

VICTOR ADOLFO ROMERO CANO

*MAPEAMENTO E LOCALIZAÇÃO
SIMULTÂNEOS PARA MULTIRROBÔS
COOPERATIVOS*

São Paulo – SP

2010

VICTOR ADOLFO ROMERO CANO

***MAPEAMENTO E LOCALIZAÇÃO
SIMULTÂNEOS PARA MULTIRROBÔS
COOPERATIVOS.***

Dissertação apresentada à Escola Politécnica
da Universidade de São Paulo para obtenção
do título de Mestre em Engenharia Elétrica.

São Paulo – SP

2010

Dissertação de Mestrado sob o título “*Mapeamento e Localização Simultâneos para Multirobôs Cooperativos*”, defendida por Victor Adolfo Romero Cano e aprovada em 5 de outubro de 2010, em São Paulo, São Paulo, pela banca examinadora constituída pelos doutores:

Prof. Dr. Oswaldo Luiz do Valle Costa
Departamento de Engenharia de Telecomunicações e
Controle - POLI USP
Orientador

Profa. Dra. Anna Helena Reali Costa
Departamento de Engenharia de Computação e
Sistemas Digitais - POLI USP

Prof. Dr. Valdir Grassi Junior
Departamento de Engenharia Elétrica - Escola de
Engenharia de São Carlos USP

VICTOR ADOLFO ROMERO CANO

***MAPEAMENTO E LOCALIZAÇÃO
SIMULTÂNEOS PARA MULTIRROBÔS
COOPERATIVOS.***

Dissertação apresentada à Escola Politécnica
da Universidade de São Paulo para obtenção
do título de Mestre em Engenharia Elétrica.

Área de Concentração:
Engenharia de Sistemas

Orientador:
Prof. Dr. Oswaldo Luiz do Valle Costa

São Paulo – SP

2010

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 22 de outubro de 2010.

Assinatura do autor _____

Assinatura do orientador _____

FICHA CATALOGRÁFICA

Romero Cano, Victor Adolfo

**Mapeamento e localização simultâneos para multirrobo
cooperativos / V.A. Romero Cano. — ed.rev. — São Paulo, 2010.
66 p.**

**Dissertação (Mestrado) - Escola Politécnica da Universidade
de São Paulo. Departamento de Engenharia de Telecomunica-
ções e Controle.**

**1. Robótica (Simulação computacional) 2. Inteligência arti-
ficial 3. Probabilidade aplicada I. Universidade de São Paulo.
Escola Politécnica. Departamento de Engenharia de Telecomu-
nicações e Controle II. t.**

*Dedico esta dissertação a Deus,
fonte e motivação de tudo o que sou.*

Agradecimentos

Dedico meus sinceros agradecimentos:

- ao professor Oswaldo Luiz do Valle Costa, pela orientação e incentivo;
- à professora Anna Helena Reali Costa pela grande ajuda no desenvolvimento do projeto, e pela amizade e supervisão durante o estágio no programa de aperfeiçoamento no ensino (PAE).
- ao professor Jaime da Cruz pelas suas sugestões ao trabalho e pelas conversas que encheram de animo o espírito;
- aos meus pais Adelina e Jorge Antonio, cujo exemplo de honestidade e trabalho tem sido um norteador para a minha vida;
- aos meus irmãos Laura, Yelitza, Luis Alberto e Jairo que mesmo à distância sempre souberam me dar sua força;
- a minha namorada Johana, que tem me dado seu apoio e amor nos momentos mais difíceis;
- aos meus amigos Ana Lucia, Ana Maria, Ester, Katia, Lina, Sara, Wislenny, André, Fabio, Glauco, Jahn Pierre, John e Ruben que com sua companhia fizeram destes dois anos de mestrado um tempo inesquecível.
- aos meus colegas e amigos de sala Zoraida, Cristiani e Hector pelas conversas e trocas de idéias.
- ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo apoio a este projeto. proc. n. 136207/2008-1.

*“Por isso mesmo, dedikai todo o esforço em juntar à vossa fé a fortaleza,
à fortaleza o conhecimento, ao conhecimento o domínio próprio,
ao domínio próprio a constância, à constância a piedade,
à piedade a fraternidade, e à fraternidade, o amor.”*

II São Pedro, 1, 5-7. Bíblia Sagrada.

Resumo

Neste trabalho foi desenvolvido um estudo comparativo entre duas estratégias básicas para a combinação de mapas parciais baseados em marcos para sistemas multirobô: a estratégia por associação de marcos e a estratégia por distância relativa entre os robôs (também conhecida por *rendezvous*). O ambiente simulado corresponde a um entorno plano povoado de árvores que são mapeadas por uma equipe de dois robôs móveis equipados com sensores laser para medir a largura e localização de cada árvore (marco). Os mapas parciais são estimados usando o algoritmo FastSLAM. Além do estudo comparativo propõe-se também um algoritmo alternativo de localização e mapeamento simultâneos para multirrobôs cooperativos, utilizando as observações entre os robôs não só para o cálculo da transformação de coordenadas, mas também no desenvolvimento de um processo sequencial para atualizar o alinhamento entre os mapas, explorando de forma mais eficiente as observações entre robôs. Os experimentos realizados demonstraram que o algoritmo proposto pode conduzir a resultados significativamente melhores em termos de precisão quando comparado com a abordagem de combinação de mapas tradicional (usando distância relativa entre os robôs).

Abstract

In this text a comparative survey between the two basic strategies used to combine partial landmark based maps in multi-robot systems, data association and inter-robot observations (known as rendezvous), is presented. The simulated environment is a flat place populated by trees, which are going to be mapped by a two-mobile robot team equipped with laser range finders in order to measure every tree (landmark) location and width. Partial maps are estimated using the algorithm FastSLAM. Besides the comparative study it is also proposed an alternative algorithm for Simultaneous Localization and Mapping (SLAM) in multi-robot cooperative systems. It uses observations between robots (detections) not only for calculating the coordinate transformation but also in the development of a sequential process for updating the alignment between maps, exploiting in a more efficient way the inter-robot observations. The experiments showed that the algorithm can lead to significantly better results in terms of accuracy when compared with the traditional approach of combining maps (using the relative distance between robots).

Lista de Figuras

1	O Problema do SLAM, extraído de (WHYTE; BAILEY, 2006). O robô efetuando SLAM estima a sua própria localização e a dos marcos que representam o ambiente.	p. 22
2	Postura do robô no instante k-1	p. 25
3	Robô medindo posição e orientação de um marco	p. 28
4	Rede Bayesiana dinâmica que modela o problema do SLAM na sua forma fatorada, extraído de (THRUN; BURGARD; FOX, 2005)	p. 31
5	Primeiros instantes da exploração do robô. No instante k o robô encontra o primeiro marco	p. 32
6	Exemplo do SLAM para um só robô no caso de um ambiente com quatro marcos. Em k=1 são feitas 3 observações e em k=2 são feitas duas. . .	p. 35
7	Mapas parciais, construídos depois do SLAM efetuado por cada robô. Esquerda robô 1 e direita robô 2.	p. 38
8	Resultados do processo de alinhamento entre mapas parciais usando associações entre marcos, isto é, a transformação de coordenadas e as correspondências.	p. 39
9	Transformação de coordenadas entre os sistemas de coordenadas dos robôs a partir de uma correspondência entre os mapas.	p. 40
10	Ângulo Θ da transformação de coordenadas entre os sistemas de coordenadas dos robôs a partir de duas correspondências entre os mapas. . . .	p. 41
11	Triângulos retângulos formados pelo par de correspondências escolhido	p. 41
12	Processo de alinhamento usando detecções.	p. 43
13	Esquema do algoritmo	p. 47

14	Mapa global usando associações entre marcos (Experimento 1). As estrelas são as árvores mapeadas só pelo robô um, os círculos as árvores mapeadas só pelo robô dois e os quadros são a estimativa combinada dos marcos considerados correspondentes depois de aplicada a transformação.	p. 54
15	Mapa global usando detecções entre robôs (Experimento 1).	p. 54
16	Mapa e percursos do segundo experimento. Pouca sobreposição entre mapas parciais.	p. 54
17	Mapa global usando associações entre marcos (segundo experimento).	p. 55
18	Mapa global usando detecções entre robôs (segundo experimento).	p. 55
19	Ambiente da simulação (estrelas), primeiro (acima) e segundo (embaixo) percurso de cada robô. Na figura da esquerda, a elipse superior rodeia o ponto de encontro da primeira detecção e a inferior a do segundo.	p. 58
20	Mapa global obtido pelo robô R1 usando a primeira detecção.	p. 59
21	Mapas finais, o da direita usando a metodologia padrão e o da esquerda usando o modelo de detecção para SLAM aqui proposto.	p. 59

Lista de Tabelas

- 1 M partículas que representam M possíveis percursos do robô j , cada um deles com seu mapa associado. Cada partícula tem um peso w associado. p. 37
- 2 Comparação entre os métodos de combinação para diferentes casos de intensidade de ruído. p. 56

Lista de Símbolos

u_k - vetor de controle aplicado no instante $k-1$ para levar o robô até o estado x_k no instante k , p. 21

z_k - medida de posição relativa de um marco em relação ao robô no tempo k , p. 21

x_k - vetor que descreve a posição e orientação do robô (postura) no tempo k , p. 21

m_i - vetor que descreve a localização do i -ésimo marco, p. 22

m - mapa do ambiente ou conjunto de todos os marcos, p. 22

M - número de partículas ou amostras do estado do sistema (postura do robô e o mapa) no momento k , p. 30

$w_k^{[i]}$ - Peso da partícula i no instante k , p. 31

$x_k^{j[i]}$ - partícula i do vetor que descreve a posição e orientação (postura) do robô j no instante k , com $i=1,\dots,M$, p. 36

N - número de marcos (árvores) do mapa real, p. 36

N_j - número de marcos mapeados pelo robô j , p. 36

$d = (d_x, d_y), \Theta$ - transformação entre os sistemas de coordenadas de dois robôs que vão combinar seus mapas, p. 42

Sumário

1	Introdução	p. 14
2	Revisão da literatura	p. 17
2.1	Algoritmos de SLAM mais amplamente utilizados	p. 17
2.2	Abordagens para o problema de SLAM multirrobo	p. 18
2.2.1	Abordagem Centralizada	p. 19
2.2.2	Abordagem descentralizada	p. 20
3	Formulação do problema	p. 21
3.1	SLAM para um único robô	p. 21
3.2	Os Modelos	p. 24
3.2.1	O Modelo de Movimento	p. 25
3.2.2	O Modelo de Observação ou de Medida	p. 27
3.3	Formulação do problema para multirrobo	p. 29
4	SLAM usando um filtro de partículas reduzido	p. 30
4.1	SLAM para um robô	p. 30
4.2	Exemplo	p. 34
5	SLAM multirrobo	p. 36
5.1	Técnicas para o alinhamento de mapas	p. 37
5.1.1	Alinhamento de mapas usando associações entre marcos	p. 39
5.1.2	Alinhamento de mapas usando detecções	p. 42

5.2	Obtenção do mapa global	p. 44
6	Uma proposta alternativa para FastSLAM multirrobo	p. 46
6.1	Detalhamento do Algoritmo	p. 47
6.1.1	Cálculo da densidade de detecção $P(x_k^2 = x x_k^1 = x', d_2)$	p. 50
7	Resultados	p. 52
7.1	Resultados do estudo comparativo	p. 53
7.1.1	Experimento 1	p. 53
7.1.2	Experimento 2	p. 53
7.1.3	Experimento 3	p. 56
7.2	Resultados da avaliação do algoritmo proposto	p. 57
8	Conclusões e Propostas Futuras	p. 61
8.1	Conclusões	p. 61
8.2	Perspectiva de Trabalhos Futuros	p. 61
	Referências	p. 63

1 *Introdução*

A navegação é uma das capacidades mais exigentes de um robô móvel. O êxito na navegação depende do sucesso nos quatro blocos que a constituem: percepção (o robô interpreta seus sensores e extrai informação útil), localização (o robô determina sua posição no ambiente, isto é, o robô se pergunta: onde estou?), cognição (o robô tem que decidir como agir para conseguir seus objetivos); e controle de movimento (o robô regula as saídas de seus atuadores para seguir a trajetória desejada).

Durante a localização o robô faz a estimação da sua posição com base em medidas de sensores que têm ruídos e modelos de movimento que são imprecisos devido a situações não modeladas, como é o caso de deslizamento das rodas. Essa localização é feita em relação a uma representação do ambiente ou mapa, o qual pode ser providenciado ao robô, ou pode ser construído em tempo real. O problema de o robô criar o mapa e simultaneamente localizar-se nele é chamado de SLAM (*Simultaneous Localization and Mapping*).

Os mecanismos de localização mais usados atualmente são a odometria, sensores especializados como o GPS, e outros como a câmera zenital. Na integração da informação gerada por esses mecanismos, usam-se métodos probabilísticos como os processos de decisão de Markov, o filtro de Kalman e filtros de partículas (também conhecidos como métodos sequenciais de Monte Carlo). As abordagens básicas para o problema de mapeamento e localização simultâneos são o EKF-SLAM (*Extended Kalman Filter - Simultaneous Localization and Mapping*) e FastSLAM (WHYTE; BAILEY, 2006).

Em trabalhos recentes tem sido tratado o caso em que multirrobôs são utilizados ((FOX et al., 2006), (WU et al., 2009) e (SEOK; LEE, 2009)). Algumas razões para a utilização deste tipo de sistemas são as seguintes:

- É mais barato fazer uma equipe de robôs simples que um só robô complexo.
- A capacidade de troca de informação entre robôs diminui a incerteza no processo de estimação.

- Diminuição do tempo de realização das tarefas.

Os sistemas cooperativos possuem boas características com relação à velocidade na execução de tarefas de exploração e na diminuição da incerteza no processo de estimação. Portanto, são atraentes para a realização de tarefas de resgate em catástrofes, naturais ou não, e mapeamento de terreno. Por esse motivo, tais sistemas têm despertado na comunidade um recente interesse no problema do SLAM para sistemas multirrobô. Tornase, desse modo, importante desenvolver trabalhos que analisem cada fase deste processo, especialmente uma fase tão importante como é a combinação dos mapas gerados pelo sistema multirrobô.

O desafio consiste, então, em encontrar estratégias para combinar eficientemente a informação coletada pelos sensores dos diferentes robôs. O SLAM multirrobô pode ser feito de duas formas diferentes: usando um só mapa (FENWICK; NEWMAN; LEONARD, 2002), que é atualizado por todos os membros do time, ou cada robô constrói um mapa parcial do ambiente e em certo momento esses mapas são combinados em um só.

No caso desta pesquisa foi utilizada a segunda metodologia, cujo processo de combinação pode ainda ser dividido em duas fases:

- a primeira é chamada de alinhamento, na qual se calcula a transformação de coordenadas que permitirá colocar a postura do robô dois e seus respectivos marcos no sistema de coordenadas do robô um;
- na segunda, uma vez que o mapa do robô dois tenha sido alinhado com o mapa do robô um, as estimativas dos marcos que se correspondem precisam ser integradas com a finalidade de gerar um mapa global.

Nesse sentido, o principal objetivo do presente trabalho é comparar algumas metodologias usadas na combinação de mapas baseados em marcos para sistemas multirrobô e propor um algoritmo que explore de uma forma eficiente as observações entre robôs (também chamadas detecções) no processo de combinação de mapas. O algoritmo proposto é relevante dado que utiliza a crença que um robô tem sobre a postura do outro com o fim de diminuir a incerteza no processo de combinação de mapas, fato que até o momento é desconsiderado na solução do SLAM para multirrobôs.

Para fins de simulação, foi criada uma interface gráfica que permite ao usuário definir o mapa real e os percursos dos robôs. Com essa informação, um simulador do algoritmo FastSLAM2.0 fornecido pelos professores Tim Bailey e Juan Nieto (BAILEY; NIETO, 2006)

foi usado para realizar SLAM para cada robô, obtendo deste jeito os mapas parciais do ambiente. Posteriormente, duas estratégias foram usadas com o objetivo de combinar os mapas parciais estimados pelos robôs. É importante salientar que foi introduzido erro para refletir a incerteza tanto na estimação dos mapas parciais quanto na estimação do mapa global. Foi abordada a combinação dos mapas de dois robôs, entretanto, todas as idéias aqui apresentadas podem ser facilmente generalizadas para mais de dois robôs.

