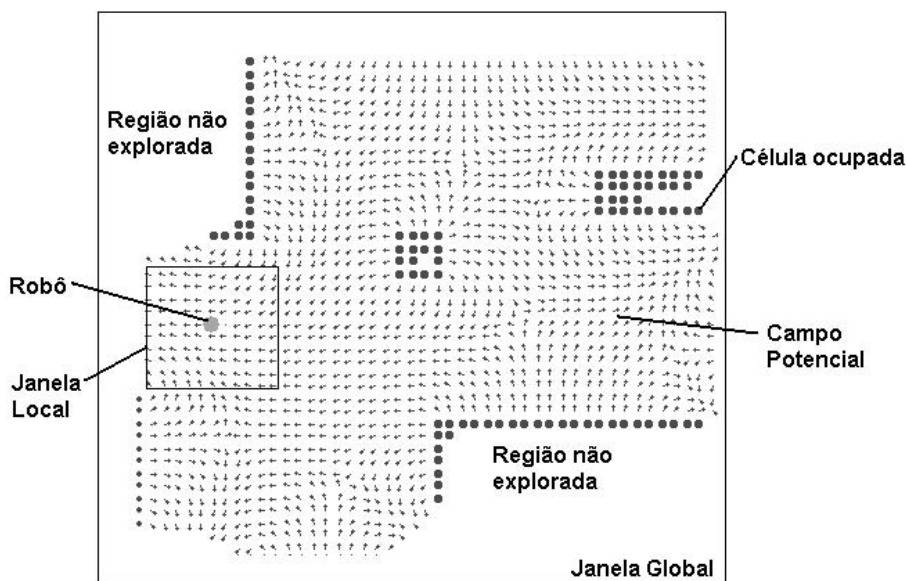


Figura 3.1: Exploração de um ambiente usando o campo potencial.

3.2 Arquitetura do Agente Exploratório

O agente exploratório proposto por Edson (PRESTES, 2003) utiliza os campos potenciais harmônicos para navegar e explorar o ambiente. Para definir esse agente, no mundo real representado pelo robô Nomad 200, Edson propôs uma arquitetura similar à arquitetura *blackboard* (PETTERSSON, 1997).

Esta arquitetura é composta por uma estrutura de dados chamada *blackboard* e diversos módulos especialistas, chamados *fontes de conhecimento* que atuam independentemente sobre os dados, atualizando seus valores e verificando sua consistência.

Os dados armazenados no *blackboard* são compostos de dois componentes essenciais e complementares:

- as coordenadas do robô em um sistema de referência global;
- a representação do mundo baseada em grades de ocupação.

Os módulos especialistas interagem com estes dois componentes atualizando-os ou extraindo informações que são usadas nas fases subsequentes de planejamento e ação. Os módulos especialistas são:

- **Manutenção do mapa:** faz a atualização do mapa baseado nas leituras sensoriais e trata, adicionalmente, do preenchimento de lacunas. A representação do ambiente é a mesma descrita na Seção 3.1.2;
- **Localização:** é fundamental para a construção de um mapa confiável. Envolve: integração de caminho, responsável por realizar a localização do robô baseada nas informações fornecidas por seus odômetros; auto-localização, responsável por reduzir os erros de odometria gerados pelo módulo de integração de caminho;

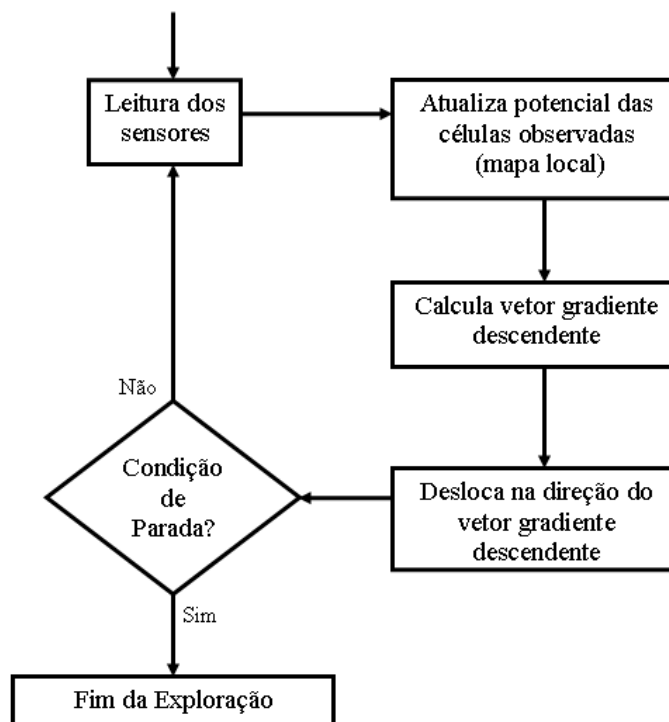


Figura 3.2: Fluxograma da Exploração

- **Controle e atuação do robô:** responsável por atualizar o campo potencial (função heurística). Define a condição de parada, responsável por determinar quando o robô deve parar sua atuação (quando há a identificação de um objetivo ou quando o ambiente foi completamente explorado - objetivo é apenas construir o mapa do ambiente). A condição de parada (fim da exploração) consiste em verificar se existe alguma célula inacessível com estado igual a *espaço livre* ao lado de alguma célula com estado igual a *não explorado*.

O módulo de controle também é responsável pelo processamento da ação. A ação corresponde ao deslocamento do agente seguindo o gradiente do potencial. Neste caso, o gradiente apontará para a direção da região desconhecida mais próxima ou para a maior no caso de existir mais de uma região desconhecida à mesma distância. Se nenhum obstáculo for detectado, o limite de sua região de ativação é igual a zero e o potencial é igual a zero em todas as células. Neste caso, ou em qualquer situação onde não existe gradiente para guiar o agente, ele simplesmente segue em linha reta para frente.

A Figura 3.3 mostra a arquitetura com seus módulos especialistas. Todos os especialistas são ativados a partir do módulo de interface com o usuário. Este módulo é responsável por especificar a tarefa a ser realizada (exploração, aquisição de mapa, planejamento de caminhos, localização de objeto); ajustar os parâmetros do sistema e enviar comandos ao robô.

Em seus experimentos, Edson realizou a movimentação do robô através de um único comando que definia sua velocidade linear e seu ângulo de rotação. Este comando permitiu realizar o movimento combinado, rotação em conjunto com translação,

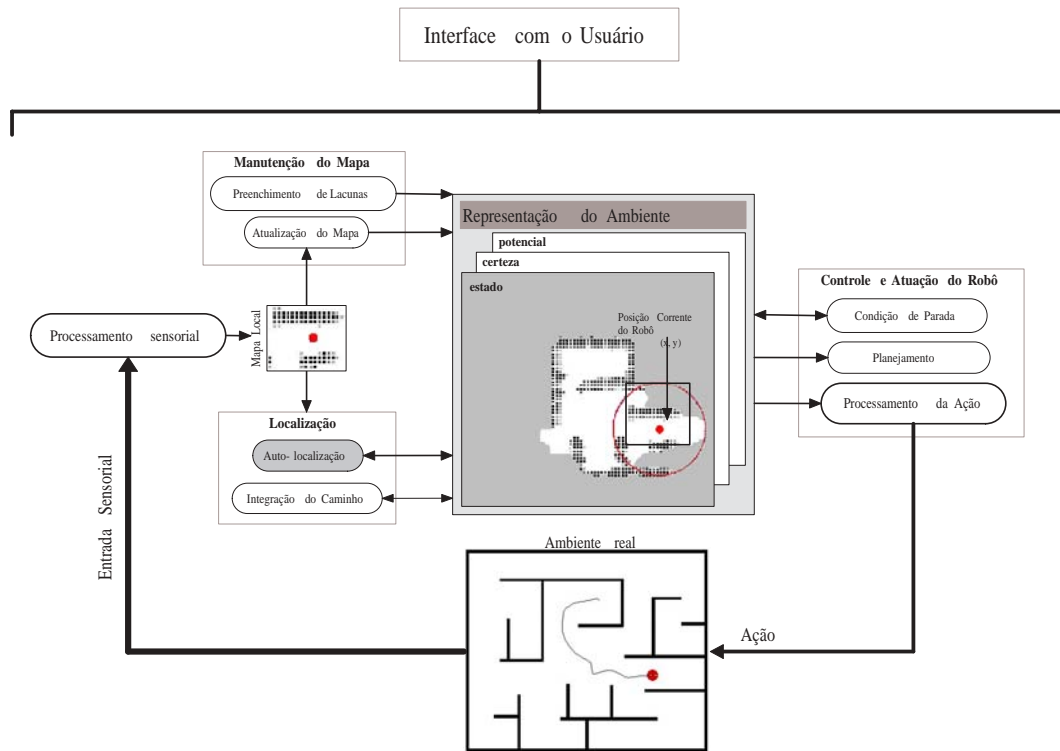


Figura 3.3: Arquitetura do Agente Exploratório (PRESTES, 2003)

gerando trajetórias suaves, ou seja, resultou em uma trajetória sem mudanças abruptas na orientação do robô, um dos fatores que aumenta os erros de odometria.

Outro fator que contribuiu para o sucesso dos experimentos sem o módulo de auto-localização foi a baixa velocidade de deslocamento do robô. Entretanto, ela foi reduzida justamente para evitar a ocorrência de erros. Com o módulo de auto-localização, a velocidade de exploração poderá ser aumentada, permitindo um tempo de execução menor para as mesmas tarefas.

4 FILTRO DE KALMAN

Na arquitetura do agente exploratório (ver Seção 3.2), o módulo de auto-localização é fundamental para o sucesso da exploração de ambientes. Esse módulo deve corrigir os erros de localização do robô no mapa gerado durante o processo exploratório. Para implementá-lo, optou-se por uma estratégia usando o Filtro de Kalman.

Neste capítulo, são apresentados os princípios fundamentais do Filtro de Kalman, começando por um breve histórico e por algumas definições básicas para o entendimento. Este capítulo serve também de base para compreensão do próximo, que mostra como o Filtro de Kalman é empregado no módulo de auto-localização.

4.1 Histórico

Em 1960, Rudolph Emil Kalman publica o artigo intitulado *A New Approach to Linear Filtering and Prediction Problems* (KALMAN, 1960) que descreve uma solução recursiva para o problema da filtragem linear de dados discretos. Naquela época, devido à baixa capacidade de processamento das máquinas, os experimentos que utilizavam o filtro limitavam-se a resolver problemas lineares.

Com o passar do tempo e com o avanço da computação digital, foram descobertas outras aplicações para o Filtro de Kalman (FK). Suas aplicações vão desde a determinação do prazo de validade de alimentos perecíveis (ARAÚJO, 2000), até aplicações na engenharia espacial. Em (ZANARDI; LÓTA, 2001), por exemplo, o filtro é usado para calcular a orientação do veículo espacial em relação a um sistema de referência com origem na Terra, a partir de dados de sensores a bordo de um satélite.

Particularmente interessante é o uso do FK na navegação autônoma ou assistida. Nos trabalhos nesta área (CSORBA, 1997; GUIVANT; NEBOT, 2001; DISSANAYAKE et al., 2001; WELCH; BISHOP, 2001; WELCH, 2001; ARAÚJO, 2000; ZANARDI; LÓTA, 2001), o FK é utilizado para resolver o problema da localização do robô e dos objetos no mapa do ambiente. Normalmente, ele é aplicado simultaneamente à construção do mapa. A Figura 4.1 ilustra uma aplicação típica usando o Filtro de Kalman.

O Filtro de Kalman, também chamado de Estimador Linear Ótimo, é uma ferramenta poderosa para trabalhar com sistemas que apresentam grande ocorrência de ruído, desde que a magnitude deste ruído seja conhecida. Quando o sistema modelado é linear, usa-se o Filtro de Kalman. Para sistemas não lineares a solução é chamada Filtro de Kalman Estendido (FKE) que realiza a linearização do modelo usando derivadas parciais.

Na próxima seção é apresentado um estudo sobre a definição teórica do Filtro de Kalman baseado nos trabalhos (WELCH, 2001), (WELCH; BISHOP, 2001), (CSORBA, 1997) e (GUIVANT; NEBOT, 2001).

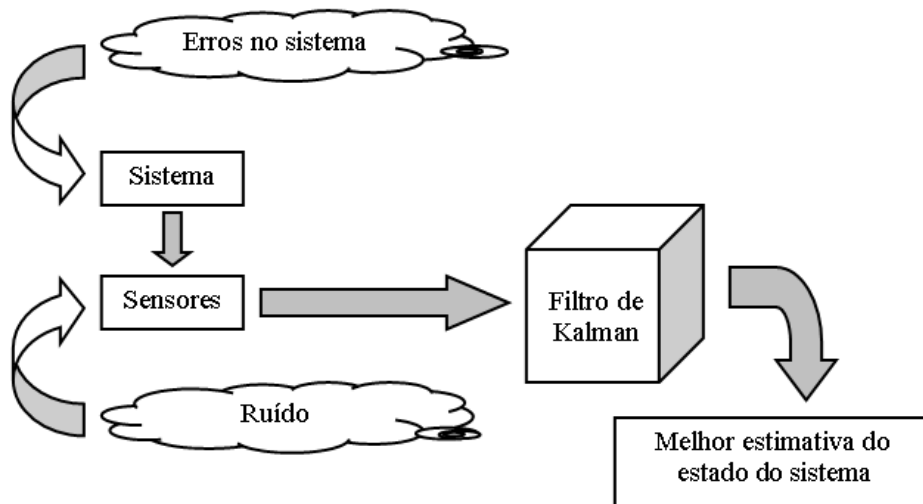


Figura 4.1: Aplicação típica usando Filtro de Kalman

4.2 Definições

Antes de entrar na definição do Filtro de Kalman propriamente dita, é importante considerar algumas definições básicas ao estudo.

4.2.1 Definição de Modelo Determinístico

Entende-se por modelo determinístico a função que descreve matematicamente um sistema físico. Essa função identifica as variáveis de interesse, as entradas e as saídas do sistema em questão. De posse do modelo, é possível analisar como o sistema se comporta quando mudam os valores de suas variáveis ou de suas entradas.

Porém, nenhum modelo de sistema é perfeito. Qualquer modelo analisa apenas características de interesse direto, ou seja, de interesse para a aplicação que se está analisando. Por exemplo, embora exista um número muito grande de maneiras para analisar precisamente a curva de um veículo, apenas um número pequeno pode ser incluído em um modelo útil.

O objetivo do modelo é representar a maneira dominante ou crítica da resposta do sistema, logo, muitos efeitos, apesar de conhecidos, não serão modelados. Os modelos precisam ser simplificados para que fiquem apenas com o essencial, a fim de gerar um algoritmo computacionalmente possível.

