

Departamento de Engenharia Electrotécnica
Faculdade de Ciências e Tecnologia
Universidade de Coimbra



MRSLAM – Localização e Mapeamento Simultâneos com Múltiplos Robôs

João Alexandre Simões Martins

Estado da Arte

Novembro de 2012

Departamento de Engenharia Electrotécnica
Faculdade de Ciências e Tecnologia
Universidade de Coimbra



MRSLAM – Localização e Mapeamento Simultâneos com Múltiplos Robôs

Orientador:

Prof. Doutor Rui Rocha

Co-Orientador:

Eng.º David Portugal

Estado da Arte

Novembro de 2012

Índice

<i>Lista de Figuras</i>	4
<i>Lista de Tabelas</i>	5
<i>Lista de Acrónimos</i>	6
<i>Resumo</i>	7
1 <i>Introdução</i>	7
2 <i>Tipos de Representação de Ambientes</i>	11
3 <i>Revisão da literatura</i>	14
3.1 Algoritmos de SLAM populares	15
3.1.1. Filtro de Kalman	16
3.1.2. FastSLAM (Rao-Blackwellized Particle Filter)	19
3.1.3. Generalização do problema de SLAM para múltiplos robôs	20
3.2 Classificação de abordagens para o problema de <i>MRSLAM</i>	25
4 <i>Conclusão</i>	29
5 <i>Referências</i>	31

Lista de Figuras

Figura 1. Representação das etapas de navegação de um robô móvel [SN04].	7
Figura 2. Diferentes etapas de um método de SLAM clássico [Sou03].	8
Figura 3. Relação entre Fiabilidade e Complexidade em sistemas robóticos [R06].	9
Figura 4. Exemplo de Mapa Métrico criado por robôs [GSB07].	11
Figura 5. Exemplo de Mapa Topológico [Spa12].	12
Figura 6. Sala com objecto no meio [Pra07].	13
Figura 7. Mapa de grelhas de ocupação representando a sala da Figura 6 [Pra07].	13
Figura 9. Representação do processo SLAM [SC86].	15
Figura 8. Representação do problema SLAM.	15
Figura 10. Ilustração do processo de FastSLAM [Díaz08].	19
Figura 11. Representação de um sistema centralizado coordenando uma equipa de robôs [LWW+10].	25
Figura 12. Exemplo de uma tarefa de exploração de um ambiente limitado com fronteiras (<i>Boundery</i>) 2D usando múltiplos robôs (R_i) para detectar objectos móveis e/ou estáticos (O_j) [Int2].	27

Lista de Tabelas

Tabela 1. Tabela comparativa entre o Mapas Métricos e Mapas Topológicos.	14
Tabela 2. Vantagens e desvantagens das soluções EKF e FastSLAM.	20
Tabela 3. Principais características entre os algoritmos estudados.	24
Tabela 4. Características principais dos diferentes tipos de sistemas centralizados conhecidos.	26
Tabela 5. Características principais dos diferentes tipos de sistemas distribuídos conhecidos.	28
Tabela 6. Vantagens e desvantagens entre Sistemas Centralizados e Sistemas Distribuídos.	29

Lista de Acrónimos

C-SAM	<i>Collaborative Smoothing and Mapping</i>
EKF	<i>Extended Kalman Filter</i>
FastSLAM	<i>Fast Simultaneous Localization and Mapping</i>
GNSS	<i>Global Navigation Satellite Systems</i>
ICP	<i>Iterated Closest Point</i>
ImpICP	<i>Improved Iterated Closest Point</i>
LIDAR	<i>Light Detection And Ranging</i>
MMW	<i>Millimeter-Wave</i>
MRSLAM	<i>Multi-Robot Simultaneous Localization and Mapping</i>
RBPF	<i>Rao-Blackwellized Particle Filter</i>
RANSAC	<i>Random Sample Consensus</i>
SEIF	<i>Sparse Extended Information Filter</i>
SLAM	<i>Simultaneous Localization and Mapping</i>
SVD	<i>Singular Value Decomposition</i>

Resumo

A maturidade das técnicas de Localização e Mapeamento Simultâneo (*SLAM*) com um único robô é actualmente reconhecida em vários trabalhos [SC86,SSC90,GN01,BW06,KSC08]. Todavia, a extensão destas técnicas para múltiplos robôs, de modo a desempenharem tarefas cooperativas de *SLAM* e exploração em ambientes desconhecidos e/ou dinâmicos é ainda um grande desafio. Embora possam potencialmente melhorar a eficiência, precisão e robustez desta operação, os sistemas de múltiplos robôs móveis possuem várias fontes de complexidade, que requerem esforço na implementação de filtros probabilísticos para a estimação da posição dos diferentes robôs e fusão dos mapas parciais.