Esse trabalho está organizado da seguinte forma: no capítulo 2 é feita uma revisão da literatura; no capítulo seguinte, formula-se o problema de SLAM tanto para o caso de um único robô quanto para o caso multirrobô. No capítulo 4, o filtro de partículas aplicado em SLAM é apresentado.

Os principais resultados da dissertação estão nos capítulos 5, 6 e 7. No capítulo 5 são descritas as características do SLAM multirrobô, no capítulo 6 uma proposta alternativa na abordagem desse problema é apresentada, no capítulo 7 apresentam-se os resultados das simulações e, por fim, no capítulo 8, são apresentadas as conclusões e propostas futuras.

2 *Revisão da literatura*

A seguir, os principais temas abordados neste trabalho são sucintamente descritos, iniciando com um apanhado da literatura na área sobre os algoritmos de SLAM, seguido de uma descrição das metodologias de combinação de mapas mais comumente utilizadas em SLAM multirrobô.

2.1 Algoritmos de SLAM mais amplamente utilizados

O problema do SLAM começou a ser abordado a partir de (SMITH; CHEESEMAN, 1986) que apresenta um método para estimar as relações entre sistemas de coordenadas mantendo uma estimativa do erro que se propaga ao fazer as transformações de coordenadas, fornecendo dessa forma, uma modelagem explícita da incerteza no processo de estimação. Em (SMITH; SELF; CHEESEMAN, 1990) apresenta-se pela primeira vez o EKF como solução ao problema do SLAM, tornando-o uma abordagem clássica na solução deste problema. Muitos trabalhos usando EKF-SLAM foram desenvolvidos, entre os quais podem ser mencionados (GUIVANT; NEBOT, 2001) e (DISSANAYAKE; NEWMAN; CLARK, 2002).

Como uma alternativa às hipóteses na abordagem do EKF (distribuições gaussianas e modelos lineares) e suas limitações quanto a sua complexidade quadrática e a sensibilidade às falhas na associação de dados, surge o algoritmo FastSLAM apresentado em (MONTEMERLO et al., 2002) e (MONTEMERLO et al., 2003), que é o primeiro a considerar modelos não-lineares e distribuições não-gaussianas na solução do problema do SLAM, com a vantagem de ter uma menor complexidade e maior robustez na associação de dados. Trabalhos usando esta abordagem e versões melhoradas da mesma encontram-se em (MONTEMERLO; THRUN, 2003), (KIM; SAKTHIVEL; CHUNG, 2008) e (ZHANG; MENG; CHEN, 2009).

Tanto o EKF quanto o FastSLAM foram generalizados para o problema multirrobô.

Porém, além desses, outros algoritmos foram usados para a solução do problema de SLAM em sistemas multirrobô como é o caso dos trabalhos apresentados a seguir:

- (THRUN; LIU, 2003) soluciona o problema de SLAM multirrobô usando *Sparse Extended Information filters*, nos quais os mapas e as posturas dos robôs são representados através de campos aleatórios de Markov gaussianos. O algoritmo foi testado para o caso de oito robôs com dados obtidos do *Victoria Park* em Sydney, um *dataset* usado amplamente na comunidade do SLAM para avaliar os algoritmos;
- (MARTINEZ-CANTIN; CASTELLANOS; FREITAS, 2007) generaliza o filtro de partículas marginal para o caso multirrobô. Eles apresentam esta abordagem como uma nova direção no projeto de métodos baseados em partículas para realização de SLAM;
- (FOX et al., 2006) apresenta um sistema que permite que uma equipe de robôs explore ambientes de forma eficiente usando *Graph-SLAM*, um algoritmo baseado em otimização na solução do problema do SLAM ¹.
- (ANDERSON; NYGARDS, 2008) introduz o uso da técnica *Square Root Information Smoothing* (SRIS) para combinar mapas criados por diferentes robôs e obter a trajetória seguida por cada um deles.
- (ANDERSON; NYGARDS, 2009) propõe uma solução ao problema de alinhar e combinar mapas construídos por vários robôs usando observações feitas entre eles durante seus percursos. Essa mesma solução é aplicada em *collaborative smoothing and mapping* (C-SAM), que é uma abordagem de *smoothing* usada para combinar mapas criados por diferentes robôs independentemente ou em equipes. A principal contribuição desse trabalho é um algoritmo para solucionar o problema de associação de dados e eliminar falsas observações quando o alinhamento dos mapas é feito durante um encontro (*rendezvous*). Este trabalho é similar ao desenvolvido em (ZHOU; ROUMELIOTIS, 2006), porém, ao invés de usar o EKF (filtragem) para solucionar o problema de SLAM, usa-se o SRIS (*smoothing*).

2.2 Abordagens para o problema de SLAM multirrobô

O problema do SLAM multirrobô apresenta as seguintes características gerais:

¹ver (THRUN; BURGARD; FOX, 2005) para uma explicação detalhada do uso do algoritmo *Graph-SLAM* na solução do problema do SLAM

- o número de mapas usados no processo;
- o algoritmo de SLAM para um único robô a partir do qual o algoritmo multirrobô é generalizado;
- a necessidade de os robôs conhecerem suas posições iniciais;
- a metodologia usada para definir características comuns entre mapas.

A solução dada para cada característica permite analisar e classificar as diferentes abordagens do SLAM multirrobô. Quando um único mapa é usado, a abordagem chama-se centralizada. No caso em que cada robô constrói seu próprio mapa local e em determinado momento é realizado um processo de combinação, a abordagem chama-se de descentralizada. Uma revisão da literatura sobre ambos os casos é apresentada nas seções 2.2.1 e 2.2.2.

2.2.1 Abordagem Centralizada

(FENWICK; NEWMAN; LEONARD, 2002) e (TONG et al., 2008) apresentam algoritmos de SLAM que combinam informação sensorial e de navegação vindo de uma equipe de dois robôs, cujas posições iniciais são conhecidas, usando um único EKF para solucionar o problema do SLAM. Essa solução estima o estado conjunto, que, corresponde a um vetor de estado contendo as estimativas das posturas de todos os robôs da equipe e das posições dos marcos. Para tanto, um único mapa baseado em marcos é criado e atualizado. No primeiro, um teorema de convergência para SLAM multirrobô é provado pela primeira vez com o objetivo de quantificar os ganhos de desempenho da colaboração, permitindo, assim, a determinação do número de robôs cooperativos necessários para efetuar uma tarefa. No outro, é feita uma proposta de abordagem de definição dos pontos de destino com base nas características de incerteza do mapa criado.

Já no trabalho de (HOWARD, 2006) é apresentada uma solução em que um único mapa baseado em marcos é atualizado por uma equipe de quatro robôs, usando uma generalização do filtro de partículas proposto por (HAHNEL et al., 2003) para um único robô. (HOWARD, 2006) aborda o problema do SLAM multirrobô, primeiramente, assumindo que a posição inicial dos robôs é conhecida, e em seguida, apresenta uma solução para o problema quando as posições iniciais não são conhecidas. Nessa abordagem, as seqüências de ações/observações (previamente armazenadas) de cada robô são combinadas em um único mapa do ambiente quando ocorrem encontros entre eles. Finalmente, (HOWARD, 2006)

usa o conceito de robôs virtuais voltando no tempo com o objetivo de integrar, no mapa, os dados obtidos por cada robô antes do primeiro encontro. Esse algoritmo foi testado em um ambiente interior usando robôs homogêneos e *occupancy grids* como representação do ambiente. Os robôs foram equipados com odômetros e *laser range finders* para fazer as observações.

Por outro lado, (NIETO; GUIVANT; NEBOT, 2003) aborda o problema multirrobô usando FastSLAM, em que um robô é localizado usando localização de Monte Carlo no mapa construído por outro robô usando FastSLAM.

2.2.2 Abordagem descentralizada

Outro caso de SLAM multirrobô é aquele no qual cada robô inicia seu percurso de uma posição desconhecida criando um mapa local próprio. Em certo momento da navegação, os robôs efetuam um processo de alinhamento e combinação dos mapas criados. Nesse caso, a transformação de coordenadas necessária para efetuar tal combinação pode ser obtida a partir de duas fontes de informação. Primeiro, as associações entre os marcos mapeados pelos robôs, como em (DEDEOGLU; SUKHATME, 2000), (THRUN; LIU, 2003), (MARTINEZ-CANTIN; CASTELLANOS; FREITAS, 2007) e (SEOK; LEE, 2009), e segundo, as observações entre robôs, como em (ZHOU; ROUMELIOTIS, 2006), (HOWARD, 2006), (ANDERSON; NYGARDS, 2009), (WU et al., 2009) e (OZKUCUR; AKIN, 2010). Um estudo comparativo destas duas metodologias quando os mapas locais são criados usando o algoritmo FastSLAM (ver (MONTEMERLO; THRUN, 2010) para uma explicação detalhada do algoritmo) é apresentado em (ROMERO; COSTA, 2010).

A estimação das correspondências entre marcos, na combinação usando associações, pode ser realizada por meio de várias técnicas como: *RANSAC* (*Random Sample consensus*), *SVD* (*Singular Value Decomposition*), *ICP* (*Iterated Closed Point*) e *ImpICP* (*Improved ICP*). (BALLESTA et al., 2008) apresenta um estudo comparativo avaliando o desempenho destas técnicas concluindo que os melhores resultados são os obtidos com RANSAC e ImpICP. RANSAC caracterizou-se por valores de erro menores e ImpICP por uma melhor estabilidade, isto é, menor número de falhas no cálculo da transformação de coordenadas. Nos trabalhos em que são usadas as observações entre robôs para fazer o alinhamento, o critério mais usado para definir se dois marcos são correspondentes é a distância de *Mahalanobis* no caso de abordagens EKF e a distância euclidiana em abordagens FastSLAM.

3 *Formulação do problema*

Neste capítulo são descritos os princípios, características e terminologias relacionados ao problema do SLAM. Na seção 3.1, o problema do SLAM com somente um robô é apresentado. Em seguida, na seção 3.2 são apresentados os modelos de movimento e de medida. Finalmente, na seção 3.3 formula-se o problema para multirrobôs.

3.1 SLAM para um único robô

Formula-se o problema do SLAM para um só robô e depois se generaliza para o problema com multirrobôs. A seguir serão definidos alguns conceitos usados nesta dissertação:

- Mapa: representação espacial, composta por um conjunto de marcos, que um dado robô possui sobre o ambiente que percorre.
- Marcos: diferentes características estáticas do ambiente que podem ser reconhecidas pela percepção do robô. Cada marco representa um objeto no ambiente.
- Observação: medida da posição relativa de um marco em relação ao robô.
- Postura: posição e orientação do robô.

Considere-se um robô móvel, deslocando-se por um ambiente e efetuando observações relativas de um número (N) desconhecido de marcos, usando um sensor localizado no robô (ver figura 1). No instante k , definem-se as seguintes variáveis:

Conhecidas pelo robô:

u_k : vetor de controle aplicado no instante $k-1$ para levar o robô até o estado x_k no instante k ;

z_k : medida de posição relativa de um marco em relação ao robô no tempo k .

Estimadas pelo robô:

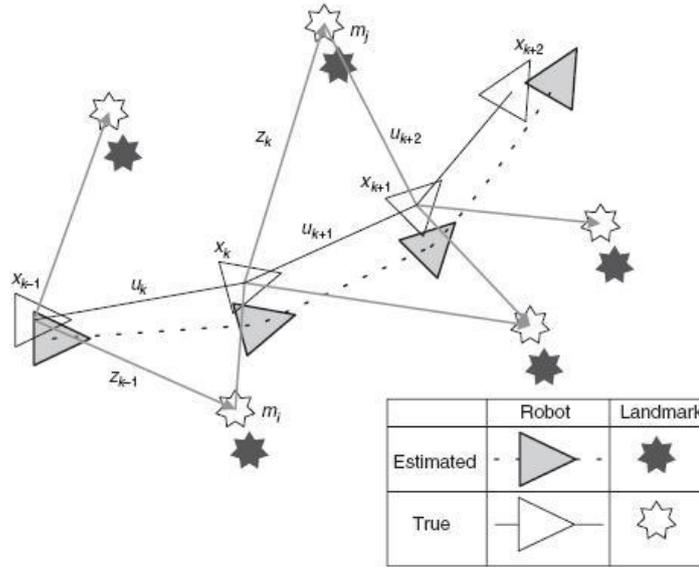


Figura 1: O Problema do SLAM, extraído de (WHYTE; BAILEY, 2006). O robô efetuando SLAM estima a sua própria localização e a dos marcos que representam o ambiente.

x_k : vetor que descreve a posição e orientação do robô (postura);

m_i : vetor que descreve a localização do i -ésimo marco;

Também serão definidos os seguintes conjuntos:

$\mathbf{x}_{0:k} = \{\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_k\}$: histórico de localizações do robô;

$\mathbf{u}_{1:k} = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k\}$: histórico das entradas de controle ¹;

$\mathbf{m} = \{\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_N\}$: mapa ou conjunto de todos os marcos;

$\mathbf{z}_{1:k} = \{\mathbf{z}_1, \mathbf{z}_2, \dots, \mathbf{z}_k\}$: conjunto de todas as observações;

Na abordagem probabilística do SLAM, requer-se calcular a probabilidade

$$P(x_k, m | z_{1:k}, u_{1:k}) \quad (3.1)$$

para cada instante de tempo k . Ela descreve a probabilidade conjunta da localização dos marcos e da postura do robô, dadas as observações coletadas e as entradas de controle até o momento k .

Uma nova abordagem em que o problema é modelado como um processo de estados discretos regido por uma relaxação da condição de Markov é apresentada em (SELVATICI;

¹o termo 'entradas de controle' é genérico e corresponde à informação sobre o movimento do robô gerada usando sensores internos (odômetros, sensores inerciais) ou observando os comandos de controle executados pelo robô

COSTA, 2006). Nesta pesquisa segue-se a modelagem tradicional na qual a navegação do robô e o sensoriamento são modelados através dos parâmetros de um Modelo Oculto de Markov.

A postura desconhecida do robô x_k é modelada como uma cadeia de Markov com probabilidade de transição de estado

$$P(x_k | x_{k-1}, u_k), \quad (3.2)$$

probabilidade que também é chamada de modelo de movimento já que descreve como os controles definem o movimento do robô e, além disso, como o ruído nos controles aumenta a incerteza na estimação da sua postura.

As observações z_k são também governadas por uma função probabilística chamada de modelo de observação que descreve o sensor externo do robô, suas características de ruído e a relação da observação atual com o marco observado e a postura do robô. Escreve-se o modelo de observação como:

$$P(z_k | x_k, m). \quad (3.3)$$

A probabilidade da equação (3.1) é então calculada recursivamente. Essa regra de atualização é conhecida como o filtro de Bayes para SLAM e sua derivação é apresentada a seguir:

Reescreve-se a probabilidade (3.1) usando a regra de Bayes:

$$P(x_k, m | z_{1:k}, u_{1:k}) = \frac{P(z_k | x_k, m, z_{1:k-1}, u_{1:k}) P(x_k, m | z_{1:k-1}, u_{1:k})}{P(z_k | z_{1:k-1}, u_{1:k})}.$$

Dado que se está calculando uma probabilidade, o denominador do cálculo anterior pode ser considerado como uma variável de normalização, denotada por η . Logo,

$$P(x_k, m | z_{1:k}, u_{1:k}) = \eta P(z_k | x_k, m, z_{1:k-1}, u_{1:k}) P(x_k, m | z_{1:k-1}, u_{1:k}).$$

Aplicando a condição de Markov, tem-se que a observação só depende de x_k e m :

$$P(x_k, m | z_{1:k}, u_{1:k}) = \eta P(z_k | x_k, m) P(x_k, m | z_{1:k-1}, u_{1:k}).$$

Usando o teorema de probabilidade total:

$$P(x_k, m | z_{1:k}, u_{1:k}) = \eta P(z_k | x_k, m) \int P(x_k, m | x_{k-1}, z_{1:k-1}, u_{1:k}) P(x_{k-1} | z_{1:k-1}, u_{1:k}) dx_{k-1}.$$

Expandindo-se o primeiro termo da integral usando a definição de probabilidade condicional, tem-se que ² :

$$P(\bullet) = \eta P(z_k | x_k, m) \int P(x_k | m, x_{k-1}, z_{1:k-1}, u_{1:k}) P(m | x_{k-1}, z_{1:k-1}, u_{1:k}) P(x_{k-1} | z_{1:k-1}, u_{1:k}) dx_{k-1}.$$

Aplicando a condição de Markov no primeiro termo da integral, tem-se que

$$P(\bullet) = \eta P(z_k | x_k, m) \int P(x_k | x_{k-1}, u_k) P(m | x_{k-1}, z_{1:k-1}, u_{1:k}) P(x_{k-1} | z_{1:k-1}, u_{1:k}) dx_{k-1}.$$

Combinando os dois últimos termos da integral (teorema da probabilidade composta), tem-se que:

$$P(x_k, m | z_{1:k}, u_{1:k}) = \eta P(z_k | x_k, m) \int P(x_k | x_{k-1}, u_k) P(m, x_{k-1} | z_{1:k-1}, u_{1:k}) dx_{k-1}.$$

Dado que u_k não oferece informação sobre m nem sobre x_{k-1} , o filtro de Bayes que atualiza a probabilidade posterior do SLAM fica da seguinte forma:

$$P(x_k, m | z_{1:k}, u_{1:k}) = \eta P(z_k | x_k, m) \int P(x_k | x_{k-1}, u_k) P(m, x_{k-1} | z_{1:k-1}, u_{1:k-1}) dx_{k-1}. \quad (3.4)$$

A equação(3.4) proporciona um método recursivo para calcular a probabilidade posterior sobre o estado x_k do robô e o mapa m no instante k com base em todas as observações $z_{1:k}$ e todas as entradas de controle $u_{1:k}$ até o instante k . A recursão é uma função do modelo de movimento $P(x_k | x_{k-1}, u_k)$ e o modelo de observação $P(z_k | x_k, m)$ definidos previamente.