4.2.2 Definição de Ruído

Por ruído entende-se o somatório de todas as influências não-sistemáticas sobre o comportamento de um sistema. Em outras palavras, ruído corresponde a toda influência que não pode ser abarcada pelo modelo determinístico do sistema.

Apesar de ser imprevisível o ruído é passível de modelamento. O modelo mais comum de ruído, que é base do método de Kalman, é o ruído gaussiano, também conhecido como *ruído branco*, definido pela seguinte distribuição

$$f(x, x_m, q) = \frac{e^{-1/2 \frac{(x-x_m)^2}{q^2}}}{\sqrt{2\pi q^2}} \quad (4.1)$$

onde x_m é a média dos valores possíveis de x e q^2 sua variância. A distribuição gaussiana é resultado da combinação de diversas fontes de influência independentes e de média finita.

4.2.3 Definição do Filtro de Kalman

Considere um sistema que evolui no tempo de acordo com o modelo

$$\mathbf{x}(t+1) = \mathbf{f}[\mathbf{x}(t), \mathbf{u}(t), \mathbf{v}(t)], \quad (4.2)$$

onde $\mathbf{x}(t)$ é o vetor do estado do sistema no tempo t , $\mathbf{u}(t)$ é a ação tomada no tempo t e $\mathbf{v}(t)$ é o ruído associado à dinâmica.

O vetor $\mathbf{u}(t)$ corresponde à ação, ou um conjunto de ações, que atua sobre o sistema no instante t e muda o seu estado.

O ruído associado à dinâmica é assumido como sendo gaussiano de média zero, com componentes independentes, e covariância

$$\langle v_i(t)v_j(t') \rangle = \mathbf{Q}_i(t)\delta_{ij}\delta_{tt'}. \quad (4.3)$$

onde $\langle \rangle$ representa o valor esperado sobre as variáveis $v_i(t)$ e $v_j(t')$ e δ_{ij} é a função delta de Kronecker (retorna zero se os índices são diferentes e um se são iguais). $\mathbf{Q}_i(t)$ é a covariância da variável $v_i(t)$.

Desta forma, a dinâmica modelada não é determinística e, portanto, haverá sempre uma incerteza quanto ao real estado do sistema após uma ação tomada. Por este motivo, há a necessidade de fazer estimativas sobre o estado do sistema.

O processo de estimativa do estado do sistema começa sempre a partir de uma medida conhecida de seu valor médio e sua variância no tempo inicial ($t = 0$). A partir daí, para cada tempo, o formalismo produz um valor estimado para o estado do sistema e a variância em torno desta estimativa.

A teoria do Filtro de Kalman é linear. Mesmo para o caso de uma dinâmica arbitrária como a da equação 4.2, deve-se supor que o erro de estimativa e o ruído são suficientemente pequenos para que se possa linearizar as equações. Assim podemos dizer que

$$\mathbf{x}(t+1) = \mathbf{f}[\mathbf{x}_e(t), \mathbf{u}(t), 0] + \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \delta \mathbf{x} + \frac{\partial \mathbf{f}}{\partial \mathbf{v}} \mathbf{v}(t) \quad (4.4)$$

onde $\mathbf{x}_e(t)$ é alguma estimativa e $\delta \mathbf{x}$ o seu erro. A expressão $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}$ denota a matriz Jacobiana de elementos $\mathbf{J}_{ij} = \frac{\partial f_i}{\partial x_j}$, onde i é a linha e j é a coluna da matriz.

Uma outra suposição importante é que o erro $\delta \mathbf{x}$ e o ruído \mathbf{v} são variáveis aleatórias não correlacionadas entre si.

Uma das principais vantagens do Filtro de Kalman é estimar variáveis sem que para isso precise guardar todos os estados anteriores. Para tanto, é definido que cada tempo possui duas estimativas: *a priori* e *a posteriori* da medida do estado.

A estimativa *a priori* do estado no tempo $t+1$ é obtida a partir da estimativa *a posteriori* no tempo t e da ação, usando o modelo como se este fosse determinístico (sem ruído, $v(t) = 0$).

Portanto, a estimativa *a priori* é dada por

$$\hat{\mathbf{x}}(t+1|t) = \mathbf{f}[\hat{\mathbf{x}}(t|t), \mathbf{u}(t), 0], \quad (4.5)$$

onde $\hat{\mathbf{x}}(t|t)$ é a estimativa *a posteriori*. A estimativa *a posteriori* ao tempo $t+1$ é a estimativa após a medida do estado e executada a ação.

A Figura 4.2 ilustra estas variáveis temporais. A primeira, $\hat{\mathbf{x}}(t|t)$, é a estimativa *a posteriori* do tempo t . A segunda, $\hat{\mathbf{x}}(t+1|t)$, é a estimativa *a priori* do tempo $t+1$ e a terceira, $\hat{\mathbf{x}}(t+1|t+1)$, é a estimativa *a posteriori* do tempo $t+1$ e assim sucessivamente.

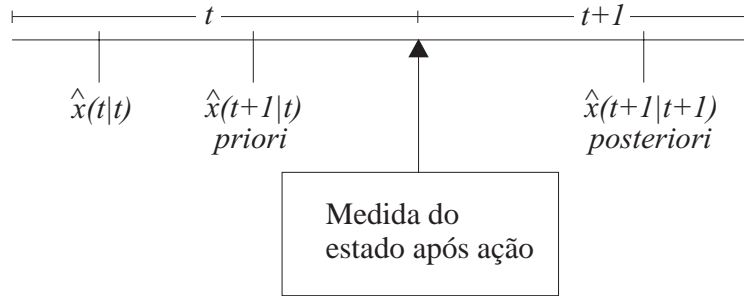


Figura 4.2: Estimativa *a priori* e *a posteriori*

O erro desta estimativa é definido por

$$\delta \mathbf{x}(t+1|t) = \mathbf{x}(t+1) - \hat{\mathbf{x}}(t+1|t) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \delta \mathbf{x} + \frac{\partial \mathbf{f}}{\partial \mathbf{v}} \mathbf{v}(t) \quad (4.6)$$

possuindo covariância

$$\mathbf{P}(t+1|t) = \langle \delta \mathbf{x}(t+1|t) \delta \mathbf{x}(t+1|t)^T \rangle \quad (4.7)$$

$$\mathbf{P}(t+1|t) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \mathbf{P}(t|t) \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^T + \frac{\partial \mathbf{f}}{\partial \mathbf{v}} \mathbf{Q}(t) \left(\frac{\partial \mathbf{f}}{\partial \mathbf{v}} \right)^T \quad (4.8)$$

Se não há como medir o estado do sistema, a variância tende a aumentar e conseqüentemente a incerteza sobre o estado do sistema também. Isso porque a cada tempo um ruído branco é adicionado.

Se há como medir o estado do sistema, então é possível avaliar o erro da estimativa de forma mais precisa. Na prática, as medidas do estado refletem uma quantidade parcial de informação vinda dos sensores, normalmente corrompida com uma certa dose de ruído. Tal como ocorre com as ações do robô, as medidas dos sensores também devem ser modeladas.

Considere que o modelo para a medida $\mathbf{z}(t)$ do estado $\mathbf{x}(t)$ é dado por

$$\mathbf{z}(t) = \mathbf{h}[\mathbf{x}(t), \mathbf{w}(t)], \quad (4.9)$$

onde $\mathbf{w}(t)$ é um ruído gaussiano tal que

$$\langle w(t) \rangle = 0 \quad (4.10)$$

$$\langle w_i(t) w_j(t') \rangle = \mathbf{R}_i \delta_{ij} \delta t t' \quad (4.11)$$

Da mesma forma que no caso da equação de evolução do estado do sistema (eq. 4.4), podemos linearizar a expressão para a medida em torno de uma estimativa do estado do sistema \mathbf{x}_e e do ruído associado ao modelo do sistema \mathbf{w} :

$$\mathbf{z}(t) = \mathbf{h}[\mathbf{x}_e(t), 0] + \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \delta \mathbf{x} + \frac{\partial \mathbf{h}}{\partial \mathbf{w}} \mathbf{w}(t) \quad (4.12)$$

A medida $\mathbf{z}(t + 1)$ pode ser estimada *a priori* a partir da estimativa *a priori* do estado do sistema. Ou seja, sabendo que o robô vai encontrar uma dada marca à frente, se estima o resultado da medida como

$$\hat{\mathbf{z}}(t + 1|t) = \mathbf{h}[\hat{\mathbf{x}}(t + 1|t), 0] \quad (4.13)$$

Observe que considera-se uma medida perfeita, sem ruído.

A diferença entre o valor obtido na medida e o valor estimado da medida é chamada de vetor *inovação*,

$$\delta \mathbf{z}(t + 1|t) = \mathbf{z}(t + 1) - \hat{\mathbf{z}}(t + 1|t) = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \delta \mathbf{x} + \frac{\partial \mathbf{h}}{\partial \mathbf{w}} \mathbf{w}(t) \quad (4.14)$$

A inovação é a única fonte de informação disponível para aprimorar a estimativa do estado do sistema ao tempo $t+1$. O objetivo do Filtro de Kalman é usar esta informação de uma forma ótima.

Assim, a melhor estimativa linear do estado x do sistema usando o modelo dinâmico e a medida observada é a estimativa *a posteriori* dada por

$$\hat{\mathbf{x}}(t + 1|t + 1) = \hat{\mathbf{x}}(t + 1|t) + \mathbf{K} \delta \mathbf{z}(t + 1|t) \quad (4.15)$$

onde a matriz \mathbf{K} é o chamado *ganho de Kalman* e é escolhido de forma a minimizar a variância do erro final da estimação para cada uma das componentes do estado do sistema. Esse erro final tem a seguinte expressão:

$$\delta x_i(t + 1|t + 1) = \delta x_i(t + 1|t) + \sum_j \mathbf{K}_{ij} \delta z_j(t + 1|t) \quad (4.16)$$

A matriz covariância do erro é

$$\mathbf{P}_{ij}(t + 1|t + 1) = \mathbf{P}_{ii}(t + 1|t) + 2 \sum_j \mathbf{K}_{ij} \langle \delta x_i \delta z_j \rangle + \sum_{jk} \mathbf{K}_{ij} \mathbf{K}_{ik} \langle \delta z_j \delta z_k \rangle \quad (4.17)$$

Na literatura, freqüentemente encontramos a expressão filtro linear ótimo para designar o Filtro de Kalman. Para que o filtro seja ótimo, deve-se minimizar a variância do erro (a diagonal da matriz covariância em relação aos elementos do FK),

$$\frac{\partial \mathbf{P}_{ii}}{\partial \mathbf{K}_{il}} = 0 \quad (4.18)$$

que resulta em

$$\langle \delta x_i \delta z_i \rangle + \sum_k \mathbf{K}_{ik} \langle \delta z_i \delta z_k \rangle = 0 \quad (4.19)$$

ou, em notação de matriz,

$$\mathbf{K} = \mathbf{S}_{xz}(t + 1|t) \mathbf{S}_{zz}^{-1}(t + 1|t) \quad (4.20)$$

onde

$$\mathbf{S}_{xz}(t+1|t) = \langle \delta \mathbf{x} \delta \mathbf{z}^T \rangle \quad (4.21)$$

$$\mathbf{S}_{zz}(t+1|t) = \langle \delta \mathbf{z} \delta \mathbf{z}^T \rangle \quad (4.22)$$

A implementação deste resultado é mostrada mais adiante.

4.2.4 Fases do Filtro

De posse da definição do Filtro, é possível delimitar perfeitamente duas fases de processamento. A primeira, chamada de *Fase de Predição*, é responsável por estimar qual será o estado do sistema se uma determinada ação for aplicada a ele. Note-se que a única informação do estado atual do sistema é fornecida no momento da medição, também chamada de *observação*. A segunda fase é conhecida como *Fase de Atualização* e é onde ocorre a comparação entre o que foi estimado e o que realmente foi observado. O erro da estimativa do Filtro, a *inovação*, é o que vai ser utilizado para o cálculo da estimativa *a posteriori*. A Figura 4.3 ilustra todo este processo.

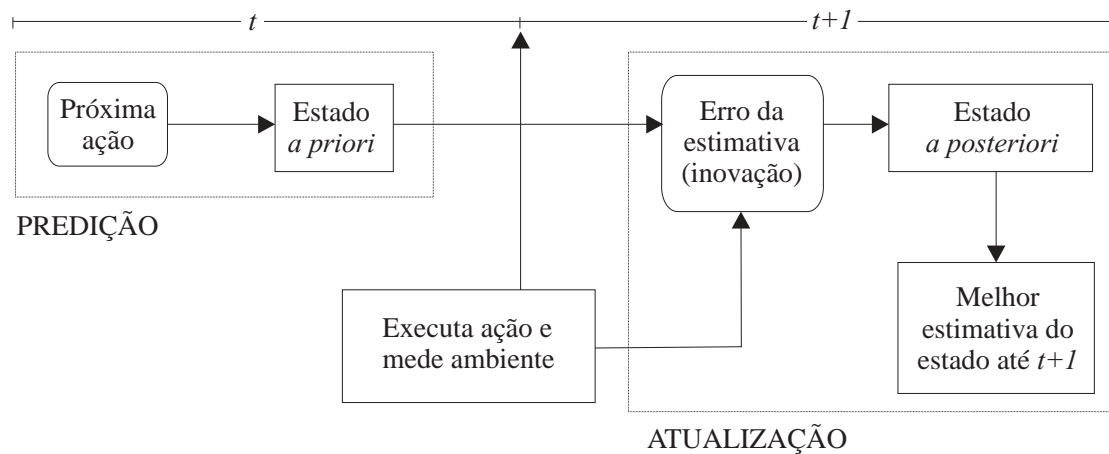


Figura 4.3: Processo de estimativa *a priori* e *a posteriori*

Uma iteração do processo de estimativa do Filtro de Kalman é ilustrada na Figura 4.4. O objetivo é formular uma estimativa \hat{x} de um estado x no instante $t+1$, tendo por base apenas a estimativa no instante t e uma observação feita no instante $t+1$.

4.3 Aplicação na Robótica Móvel

O sistema que interessa para o estudo do Filtro de Kalman aplicado à Robótica Móvel é descrito como um ou mais robôs e um conjunto de objetos. Entende-se por objetos qualquer obstáculo (paredes, portas, mesas, cadeiras) que estiver no ambiente. O estado deste sistema é, portanto, a concatenação do estado do robô com o estado dos objetos.