É importante salientar que o problema de mapeamento pode ser formulado como o cálculo da densidade condicional $P(m | x_{0:k}, z_{1:k}, u_{1:k})$. Neste caso assume-se que a localização do robô x_k é conhecida (ou pelo menos determinística) em cada instante de tempo. Um mapa é então construído juntando as observações de diferentes localizações. Inversamente, o problema de localização pode ser formulado como o cálculo da probabilidade $P(x_k | z_{1:k}, u_{1:k}, m)$. Aqui assume-se que a localização dos marcos é conhecida com certeza, e o objetivo é calcular uma estimativa da localização do robô com relação a esses marcos.

3.2 Os Modelos

Apresenta-se agora a dedução dos modelos de movimento (3.2) e de observação (3.3).

²• representará o termo entre parêntesis da 1ª igualdade

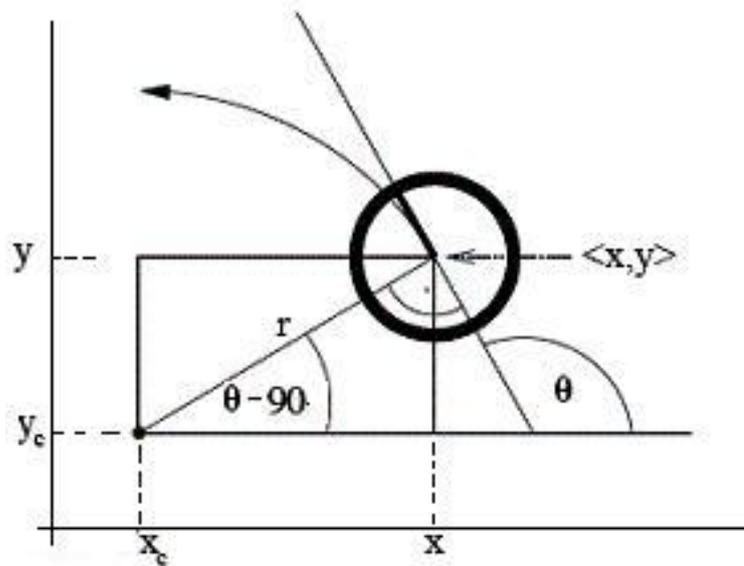


Figura 2: Postura do robô no instante k-1

3.2.1 O Modelo de Movimento

A abordagem probabilística do problema do SLAM leva em conta o fato de que os sinais de controle e as medidas de realimentação de todo o sistema estarem sujeitas a incertezas e, por isso, enxerga-se a postura do robô como uma probabilidade e não como um dado determinístico.

O desenvolvimento do modelo cinemático aqui apresentado é baseado no capítulo 5 do livro *Probabilistic Robotics* de Sebastian Thrun, Wolfram Burgard e Dieter Fox (THRUN; BURGARD; FOX, 2005). Este modelo é geral e pode ser usado tanto para robôs diferenciais quanto para robôs em configuração Ackerman e sincro-drive.

Seja $x_{k-1} = (x y \theta)^T$ a postura inicial do robô, e $u_k = (v w)^T$ a velocidade linear e angular comandadas durante o intervalo de tempo Δt no qual supõe-se que o robô vai de x_{k-1} a $x_k = (x' y' \theta')^T$ através de uma trajetória circular (devido à suposição de velocidades v e w constantes durante Δt). Ver figura 2.

No momento $k - 1$ o centro da circunferência que o robô descreve instantaneamente encontra-se em:

$$\begin{aligned} x_c &= x - r \cos(\theta - 90^\circ) \\ &= x - \frac{v}{w} \cos(-(90^\circ - \theta)) \\ &= x - \frac{v}{w} \cos(90^\circ - \theta). \end{aligned}$$

Logo

$$\boxed{x_c = x - \frac{v}{w} \sin(\theta)} \quad (3.5)$$

e

$$\begin{aligned} y_c &= y - r \sin(\theta - 90^\circ) \\ &= y - \frac{v}{w} \sin(-(90^\circ - \theta)) \\ &= y + \frac{v}{w} \sin(90^\circ - \theta). \end{aligned}$$

Logo

$$\boxed{y_c = y - \frac{v}{w} \cos(\theta)}. \quad (3.6)$$

Depois de um intervalo de tempo Δt , sendo $\Delta\theta = w\Delta t$ a mudança na orientação, o robô terá a postura $x_k = (x' y' \theta')^T$, obtida mediante a aplicação das identidades para ângulos complementares e ângulos opostos:

$$\theta' = \theta + \Delta\theta$$

$$\begin{aligned} x' &= x_c + r \cos(\theta + \Delta\theta - 90^\circ) = x_c + \frac{v}{w} \cos(90^\circ - (\theta + \Delta\theta)) = x_c + \frac{v}{w} \sin(\theta + \Delta\theta) \\ y' &= y_c + r \sin(\theta + \Delta\theta - 90^\circ) = y_c - \frac{v}{w} \sin(90^\circ - (\theta + \Delta\theta)) = y_c + \frac{v}{w} \cos(\theta + \Delta\theta) \end{aligned}$$

Assim obtém-se o modelo cinemático do robô, dado por:

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} -\frac{v}{w} \sin(\theta) + \frac{v}{w} \sin(\theta + w\Delta t) \\ -\frac{v}{w} \cos(\theta) + \frac{v}{w} \cos(\theta + w\Delta t) \\ w\Delta t \end{bmatrix}. \quad (3.7)$$

As velocidades comandadas estão sujeitas a ruído, e portanto serão representadas como:

$$\begin{bmatrix} \hat{v} \\ \hat{w} \end{bmatrix} = \begin{bmatrix} v \\ w \end{bmatrix} + \begin{bmatrix} \epsilon_{\alpha_1 v^2 + \alpha_2 w^2} \\ \epsilon_{\alpha_3 v^2 + \alpha_4 w^2} \end{bmatrix} \quad (3.8)$$

onde ϵ_{b^2} corresponde a uma variável aleatória de erro aditivo (ruído) com distribuição de probabilidade normal de média zero e variância b^2 . No modelo a variância do erro é proporcional às velocidades comandadas e os parâmetros $\alpha_1 - \alpha_4$ são parâmetros de erro específicos para cada robô. Eles modelam a precisão de atuação do robô, permitindo obter um desvio padrão b que depende das velocidades comandadas. Quanto menor for

a precisão, maiores serão estes parâmetros. A distribuição normal com média zero e variância b^2 é dada pela seguinte função de densidade:

$$\epsilon_{b^2}(a) = \frac{1}{\sqrt{2\pi b^2}} e^{-\frac{1}{2} \frac{a^2}{b^2}}.$$

Até o momento tem-se suposto que o robô descreve durante o intervalo Δt uma trajetória perfeitamente circular de raio $r = \frac{\hat{v}}{\hat{w}}$, o que não acontece na realidade, devido à incerteza própria da atuação e a variação de (x_c, y_c) e de r . Portanto, para inserir este efeito no modelo, modifica-se a orientação no instante k por:

$$\theta' = \theta + \hat{w}\Delta t + \hat{\gamma}\Delta t \quad (3.9)$$

onde

$$\hat{\gamma} = \epsilon_{\alpha_5 v^2 + \alpha_6 w^2}. \quad (3.10)$$

com $\alpha_5 - \alpha_6$ parâmetros específicos que determinam a variância do ruído rotacional. Desta forma chega-se ao modelo cinemático para cada robô levando em conta as incertezas próprias do sistema:

$$\begin{bmatrix} x' \\ y' \\ \theta' \end{bmatrix} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} + \begin{bmatrix} -\frac{\hat{v}}{\hat{w}} \sin(\theta) + \frac{\hat{v}}{\hat{w}} \sin(\theta + \hat{w}\Delta t) \\ -\frac{\hat{v}}{\hat{w}} \cos(\theta) + \frac{\hat{v}}{\hat{w}} \cos(\theta + \hat{w}\Delta t) \\ \hat{w}\Delta t + \hat{\gamma}\Delta t \end{bmatrix}. \quad (3.11)$$

Esse modelo permitirá gerar amostras aleatórias da postura do robô dada uma certa ação de controle e uma posição inicial.

3.2.2 O Modelo de Observação ou de Medida

O sensor com o qual serão efetuadas as observações fornece uma medida da posição e orientação do marco j em relação ao robô. O sensor também possui um extrator de características que consegue identificar cada marco detectado no ambiente com uma assinatura s . O modelo é formulado para o caso em que a i -ésima característica (distância, orientação e assinatura) corresponde ao j -ésimo marco no mapa (ver figura 3).

A função (modelo de medida) que relaciona a observação com a postura do robô e a posição do marco observado fica da seguinte forma:

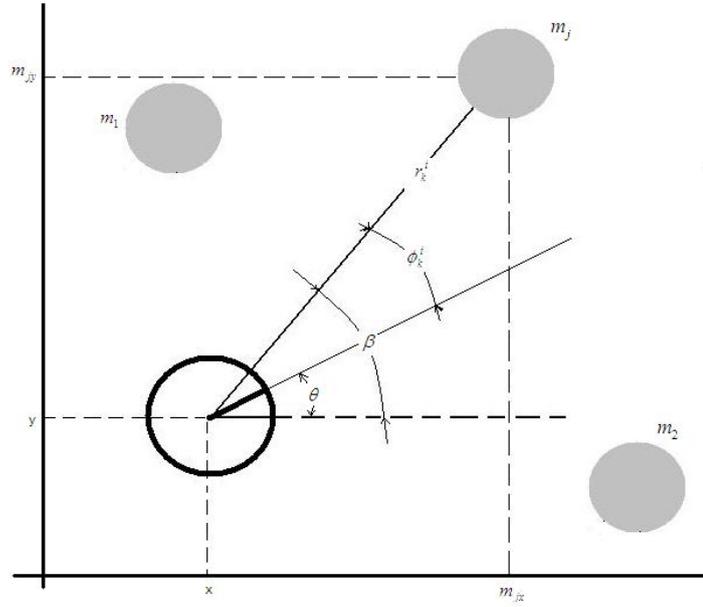


Figura 3: Robô medindo posição e orientação de um marco

$$\begin{bmatrix} r_k^i \\ \phi_k^i \\ s_k^i \end{bmatrix} = \begin{bmatrix} \sqrt{(m_{j,x} - x)^2 + (m_{j,y} - y)^2} \\ \text{atan2}(m_{j,y} - y, m_{j,x} - x) - \theta \\ s_j \end{bmatrix} + \begin{bmatrix} \epsilon_{\sigma_r^2} \\ \epsilon_{\sigma_\phi^2} \\ \epsilon_{\sigma_s^2} \end{bmatrix}. \quad (3.12)$$

No instante k o robô pode medir certo número de características e deve associar cada uma delas a um dos marcos do mapa estimado até o momento. No caso em que a característica for identificada como nova, um novo marco é inserido no mapa. Portanto as variáveis r_k^i e ϕ_k^i (característica i) representam a distância e orientação do marco j em relação ao robô, respectivamente, dado o fato que a característica i corresponde ao marco j do mapa. A assinatura para a característica i é s_k^i . Além disso $x_k = (x \ y \ \theta)'$ é a postura do robô no momento k , $(m_{j,x}, m_{j,y})$ são as coordenadas do marco j e $\epsilon_{\sigma_r^2}$, $\epsilon_{\sigma_\phi^2}$ e $\epsilon_{\sigma_s^2}$ são variáveis de erro gaussiano com desvio padrão σ_r^2 , σ_ϕ^2 e σ_s^2 respectivamente.

No modelo, atan2 é a função arco tangente de dois argumentos, que generaliza o arco tangente de y/x ao R^2 :

$$\text{atan2}(y, x) = \begin{cases} \text{atan}(y/x) & \text{se } x > 0 \\ \text{sign}(y)(\pi - \text{atan}(|y/x|)) & \text{se } x < 0 \\ 0 & \text{se } x = y = 0 \\ \text{sign}(y)\pi/2 & \text{se } x = 0, y \neq 0 \end{cases} \quad (3.13)$$

3.3 Formulação do problema para multirrobo

Nesta seção considera-se uma equipe de L robôs equipados com sensores que lhes permitem obter informação do ambiente de acordo com o modelo de medida (3.12). Os robôs deslocam-se em um ambiente povoado por certo número N de marcos estacionários cujas coordenadas são denotadas como: $m = (m_1 \dots m_N)$. A postura de cada robô é uma função do tempo e será denotada por x_k^j ; aqui k é o índice do tempo, e j ($j \in \{1, \dots, L\}$) é o índice do robô. Denotam-se as posturas de todos os robôs no tempo k como $X_k = (x_k^1 \dots x_k^L)$.

A cada instante de tempo, cada robô executa um comando de movimento ou ação de controle denotada por u_k^j , que subsequente altera sua postura. Essa transição é governada pelo modelo de movimento (3.11).

Cada robô pode também obter informação da posição dos marcos perto dele em relação à sua própria posição. O vetor da medida i efetuada pelo robô j no momento k será denotado por $z_k^{i,j}$. As observações são governadas pelo modelo de medida (3.12). Além dos marcos, os robôs também podem medir a posição entre eles e outros membros do time. A observação feita pelo robô j sobre a postura do robô i em relação a si próprio será denotada $^j d_i$ e a relação que a descreve será apresentada na seção quando a combinação de mapas usando observações entre robôs for abordada. Estas observações entre robôs também serão chamadas como detecções e o termo observação referirá-se unicamente a medidas em relação a marcos. Denota-se o conjunto de todas as detecções como $D_{1:k}$.

O objetivo do SLAM multirrobo é estimar a probabilidade posterior

$$p(X_k, m | Z_{1:k}, U_{1:k}, D_{1:k})$$

sobre todas as posturas X_k dos robôs e todas as localizações m dos marcos a partir dos dados disponíveis, $Z_{1:k}$, $U_{1:k}$ e $D_{1:k}$. Aqui $Z_{1:k}$ é o conjunto das medidas obtidas pelos robôs do tempo 1 ao tempo k e $U_{1:k}$ o conjunto de todas as entradas de controle.

4 *SLAM usando um filtro de partículas reduzido*

Neste capítulo é apresentada a abordagem do problema do SLAM baseada em filtros de partículas e chamada de *Fast-SLAM*. O algoritmo foi desenvolvido em (MONTEMERLO et al., 2002) e (MONTEMERLO et al., 2003). Na seção 4.1 é discutido o algoritmo para um único robô, terminando com exemplo na seção 4.2.