Considere um sistema com um robô e N objetos representado em um plano cartesiano. O robô e os objetos possuem além da coordenada cartesiana uma orientação neste plano indicando a direção que estão se movendo. Partindo da suposição que os objetos são

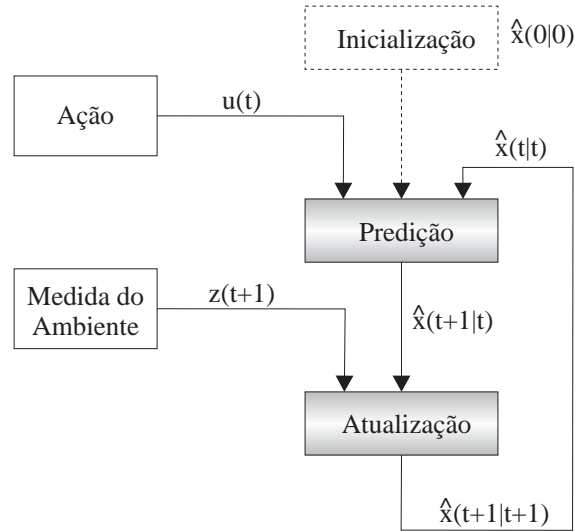


Figura 4.4: Fases do Filtro de Kalman

fixos, não se movimentam no ambiente, e pontuais, pode-se dizer que a orientação para os objetos será sempre nula. Só interessa, portanto, a orientação do robô no ambiente.

O vetor de estado do sistema é, portanto,

$$\mathbf{x} = \begin{bmatrix} x_r \\ y_r \\ \phi \\ x_1 \\ y_1 \\ 0 \\ \vdots \\ x_N \\ y_N \\ 0 \end{bmatrix} \quad (4.23)$$

onde (x_r, y_r, ϕ) é o estado do robô (sua posição e orientação) e $(x_i, y_i, 0)$ é o estado de cada objeto i , com i variando de 1 até N . O vetor de estado é, portanto, de dimensão $3(N + 1)$

O modelo do sistema é a equação de movimento do robô para uma dada ação ao tempo t , uma vez que os objetos estão sendo considerados como fixos.

Este modelo, também chamado modelo cinemático, é

$$\begin{bmatrix} x_r(t+1) \\ y_r(t+1) \\ \phi(t+1) \end{bmatrix} = \begin{bmatrix} x_r(t) \\ y_r(t) \\ \phi(t) \end{bmatrix} + \begin{bmatrix} (V(t) + v_v(t))\cos(\phi(t) + \gamma(t) + v_\gamma(t)) \\ (V(t) + v_v(t))\sen(\phi(t) + \gamma(t) + v_\gamma(t)) \\ \gamma(t) + v_\gamma(t) \end{bmatrix} \quad (4.24)$$

onde $V(t)$ é a velocidade do robô ao tempo t , $\gamma(t)$ é o ângulo que ele rotaciona ao tempo t e $v_v(t)$ e $v_\gamma(t)$ são, respectivamente, o erro na velocidade e o erro no ângulo de rotação introduzidos no tempo t .

O próximo item a definir é como será o modelo de medida do ambiente. Na Figura 4.5, o círculo representa o robô e o retângulo representa o objeto que está sendo observado

por um dos seus sensores. Considera-se o ponto central do círculo como a coordenada cartesiana correspondente ao robô no plano que representa o ambiente. O objeto encontra-se a uma distância d da posição do robô e forma um ângulo θ com o vetor de orientação do robô.

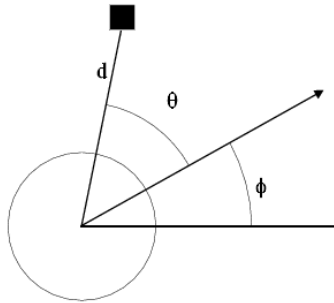


Figura 4.5: Processo de observação do ambiente

A relação da posição do objeto observado com a posição do robô é dada por

$$\begin{bmatrix} d_i(t) \\ \theta_i(t) \end{bmatrix} = \begin{bmatrix} \sqrt{(\Delta x_i)^2 + (\Delta y_i)^2} + w_d(t) \\ \text{tg}^{-1}\left(\frac{\Delta y_i}{\Delta x_i}\right) - \phi(t) + w_\theta(t) \end{bmatrix} \quad (4.25)$$

onde $\Delta x_i = x_r - x_i$, $\Delta y_i = y_r - y_i$ e w_d e w_θ são, respectivamente, erro na medição da distância e erro na medição do ângulo que forma o objeto com a orientação do robô.

Para ilustrar, as jacobianas para as equações de movimento vistas na equação 4.4 são descritas considerando que o ambiente tenha apenas dois objetos. A primeira matriz, $\frac{\partial \mathbf{f}}{\partial \mathbf{x}}$, é

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} 1 & 0 & -V(t)\text{sen}(\phi(t) + \gamma(t)) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & V(t)\text{cos}(\phi(t) + \gamma(t)) & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.26)$$

A segunda matriz, $\frac{\partial \mathbf{f}}{\partial \mathbf{v}}$, mostra a dependência do modelo com os ruídos. É uma matriz $[3(N + 1)] \times [2]$ pois há somente dois termos de ruído, um para a velocidade e outro para o ângulo de rotação. Logo,

$$\frac{\partial \mathbf{f}}{\partial \mathbf{v}} = \begin{bmatrix} \cos(\phi(t) + \gamma(t)) & -V(t)\text{sen}(\phi(t) + \gamma(t)) \\ \text{sen}(\phi(t) + \gamma(t)) & V(t)\cos(\phi(t) + \gamma(t)) \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (4.27)$$

Da mesma forma, as jacobinas para o modelo de medida são descritas a seguir. A matriz $\frac{\partial \mathbf{h}}{\partial \mathbf{x}}$ é uma matriz $[2N] \times [3(N + 1)]$. É definida como

$$\frac{\partial \mathbf{h}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\Delta x_1}{d_1} & \frac{\Delta y_1}{d_1} & 0 & \frac{-\Delta x_1}{d_1} & \frac{-\Delta y_1}{d_1} & 0 & 0 & 0 & 0 \\ \frac{-\Delta y_1}{d_1} & \frac{\Delta x_1}{d_1} & -1 & \frac{\Delta y_1}{d_1} & \frac{-\Delta x_1}{d_1} & 0 & 0 & 0 & 0 \\ \frac{\Delta x_2}{d_2} & \frac{\Delta y_2}{d_2} & 0 & 0 & 0 & 0 & \frac{-\Delta x_2}{d_2} & \frac{-\Delta y_2}{d_2} & 0 \\ \frac{-\Delta y_2}{d_2} & \frac{\Delta x_2}{d_2} & -1 & 0 & 0 & 0 & \frac{\Delta y_2}{d_2} & \frac{-\Delta x_2}{d_2} & 0 \end{bmatrix} \quad (4.28)$$

onde $d_i = \sqrt{(\Delta x_i)^2 + (\Delta y_i)^2}$.

A matriz $\frac{\partial \mathbf{h}}{\partial \mathbf{w}}$ é a matriz identidade de dimensão $[2N] \times [2N]$, pois a medida da posição de cada objeto envolve dois parâmetros, distância e ângulo, cada um deles tendo seu ruído independente. Portanto,

$$\frac{\partial \mathbf{h}}{\partial \mathbf{w}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.29)$$

4.3.1 Usando as Matrizes

A definição teórica do Filtro de Kalman mostra que a partir de uma estimativa de estado $\hat{\mathbf{x}}(t|t)$ com uma covariância conhecida $\mathbf{P}(t|t)$ podemos gerar estimativas para o próximo valor temporal, $t + 1$. A estimativa *a priori* é

$$\hat{\mathbf{x}}(t + 1|t) = \mathbf{f}(\hat{\mathbf{x}}(t|t), \mathbf{u}(t), 0), \quad (4.30)$$

resultado da propagação do modelo sem ruído a partir do estado estimado anterior. O erro teórico desta estimativa é

$$\delta \mathbf{x}(t + 1|t) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \delta \mathbf{x}(t|t) + \frac{\partial \mathbf{f}}{\partial \mathbf{v}} \mathbf{v}(t) \quad (4.31)$$

Com a estimativa *a priori* do estado é possível saber a estimativa *a priori* da medida,

$$\hat{\mathbf{z}}(t + 1|t) = \mathbf{h}(\hat{\mathbf{x}}(t + 1|t), 0), \quad (4.32)$$

cujos erro, também chamado de *inovação* é dado por

$$\delta \mathbf{z}(t+1|t) = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \delta \mathbf{x}(t+1|t) + \frac{\partial \mathbf{h}}{\partial \mathbf{w}} \mathbf{w}(t+1) \quad (4.33)$$

Observa-se que nesta última expressão aparece o erro da equação 4.31. Pode-se então reescrever a equação 4.33 como segue

$$\delta \mathbf{z}(t+1|t) = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \delta \mathbf{x}(t|t) + \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \frac{\partial \mathbf{f}}{\partial \mathbf{v}} \mathbf{v}(t) + \frac{\partial \mathbf{h}}{\partial \mathbf{w}} \mathbf{w}(t+1) \quad (4.34)$$

A estimativa final (*a posteriori*) é obtida de

$$\hat{\mathbf{x}}(t+1|t+1) = \hat{\mathbf{x}}(t+1|t) + \mathbf{K} \delta \mathbf{z}(t+1|t) \quad (4.35)$$

onde o ganho de Kalman \mathbf{K} representa o uso ótimo da inovação para extrair informação da medida. \mathbf{K} é o produto de duas matrizes de covariância

$$\mathbf{K} = \langle \delta \mathbf{x}(t+1|t) \delta \mathbf{z}(t+1|t)^T \rangle \langle \delta \mathbf{z}(t+1|t) \delta \mathbf{z}(t+1|t)^T \rangle^{-1} \quad (4.36)$$

Usando as expressões da seção anterior, são calculadas cada uma das covariâncias. Primeiramente,

$$\mathbf{S}_{xz}(t+1|t) = \langle \delta \mathbf{x}(t+1|t) \delta \mathbf{z}(t+1|t)^T \rangle \quad (4.37)$$

O cálculo é simplificado se for levado em consideração o fato de que a matriz $\frac{\partial \mathbf{h}}{\partial \mathbf{w}}$ é igual à identidade, ou seja, pode ser igualada a 1 na equação 4.33. Além disso, as variáveis aleatórias $\delta \mathbf{x}(t|t)$, $\mathbf{v}(t)$ e $\mathbf{w}(t+1)$ são tais que

$$\langle \delta \mathbf{x}(t|t) \mathbf{v}(t)^T \rangle = 0 \quad (4.38)$$

$$\langle \delta \mathbf{x}(t|t) \mathbf{w}(t+1)^T \rangle = 0 \quad (4.39)$$

$$\langle \mathbf{v}(t) \mathbf{w}(t+1)^T \rangle = 0 \quad (4.40)$$

e as matrizes de covariância \mathbf{P} , \mathbf{Q} e \mathbf{R} são tais que

$$\langle \delta \mathbf{x}(t|t) \delta \mathbf{x}(t|t)^T \rangle = \mathbf{P}(t|t) \quad (4.41)$$

$$\langle \mathbf{v}(t|t) \mathbf{v}(t|t)^T \rangle = \mathbf{Q}(t) \quad (4.42)$$

$$\langle \mathbf{w}(t+1) \mathbf{w}(t+1)^T \rangle = \mathbf{R}(t+1) \quad (4.43)$$

onde \mathbf{Q} é a chamada matriz de covariância do ruído do sistema e \mathbf{R} é a matriz de covariância do ruído da medida.

Portanto, reescrevendo a equação tem-se

$$\mathbf{S}_{xz}(t+1|t) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \mathbf{P}(t|t) \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^T \left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right)^T + \frac{\partial \mathbf{f}}{\partial \mathbf{v}} \mathbf{Q}(t) \left(\frac{\partial \mathbf{f}}{\partial \mathbf{v}} \right)^T \left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right)^T \quad (4.44)$$

Enquanto que

$$\mathbf{S}_{zz}(t+1|t) = \langle \delta \mathbf{z}(t+1|t) \delta \mathbf{z}(t+1|t)^T \rangle \quad (4.45)$$

resulta

$$\mathbf{S}_{zz}(t+1|t) = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \mathbf{P}(t|t) \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^T \left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right)^T + \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \frac{\partial \mathbf{f}}{\partial \mathbf{v}} \mathbf{Q}(t) \left(\frac{\partial \mathbf{f}}{\partial \mathbf{v}} \right)^T \left(\frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right)^T + \mathbf{R}(t+1) \quad (4.46)$$

As matrizes calculadas nas equações 4.44 e 4.46, \mathbf{S}_{xz} e \mathbf{S}_{zz} , combinadas, geram o ganho de Kalman. A primeira é uma matriz de dimensão $[3(N + 1)] \times [2N]$ enquanto a outra tem dimensão $[2N] \times [2N]$.

A covariância da estimativa *a posteriori* é atualizada como

$$\mathbf{P}(t + 1|t + 1) = \mathbf{P}(t + 1|t) - \mathbf{K}\mathbf{S}_{zz}(t + 1|t)\mathbf{K}^T \quad (4.47)$$

Finalmente, é possível enumerar o conjunto de equações que define o algoritmo do Filtro de Kalman. A seguir é apresentado um resumo das equações divididas em suas respectivas fases.

Fase de Predição: Definição da ação e cálculo das variáveis *a priori*.

$$\hat{\mathbf{x}}(t + 1|t) = \mathbf{f}(\hat{\mathbf{x}}(t|t), \mathbf{u}(t), 0) \quad (4.48)$$

$$\hat{\mathbf{z}}(t + 1|t) = \mathbf{h}(\hat{\mathbf{x}}(t + 1|t), 0) \quad (4.49)$$

$$\mathbf{P}(t + 1|t) = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \mathbf{P}(t|t) \left(\frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right)^T + \frac{\partial \mathbf{f}}{\partial \mathbf{v}} \mathbf{Q}(t) \left(\frac{\partial \mathbf{f}}{\partial \mathbf{v}} \right)^T \quad (4.50)$$

Fase de Atualização: Realização da medida sensorial e cálculo das variáveis *a posteriori*.

$$\delta \mathbf{z}(t + 1|t) = \mathbf{z}(t + 1) - \hat{\mathbf{z}}(t + 1|t) \quad (4.51)$$

$$\mathbf{K} = \mathbf{S}_{xz} \mathbf{S}_{zz}^{-1} \quad (4.52)$$

$$\hat{\mathbf{x}}(t + 1|t + 1) = \hat{\mathbf{x}}(t + 1|t) + \mathbf{K} \delta \mathbf{z}(t + 1|t) \quad (4.53)$$

$$\mathbf{P}(t + 1|t + 1) = \mathbf{P}(t + 1|t) - \mathbf{K}\mathbf{S}_{zz}(t + 1|t)\mathbf{K}^T \quad (4.54)$$

4.3.2 Adaptações de implementação

A principal dificuldade para a implementação do Filtro de Kalman no mapeamento de ambientes é o fato de que a dimensão das matrizes aumenta com o quadrado do número de objetos detectados. Além disso, o aumento da dimensão das matrizes torna o problema mais crítico para o cálculo de suas inversas. A inversão de matrizes, em geral, é uma operação custosa e nem sempre possível (algumas matrizes não possuem inversa).