4.1 SLAM para um robô

Serão apresentadas inicialmente algumas definições: denota-se por M o número de partículas ou amostras do estado do sistema (postura do robô e o mapa) no momento k e por $x_k^{[i]}$ a i 'ésima partícula.

A elevada dimensionalidade do espaço de estados do SLAM faz com que a aplicação direta de filtros de partículas seja computacionalmente inviável. Contudo, é possível reduzir o espaço de amostragem aplicando a técnica de *Rao-Blackwellization* (R-B), através da qual um estado conjunto é fracionado de acordo com a regra do produto $P(x_1, x_2) = P(x_2|x_1)P(x_1)$ e se $P(x_2|x_1)$ pode ser representada analiticamente, só $P(x_1)$ precisa ser amostrada: $x_1^{[i]} \sim P(x_1)$. A probabilidade conjunta é então representada pelo conjunto $\{x_1^{[i]}, P(x_2|x_1^{[i]})\}_{i=1}^M$. O estado conjunto do SLAM pode então ser fatorado em um componente do robô e um componente condicional do mapa:

$$P(x_{0:k}, m|u_{1:k}, z_{1:k}) = P(m|x_{0:k}, z_{1:k})P(x_{0:k}|u_{1:k}, z_{1:k}) \quad (4.1)$$

Aqui a probabilidade é sobre a trajetória $x_{0:k}$ ao invés de ser sobre a postura x_k (como é na maioria das abordagens do SLAM) e, com isso, quando condicionados à trajetória, os marcos do mapa tornam-se independentes (ver figura 4). Isto permite fatorar a probabilidade posterior do SLAM em um produto de termos mais simples.

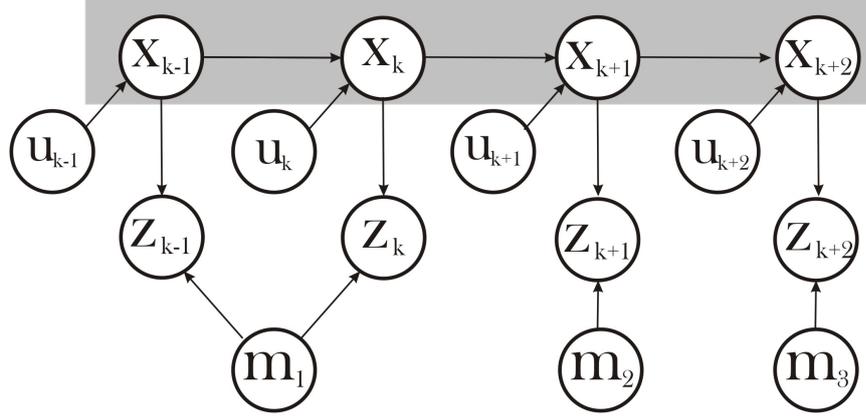


Figura 4: Rede Bayesiana dinâmica que modela o problema do SLAM na sua forma fatorada, extraído de (THRUN; BURGARD; FOX, 2005)

A estrutura essencial do FastSLAM é então um estado reduzido (*Rao-Blackwellized state*), onde a trajetória é representada por amostras ponderadas e o mapa é calculado analiticamente.

Conforme o ambiente é explorado, novas características são observadas e devem ser adicionadas no mapa (ver figura 5).

Cada marco observado pela primeira vez é inicializado para as M partículas geradas a partir do modelo de movimento. Tomando como certa a postura gerada na fase de predição do filtro de partículas (FP) reduzido, os parâmetros iniciais da gaussiana que representa cada marco (ver figura 3) são dados por:

$$g_k(x_k, z) = \begin{bmatrix} m_{jx} \\ m_{jy} \end{bmatrix} = \begin{bmatrix} x_k + r_k^j \cos(\theta_k + \phi_k^j) \\ y_k + r_k^j \sin(\theta_k + \phi_k^j) \end{bmatrix} \quad (4.2)$$

e

$$P_k = (\Delta g_k Q_k^{-1} \Delta g_k)^{-1} \quad (4.3)$$

onde $\Delta g_k = h'(m_j, x_k)$ é o jacobiano do modelo de movimento, Q_k^{-1} é a matriz de covariância do erro de observação e P_k é a covariância inicial da estimação. Deste modo a probabilidade conjunta da postura do robô e a posição dos marcos no instante k , é representada pelo conjunto $\{w_k^{[i]}, x_{0:k}^{[i]}, P(m|x_{0:k}^{[i]}, z_{1:k})\}_{i=1}^M$, onde $w_k^{[i]}$ é o peso da partícula i no instante k . O mapa m acompanhando cada partícula é composto de distribuições gaussianas independentes:

$$P(m|x_{0:k}^{(i)}, z_{1:k}) = \prod_{j=1}^N P(m_j|x_{0:k}^{(i)}, z_{1:k})$$

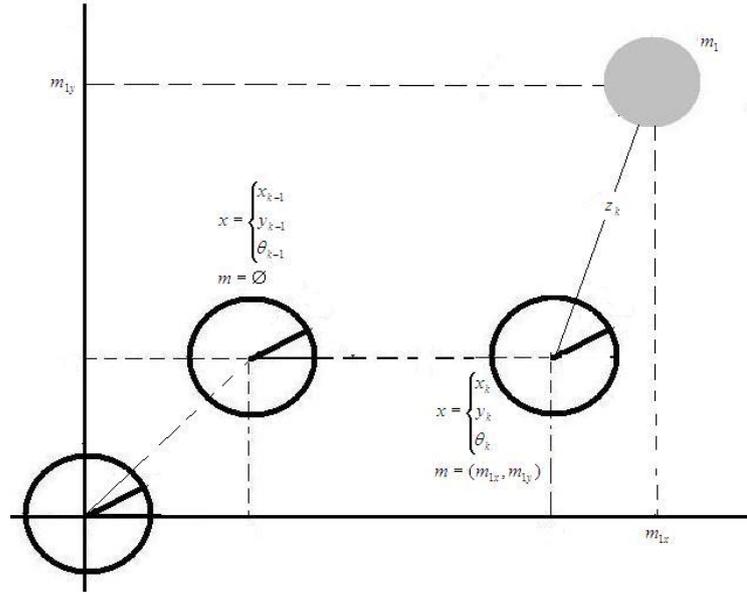


Figura 5: Primeiros instantes da exploração do robô. No instante k o robô encontra o primeiro marco

sendo N o número de marcos. No FastSLAM, uma estimação recursiva é feita usando filtragem de partículas para a postura e EKF para o mapa.

A atualização do mapa, para uma trajetória (partícula) dada $x_{0:k}^{[i]}$, é trivial. Cada marco observado é processado individualmente como uma observação a partir de uma postura conhecida na fase de atualização de um EKF. Os marcos não observados não são atualizados. Propagar as partículas da postura, por sua vez, é mais complexo como será discutido adiante.

Assume-se que no tempo $k-1$, o estado conjunto é representado por $\{w_{k-1}^{[i]}, x_{0:k-1}^{[i]}, P(m|x_{0:k-1}^{[i]}, z_{1:k-1})\}_{i=1}^M$.

Logo, a forma geral do filtro de partículas R-B para o SLAM é como segue:

1. para cada partícula, calcula-se a probabilidade *a priori* ou *proposal distribution* (π), condicionada à história da partícula específica e obtêm-se uma amostra a partir dela;

$$x_k^{[i]} \sim \pi(x_k | x_{0:k-1}^{[i]}, z_{1:k}, u_k). \quad (4.4)$$

Esta nova amostra é adicionada à história da partícula $x_{0:k}^{[i]} \triangleq \{x_{0:k-1}^{[i]}, x_k^{[i]}\}$;

2. ponderam-se as amostras de acordo com a função de pesos (*importance function*)

$$w_k^{[i]} = w_{k-1}^{[i]} \frac{P(z_k | x_{0:k}^{[i]}, z_{1:k-1}) P(x_k^{[i]} | x_{k-1}^{[i]}, u_k)}{\pi(x_k^{[i]} | x_{0:k-1}^{[i]}, z_{1:k}, u_k)}. \quad (4.5)$$

Os termos do numerador desta equação são o modelo de observação e o modelo de movimento. Aqui, o modelo de observação difere de (3.3) já que R-B requer que a dependência sobre o mapa seja eliminada:

$$P(z_k | x_{0:k}, z_{1:k-1}) = \int P(z_k | x_k, m) P(m | x_{0:k-1}, z_{1:k-1}) dm; \quad (4.6)$$

3. se for necessário, fazer re-amostragem. A re-amostragem é feita selecionando partículas com substituição a partir do conjunto $x_{0:k}^{[i]}$ incluindo seus mapas associados com probabilidade de seleção proporcional a $w_k^{[i]}$. Às partículas selecionadas é dado um peso uniforme, $w_k^{[i]} = 1/M$;
4. se a característica medida corresponde a um marco já inserido no mapa, atualizam-se as coordenadas do marco usando um EKF e a postura do robô já estimada. Considerando-se que o marco é fixo, teremos que a predição (média) do filtro corresponde às coordenadas estimadas no momento anterior (não tem dinâmica) e o modelo de observação é a equação (3.12).

As duas versões de FastSLAM na literatura, Fast SLAM 1.0 (MONTEMERLO et al., 2002) e FastSLAM 2.0 (MONTEMERLO et al., 2003), diferem só em termos da probabilidade *a priori* (passo 1) e, conseqüentemente, em seus pesos (passo 2). FastSLAM 2.0 é de longe a solução mais eficiente conforme (MONTEMERLO et al., 2003), quando a precisão do controle for baixa em relação à precisão dos sensores de observação do robô.

Em FastSLAM 1.0, a probabilidade *a priori* é o modelo de movimento

$$x_k^{[i]} \sim P(x_k | x_{k-1}^{[i]}, u_k). \quad (4.7)$$

Portanto, olhando para (4.5), vemos que as amostras são ponderadas de acordo com o modelo de observação marginalizado:

$$w_k^{[i]} = w_{k-1}^{[i]} P(z_k | x_{0:k}^{[i]}, z_{1:k-1}) \quad (4.8)$$

Em FastSLAM 2.0, a probabilidade *a priori* inclui a observação atual:

$$x_k^{[i]} \sim P(x_k | x_{0:k-1}^{[i]}, z_{1:k}, u_k) \quad (4.9)$$

onde

$$P(x_k | x_{0:k-1}^{[i]}, z_{1:k}, u_k) = \frac{1}{C} P(z_k | x_k, x_{0:k-1}^{[i]}, z_{1:k-1}) P(x_k | x_{k-1}^{[i]}, u_k)$$

e C é uma constante de normalização. O peso de acordo com (4.5) é $w_k^{[i]} = w_{k-1}^{[i]} C$.

Estatisticamente, FastSLAM 1.0 e 2.0 sofrem degeneração devido a sua incapacidade de esquecer o passado. A marginalização do mapa em (4.6) introduz dependência na história da postura e das observações, e assim, quando a re-amostragem empobrece esta história, a precisão estatística é perdida. No entanto, os resultados empíricos de FastSLAM 2.0 em ambientes reais externos (MONTEMERLO et al., 2003) mostram que na prática, o algoritmo é capaz de gerar um mapa preciso.

4.2 Exemplo

Considere-se um robô realizando tarefas de exploração percorrendo um ambiente com 4 marcos (ver figura 6).

1. Quando o robô inicia seu percurso, não tem conhecimento sobre o mapa. É retirada uma amostra da nova postura do robô a partir da probabilidade *a priori* e o robô mede os marcos 1, 2 e 3.
2. É avaliado se os marcos já foram observados. Entretanto como eles ainda não foram observados, é inicializado um EKF para cada um deles (ver equações (4.2) e (4.3)).
3. Agora o mapa do ambiente produzido pelo robô é: $m = \{(m_{1,x}, m_{1,y}), (m_{2,x}, m_{2,y}), (m_{3,x}, m_{3,y})\}$. A partir da predição do estado e da medida, calculam-se os pesos.
4. Com as amostras da postura atualizadas e uma nova ação de controle (e uma nova observação no caso de FastSLAM 2.0) gera-se uma nova predição da postura.
5. Em duas observações feitas no tempo $k = 2$ medem-se os marcos 3 e 4. Logo, temos duas medidas z_k^i (z_2^1 e z_2^2) com $i = 1, 2$, no momento o mapa está composto por 3 marcos ($N = 3$) m_j com $j = \{1, 2, 3\}$.
6. É avaliado se os marcos são novos ou já observados. As correspondências são definidas, determinando que:

a medida z_2^2 ($i = 2$) corresponde a um novo marco. Um novo marco é inicializado;

a medida z_2^1 ($i = 1$) corresponde ao marco 3 (ver figura 6). Logo o modelo de medida para esta observação fica:

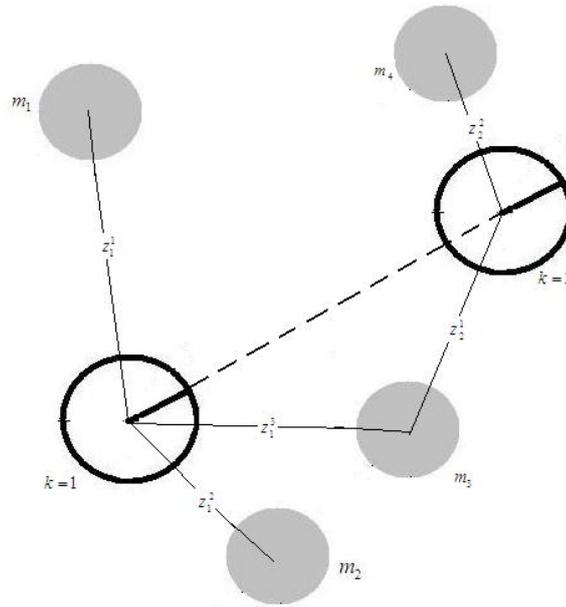


Figura 6: Exemplo do SLAM para um só robô no caso de um ambiente com quatro marcos. Em $k=1$ são feitas 3 observações e em $k=2$ são feitas duas.

$$\begin{bmatrix} r_2^1 \\ \phi_2^1 \\ s_2^1 \end{bmatrix} = \begin{bmatrix} \sqrt{(m_{3,x} - x)^2 + (m_{3,y} - y)^2} \\ \text{atan2}(m_{3,y} - y, m_{3,x} - x) - \theta \\ s_3 \end{bmatrix} + \begin{bmatrix} \epsilon_{\sigma_r^2} \\ \epsilon_{\sigma_\phi^2} \\ \epsilon_{\sigma_s^2} \end{bmatrix}.$$

7. Os marcos 1 e 2 não são modificados já que não foram observados; calculam-se os pesos e se faz re-amostragem. Neste ponto o mapa fica composto por $N=4$ marcos.

O procedimento acima segue para cada nova medida até o fim da exploração.

5 *SLAM multirrobo*

Como foi explicado na introdução, o SLAM multirrobo distribuído aborda o problema de integrar em um único mapa os diferentes mapas que uma equipe de robôs móveis gerou ao explorar independentemente um ambiente.

No SLAM uma das principais tarefas é a associação de dados. Existem três tipos de associação de dados: o primeiro tipo é local, isto é, cada robô identifica os marcos do seu mapa de tal forma que ele saiba que marco do seu mapa deve ser atualizado pela observação efetuada; o segundo tipo é chamado de associação de dados global e é relacionado com o fechamento de voltas; finalmente, o terceiro tipo de associação implica conhecer as correspondências entre os marcos de mapas de diferentes robôs. O primeiro e terceiro tipo de associação de dados são processos de estimação independentes e, portanto um não implica no outro, sobretudo considerando um identificador tão ambíguo como é o caso da largura das árvores, no qual vários marcos podem possuir uma assinatura igual ou parecida.

Como o objetivo da pesquisa é enfatizar a combinação dos mapas parciais, o processo de SLAM para cada robô é realizado assumindo a associação de dados (local) conhecida.

Considerem-se dois robôs com modelos de controle e observação ruidosos, cada um deles fazendo FastSLAM (MONTEMERLO et al., 2003) durante certo tempo. A incerteza é inserida por variáveis aleatórias normais com desvios padrão σ_V (velocidade) e σ_G (ângulo de direção) nas entradas de controle e por variáveis aleatórias normais com desvios padrão σ_R (distância) e σ_B (orientação) nas observações dos marcos. Seja:

- $x_k^{j[i]}$: partícula i do vetor que descreve a posição e orientação (postura) do robô j no instante k , com $i=1,\dots,M$.
- N : número de marcos (árvores) do mapa real.
- N_j : número de marcos mapeados pelo robô j .

Tabela 1: M partículas que representam M possíveis percursos do robô j , cada um deles com seu mapa associado. Cada partícula tem um peso w associado.

Robô j					
P	Trajectoria	Marco1	Marco2	...	Marco N
1	$x_{1:k}^{1[1]} = \{(xy\theta)^T\}_{1:k}^{1[1]}$	$\mu_1^{[1]}, \Sigma_1^{[1]}$	$\mu_2^{[1]}, \Sigma_2^{[1]}$...	$\mu_{N_j}^{[1]}, \Sigma_{N_j}^{[1]}$
2	$x_{1:k}^{1[2]} = \{(xy\theta)^T\}_{1:k}^{1[2]}$	$\mu_1^{[2]}, \Sigma_1^{[2]}$	$\mu_2^{[2]}, \Sigma_2^{[2]}$...	$\mu_{N_j}^{[2]}, \Sigma_{N_j}^{[2]}$
	\vdots	\vdots	\vdots	\vdots	\vdots
M	$x_{1:k}^{1[M]} = \{(xy\theta)^T\}_{1:k}^{1[M]}$	$\mu_1^{[M]}, \Sigma_1^{[M]}$	$\mu_2^{[M]}, \Sigma_2^{[M]}$...	$\mu_{N_j}^{[M]}, \Sigma_{N_j}^{[M]}$

É importante destacar que cada robô pode eventualmente mapear um conjunto menor de marcos que os existentes. Definem-se as seguintes entradas do sistema:

1. o mapa real, que corresponde a uma matriz $3 \times N$ que contém em cada coluna as coordenadas (x, y) de cada marco e sua assinatura (largura da árvore);
2. um conjunto de pontos que descreve o percurso de cada robô (controle).

Com o objetivo de criar um mapa global do ambiente a partir dos mapas parciais, o primeiro passo é calcular a transformação de coordenadas entre os sistemas de coordenadas dos robôs. Esse processo pode ser efetuado a partir de duas metodologias: primeiro, usando os próprios mapas, isto é, os marcos estimados e suas assinaturas, e segundo, usando a medição da distância relativa entre os robôs no momento em que a integração é feita. O primeiro caso é abordado na seção 5.1.1 e o segundo na seção 5.1.2.

5.1 Técnicas para o alinhamento de mapas

Descreve-se a seguir cada uma das etapas do processo de estimação da transformação de coordenadas.

Têm-se M partículas para cada robô, cada uma constituída por uma possível postura do robô j e N_j estimativas dos marcos ($j = 1, 2$), essas estimativas dos marcos por sua vez correspondem a uma coordenada (x, y) e uma matriz de covariância (ver tabela 1).

Como já foi mencionado, tem-se M realizações do mapa de cada robô. Porém, para realizar o processo de alinhamento, precisa-se de um só mapa para cada robô. Em (BALLESTA et al., 2008) escolhe-se o mapa com o maior peso, entretanto, aqui é proposta a combinação das estimativas dos M mapas de cada robô. Deve-se notar que as M

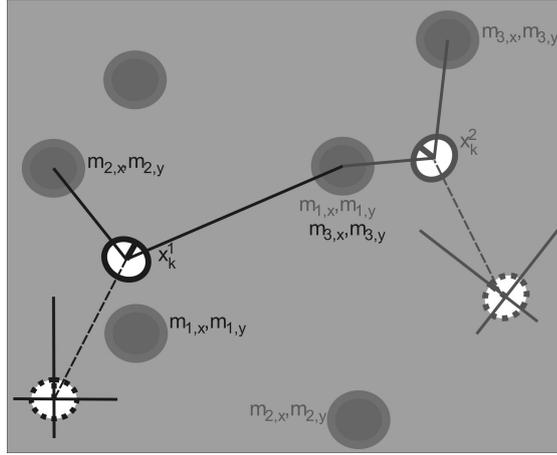


Figura 7: Mapas parciais, construídos depois do SLAM efetuado por cada robô. Esquerda robô 1 e direita robô 2.

estimativas $\mu_i^{[l]}$ do marco i , $i = 1, \dots, N_j$, associadas a cada partícula l , $l = 1, 2 \dots M$, sendo j o índice do robô, são independentes entre si e podem ser combinadas como segue:

$$\mu_i = w_1 \mu_i^{[1]} + w_2 \mu_i^{[2]} + \dots + w_M \mu_i^{[M]}. \quad (5.1)$$

Calcula-se a seguir a matriz de covariância. Seja:

$$\begin{aligned} v &= E(\mu_i) \\ &= w_1 E(\mu_i^{[1]}) + w_2 E(\mu_i^{[2]}) + \dots + w_M E(\mu_i^{[M]}). \end{aligned}$$

A diferença entre as variáveis aleatórias e os valores esperados é dada por:

$$(\mu_i - v) = w_1 (\mu_i^{[1]} - E(\mu_i^{[1]})) + w_2 (\mu_i^{[2]} - E(\mu_i^{[2]})) + \dots + w_M (\mu_i^{[M]} - E(\mu_i^{[M]})).$$

Da independência entre as variáveis aleatórias, a matriz de covariância de cada um dos marcos do mapa resultante é dada por:

$$\Sigma_i = E((\mu_i - v)(\mu_i - v)) = \sum_{p=1}^M w_p^2 \Sigma_i^{[p]}. \quad (5.2)$$

Este processo de combinação (equações (5.1) e (5.2)) é feito para cada um dos marcos de cada robô. Após isso, cada uma das técnicas de alinhamento é usada independentemente para calcular a transformação de coordenadas entre os sistemas de coordenadas de cada robô (seções 5.1.1 e 5.1.2).

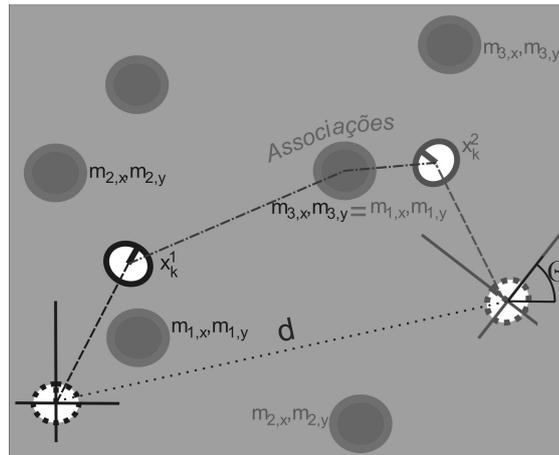


Figura 8: Resultados do processo de alinhamento entre mapas parciais usando associações entre marcos, isto é, a transformação de coordenadas e as correspondências.

5.1.1 Alinhamento de mapas usando associações entre marcos

Este método presuppõe que haja marcos comuns mapeados pelos robôs. Neste ponto do processo de estimação tem-se dois mapas parciais do ambiente (ver figura 7) os quais possuem sistemas de coordenadas diferentes (a origem de cada mapa é o ponto de partida do seu respectivo robô). A partir da informação disponível (ver tabela 1) estima-se a transformação de coordenadas entre os mapas parciais. Desta forma, usa-se como sistema de coordenadas global um dos sistemas de coordenadas local. O método a ser usado é o RANSAC (*Random Sample Consensus*) que segundo (BALLESTA et al., 2008) é um bom método para alinhar mapas baseados em marcos.

O RANSAC é um estimador robusto de parâmetros de modelos matemáticos. Os estimadores de parâmetros clássicos, como mínimos quadrados, baseiam-se em todos os dados que tomam como entrada. Desse modo, se grande parte dos dados vier de uma fonte em que haja erros inerentes ou ruidosos (algum tipo de *outlier*), o modelo final estimado será prejudicado. O RANSAC, porém, foi desenvolvido com o intuito de evitar que *outliers* atrapalhem a computação do modelo final. A partir de diferentes modelos gerados pelo algoritmo e utilizando-se alguma condição de parada, obtém-se o modelo mais representativo entre os avaliados. No caso de associação entre os marcos dos dois mapas, o algoritmo é implementado da seguinte forma:

1. obtém-se uma lista de possíveis correspondências (m, m') entre os dois mapas. O critério seguido para associar correspondências com base nas assinaturas é a distância euclidiana e é desenvolvido como segue:
 - (a) toma-se o primeiro marco do primeiro mapa (marco um);

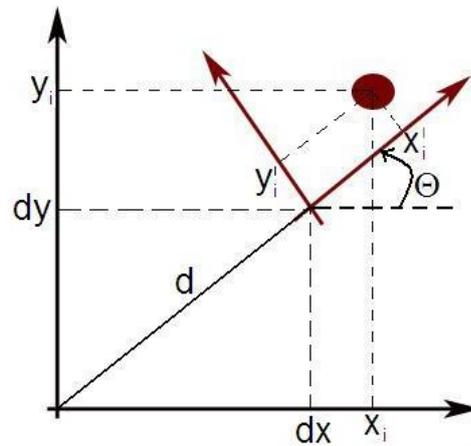


Figura 9: Transformação de coordenadas entre os sistemas de coordenadas dos robôs a partir de uma correspondência entre os mapas.

- (b) cria-se uma lista dos marcos do segundo mapa com uma assinatura similar à do marco um, isto é, árvores cuja largura seja aproximadamente igual; usou-se nas simulações um limiar de $\pm 3\text{cm}$;
 - (c) toma-se o segundo marco do primeiro mapa (marco dois);
 - (d) cria-se uma lista dos marcos do segundo mapa com uma assinatura similar à do marco dois;
 - (e) calcula-se a distância entre o marco um e o marco dois (no sistema de coordenadas do robô um);
 - (f) a partir das listas dos itens (b) e (d), cria-se uma lista dos S pares de marcos do mapa dois que têm entre eles uma distância aproximadamente igual à distância calculada no item (e). Usou-se um limiar de $\pm 1\text{m}$;
 - (g) obtém-se desta forma, S possíveis correspondências entre alguns marcos do mapa dois com o primeiro marco do mapa um e S possíveis correspondências entre alguns marcos do mapa dois com o segundo marco do mapa um, Repete-se o processo para cada possível combinação entre pares de marcos do mapa um, isto é: $\frac{N_1!}{2!(N_1-2)}$ vezes. Fica pronta, desta forma, a lista de correspondências.
2. Depois, dois pares de correspondências ($[(x_i, y_i), (x'_i, y'_i)]e[(x_j, y_j), (x'_j, y'_j)]$) são selecionados aleatoriamente da lista anterior. Os dois pares de correspondências são usados para calcular os parâmetros de alinhamento (dx, dy, Θ). A partir da figura 9 pode-se obter geometricamente as equações (5.3) e (5.4) que calculam as componentes x e y da transformação.

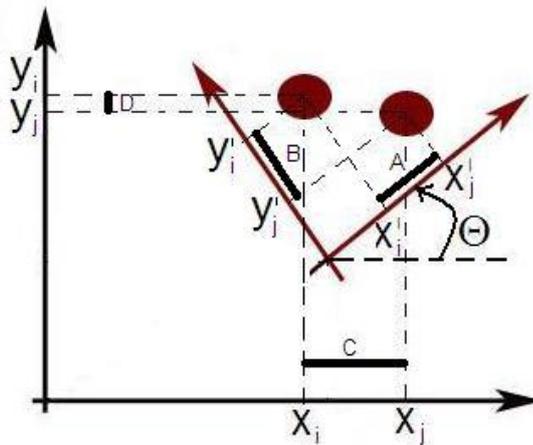


Figura 10: Ângulo Θ da transformação de coordenadas entre os sistemas de coordenadas dos robôs a partir de duas correspondências entre os mapas.

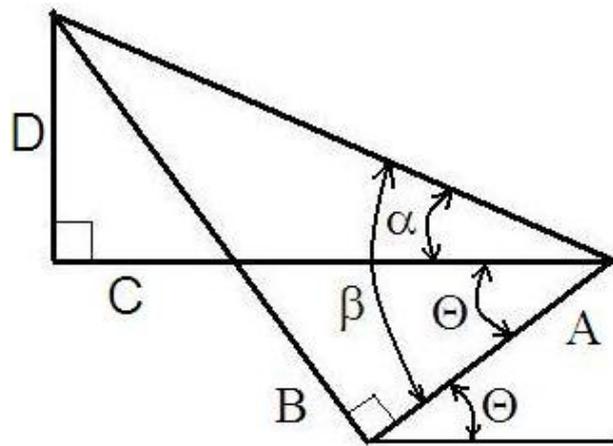


Figura 11: Triângulos retângulos formados pelo par de correspondências escolhido

$$d_x = x_i - x'_i \cos \Theta + y'_i \sin \Theta \quad (5.3)$$

$$d_y = y_i - y'_i \cos \Theta - x'_i \sin \Theta \quad (5.4)$$

$$d^2 = d_x^2 + d_y^2 \quad (5.5)$$

$$\Theta = \arctan \frac{BC - AD}{AC + BD}, \quad (5.6)$$

sendo $A = (x'_j - x'_i)$, $B = (y'_i - y'_j)$, $C = (x_j - x_i)$ e $D = (y_i - y_j)$.

Com a equação (5.6) calcula-se o ângulo da transformação que pode ser derivado da seguinte forma: na figura 10 pode-se observar como o par de correspondências es-

colhido forma dois triângulos retângulos que compartilham uma mesma hipotenusa (distância entre os marcos, que é igual nos dois sistemas de coordenadas dada a correspondência) e cujos catetos são conhecidos. Da figura 11 calcula-se o ângulo Θ da seguinte forma:

$$\begin{aligned}\Theta &= \beta - \alpha \\ &= \arctan\left(\frac{B}{A}\right) - \arctan\left(\frac{D}{C}\right)\end{aligned}$$

Dado que:

$$\arctan(x) - \arctan(y) = \arctan\left(\frac{x - y}{1 + xy}\right)$$

tem-se:

$$\begin{aligned}\Theta &= \arctan\left(\frac{\frac{B}{A} - \frac{D}{C}}{1 + \frac{BD}{AC}}\right) \Rightarrow \\ \Theta &= \arctan\left(\frac{BC - AD}{AC + BD}\right).\end{aligned}$$

3. O terceiro passo consiste em procurar possíveis correspondências que apoiam a transformação (d, Θ) calculada no passo anterior. A cada ponto do conjunto m' é aplicada a transformação calculada e procura-se sua correspondência no conjunto m estabelecendo que a distância euclidiana entre ambos seja menor do que o limiar t_h (0.7m nas simulações). Estas correspondências são chamadas de apoios. Finalmente, repete-se o segundo e o terceiro passo um certo número de vezes pré-determinado. A transformação com maior número de apoios é escolhida.

Como resultado do processo de alinhamento obtém-se as seguintes informações (ver figura 8):

- $d = (d_x, d_y)$, Θ : transformação entre os sistemas de coordenadas do robô 2 e 1.
- N_{jna} : coordenadas dos marcos mapeados pelo robô j que não estão no mapa do outro robô.
- N_{ja} : coordenadas dos marcos mapeados pelo robô j que também estão no mapa do outro robô.

5.1.2 Alinhamento de mapas usando detecções

A metodologia usando detecções, apresentada em (ZHOU; ROUMELIOTIS, 2006) não assume sobreposição entre os mapas parciais e, ao invés disso, usa uma única medida ruidosa de distância relativa entre os robôs no momento da integração dos mapas. Para

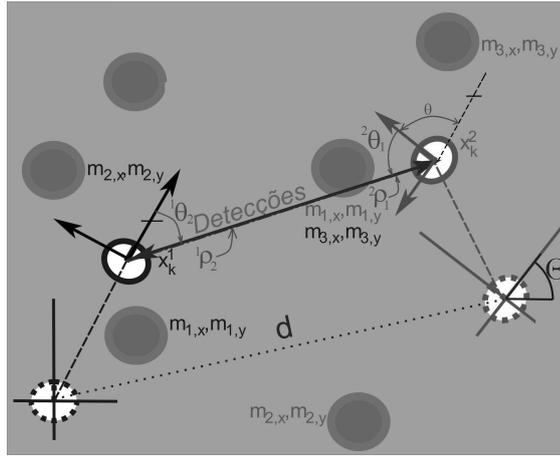


Figura 12: Processo de alinhamento usando detecções.

isso é preciso implementar um sistema de planejamento que possibilite os encontros, isto é, que permita os robôs atingirem uma posição dentro dos seus campos de visão para conseguir, desta forma, compartilhar as informações necessárias para o alinhamento (ver (ROY; DUDEK, 2001)). Consideram-se as características de ruído da detecção como sendo as mesmas do sensor de observação dos marcos, assumindo que o robô consegue distinguir entre um marco e o outro robô. Ver também (OZKUCUR; AKIN, 2010).