Uma forma de implementar o Filtro de Kalman sem ter que inverter matrizes na sua dimensão completa é considerar uma única medida a cada tempo. Ou seja, o que na prática é uma ação ao tempo t seguida de uma série de medidas simultâneas de posição dos objetos ao tempo $t + 1$ pode ser implementado como uma ação ao tempo t seguida de uma sequência de medições nos tempos t_i , onde a cada um desses tempos somente um objeto é observado e nenhuma ação é tomada. Ou seja,

$$(t) \rightarrow a \rightarrow (t + 1 : n) \quad (4.55)$$

passa a ser

$$(t) \rightarrow a \rightarrow (t_1 : m_1) \rightarrow (t_2 : m_2) \rightarrow \dots \rightarrow (t_n : m_n) \quad (4.56)$$

onde a significa uma ação, n o número de medidas e m_i significa uma única medida. Em outras palavras, ao invés da forma completa de $\frac{\partial \mathbf{h}}{\partial \mathbf{x}}$ usa-se somente as linhas que correspondem às variáveis de medida de um determinado objeto. Por exemplo, se a medida foi referente ao segundo objeto, a matriz será somente

$$\frac{\partial \mathbf{h}}{\partial \mathbf{x}_2} = \begin{bmatrix} \frac{\Delta x_2}{d_2} & \frac{\Delta y_2}{d_2} & 0 & 0 & 0 & 0 & -\frac{\Delta x_2}{d_2} & -\frac{\Delta y_2}{d_2} & 0 \\ -\frac{\Delta y_2}{d_2} & \frac{\Delta x_2}{d_2} & -1 & 0 & 0 & 0 & \frac{\Delta y_2}{d_2} & -\frac{\Delta x_2}{d_2} & 0 \end{bmatrix} \quad (4.57)$$

Passando para o caso do robô, o movimento será uma sequência genérica de ações e medidas únicas

$$(t_0 : m_0) \rightarrow a_0 \rightarrow (t_1 : m_1) \rightarrow a_1 \rightarrow (t_2 : m_2) \rightarrow \dots \rightarrow (t_n : m_n) \quad (4.58)$$

podendo algumas ações a_i serem iguais a zero, ou seja, não haver ação entre o tempo t_i e o tempo t_{i+1} . Pode existir também m_i igual a zero, ou seja, nada ter sido detectado pela medida do ambiente no tempo t_i .

Sendo assim, é possível enumerar algumas regras para que o sistema funcione com uma medição por unidade de tempo:

- Cada vez que houver uma ação deve-se calcular uma estimativa *a priori* do novo estado.
- Se não houver uma ação, a estimativa *a priori* de um dado tempo é igual a estimativa *a posteriori* do tempo anterior.
- Para um dado tempo, se houver uma medida, o filtro de Kalman deve ser usado para gerar a estimativa *a posteriori*.
- Se não houver uma medida, o Filtro de Kalman não é usado e a estimativa *a posteriori* é igual a estimativa *a priori* para aquele tempo.

5 MÓDULO DE AUTO-LOCALIZAÇÃO

Neste capítulo, é proposta uma solução para o problema da auto-localização para um robô que realiza simultaneamente a exploração e o mapeamento de ambientes. Esta solução utiliza o Filtro de Kalman para estimar a posição real de um robô e ajustá-la no mapa global resultante da exploração.

Do ponto de vista do Agente Exploratório definido por Edson Prestes (PRESTES, 2003), esta é uma proposta para implementar o módulo de auto-localização. Este módulo foi implementado e testado em simulação utilizando as adaptações de implementação descritas na Seção 4.3.2.

O capítulo está organizado da seguinte maneira. Na Seção 5.1 é apresentado o Simulador do Agente Exploratório, suas funções e características de implementação. Na seção seguinte, é apresentada na forma de exemplos a principal motivação para este trabalho. A Seção 5.3 descreve como foi feita a associação deste módulo de auto-localização com o método de exploração. Por fim, a Seção 5.4 apresenta os experimentos que foram realizados em simulação e faz comparações entre os mapas gerados, com e sem marcas de Kalman.

5.1 O Simulador do Agente Exploratório

O ponto de partida da implementação do módulo de auto-localização é um simulador desenvolvido para plataforma Windows com a ferramenta *Borland Delphi*, versão 7.0. Este simulador foi desenvolvido com o objetivo de testar o método exploratório (PRESTES, 2003) e é através dele que o usuário monta o ambiente que será simulado, altera parâmetros do sistema e executa a simulação da exploração.

Primeiramente, a interface disponibiliza objetos de formas geométricas variadas. Os objetos podem ser colocados em qualquer ponto da janela que representa o ambiente e podem ser redimensionados e movimentados pelo usuário. A movimentação dos objetos pode acontecer inclusive durante a exploração. Esse comportamento torna o objeto dinâmico para o robô.

Além de objetos, o usuário pode inserir no ambiente um objeto especial chamado *objetivo*. Quando este objeto está presente, o processo exploratório é executado com o propósito de encontrar o objetivo. Se não está presente, o propósito é simplesmente construir um mapa de todo o ambiente simulado.

A Figura 5.1 mostra um exemplo de ambiente montado pelo usuário com os objetos disponíveis para simulação.

O Simulador permite também que o usuário altere parâmetros de execução, tais como velocidade, ângulo inicial e alcance da visão do robô. Também permite alterar parâmetros do método exploratório, número de iterações para estabilização, tamanho das

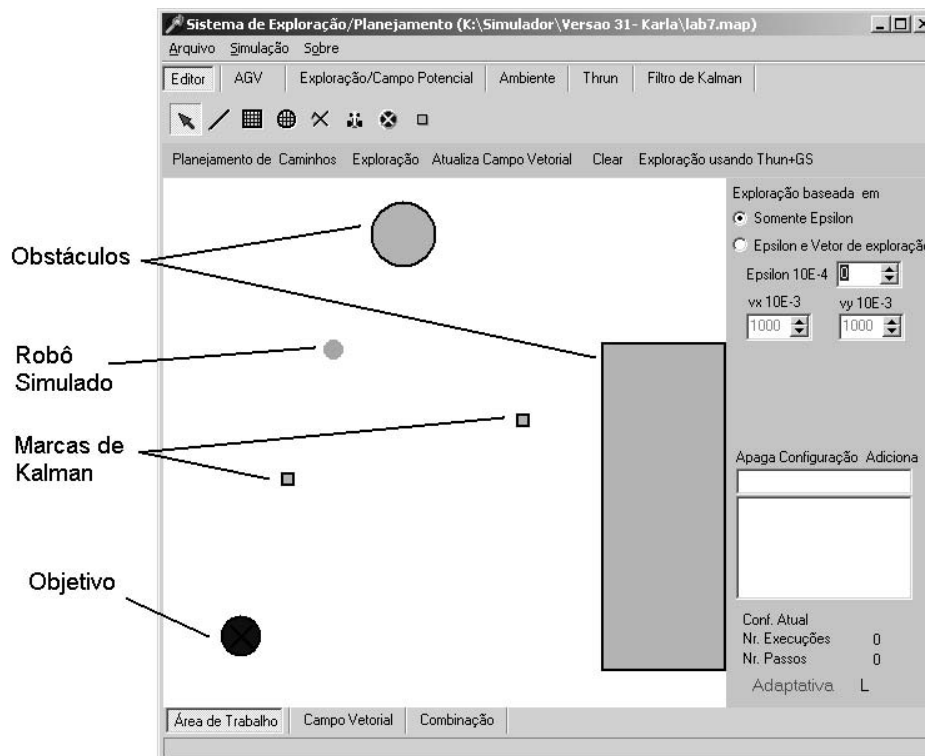


Figura 5.1: Tela do Simulador com um ambiente montado pelo usuário.

células da grade, entre outros.

Apesar de ter sido desenvolvido para simulação, existe a possibilidade de trabalhar com um robô real utilizando tanto ambiente simulado quanto ambiente real no Simulador. A comunicação é feita através de um componente *Delphi* desenvolvido para este fim e um cabo acoplado ao robô.

O robô que se comunica com o Simulador é o mini-robô Khepera, fabricado por K-Team, Suíça. O robô possui 8 sensores de proximidade e luminosidade (infravermelho), 55mm de diâmetro e 30mm de altura, pesando aproximadamente 70g. O tamanho reduzido permite experimentos em uma pequena área, e facilita a criação de ambientes complexos e cheios de obstáculos de forma simples e com baixo custo. Além disso, a comunicação se faz através da porta serial do computador, o que possibilita implementar sistemas independentemente de plataforma, sistema operacional ou linguagem de programação.

Neste trabalho, os testes foram realizados apenas em simulação, sem envolver o robô real. Entretanto, o Simulador encontra-se preparado para comunicação com o robô Khepera. Os testes não foram feitos no robô real devido, principalmente, a limitações de hardware. Os erros de odometria são grandes demais devido ao cabo permanentemente conectado ao robô, inviabilizando, assim, os testes.

5.2 Motivação do uso de auto-localização

Considere um robô que percorre um ambiente com o objetivo de mapeá-lo. Há duas fontes de erro que são simuladas e que deverão ser tratadas pelo módulo de auto-localização:

- erros de movimentação do robô: são os erros de odometria. Na simulação, são inseridos na velocidade do robô e no ângulo de rotação, ou seja, o robô segue a dinâmica definida pela equação 4.24, onde as variáveis aleatórias v_v e v_γ são geradas a cada instante de tempo a partir de um gerador pseudo-aleatório gaussiano com variâncias $\sigma_v = \langle v_v^2 \rangle$ e $\sigma_\gamma = \langle v_\gamma^2 \rangle$ respectivamente. A partir daqui chamaremos σ_v e σ_γ de níveis de ruído;
- erros de leitura dos sensores: são erros de posicionamento do objeto. Eles são inseridos na medida da distância que o objeto se encontra do robô e na medida do ângulo que o objeto forma com a orientação do robô (ver Figura 4.5).

Suponha que o robô comece a exploração do ambiente e sofra um erro pequeno no seu ângulo de orientação. Como esse erro é incremental e não há formas de corrigi-lo, o robô segue construindo um mapa que é levemente diferente do mapa real. Com o avanço da exploração, no entanto, este mapa vai se deformando pois o robô não se encontra na posição que estima estar.

Na Figura 5.2, há quatro experimentos, cada um inserindo um nível diferente de ruído na orientação do robô. O erro de velocidade é no máximo 0,01% da velocidade do robô e não é incremental como o erro na orientação. O nível de ruído na velocidade é o mesmo para todos os experimentos deste capítulo.

O mapa interno do robô está sobreposto ao ambiente simulado, com o campo potencial representado pelas setas e os objetos identificados representados por células preenchidas. O círculo com preenchimento representa o robô real e o círculo sem preenchimento representa a estimativa da posição do robô fornecida pelo Filtro de Kalman.

A Figura 5.2 mostra a deformação do mapa com o aumento do nível de ruído no ângulo de orientação do robô. Note a deformação das paredes, por exemplo. Quando o erro de posição do robô é grande, as células que representam as paredes não formam uma linha reta como deveriam.

Durante os experimentos, foi possível constatar que as maiores deformações do mapa ocorrem com a inserção de erros na orientação do robô. Experimentos onde o robô sofre apenas erro na velocidade deslocaram o mapa, mas não o deformaram.

Suponha que, em determinado momento, o robô perceba o erro de seus odômetros e corrija sua posição. Mesmo que ele corrija todo o mapa gerado até aquele momento com o mesmo deslocamento, ele ajusta os objetos mais próximos de sua posição, mas desajusta os que foram identificados no início da exploração, quando o erro ainda era pequeno.

Portanto, a solução de simplesmente ajustar todo o mapa de acordo com o erro identificado no robô não soluciona o problema. É muito difícil para o robô identificar o objeto que foi mapeado erroneamente durante a exploração. Essa identificação só será possível se numa segunda observação deste mesmo objeto o robô estiver com sua posição corrigida ou se forem feitos vários ciclos de exploração sobre o mesmo ambiente.

Então, pode-se deduzir que, na presença de ruído, construir um mapa mais fiel ao ambiente real exige vários momentos de ajuste durante a exploração. Esses momentos, fazendo uma analogia com o sistema sensorial humano, são como *pontos de referência*. Quando não possui a capacidade de enxergar o ambiente que está percorrendo, o homem procura por portas, cadeiras, paredes para se localizar.

Note que percorrer um corredor às escuras é muito diferente de percorrer uma sala cheia de obstáculos sob a mesma situação. Portanto, quanto mais complexo for o ambiente maior será a quantidade de *pontos de referência* necessários.