1. Considerem-se dois robôs, cada um deles efetuando FastSLAM individualmente durante certo tempo. No final dessa etapa, obtém-se para cada robô M partículas constituídas por uma possível postura do robô j e N_j estimativas dos marcos ($j = 1, 2$). Essas estimativas dos marcos por sua vez correspondem a uma coordenada (x, y) e uma matriz de covariância (ver tabela 1).
2. Neste ponto do processo de estimação têm-se dois mapas parciais do ambiente obtidos como apresentado no início da seção 3 (equações (5.1) e (5.2)) e os quais possuem sistemas de coordenadas diferentes.
3. No momento em que se começa a integração dos mapas, cada um dos robôs mede sua posição relativa em relação ao outro robô (distâncias $({}^1\rho_2, {}^2\rho_1)$ com desvio padrão σ_R e orientações $({}^1\theta_2, {}^2\theta_1)$ com desvio padrão σ_B); (ver figura 12).
4. Com os ângulos medidos $({}^1\theta_2, {}^2\theta_1)$, calcula-se a orientação relativa entre os robôs (θ) dada pela equação (5.7). (ver figura 12).

$$\theta = \pi + {}^1\theta_2 - {}^2\theta_1 \quad (5.7)$$

5. Com os dados obtidos, calcula-se a transformação de coordenadas (d, Θ) , dada pelas equações (5.8), (5.9) e (5.10), a qual permite expressar as coordenadas dos marcos mapeados pelo robô dois no sistema de coordenadas do robô um.

$$dx = x_1 + \rho \cos(\phi_1 + \theta_2) - (x_2 \cos \Theta - y_2 \sin \Theta) \quad (5.8)$$

$$dy = y_1 + \rho \sin(\phi_1 + \theta_2) - (x_2 \sin \Theta + y_2 \cos \Theta) \quad (5.9)$$

$$\Theta = \phi_1 + \theta - \phi_2. \quad (5.10)$$

Nessas equações, ϕ_1 e ϕ_2 são as orientações finais do robô um e dois respectivamente e $\rho = ({}^1\rho_2 + {}^2\rho_1)/2$.

Como resultado do processo de alinhamento obtém-se a transformação de coordenadas entre os sistemas de coordenadas dos robôs: (d, Θ) .

5.2 Obtenção do mapa global

Depois de calculada a transformação de coordenadas, juntam-se as estimativas dos marcos feitas por cada robô em um só mapa. Dessa maneira, os marcos que estão só em um mapa parcial são inseridos no mapa global aplicando a transformação de coordenadas e as estimativas dos pares de marcos (sua média e matriz de covariância) que foram determinadas como correspondentes são misturadas por meio de um filtro de Kalman.

Considerando cada par (i) de marcos correspondentes, usa-se a estimativa do robô um (μ_i^1) como uma predição ou melhor tentativa do estado do marco i, sendo i o índice da correspondência, e a estimativa do robô dois (μ_i^2) como uma medida desse estado. Cada vetor aleatório é descrito apropriadamente por sua matriz de covariância (Σ_i^1 e Σ_i^2) e sua média (μ_i^j). Dessa forma, as equações para integrar as estimativas dos marcos em comum para os dois robôs ficam da seguinte forma:

$$\hat{\mu}_i^G = \mu_i^1 \quad (5.11)$$

$$\hat{\Sigma}_i^G = \Sigma_i^1 \quad (5.12)$$

$$\tilde{y} = \mu_i^2 - \mu_i^1 \quad (5.13)$$

$$S_i = \Sigma_i^1 + \Sigma_i^2 \quad (5.14)$$

$$k_i = \Sigma_i^1 (\Sigma_i^1 + \Sigma_i^2)^{-1} \quad (5.15)$$

$$\mu_i^G = \mu_i^1 + k_i (\mu_i^2 - \mu_i^1) \quad (5.16)$$

$$\Sigma_i^G = (I - k_i) \Sigma_i^1 \quad (5.17)$$

onde μ_i^G é a coordenada do marco i no mapa global, e Σ_i^G a respectiva matriz de covariância. Obtém-se desta forma o mapa global a partir dos mapas parciais criados pelos robôs.

Depois da integração, cada robô fica com as mesmas M partículas, porém expandidas com os marcos mapeados pelo robô com o qual foi feita a combinação. É importante salientar que cada robô integra o mapa conjunto enviado pelo outro robô com cada um dos mapas que possui e após disso continua fazendo FastSLAM até que um outro processo de combinação seja realizado. Finalmente chegara-se a um ponto em que cada robô terá um mapa do ambiente completo e, portanto similar a o dos outros robôs, fazendo desnecessária a realização de outros processos de combinação.

6 *Uma proposta alternativa para FastSLAM multirrobô*

Propõe-se neste capítulo um algoritmo alternativo de combinação de mapas criados usando a técnica de FastSLAM. Uma descrição do algoritmo é apresentada na seção 6.1.

Em (ZHOU; ROUMELIOTIS, 2006) a propagação da incerteza devida à aplicação da transformação de coordenadas é realizada por meio da atualização da matriz de covariância e o vetor de médias representando o estado conjunto (postura do robô e posição dos marcos) depois da combinação. Já em (OZKUCUR; AKIN, 2010) uma adaptação para o método de FastSLAM é apresentada e essa propagação realiza-se ao combinar um mapa conjunto do robô dois (R2) com o mapa de cada partícula do robô um (R1). Por facilidade na nomenclatura nomeiam-se os robôs procedendo a combinação como robô R1 e robô R2. É importante ressaltar que cada robô fará o processo de combinação no momento da detecção, fazendo ele próprio o papel de robô R1 (robô que faz a combinação) e o de robô R2 dois (robô detectado). Depois do processo de combinação, cada robô continua fazendo FastSLAM atualizando as M realizações do mapa global obtidas.

O algoritmo aqui proposto é baseado na adaptação do método de combinação de mapas usando o EKF-SLAM para o FastSLAM feita em (OZKUCUR; AKIN, 2010). Propõe-se o uso das observações entre robôs não só para o cálculo da transformação de coordenadas, mas também no desenvolvimento de um processo que de forma sequencial atualize o alinhamento entre os mapas. Em (OZKUCUR; AKIN, 2010) o mapa conjunto a ser alinhado corresponde à média ponderada dos M mapas do robô dois, enquanto neste trabalho as observações entre robôs são usadas para calcular este mapa conjunto. Na figura 13 o algoritmo de combinação de mapas proposto é representado. Nela, tanto as setas finas quanto as largas representam o fluxo da informação que aparece nas elipses. Dessas informações a crença que R1 tem sobre R2 só estará disponível a partir da segunda detecção.

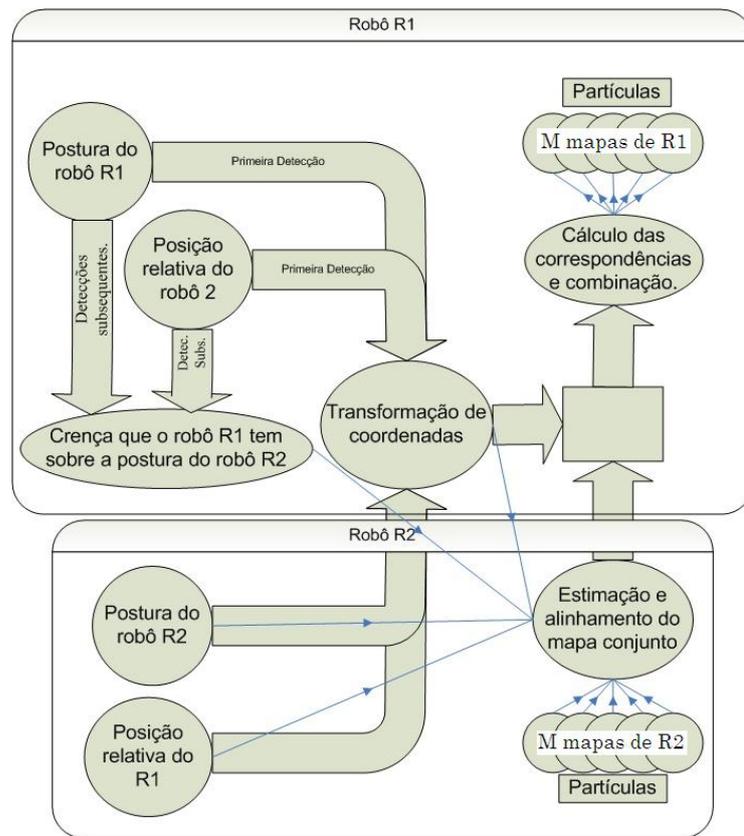


Figura 13: Esquema do algoritmo

6.1 Detalhamento do Algoritmo

No momento em que os dois robôs entram em comunicação, precisa-se obter uma estimativa do mapa que o robô detectado (R2) possui a fim de combinar essa estimativa (mapa conjunto) com os M mapas do robô R1. Considera-se como motivação do algoritmo aqui proposto a seguinte pergunta: será que o alinhamento entre os mapas pode ser melhorado ao considerar as detecções no cálculo do mapa conjunto (mapa a ser alinhado)?

Seja $p(x_{0:k}^2 \mid z_{1:k}^2, u_{1:k}^2)$ a probabilidade que descreve a postura do robô R2 baseando-se unicamente em seu próprio processo de SLAM e considerando como certo o mapa m estimado até o momento k . Define-se a seguir o modelo de detecção para SLAM multirrobô que apresenta a mesma probabilidade comentada acima, só que considerando as observações de posição relativa efetuadas pelos robôs. Isto é feito por uma adaptação do modelo de detecção apresentado em (FOX et al., 2000) e (ODAKURA; BIANCHI; COSTA, 2009) para localização ao problema de SLAM multirrobô aqui abordado.

O algoritmo proposto soluciona o problema de alinhamento de mapas tanto para posições iniciais conhecidas quanto para posições iniciais desconhecidas (*known and un-*

known initial correspondence).

No caso de posições iniciais desconhecidas a primeira detecção é usada para calcular a transformação de coordenadas como explicado na seção 5.1.2 e os mapas são combinados pela primeira vez. Após isso cada robô continua fazendo SLAM individualmente com o mapa (M realizações) obtido até que uma segunda detecção seja feita. Neste momento o modelo de detecção para SLAM multirrobô aqui proposto é aplicado.

Seja G_j o sistema de coordenadas local do robô j , k o instante de tempo anterior à detecção e $k + 1$ o instante no qual a detecção é feita e processada. Requer-se que no instante $k + 1$ o robô R2 envie para R1 uma estimativa do próprio mapa para R1 alinhá-lo e combiná-lo com cada uma das M realizações que ele tem de seu mapa.

Na metodologia apresentada em (OZKUCUR; AKIN, 2010) o mapa enviado é a média dos mapas que R2 tem no instante k ponderada pelos pesos de cada partícula e R1 faz o alinhamento usando unicamente a segunda detecção. Propõe-se aqui atualizar os pesos de R2 antes de obter o mapa conjunto, usando as duas detecções já efetuadas: a primeira para colocar os mapas de R2 no sistema de coordenadas $G1$ e a segunda para atualizar os pesos de R2. Explica-se agora, como a segunda detecção (e de fato todas as seguintes) é integrada na crença de R2. Para isso consideram-se as M posturas x de R2 (calculadas por ele próprio) no sistema de coordenadas $G1$ (transformação feita usando a primeira detecção) como o espaço de todas as possíveis posturas de R2 e com uma probabilidade *a priori* dada pelos pesos (w) produto do FastSLAM. A probabilidade sobre as partículas de R2 já atualizada ($P(x_k^2 = x)$) considerando a última detecção pode ser escrita como segue:

$$P(x_{k+1}^2 = x) := P(x_k^2 = x | Z_{1:k}, U_{1:k}, D_{1:k}),$$

onde $Z_{1:k}$, $U_{1:k}$ e $D_{1:k}$ são as informações coletadas por todos os robôs como explicado na seção 3.3. $P(x_k^2 = x | Z_{1:k}, U_{1:k}, D_{1:k})$ pode ser fatorada nas distribuições sobre a postura do robô detectado (R2) segundo as informações coletadas por R1 e as coletadas por R2:

$$P(x_{k+1}^2 = x) = P(x_k^2 = x | z_{1:k}^2, u_{1:k}^2) P(x_k^2 = x | z_{1:k}^1, u_{1:k}^1, d_2).$$

Aplicando a equação de Chapman-Kolmogorov:

$$P(x_{k+1}^2 = x) = P(x_k^2 = x | z_{1:k}^2, u_{1:k}^2) \int P(x_k^2 = x | x_k^1 = x', {}^1d_2) P(x_k^1 = x' | z_{1:k}^1, u_{1:k}^1) dx'$$

Considerando as M posturas do robô R1 (x') como o espaço de todas as posturas de R1 possíveis, uma aproximação discreta da equação anterior pode ser usada, dando lugar ao modelo de detecção positiva para combinação de mapas em FastSLAM multirrobô:

$$P(x_{k+1}^2 = x) = P(x_k^2 = x | z_{1:k}^2, u_{1:k}^2) \sum_{x'} P(x_k^2 = x | x_k^1 = x', {}^1d_2) P(x_k^1 = x' | z_{1:k}^1, u_{1:k}^1). \quad (6.1)$$

Observando cada uma das partes da equação (6.1) descreve-se cada uma delas da seguinte forma:

- $P(x_{k+1}^2 = x)$: probabilidade que descreve a postura do robô dois dada toda a informação disponível. Esta probabilidade é representada pelos pesos do robô dois depois de serem atualizados pelo modelo de detecção.
- $P(x_k^2 = x | z_{1:k}^2, u_{1:k}^2)$: probabilidade que descreve a postura do robô dois dado seu próprio processo de SLAM. Esta probabilidade é representada pelos pesos do robô dois antes de serem atualizados pelo modelo de detecção.
- x : cada uma das realizações da postura do robô R2 no momento da detecção.
- x' : cada uma das realizações da postura do robô R1 no momento da detecção.
- 1d_2 : posição relativa de R2 com relação a R1.
- $P(x_k^2 = x | x_k^1 = x', {}^1d_2)$: densidade que descreve a probabilidade do robô dois ter a postura x dada a postura x' do robô R1 e a detecção. O cálculo desta probabilidade é apresentado na seção seguinte.
- $P(x_k^1 = x' | z_{1:k}^1, u_{1:k}^1)$: probabilidade que descreve a postura do robô R1 dado seu próprio processo de SLAM. Esta probabilidade é representada pelos pesos do robô R1.
- $\sum_{x'} P(x_k^2 = x | x_k^1 = x', {}^1d_2) P(x_k^1 = x' | z_{1:k}^1, u_{1:k}^1)$: valor que descreve a crença do robô R1 sobre a postura do robô R2.

6.1.1 Cálculo da densidade de detecção $P(x_k^2 = x | x_k^1 = x', {}^1d_2)$

Seja r_d o vetor de posição relativa entre o robô um e o dois (ver figura 12). Dessa forma a f.d.p de detecção pode ser derivada como segue:

$$P(x_k^2 = x | x_k^1 = x', d) = \text{prob}(x - (x' + {}^{G_1}r_d), \sigma_{sl}^2), \quad (6.2)$$

onde $\text{prob}(a, b^2)$ é uma função que retorna a probabilidade de a sob uma distribuição normal com média zero e variância b^2 . σ_{sl} é o desvio padrão do sensor laser.

Resumindo e acrescentando alguns detalhes de implementação, o algoritmo pode ser descrito pelos seguintes passos:

1. no instante $k + 1$ o robô R1 envia para o R2 uma mensagem com a transformação de coordenadas calculada na primeira detecção e um conjunto de possíveis posturas do robô R2 calculadas usando a segunda detecção:

$$\bar{x} = x' + {}^{G_1}r_d,$$

$$\bar{x} = x' + {}_1^{G_1}C({}^{G_1}\phi_1) {}^1d_2, \quad (6.3)$$

onde \bar{x} corresponde às posturas de R2 estimadas por R1, ${}_1^{G_1}C({}^{G_1}\phi_1)$ é a matriz de transformação do sistema de coordenadas solidário ao robô R1 no marco G_1 . Esta matriz é calculada usando a segunda detecção;

2. o robô R2 coloca a sua postura (M realizações) no sistema de coordenadas G_1 a partir da própria postura (estimada) e a transformação de coordenadas enviada por R1;
3. com os resultados dos itens 1 e 2 e a variância do sensor usado para as observações entre robôs, R2 calculada a densidade de detecção (6.2);
4. atualizam-se os pesos das partículas do robô R2 usando o modelo de detecção da equação (6.1);
5. depois de aplicado o modelo de detecção e atualizados os pesos do robô R2 calcula-se a média ponderada do seu mapa (mapa conjunto);
6. uma vez calculado o mapa conjunto, o robô R2 envia-o para o robô R1 que, por sua vez, define as correspondências usando a distância euclidiana. Após definidas as correspondências, a combinação é feita como explicado na seção 5.2. É importante

salientar que este processo de busca de correspondências e posterior combinação é realizado entre os marcos do mapa conjunto do robô $R2$ e o mapa de cada partícula do robô $R1$.

A partir deste momento o robô $R1$ atualiza as M partículas representando o novo estado do seu ambiente fazendo FastSLAM, isto até que uma outra detecção ocorra, momento no qual o modelo de detecção para combinação de mapas aqui proposto é aplicado.

7 *Resultados*

Considerando que os sistemas de coordenadas de cada robô seriam o início do seu percurso, foi criado um entorno gráfico para inserir o mapa e percursos dos dois robôs. Uma vez inseridas as informações, o simulador de FastSLAM 2.0 dos professores Tim Bailey e Juan Nieto ((BAILEY; NIETO, 2006)) é utilizado para gerar os mapas parciais. Os robôs simulados foram robôs tipo Ackerman com as seguintes características: uma distância entre rodas de 4m, uma velocidade de 3m/s, um ângulo máximo de giro da direção de $30 \cdot \pi / 180 \text{rad}$, uma taxa máxima de mudança no ângulo de direção de $20 \cdot \pi / 180 \text{rad/s}$ e um intervalo de tempo entre sinais de controle de 0.025s.

O processo de combinação é efetuado seguindo as metodologias apresentadas nos capítulos 5 e 6. Os três primeiros experimentos apresentados na seção 7.1 correspondem ao estudo comparativo entre as metodologias tradicionais descritas no capítulo 5. Na seção 7.2 compara-se a metodologia de detecções tradicional e a metodologia proposta no capítulo 6. Os experimentos foram feitos tomando como caso base as seguintes características de ruído (descritas no início do capítulo 5):

Ruídos de controle:

$$\sigma_V = 0.3m/s$$

$$\sigma_G = (3.0\pi/180)rad$$

Ruídos de observação e de detecção:

$$\sigma_R = 0.1m$$

$$\sigma_B = (1.0\pi/180)rad$$

7.1 Resultados do estudo comparativo

7.1.1 Experimento 1

Foi comparado o uso do mapa de peso maior com o da estimativa proposta nas equações (5.1) e (5.2) na obtenção do mapa conjunto. Os resultados obtidos levam a crer que o erro em relação à transformação de coordenadas real faz do mapa estimado da forma aqui proposta uma opção plausível para o processo de alinhamento de mapas. No mapa real, a transformação de coordenadas está dada pelo vetor $[-25.8964m \quad -33.0952m \quad 0.2663rad]$. O erro médio depois de 500 experimentos é $[0.4511m \quad 0.4063m \quad 0.0478rad]$ no caso de escolher o mapa do peso maior e $[0.4030m \quad 0.4491m \quad 0.0471rad]$ no caso do mapa ponderado. Dado que a componente y não permite definir o mapa ponderado como a melhor opção, fica a critério do projetista escolher o método que será adotado, tendo em vista as características dos mapas parciais a combinar.

Usando como mapa conjunto o resultado das equações (5.1) e (5.2), procurou-se um grau de sobreposição grande entre os mapas parciais. O mapa obtido neste primeiro experimento (usando associações entre marcos) é apresentado na figura 14. Na figura 15 apresenta-se o mapa obtido usando o método das detecções. Vê-se neste primeiro experimento que o algoritmo baseado em associações sempre encontra uma boa transformação que alinha melhor os mapas do que o algoritmo baseado em detecções, isto se percebe no número maior de marcos comuns não associados no caso de detecções quando comparado com associações.

Os resultados deixam ver como o método baseado em associações é sempre a melhor opção quando existe um grande número de marcos comuns mapeados pelos robôs.

7.1.2 Experimento 2

Na figura 16 apresentam-se os mapas e percursos do segundo experimento e nas figuras 17 e 18 os resultados. É realizada a combinação de mapas parciais com poucos marcos em comum usando tanto o método das associações quanto o das detecções. Foram observadas várias ocorrências em que a transformação de coordenadas calculada com a metodologia das associações era errada, isso devido ao fato do número de correspondências corretas usado pelo algoritmo ter diminuído. Deduz-se que a robustez do processo de estimação usando associações depende fortemente do grau de superposição entre os mapas parciais.

Quando se usam detecções este fato é desprezível, e os mapas podem ser disjuntos.

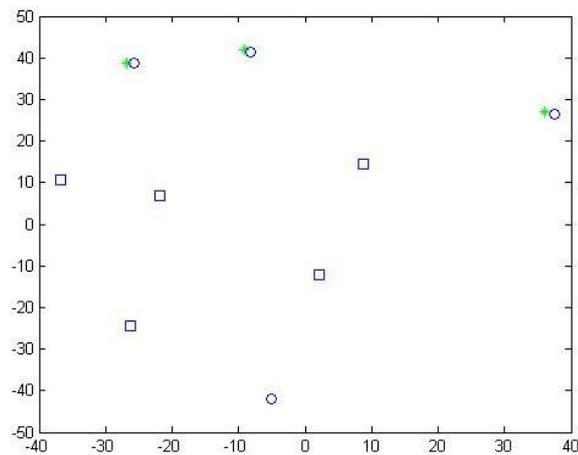


Figura 14: Mapa global usando associações entre marcos (Experimento 1). As estrelas são as árvores mapeadas só pelo robô um, os círculos as árvores mapeadas só pelo robô dois e os quadros são a estimativa combinada dos marcos considerados correspondentes depois de aplicada a transformação.

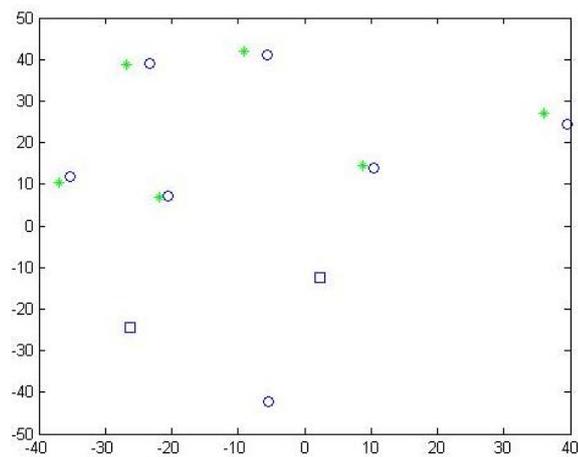


Figura 15: Mapa global usando detecções entre robôs (Experimento 1).

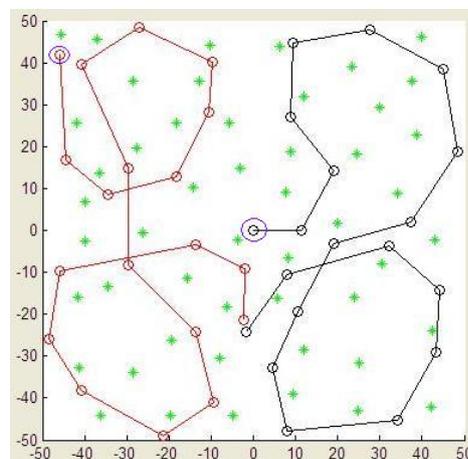


Figura 16: Mapa e percursos do segundo experimento. Pouca sobreposição entre mapas parciais.

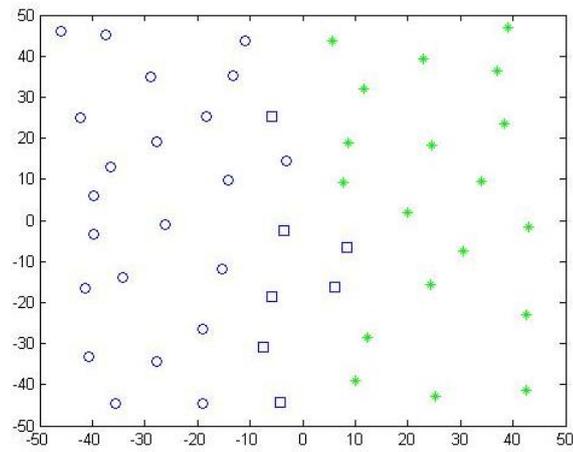


Figura 17: Mapa global usando associações entre marcros (segundo experimento).

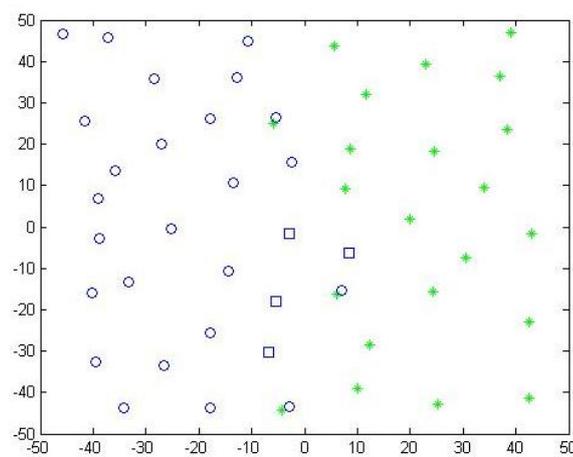


Figura 18: Mapa global usando detecções entre robôs (segundo experimento).

Contudo, o alinhamento não é tão bom quanto na primeira metodologia devido ao número maior de fontes de incerteza: na equação (5.10) o ângulo Θ é calculado a partir das posturas finais dos robôs, sendo estas estimadas usando FastSLAM2.0 e do ângulo θ entre os marcos de coordenadas solidários aos robôs, que procede dos ângulos medidos pelos robôs no momento da detecção (ver equação (5.7)).

O experimento mostra como as correspondências (quadros das figuras 17 e 18) entre os marcos (árvores) são melhor estimadas pela primeira estratégia dada a pequena distância entre marcos correspondentes (limiar de correspondência) que foi usada (0.7m). Por outro lado, não assumir sobreposição é uma vantagem da segunda estratégia no caso de mapas parciais disjuntos.

7.1.3 Experimento 3

Usando os percursos e o mapa da figura 16 (segundo experimento) simulou-se todo o processo de SLAM e de combinação dos mapas para os seguintes casos: caso base, dobrando o ruído de controle do caso base, dobrando o ruído de observação do caso base, e multiplicando por seis o ruído de observação do caso base. Após a comparação dos mapas estimados em cada caso com o mapa real, foram obtidos os resultados apresentados na tabela 2. Na segunda coluna dessa tabela apresenta-se o limiar usado para considerar correspondentes os dois marcos, depois de serem alinhados os mapas utilizando cada um dos métodos de combinação.

Tabela 2: Comparação entre os métodos de combinação para diferentes casos de intensidade de ruído.

Método de combinação	Limiar de correspondência	Test t (valor p)	Modificação aplicada na intensidade dos ruídos
Associações	0.7m	0.8198	Caso base
	0.7m	0.803	Dobrando o ruído de controle
	0.7m	0.8754	Dobrando o ruído de observação
	0.8m	0.8119	Multiplicando por seis o ruído de observação
Detecções	1.1m	0.8126	Caso base
	0.7m	0.8082	Dobrando o ruído de controle
	2.4m	6.078×10^{-5}	Dobrando o ruído de observação
	0.8m	0.8119	Multiplicando por seis o ruído de observação

Os limiares de correspondência da tabela 2 são medidas de quão perto deve estar

um marco do mapa dois (depois de alinhado) a um marco do outro mapa para serem considerados correspondentes. Portanto estes limiares indicam quão concentrados em uma região podem estar os marcos dos mapas a combinar. Eles foram definidos de tal forma que no mapa gerado fossem associados todos os marcos correspondentes. Nos experimentos percebe-se os seguintes fatos interessantes: primeiro, como é de se esperar, no caso de detecções o sensor externo pode ser tão ruim quanto esparsos os mapas a combinar, ou seja, precisa-se de um bom sensor externo para mapas com marcos concentrados e segundo, o uso de detecções pode ser tão bom (em termos de quão concentrados os mapas possam ser) como o uso de associações se o ruído de observação for menor do que o de controle, o que é muito interessante considerando a pouca quantidade de informação usada no caso de detecções quando comparado com associações.

Alem de quão concentrados possam estar os marcos dos mapas a combinar, uma característica importantíssima na hora de comparar os métodos de combinação é sua exatidão em comparação com o mapa real. Para isto cada um dos mapas globais dos casos da tabela 2 são comparados com o mapa real por meio do teste t, um método estatístico para comparação de médias. Na coluna três da tabela apresenta-se o valor p que compara cada caso com o mapa real. O teste foi feito entre as componentes x (y) de cada mapa e as componentes x (y) do mapa real, obtendo dois valores p para cada caso, dos quais, o menor é registrado na tabela por representar o caso crítico ou de maior diferença entre as médias. Verifica-se a divergência do método de detecções quando o ruído de observação é aumentado, e um bom desempenho do método quando o ruído de observação é moderado. Por outro lado, o uso de associações obtém sempre um mapa que aproxima satisfatoriamente o mapa real.

7.2 Resultados da avaliação do algoritmo proposto

Para fins de avaliação foi simulado o processo de SLAM e combinação usando o algoritmo padrão de observações entre robôs explicado na seção 5.1.2 e o algoritmo proposto no capítulo 6, os dois em um mesmo ambiente (ver figura 19). Vale a pena salientar que no caso padrão cada detecção é usada para calcular uma nova transformação de coordenadas e combinar os mapas. Isto é, os robôs fazem FastSLAM com o mapa obtido até uma nova detecção, momento no qual a transformação é de novo calculada, e o mapa conjunto de R2 obtido pela média ponderada combinado com cada mapa de R1.