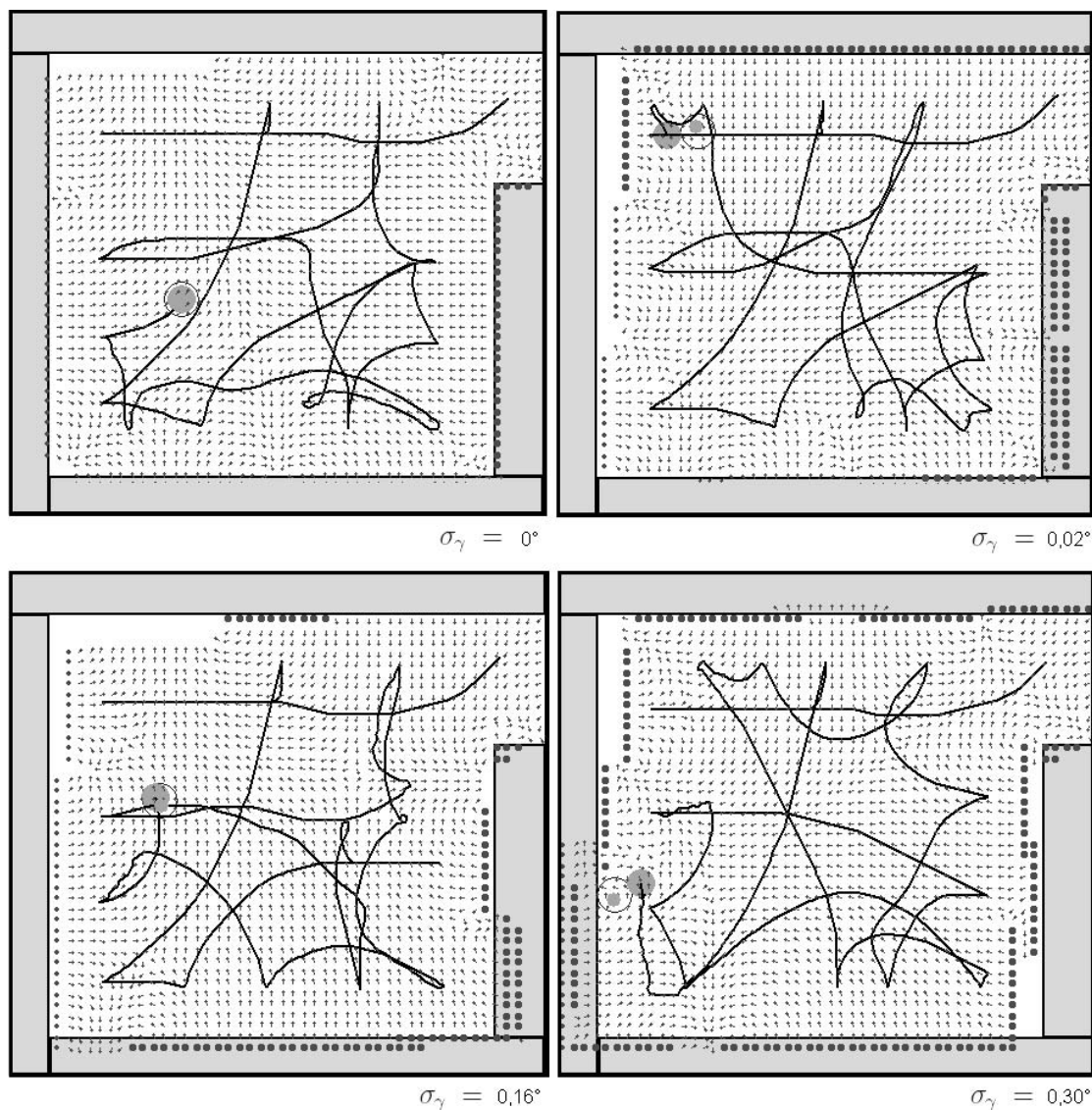


Figura 5.2: Deformação do mapa em quatro simulações de um mesmo ambiente, com diferente nível de ruído no ângulo de orientação do robô.

É nesta constatação que se baseia o módulo de auto-localização proposto. Os *pontos de referência* são inseridos no simulador como objetos especiais chamados *marcas de Kalman*. Esses objetos são reconhecidos apenas pelo módulo de auto-localização e podem estar em qualquer ponto do ambiente, inclusive dentro de outros objetos.

5.3 Associação com o Campo Potencial

Como apresentado na Seção 3.1.2, a representação usada no módulo de exploração é a grade de ocupação. Nesta representação, o ambiente é dividido em células que são classificadas como ocupadas ou não, de acordo com o retorno dos sensores. Neste tipo de representação, objetos não têm identidade, a única coisa que importa é se eles obstruem ou não o movimento do robô. Assim, objetos próximos são fundidos de forma indistinguível em um mesmo grupo de células ocupadas.

Além disso, um ambiente de poucos objetos, por exemplo, pode ser representado na grade de ocupação por centenas de células ocupadas. O filtro de Kalman, entretanto, necessita o reconhecimento preciso de objetos cuja posição serve como indicador do tipo de correção de erro que vai ser realizada.

Em outras palavras, diferentemente da representação da grade de ocupação que é discreta, as *marcas de Kalman* são representadas pela sua posição (x, y) no espaço contínuo bidimensional R^2 .

Isto representa um sério obstáculo para implementar o Filtro de Kalman diretamente usando informações da grade de ocupação. O filtro considera os objetos como sendo todos pontuais, inclusive o robô. Isso torna a representação de objetos longos muito onerosa, uma vez que um mesmo objeto será representado por vários objetos do vetor de estados do Kalman (ver Equação 4.23). A consequência disso é uma matriz de covariância muito grande. Note que a matriz de covariância aumenta quadraticamente com o número de objetos.

5.3.1 Implementação

A primeira implementação do Filtro de Kalman foi realizada considerando que o filtro teria capacidade de identificar todos os objetos presentes no ambiente, reposicionar o mapa completo e só então atualizar as células da grade. Essa implementação mostrou ser inviável para objetos longos ou em grande número pelas razões enumeradas na seção anterior.

A segunda implementação já trouxe o conceito de *marcas de Kalman* e operou de forma integrada com o método exploratório.

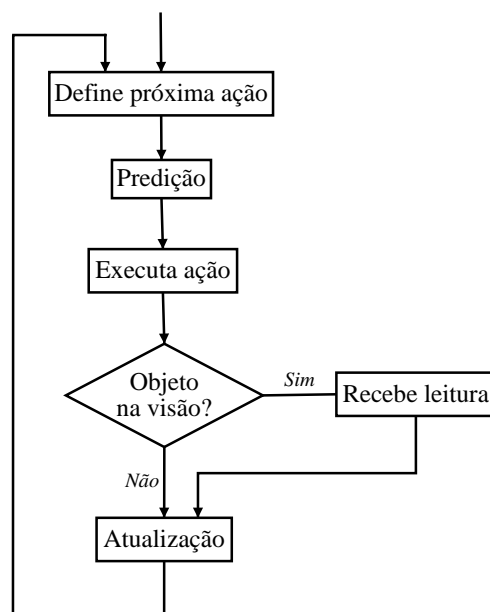


Figura 5.3: Fluxograma do Filtro de Kalman no simulador

A Figura 5.3 apresenta o fluxograma da implementação do Filtro de Kalman no simulador. As funções predição e atualização são as duas fases do filtro. A fase de predição é executada quando o robô já conhece sua próxima ação, mas não a executou ainda. Ele calcula uma estimativa de qual será seu próximo estado e após executar a ação

compara esta estimativa com o estado atual. Os ajustes de posição só são executados no momento que há uma *marca de Kalman* no alcance de visão do robô.

5.3.2 Marcas de Kalman

Existem duas características do FK que influenciam diretamente no tamanho das matrizes de covariância, ponto crítico para implementação do filtro na Robótica Móvel. São elas:

1. cada objeto longo do ambiente é identificado pelo FK como um conjunto de objetos pontuais;
2. o mapeamento dos objetos do ambiente pelo FK não é viável de implementação em ambientes com grande número de objetos.

A inserção de *marcas de Kalman* é uma forma de solucionar os dois problemas de uma única vez. As marcas são objetos pequenos, quase pontuais, que são inseridas no ambiente em pontos estratégicos. No robô real, essas marcas poderiam ser identificadas por um sensor diferente, como uma câmera. Exemplos de entidades que poderiam representar marcas de Kalman em um ambiente real são: o centro de um quadro, um canto da sala, um canto de mesa, uma luminária no teto (posição central do ambiente).

No simulador, as *marcas de Kalman* são objetos especiais do ambiente, facilmente reconhecíveis pela visão do robô, mas que não são considerados como objetos a serem mapeados. Ou seja, as *marcas de Kalman* não atualizam o mapa da grade e, conseqüentemente, não repelem o robô como os outros objetos.

Entretanto, as marcas podem estar dentro de objetos que repelem o robô. Elas serão vistas somente se estiverem próximas aos limites do objeto. É o caso de uma parede que tem uma marca de Kalman fixada e posicionada em um ponto específico.

No algoritmo de exploração há tratamento para os objetos móveis do ambiente. Entretanto, as marcas de Kalman, por serem pontos de referência, não podem se movimentar dentro do ambiente nem podem estar em um objeto que se movimenta. Se isso ocorrer, o módulo de auto-localização não conseguirá corrigir a posição do robô e, muito provavelmente, irá aumentar ainda mais o seu erro. Logo, as marcas devem estar sempre fixas no ambiente.

Pode-se considerar que cada *marca de Kalman* tem uma área de influência. Ou seja, cada marca de Kalman é responsável pela correção de posição de um certo número de objetos nas suas vizinhanças. Assim o sistema de marcas da Kalman propicia que o mapa seja corrigido localmente.

Ao fim da exploração, a ocorrência de erros no mapa gerado terá sido minimizada pelos ajustes efetuados pelo FK toda vez que o robô identifica uma *marca de Kalman*. A estratégia, portanto, não efetua diretamente a correção nos objetos mapeados, mas indiretamente os corrige uma vez que o robô se reposiciona no ambiente toda vez que identifica uma marca.

Nesse momento, é possível identificar dois níveis de mapeamento: o nível do Filtro de Kalman que mapeia apenas os pontos relevantes do ambiente através das marcas; e o nível da grade de ocupação que mapeia os objetos propriamente ditos. Ocorre, então, uma interação desses dois níveis de mapeamento a fim de produzir uma melhor estimativa da posição correta do robô no mapa. A figura 5.4 ilustra esta interação entre os dois níveis de mapeamento.

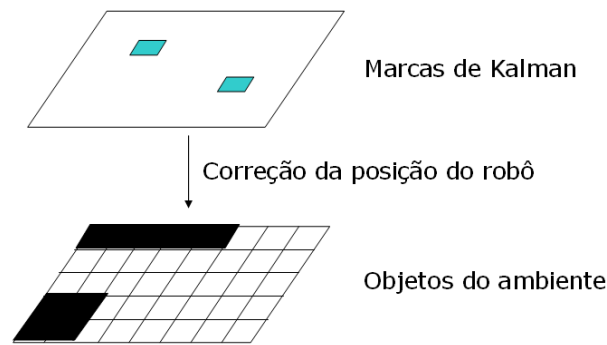


Figura 5.4: Níveis de mapeamento

5.4 Experimentos

Nesta seção, são apresentados alguns dos experimentos realizados com o módulo de auto-localização em simulação. A seção apresenta os mapas gerados durante a exploração, com e sem marcas de Kalman, na presença de diferentes níveis de ruído na orientação do robô. Mostra também um experimento onde a identidade das marcas se faz necessária. Em todos os casos, apresenta o mapa gerado sobreposto ao ambiente simulado e o gráfico dos erros em função do número de iterações.

5.4.1 Deformação do Mapa

Primeiramente, o mesmo experimento da Figura 5.2 foi realizado em outro ambiente. A Figura 5.5 mostra os mapas sobrepostos ao ambiente simulado, com os mesmos níveis de ruído na orientação do robô, sem nenhuma marca de Kalman. A Figura 5.6 mostra os respectivos mapas gerados pelo campo potencial, sem a sobreposição com o ambiente simulado. O comportamento do erro entre a posição real e a posição estimada do robô pode ser visto nos gráficos da Figura 5.7. O gráfico quando não há erro é uma linha reta junto ao eixo da iteração e não aparece nesta figura.

5.4.2 Exploração com Marcas de Kalman

O momento de maior certeza na posição correta do robô é o momento inicial da exploração. Conforme se movimenta, o robô procura pelas marcas de Kalman. As primeiras marcas localizadas provavelmente estarão na posição correta pois o erro do robô é pequeno. Ao encontrar estas marcas novamente elas serão usadas para fazer o ajuste na posição do robô.

Entranto, se uma marca é encontrada quando o robô já não se encontra na posição correta (a distância entre o robô simulado e o que representa o robô real é grande), a marca será posicionada com o referencial do robô simulado. Ao ser encontrada novamente, a marca vai corrigir a posição do robô para a mesma posição que ele estava quando a encontrou.

Isso é perfeitamente compreensível uma vez que se não conhece a marca de Kalman o robô atribui a posição para a marca relativa a sua posição atual. No ambiente real, o robô criaria suas próprias marcas nos momentos que considerasse estar na posição correta. Exatamente como o homem faz ao percorrer um caminho com a intenção de posteriormente realizar o caminho de volta.

A Figura 5.8 mostra os mapas sobrepostos ao ambiente simulado, com 5 marcas de Kalman. A Figura 5.9 mostra os respectivos mapas gerados pelo campo potencial, sem a sobreposição com o ambiente simulado. O comportamento do erro entre a posição real e a posição estimada do robô pode ser visto nos gráficos da Figura 5.10.

5.4.3 Identidade das Marcas de Kalman

Quando o erro em sua posição é muito grande, o robô não reconhece uma marca de Kalman já identificada. Isso acontece porque a identificação das marcas é feita através da posição relativa ao robô, ou seja, uma marca é reconhecida quando é detectada numa certa região. Se o robô não encontra aquela marca inserida no FK, ele cria uma nova.

Em outras palavras, uma mesma marca acaba por representar, ao mesmo tempo, duas marcas para o FK. Se a primeira está correta, ela vai corrigir a posição do robô enquanto que a segunda, que certamente está errada, vai inserir novo erro na posição.

Entretanto, numa situação real, o robô reconheceria as marcas não por sua posição e sim por sua forma, cor e tamanho. Imagine um quadro na parede cujo centro é a marca de Kalman. Ao detectar o quadro com uma câmera, por exemplo, o robô saberia que esta é a sua referência. Isso traz a necessidade de dar uma identificação única para cada marca detectada. Dessa forma, o robô saberia que a marca identificada é o quadro na parede e não um outro quadro.

Em simulação, a identidade das marcas apresentou um ótimo resultado nos casos onde o erro é muito grande. A Figura 5.11 mostra uma exploração sem identidade para as marcas e outra com identidade. A Figura 5.12 mostra os respectivos mapas gerados pelo campo potencial, sem a sobreposição com o ambiente simulado. O comportamento do erro entre a posição real e a posição estimada do robô pode ser visto nos gráficos da Figura 5.13.