Foi simulado um primeiro percurso para cada robô (figura 19) após o qual um processo

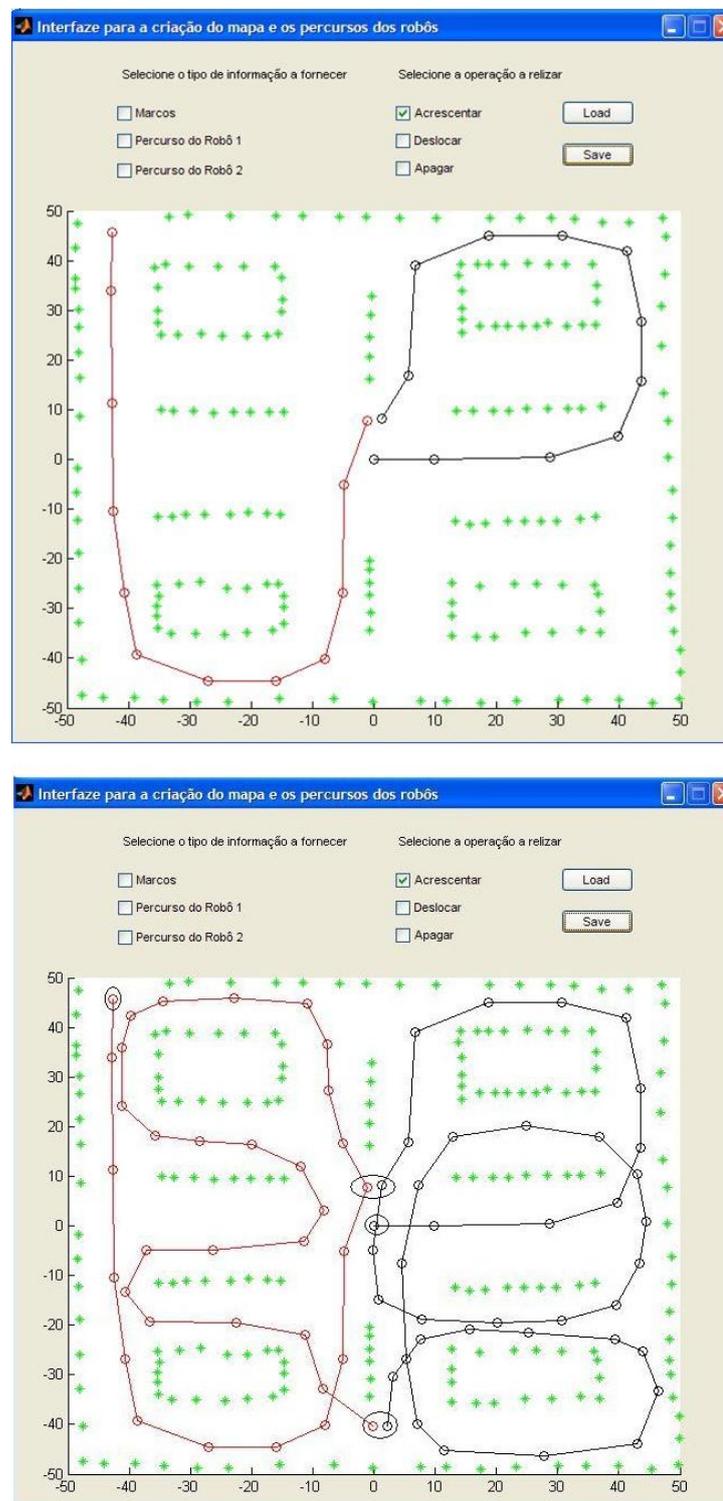


Figura 19: Ambiente da simulação (estrelas), primeiro (acima) e segundo (embaixo) percurso de cada robô. Na figura da esquerda, a elipse superior rodeia o ponto de encontro da primeira detecção e a inferior a do segundo.

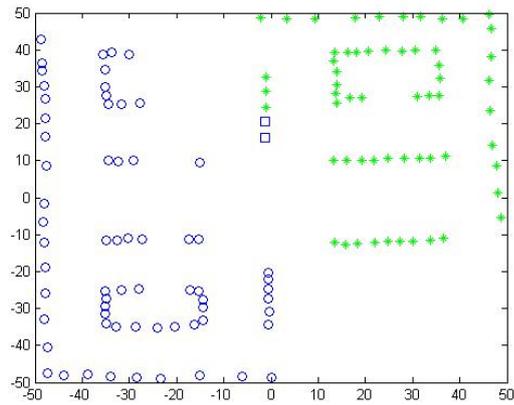


Figura 20: Mapa global obtido pelo robô R1 usando a primeira detecção.

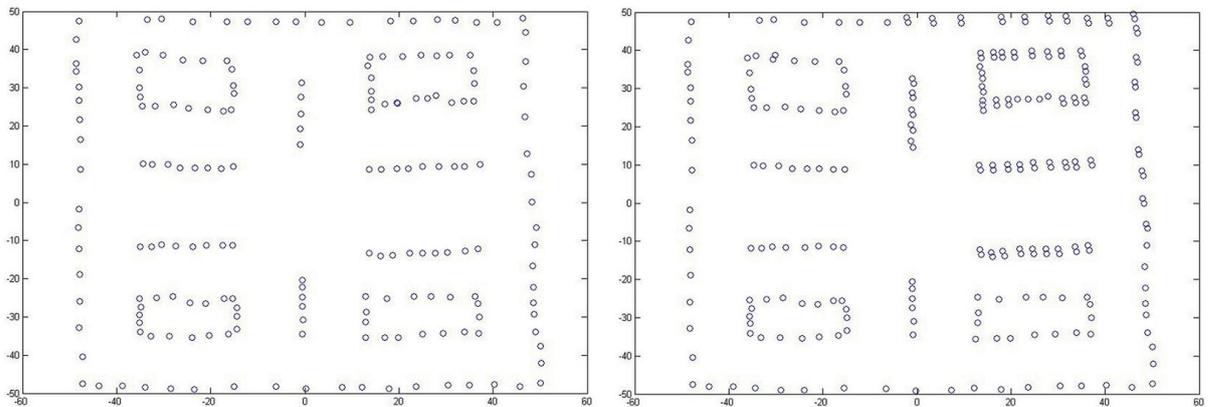


Figura 21: Mapas finais, o da direita usando a metodologia padrão e o da esquerda usando o modelo de detecção para SLAM aqui proposto.

de combinação foi realizado; calcula-se aqui a transformação de coordenadas entre os robôs e, um mapa conjunto do robô R2 obtido usando a média ponderada é combinado com o mapa de cada partícula do robô R1. O mapa de uma das partículas do robô R1 depois da combinação é apresentado na figura 20.

Cada robô efetua a combinação no seu próprio sistema de coordenadas e na sequência faz FastSLAM usando o mapa obtido na primeira detecção. Na segunda detecção (ver figura 19) o robô R1 envia ao R2 a transformação de coordenadas já calculada e uma crença sobre a postura do robô R2 (uma possível postura de R2 por cada partícula de R1 se baseando na postura estimada de R1 e a segunda detecção), informação com a qual o modelo de detecção é aplicado na estimação do mapa conjunto de R2. Esse mapa é enviado para R1 que encontra as correspondências e combina o mapa conjunto com o mapa de cada uma das suas partículas. Os mapas finais (média ponderada) obtidos seguindo a metodologia proposta no capítulo 6 e a padrão são apresentados na figura 21.

Pode-se perceber na figura 21 que a obtenção das correspondências é beneficiada dado que, para um mesmo limiar de correspondência (1.8m) o algoritmo proposto consegue associar todos os marcos que, na realidade, são correspondentes, enquanto que com a metodologia padrão obtém-se vários marcos repetidos. Como parte final da avaliação do algoritmo foram comparadas as médias dos dois mapas com o mapa real usando o teste t. Obteve-se um valor p de 0.062 para a metodologia padrão e de 0.92 para a metodologia aqui proposta. Isto mostra como o algoritmo apresentado estimou melhor o mapa real.

8 *Conclusões e Propostas Futuras*

8.1 **Conclusões**

Nesta dissertação apresentou-se um estudo comparativo das duas metodologias básicas usadas para combinar mapas baseados em marcos e criados por dois robôs percorrendo um mesmo ambiente usando o algoritmo FastSLAM. Embora a solução mais acurada em termos de alinhamento dos mapas parciais seja a associação de marcos, esta depende diretamente do grau de sobreposição entre os mapas. Sendo assim, o método não funciona para mapas disjuntos. Por outro lado, o uso da medida de posição relativa entre os robôs no momento de fazer a combinação dos mapas evita o problema mencionado já que a estimação da transformação de coordenadas depende só do estado do robô e da observação feita, e não das correspondências entre os marcos.

Os experimentos feitos para diferentes intensidades de ruído nos modelos de observação e controle permitiram observar as boas características do uso de associações e uma interessante potencialidade no uso de detecções.

Foi proposto um algoritmo alternativo de combinação de mapas que permite usar as detecções na melhoria do processo de obtenção do mapa conjunto a ser alinhado, explorando de forma mais eficiente essas detecções (na metodologia padrão a detecção é usada em um processo puramente geométrico para calcular o alinhamento entre os mapas). A contribuição da proposta encontra-se principalmente no desenvolvimento de um modelo de detecção positiva para combinação (usando detecções) de mapas em FastSLAM com multirrobôs, que é aplicado a partir da segunda detecção. Os resultados permitiram sua validação em um ambiente simulado.

8.2 **Perspectiva de Trabalhos Futuros**

Algumas sugestões para trabalhos futuros são:

- Usar métodos híbridos de combinação que poderiam explorar melhor o potencial de cada método;
- Implementar em robôs reais o estudo comparativo apresentado e o algoritmo proposto no capítulo 6.

Referências

- ANDERSON, L.; NYGARDS, J. C-sam: Multi-robot slam using square root information smoothing. In: *IEEE International Conference on robotics and Automation*. [S.l.: s.n.], 2008. p. 2798–2805.
- ANDERSON, L. A.; NYGARDS, J. On multi-robot map fusion by inter-robot observations. In: *12th International Conference on Information Fusion*. [S.l.: s.n.], 2009. p. 1712–1721.
- BAILEY, T.; NIETO, J. Fastslam simulator. In: <http://www-personal.acfr.usyd.edu.au/tbailey/>. [S.l.: s.n.], 2006.
- BAILEY, T.; NIETO, J.; NEBOT, E. Consistency of the fastslam algorithm. In: *Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA2006)*. [S.l.: s.n.], 2006. p. 424 –429. ISSN 1050-4729.
- BALLESTA, M. et al. Analysis of map alignment techniques in visual slam systems. In: *Proceedings of the 2008 IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2008)*. [S.l.: s.n.], 2008. p. 825 – 832.
- DEDEOGLU, G.; SUKHATME, G. S. Landmark-based matching algorithm for cooperative mapping by autonomous robots. In: *Proceedings of the International Symposium on Distributed Autonomous Robotic Systems (DARS 2000)*. [S.l.]: Springer-Verlag, 2000. p. 251–260.
- DISSANAYAKE, M. W. M.; NEWMAN, P.; CLARK, S. A solution to the simultaneous localization and map building problem. *IEEE Transactions on Robotics and Automation*, v. 17, n. 3, p. 229–240, 2002.
- DOUCET, A. *On Sequential Simulation-based methods for Bayesian filtering*. [S.l.: s.n.], 1998.
- FENWICK, J. W.; NEWMAN, P. M.; LEONARD, J. J. Cooperative concurrent mapping and localization. In: *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*. [S.l.: s.n.], 2002. p. 1810–1817.
- FOX, D. et al. A probabilistic approach to collaborative multi-robot localization. *Autonomous Robots on Heterogeneous Multi-Robot Systems*, v. 8, n. 3, p. 325–344, Junho 2000.
- FOX, D. et al. Distributed multirobot exploration and mapping. *Proceedings of the IEEE*, p. 1325–1339, Julho 2006.
- FRASSON, M. V. S. *Classe ABNT: confecção de trabalhos acadêmicos em L^AT_EX segundo as normas ABNT*. <http://abntex.codigolivres.org.br>: [s.n.], 2002. Versão 1.

- GUIVANT, J.; NEBOT, E. Optimization of the simultaneous localization and map building algorithm for real time implementation. *IEEE Transactions on Robotics and Automation*, v. 17, n. 3, p. 242–257, 2001.
- HAHNEL, D. et al. An efficient fastslam algorithm for generating maps of large-scale cyclic environments from raw laser range measurements. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. [S.l.: s.n.], 2003. v. 1, p. 206 – 211.
- HOWARD, A. Multi-robot simultaneous localization and mapping using particle filters. *The International Journal of Robotics Research*, v. 25, n. 12, p. 1243–1256, December 2006.
- HOWARD, A.; SUKHATME, G. S.; MATARIC, M. J. Multi-robot mapping using manifold representations. *IEEE International Conference on Robotics and Automation*, v. 4, p. 4198–4203, Abril 2004. ISSN 1050-4729.
- KIM, C.; SAKTHIVEL, R.; CHUNG, W. K. Unscented fastslam: A robust and efficient solution to the slam problem. *IEEE Transactions on Robotics*, v. 24, n. 4, p. 808–820, August 2008.
- MARTINEZ-CANTIN, R.; CASTELLANOS, J.; FREITAS, N. de. Multi-robot marginal-slam. In: *International Joint Conference on Artificial Intelligence*. [S.l.: s.n.], 2007.
- MONTEMERLO, M.; THRUN, S. Simultaneous localization and mapping with unknown data association using fastslam. In: *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*. [S.l.: s.n.], 2003. p. 1985–1991.
- MONTEMERLO, M.; THRUN, S. *FastSLAM, A Scalable Method for the Simultaneous Localization and Mapping Problem in Robotics*. [S.l.]: Springer, 2010. (Springer tracts in advanced robotics, v. 27).
- MONTEMERLO, M. et al. Fastslam: A factored solution to the simultaneous localization and mapping problem. In: *AAAI-02*. [S.l.: s.n.], 2002. p. 593–598.
- MONTEMERLO, M. et al. Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In: *IJCAI-03*. [S.l.: s.n.], 2003.
- MOURIKIS, A. I.; ROUMELIOTIS, S. I. Predicting the performance of cooperative simultaneous localization and mapping (c-slam). *The International Journal of Robotics Research*, v. 25, n. 12, p. 1273–1286, December 2006.
- NIETO, J.; GUIVANT, J.; NEBOT, E. Real time data association for fastslam. In: *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*. [S.l.: s.n.], 2003. p. 412–418.
- ODAKURA, V. V. V. A.; BIANCHI, R. A. da C.; COSTA, A. H. R. General detection model in cooperative multirobot localization. *Journal of the Brazilian Computer Society*, v. 15, n. 3, p. 33–46, 2009.
- OZKUCUR, N. E.; AKIN, H. L. Cooperative multi-robot map merging using fast-slam. In: *RoboCup 2009*. [S.l.: s.n.], 2010. p. 449–460.

- ROMERO, V. A.; COSTA, O. L. V. Map merging strategies for multi-robot fastslam: A comparative survey. In: *A ser publicado em Proceedings of the 2010 Latin American Robotics Symposium and Intelligent Robotics Meeting*. [S.l.: s.n.], 2010.
- ROUMELIOTIS, S. I.; BEKEY, G. A. Distributed multirobot localization. *IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION*, v. 18, n. 5, p. 781 – 795, Outubro 2002.
- ROY, N.; DUDEK, G. *Collaborative Robot Exploration and Rendezvous: Algorithms, Performance Bounds and Observations*. [S.l.]: Kluwer Academic Publishers, 2001. 117–136 p.
- SELVATICI, A. H. P.; COSTA, A. H. R. Relaxação do modelo oculto de markov para o problema de mapeamento e localização simulâneos. In: *Anais do XVI Congresso Brasileiro de Automática*. [S.l.]: Sociedade Brasileira de Automática, 2006. v. 1, p. 3355–3360.
- SEOK, H.; LEE, K. M. Multi-robot slam using ceiling vision. In: *The 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*. [S.l.: s.n.], 2009. p. 912 – 917.
- SMITH, R.; SELF, M.; CHEESEMAN, P. Estimating uncertain spatial relationships in robotics. Springer-Verlag New York, Inc., New York, NY, USA, p. 167–193, 1990.
- SMITH, R. C.; CHEESEMAN, P. On the representation and estimation of spatial uncertainty. *The International Journal of Robotics Research*, p. 56–68, 1986.
- THRUN, S.; BURGARD, W.; FOX, D. *Probabilistic Robotics*. [S.l.]: The MIT Press, 2005.
- THRUN, S.; LIU, Y. Multi-robot slam with sparse extended information filters. In: *Proceedings of the 11th International Symposium of Robotics Research*. [S.l.]: Springer, 2003. Volume 15/2005, p. 254–266. ISBN 978-3-540-23214-8.
- TONG, T. et al. Multi-robot cooperative map building in unknown environment considering estimation uncertainty. In: *2008 Chinese Control and Decision Conference (CCDC 2008)*. [S.l.: s.n.], 2008. p. 2896–2901.
- WHYTE, H. D.; BAILEY, T. Simultaneous localization and mapping: Part i (tutorial). *IEEE Robotics and Automation Magazine*, p. 99–108, Junho 2006.
- WU, M. et al. Cooperative multi-robot monocular-slam using salient landmarks. In: *International Asia Conference on Informatics in Control, Automation and Robotics*. [S.l.: s.n.], 2009. p. 151–155.
- ZHANG, L.; MENG, X. jiong; CHEN, Y. wu. A fastslam algorithm based on the auxiliary particle filter with stirring interpolation. In: *Proceedings of the 2009 IEEE International Conference on Information and Automation*. [S.l.: s.n.], 2009. p. 167–172.
- ZHOU, X.; ROUMELIOTIS, S. I. *Multi Robot SLAM Map Aligment with Rendezvous*. [S.l.: s.n.], Junho 2005.

ZHOU, X.; ROUMELIOTIS, S. I. Multi-robot slam with unknown initial correspondence: The robot rendezvous case. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. [S.l.: s.n.], 2006. p. 1785 – 1792.