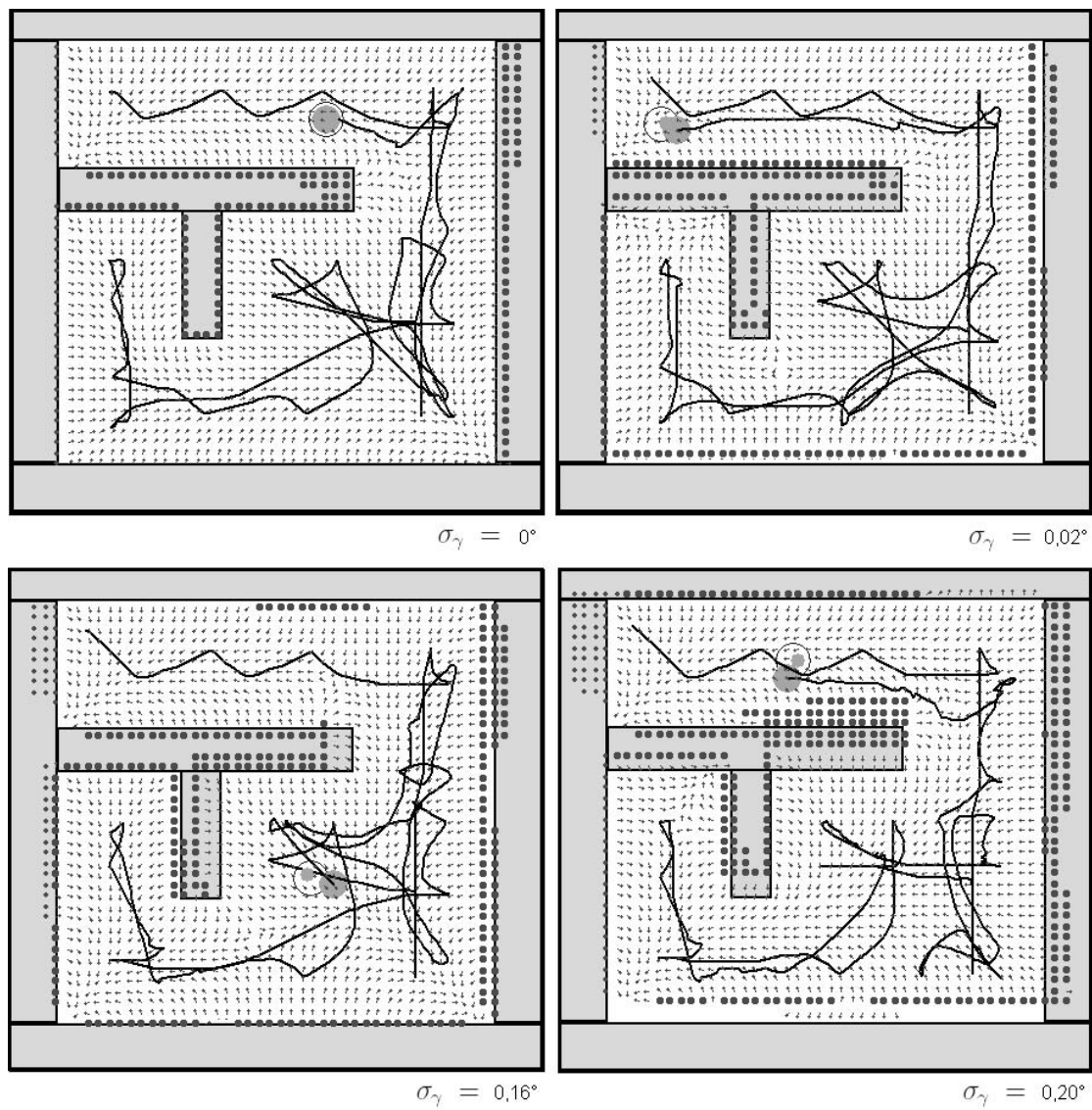


Figura 5.5: Sobreposição do ambiente simulado com o campo potencial de quatro explorações de um mesmo ambiente, com diferentes níveis de ruído no ângulo de orientação do robô.

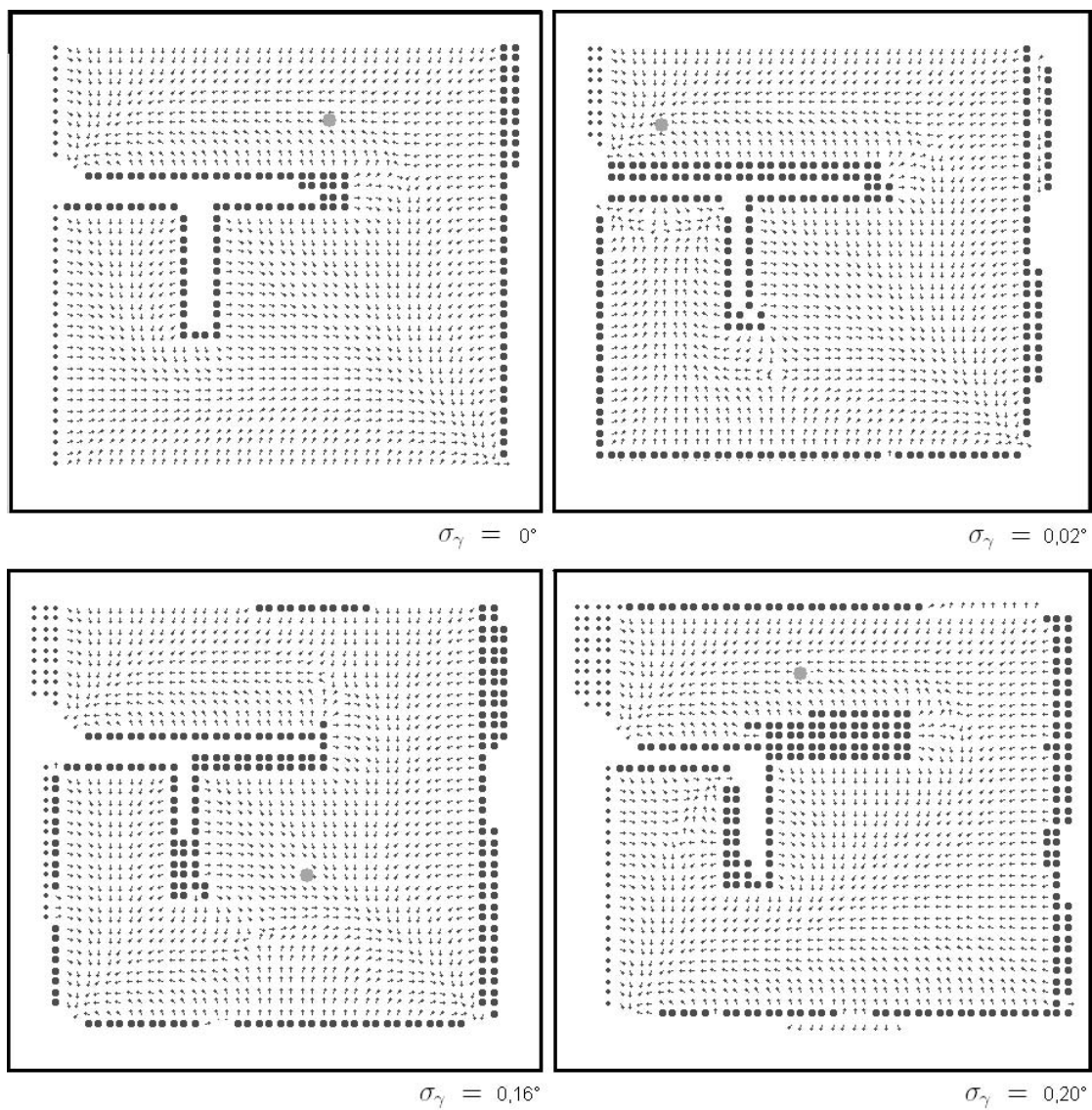


Figura 5.6: Campo potencial de quatro explorações de um mesmo ambiente, com diferentes níveis de ruído no ângulo de orientação do robô.

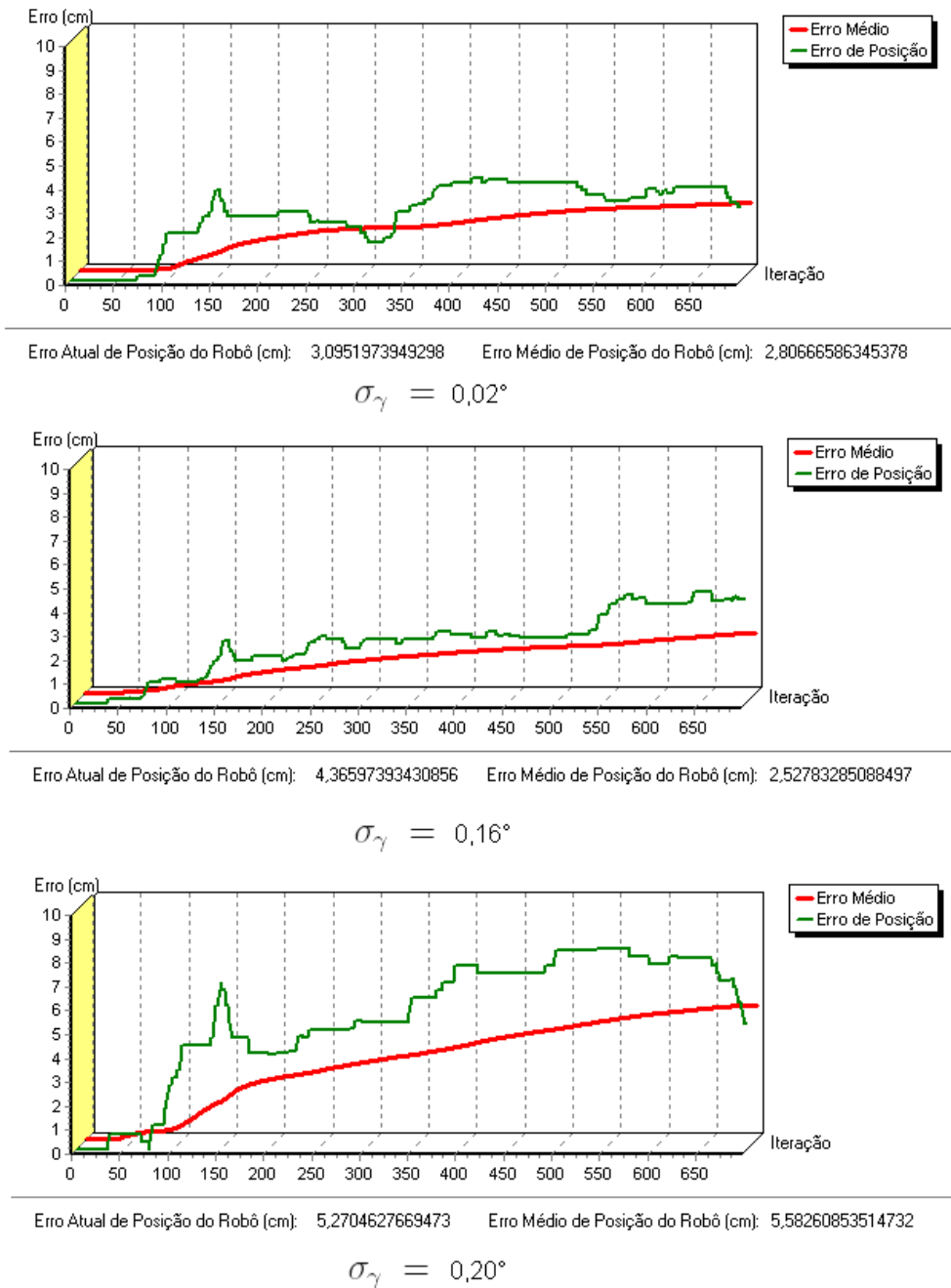


Figura 5.7: Erro e erro médio da posição do robô durante a exploração.

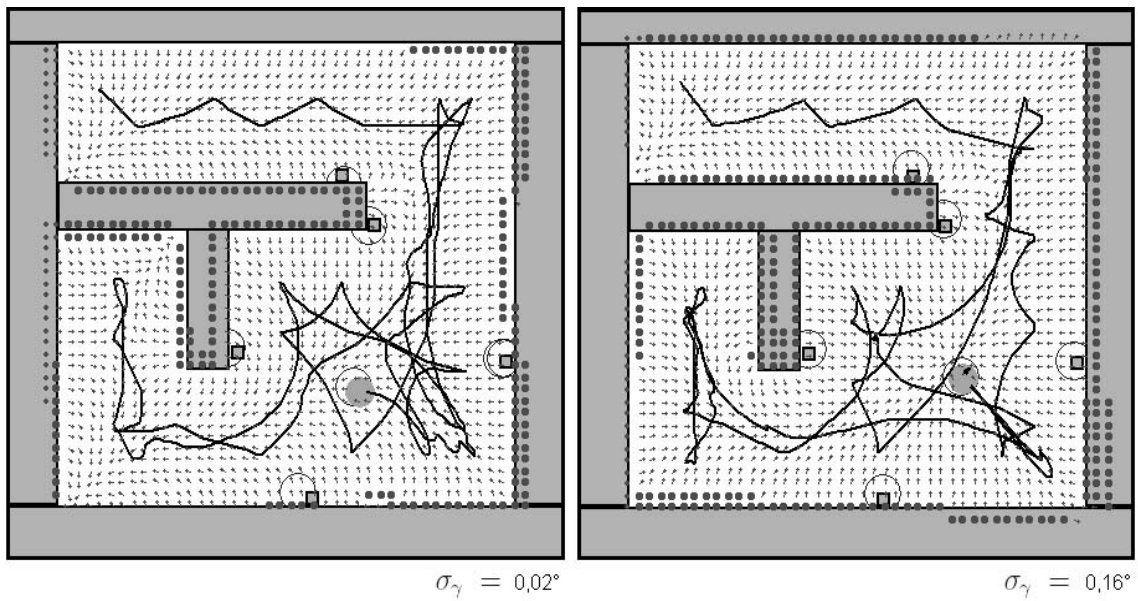


Figura 5.8: Sobreposição do ambiente simulado com o campo potencial utilizando marcas de Kalman

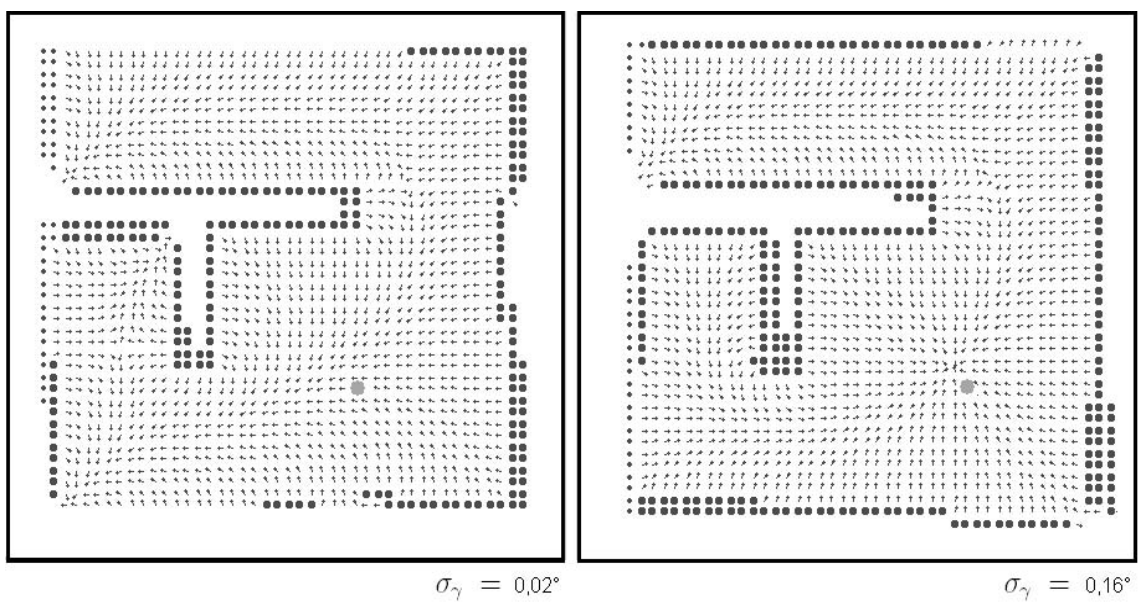


Figura 5.9: Mapa gerado com as marcas de Kalman

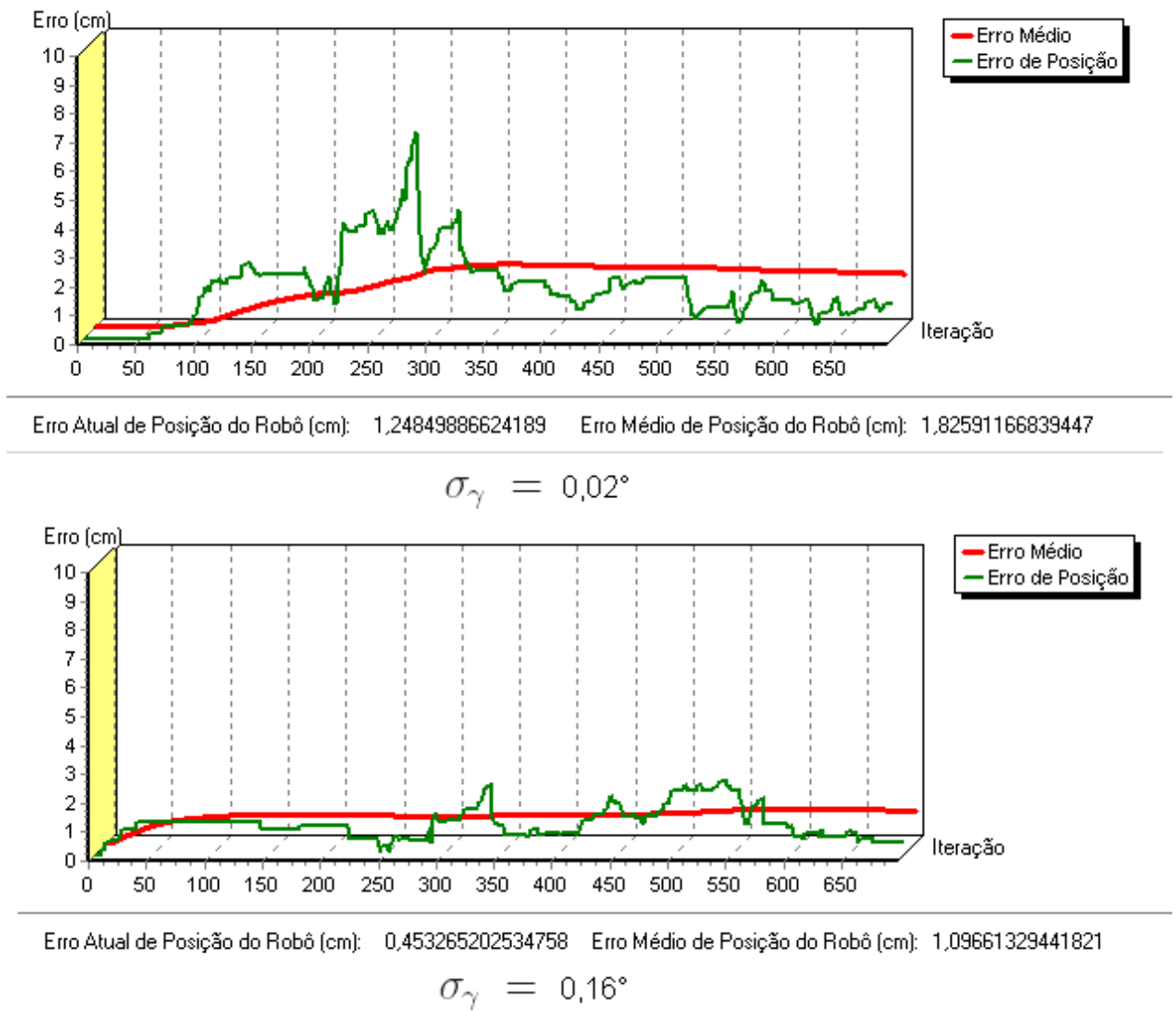


Figura 5.10: Erro e erro médio da posição do robô durante a exploração com marcas de Kalman

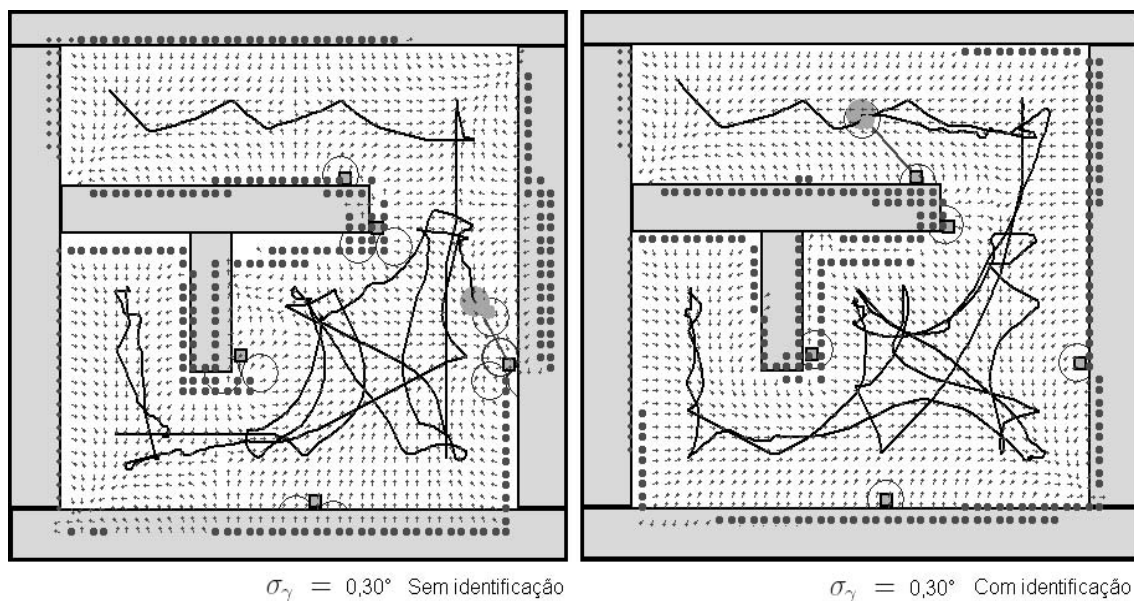


Figura 5.11: Sobreposição do ambiente simulado com o campo potencial utilizando marcas de Kalman sem e com identidade

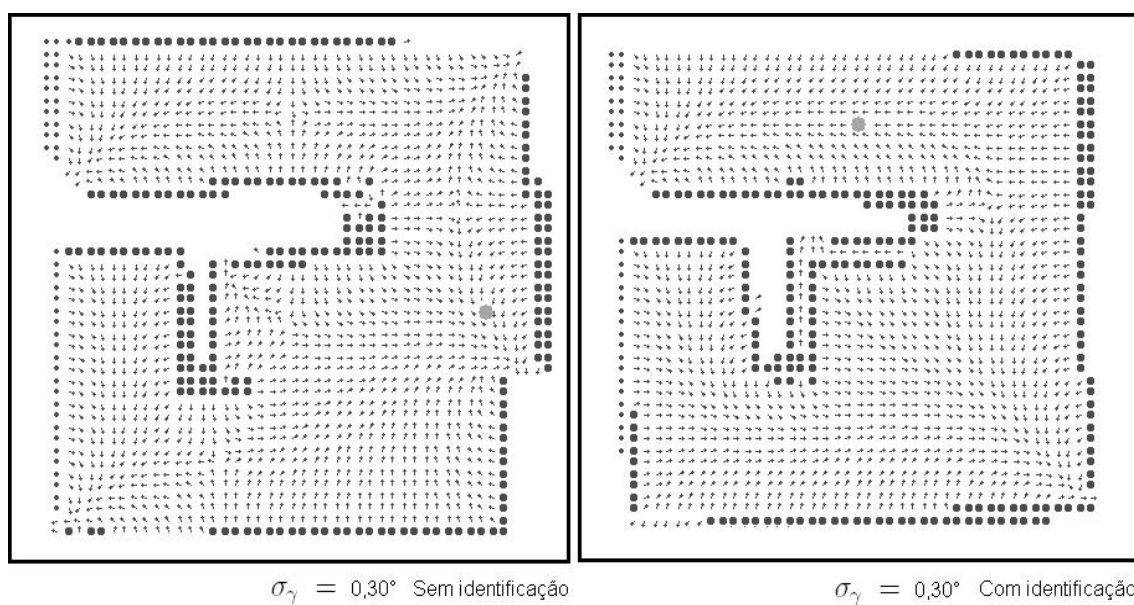


Figura 5.12: Mapa gerado com as marcas de Kalman sem e com identidade

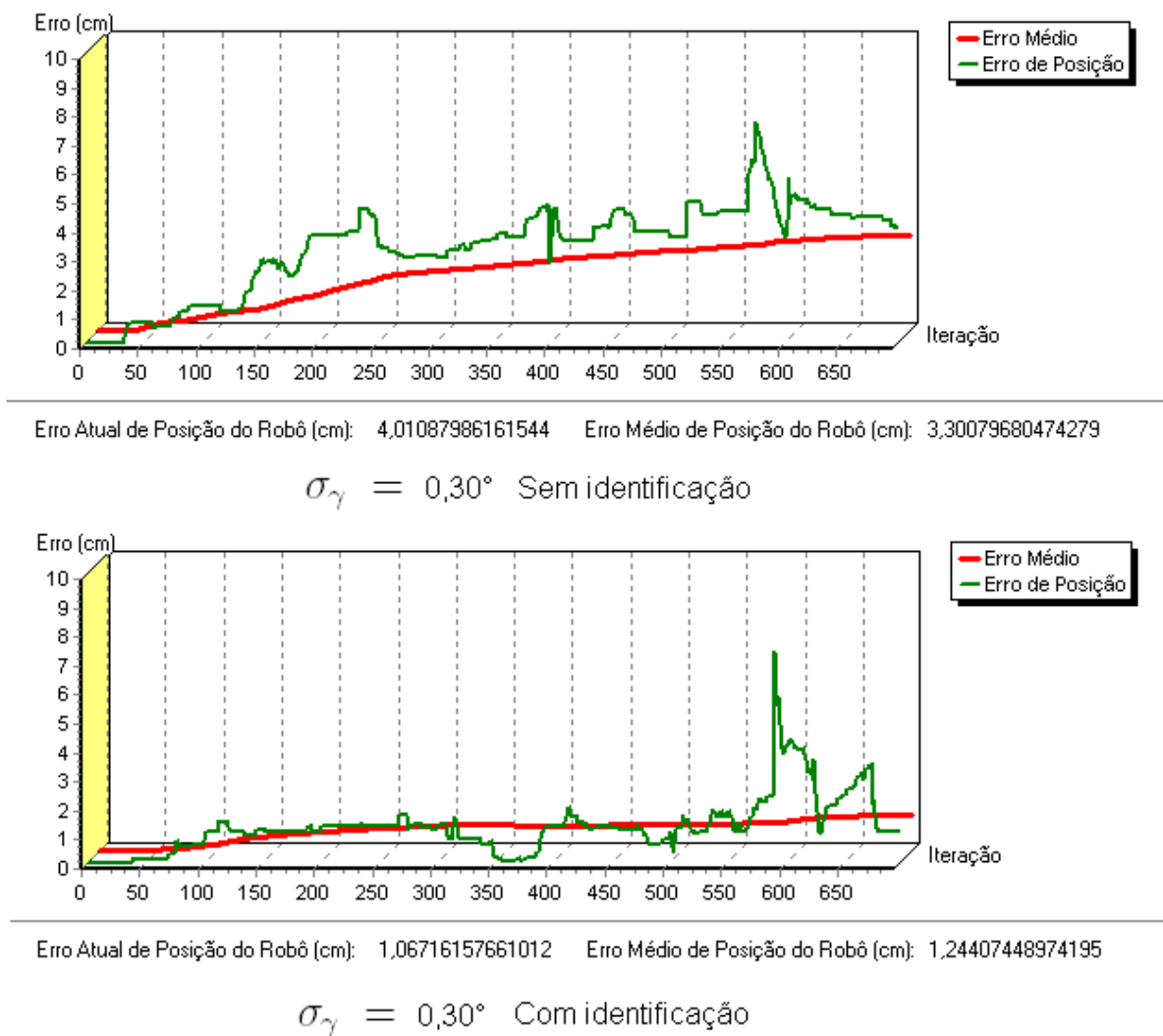


Figura 5.13: Erro e erro médio da posição do robô durante a exploração com marcas de Kalman sem e com identidade

6 CONCLUSÃO

Este trabalho apresenta uma solução para o problema de auto-localização de robôs móveis autônomos. Esta solução faz uso de um método linear de cálculo de estimativas chamado Filtro de Kalman. Do ponto de vista do Agente Exploratório definido por Edson Prestes (PRESTES, 2003), esta é uma proposta para implementar o módulo de auto-localização.

Na Seção 5.2 foi mostrado como o mapeamento do ambiente por um robô com erros de odometria pode resultar em um mapa deformado e muito diferente do mapa do ambiente real. Isto é um sério problema para um robô que pretende ter autonomia pois, além do mapa não condizente com a realidade, pode resultar na identificação errada do objetivo a ser encontrado.

A proposta leva em consideração que toda entidade que se movimenta em um ambiente conta sempre com *pontos de referência* para se localizar. Estas referências são implementadas como objetos especiais chamados de *marcas de Kalman*. O robô, ao visualizar uma marca, ajusta sua posição na janela global do ambiente. As marcas devem estar em posições estratégicas a fim de permitir que o desajuste na posição do robô não resulte em grandes deformações do mapa gerado.

A proposta de utilizar marcas no módulo de auto-localização com Filtro de Kalman surgiu para resolver dois problemas:

1. mapeamento de objetos longos: o FK identifica cada ponto do objeto longo como sendo um objeto novo. Como os ambientes percorridos pelo robô serão, na grande maioria, compostos por objetos desse tipo ocorre um crescimento quadrático da matriz de covariância, inviabilizando a implementação;
2. mapeamento de todos os objetos do ambiente: a grade de ocupação já realiza este mapeamento e, portanto, não há necessidade de mapear todos os objetos do ambiente com o FK, apenas os que são realmente relevantes para localizar o robô no mapa global, os *pontos de referência*.

Numa implementação com o robô real as marcas poderiam ser identificadas, por exemplo, por um sensor de tipo diferente. No caso do robô Nomad 200, o sonar poderia identificar os objetos enquanto a câmera identificaria as marcas de Kalman através de algum método de interpretação de imagens.

As marcas de Kalman reais poderiam ser os cantos de uma sala, uma porta ou uma mesa. Elas também poderiam estar localizadas no teto, como uma lâmpada identificando o ponto central do ambiente. Isto traz a necessidade de construir um método para reconhecer tais estruturas de acordo com o sensor que for utilizado para o reconhecimento.

Algumas considerações são importantes para a conclusão deste trabalho:

- não existe algoritmo para a Robótica Móvel Autônoma que corrija erros muito grandes;
- as marcas identificadas no início da exploração são sempre as mais confiáveis pois foram identificadas com o mínimo de desajuste na posição do robô;
- uma marca identificada quando o robô está na posição errada corrigirá o robô sempre para esta posição;
- o módulo de auto-localização deve funcionar também em ambientes com objetos móveis desde que as marcas de Kalman permaneçam fixas.

A presente proposta do módulo de auto-localização usando Filtro de Kalman tem bom desempenho por se utilizar de matrizes menores, que só reconhecem os pontos realmente relevantes do ambiente. Quanto ao mapeamento, há uma correção indireta do mapa, ou seja, o ajuste na posição do robô indiretamente ajusta a posição dos objetos. O resultado é ainda melhor com a identidade das marcas.

Sendo assim, o módulo realiza uma auto-localização híbrida pois se utiliza de técnicas de posicionamento relativo e absoluto, através da interação de diferentes níveis de mapeamento, o mapeamento das marcas de Kalman e o mapeamento da grade de ocupação, para produzir uma melhor estimativa da posição correta do robô no ambiente.

A proposta, portanto, é viável de implementação e, quando associada com o método exploratório de campos potenciais, mostrou ser eficiente para a correção da posição do robô na janela global e, conseqüentemente, para a correção do mapa construído durante a exploração.

6.1 Trabalhos Futuros

Como continuidade deste trabalho, são listados algumas sugestões de trabalhos futuros:

- elaborar uma estratégia para identificar marcas de Kalman (portas, cantos de paredes) através de câmeras (processamento gráfico) e aplicá-la em um robô real. Uma observação importante é que esta estratégia não deve identificar objetos que possam trocar de posição no ambiente durante a exploração;
- elaborar uma estratégia para limitar o número de marcas de Kalman. O aumento do número de marcas é um dos fatores críticos do módulo pois aumenta o tamanho das matrizes. Uma proposta seria atribuir às marcas um valor de importância que permitisse descartá-las à medida que marcas mais significativas surgissem no ambiente;
- efetuar testes utilizando a técnica de auto-localização proposta no simulador, interagindo com o mini-robô Khepera. Para tanto, há necessidade de adquirir um outro tipo de sensor para o robô, por exemplo a mini-câmera, e um módulo de comunicação por rádio para evitar os erros inseridos pelo cabo;
- modificação do Simulador do Nomad 200 para que implemente o módulo de auto-localização;
- testar o módulo de auto-localização no robô real Nomad 200;

- testar o módulo de auto-localização em um ambiente com objetos móveis;
- estudar o percentual de erro máximo que pode ocorrer, sem interferir no resultado final do módulo de auto-localização, ao identificar as marcas de Kalman.

REFERÊNCIAS

ARAÚJO, P. C. de. Aplicação do Filtro de Kalman na Determinação do Prazo de Validade de Alimentos Perecíveis. **Sitientibus Revista da Universidade Estadual de Feira de Santana**, [S.l.], n.23, p.45–58, jul./dez. 2000.

BAILEY, T.; NEBOT, E. Localisation in Large-Scale environments. **Robotics and Autonomous Systems**, [S.l.], v.37, p.261–281, Dec. 2001.

BETKE, M.; GURVITS, L. Mobile robot localizaton using landmarks. In: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, IROS, 1994, Munich, Germany. **Proceedings...** [S.l.: s.n.], 1994. p.135–142.

BETKE, M.; GURVITS, L. Mobile robot localization using landmarks. **IEEE Transactions on Robotics and Automation**, [S.l.], v.13, n.2, p.251–263, 1997.

BORENSTEIN, J.; KOREN, Y. Real-time obstacles avoidance for fast mobile robots. **IEEE Transactions on Systems, Man and Cybernetics**, [S.l.], v.19, n.5, p.1179–1187, Sept./Oct. 1989.

BORENSTEIN, J.; KOREN, Y. Histogramic In-Motion Mapping for Mobile Robot Obstacle Avoidance. **IEEE Journal of Robotics and Automation**, [S.l.], v.7, n.4, p.535–539, 1991.

CHATILA, R.; LACROIX, S.; BETGé-BREZETZ, S.; DEVY, M.; SIMéON, T. **Autonomous Mobile Robot Navigation for Planet Exploration the Eden Project**. [S.l.]: Laboratoire d'analyse et d'Architecture des Systèmes, 1996.

CHATILA, R.; LAUMOND, J.-P. Position referencing and consistent world modeling. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, ICRA, 1989. **Proceedings...** [S.l.: s.n.], 1989.

CHOSSET, H.; BURDICK, J. Sensor based motion exploration: the hierarchical generalized voronoi graph. **International Journal of Robotics Research**, [S.l.], v.19, n.2, p.119–148, Feb. 2000.

CONNOLLY, C.; GRUPEN, R. On the Application of Harmonic Functions to Robotics. **Journal of Robotic Systems**, [S.l.], v.10, p.931–946, 1993.

CROWLEY, J. L.; COUTAZ, J. **Navigation et modélisation pour un robot mobile**. Grenoble, France: Laboratoire d'informatique fondamentale et d'intelligence artificielle, 1985.

CSORBA, M. **Simultaneous Localisation and Map Building**. 1997. Tese (Doutorado em Ciência da Computação) — Department of Engineering Science, University of Oxford, Oxford, UK.

DAM, J. V. **Environment Modelling for Mobile Robots: neural learning for sensor fusion**. 1998. Tese (Doutorado em Ciência da Computação) — Universiteit van Amsterdam, Faculteit WINS, Amsterdam, The Netherlands.

DEVY, M.; CHATILA, R.; FILLATREAU, P.; LACROIX, S.; NASHASHIBI, F. On autonomous navigation in natural environment. **Robotics and Autonomous Systems**, [S.l.], v.16, p.5–16, 1995.

DISSANAYAKE, M. W. M. G.; NEWMAN, P.; CLARK, S.; DURRANT-WHYTE, H. F.; CSORBA, M. A Solution to the Simultaneous Localization and Map Building (SLAM) Problem. **IEEE Transactions on Robotics and Automation**, [S.l.], v.17, n.3, p.229–241, June 2001.

DUDEK, G.; FREEDMAN, P.; HADJRES, S. Using Local Information in a Non-Local Way for Mapping Graph-Like Worlds. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 1993, Los Altos, CA. **Proceedings...** [S.l.]:Morgan Kaufmann, 1993. p.1639–1647.

ELFES, A. Sonar-Based Real World Mapping and Navigation. **IEEE Journal of Robotics and Automation**, [S.l.], v.RA-3, n.3, p.249–265, 1987.

ELFES, A. Using Occupancy Grids for Mobile Robot Perception and Navigation. **Computer Magazine**, [S.l.], p.46–57, June 1989.

ENGELSON, S. **Passive map learning and visual place recognition**. 1994. Tese (Doutorado em Ciência da Computação) — Department of Computer Science, Yale University, New Haven, CT.

FEDER, H. J. S.; LEONARD, J. J.; SMITH, C. M. Adaptive mobile robot navigation and mapping. **International Journal of Robotics Research**, [S.l.], v.18, n.7, p.650–668, July 1999.

FEYNMAN, R. P.; LEIGHTON, R. B.; SANDS, M. **The Feynman Lectures on Physics**. [S.l.]: Addison-Wesley, 1972. v.2.

FOX, D.; BURGARD, W.; THRUN, S. Probabilistic Methods for Mobile Robot Mapping. In: IJCAI WORKSHOP ON ADAPTIVE SPATIAL REPRESENTATIONS OF DYNAMIC ENVIRONMENTS, 1999. **Proceedings...** [S.l.: s.n.], 1999.

GOLFARELLI, M.; MAIO, D.; RIZZI, S. Correction of Dead-Reckoning Errors in Map Building for Mobile Robots. **IEEE Transaction on Robotics and Automation**, [S.l.], v.17, n.1, p.37–47, Feb. 2001.

GUIVANT, J. E.; NEBOT, E. M. Optimization of the Simultaneous Localization and Map-Building Algorithm for Real-Time Implementation. **IEEE Transactions on Robotics and Automation**, [S.l.], v.17, n.3, p.242–257, June 2001.

HEBERT, P.; BETGÉ-BREZETS, S.; CHATILA, R. **Probabilistic map learning: necessity and difficulties**. Toulouse Cedex, France: Laboratoire d'Analyse et d'Architecture des Systèmes, 1995.

HOWARD, A.; KITCHEN, L. Generating Sonar Maps in Highly Specular Environments. In: INTERNATIONAL CONFERENCE ON CONTROL, AUTOMATION, ROBOTICS AND VISION, 4., 1996. **Proceedings...** [S.l.: s.n.], 1996.

KALMAN, R. E. A New Approach to Linear Filtering and Prediction Problems. **Transactions of the ASME—Journal of Basic Engineering**, [S.l.], v.82, n.Series D, p.35–45, 1960.

KUIPERS, B.; BYUN, Y.-T. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. **Robotics and Autonomous Systems**, [S.l.], v.8, p.47–63, 1991.

LATOMBE, J.-C. **Robot Motion Planning**. Assinippi, Norwell, Massachusetts: Kluwer Academic Publishers, 1993.

MACHADO, K. F. **Técnicas de Mapeamento de Ambientes**. 2000. 50p. Trabalho Individual (Mestrado em Ciência da Computação) — Instituto de Informática, UFRGS, Porto Alegre.

MAGUIRE, E. A.; BURGESS, N.; O'KEEFE, J. Human spatial navigation: cognitive maps, sexual dimorphism, and neural substrates. **Current Opinion in Neurobiology**, [S.l.], v.9, n.2, p.171–177, Apr. 1999.

MAJUMDER, S.; ROSENBLATT, J.; SCHEDING, S.; DURRANT-WHYTE, H. Map Building and Localization for Underwater Navigation. In: INTERNATIONAL SYMPOSIUM ON EXPERIMENTAL ROBOTICS, 2000. **Proceedings...** [S.l.: s.n.], 2000.

MATARIC, M. J. **A Model for Distributed Mobile Robot Environment Learning and Navigation**. 1990. Dissertação (Mestrado em Ciência da Computação) — MIT Artificial Intelligence Laboratory, Cambridge, MA.

MEKREZ, I. **Application de reseaux de neurones recurrences au controle d'un robot autonome**. 1997. Dissertação (Mestrado em Ciência da Computação) — Institut National Polytechnique de Grenoble, Grenoble.

ORIOLO, G.; ULIVI, G.; VANDITTELLI, M. Fuzzy Maps : a new tool for mobile robot perception and planning. **Journal of Robotic Systems**, [S.l.], v.14, n.3, p.179–197, 1997.

PETTERSSON, L. **Control System Architecture for Autonomous Agents**. [S.l.]: Department of Machine Design, KTH, 1997.

PRESTES, E. **Navegação Exploratória baseada em Problemas de Valores de Contorno**. 2003. 109p. Tese (Doutorado em Ciência da Computação) — Instituto de Informática, UFRGS, Porto Alegre.

PRESTES, E.; IDIART, M. A.; ENGEL, P. M.; TREVISAN, M. Exploration technique using potential fields calculated from relaxation methods. In: IEEE/RSJ INTERNATIONAL CONFERENCE ON INTELLIGENT ROBOTS AND SYSTEMS, IROS, 2001, Havaí, EUA. **Proceedings...** [S.l.: s.n.], 2001. p.2012–2017.

- RASCHKE, U.; BORENSTEIN, J. A comparison of Grid-type map-building techniques by index of performance. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, ICRA, 1990, Cincinnati, Ohio. **Proceedings...** [S.l.: s.n.], 1990.
- ROMERO, L.; MORALES, E.; SUCAR, E. Building Maps for Indoor Mobile Robots using Ultrasonic and Laser Range Sensors. **Computación y Sistemas**, [S.l.], Sept. 2001.
- ROY, N.; BURGARD, W.; FOX, D.; THRUN, S. Coastal Navigation - Mobile Robot Navigation with uncertainty in dynamic environments. In: IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION, ICRA, 1999, Detroit, MI. **Proceedings...** [S.l.: s.n.], 1999.
- SHATKAY, H.; KAEHLING, L. P. Learning Topological Maps with Weak Local Odometric Information. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 1997. **Proceedings...** [S.l.: s.n.], 1997. v.2, p.920-929.
- SMITH, C.; LEONARD, J. J.; BENNETT, A. A.; SHAW, C. Feature-based concurrent Mapping and Localization for autonomous underwater vehicles. In: IEEE OCEANS, 1997. **Proceedings...** [S.l.: s.n.], 1997.
- SUTTON, R.; BARTO, A. G. **Reinforcement Learning: an introduction**. Cambridge, MA: MIT Press, 1998.
- THRUN, S. Learning Maps for Indoor Mobile Robot Navigation. **Artificial Intelligence**, [S.l.], v.99, n.1, p.21-71, 1998.
- WELCH, G. **The Kalman Filter Learning Tool - Dynamic and Measurement Models**. Chapel Hill: University of North Carolina, 2001. (COMP 145).
- WELCH, G.; BISHOP, G. **An Introduction to the Kalman Filter**. Chapel Hill, NC: University of North Carolina, 2001. (TR 95-041).
- YAMAUCHI, B. A Frontier Based Exploration for Autonomous Exploration. In: IEEE INTERNATIONAL SYMPOSIUM ON COMPUTATIONAL INTELLIGENCE IN ROBOTICS AND AUTOMATION, 1997, Monterey, CA. **Proceedings...** [S.l.: s.n.], 1997. p.146-151.
- YAMAUCHI, B. Frontier-based exploration using multiple robots. In: INTERNATIONAL CONFERENCE ON AUTONOMOUS AGENTS, 2., 1998, Minneapolis. **Proceedings...** [S.l.: s.n.], 1998. p.47-53.
- ZANARDI, M. C. F. P. S.; LÓTA, V. R. D. Estimaco do Quaternio atravs do Filtro de Kalman para Comparaco com o Quaternio de Referncia. **Revista Digital da UNESP - Campus Guaratinguet**, [S.l.], v.1, n.1, p.10, set. 2001.