Neste trabalho, pretende-se desenvolver um estudo comparativo de duas estratégias para combinação de mapas parciais baseados em *landmarks* para sistemas multi-robôs: a estratégia por associação de *landmarks* e a estratégia por distância relativa entre os robôs. O ambiente a mapear corresponde a um espaço 2D com diversos obstáculos que são mapeados por uma equipa de robôs móveis cooperativos dotados de sensores laser ("*laser range finders*"), que medem distâncias e a localização de cada obstáculo, tendo em conta limitações na comunicação entre os robôs, o tipo de coordenação existente entre os agentes e o espaço de memória para armazenamento das observações.

1 Introdução

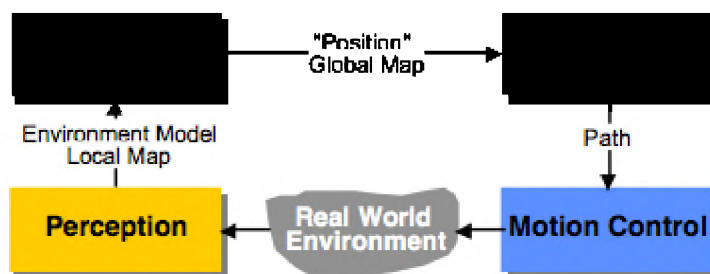


Figura 1. Representação das etapas de navegação de um robô móvel [SN04].

A navegação é uma das capacidades mais exigentes e mais importantes de um robô móvel. O sucesso na navegação depende do êxito de quatro etapas que a constituem: percepção, localização, cognição e o controlo de movimento, como é ilustrado na fig. 1. Mais concretamente, estas quatro etapas consistem no processamento da informação útil dos sensores, determinação da sua posição no ambiente, decisão de como agir para concretizar os seus objectivos e gestão das saídas dos actuadores de modo a seguir a trajectória pretendida.

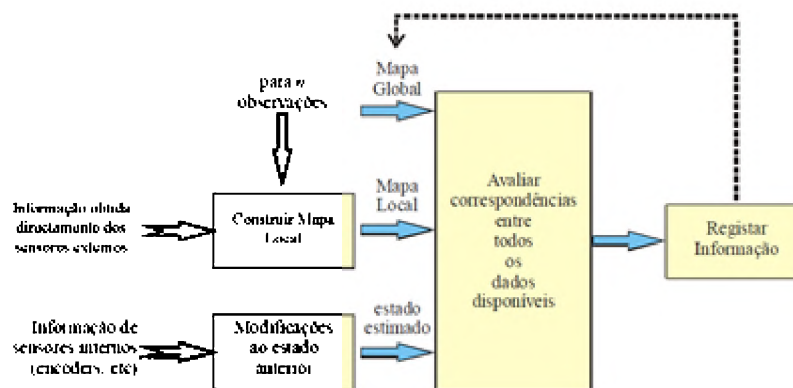


Figura 2. Diferentes etapas de um método de SLAM clássico [Sou03].

Durante a localização, o robô estima a sua posição com base nas medidas dos sensores, que possuem ruído, e em modelos de movimento que podem conter simplificações, como por exemplo não modelarem o fenómeno não linear referente ao deslizamento das rodas. A capacidade e a forma de interagir com o mundo dependem de como ele é percebido. Os sensores provêm o robô com a capacidade de “perceber” o mundo, logo a utilização dos mesmos é de grande importância na construção de um sistema autónomo. Existem diversos tipos de sensores que são utilizados na Robótica Móvel, sensores Exteroceptivos e sensores Proprioceptivos. De seguida, iremos explicar com maior detalhe cada tipo de sensores.

Sensores caracterizados como exteroceptivos são aqueles que adquirem informação a partir da energia vinda do meio. Existem dois princípios distintos de energias que são utilizadas: a electromagnética e a acústica. Os sensores exteroceptivos são agrupados como activos ou passivos, dependendo se emitem ou não energias no ambiente onde o seu sensoriamento é realizado.

Os sensores activos emitem energia no ambiente e obtêm medidas de distância e velocidade relativa a partir da reacção a esta energia. As técnicas mais utilizadas de medição de distâncias são por tempo de voo, por diferença de fase, modulação de frequência, ou por triangulação. No caso de velocidade relativa utiliza-se o efeito Doppler. Os principais sensores de energia activa na Robótica Móvel são os LIDAR's (*“Light Detection And Ranging”*), os sonares, os radares de milímetro, e os GNSS (*“Global Navigation Satellite Systems”*).

Sensores proprioceptivos são sensores que medem os parâmetros internos ao sistema. A vantagem deste tipo de sensor é de não depender do meio ambiente para as suas medidas, o que as torna robustas. Os sensores utilizados na Robótica Móvel que se enquadram nesta categoria são os odômetros e os sensores inerciais.

Em geral, a localização é feita em relação a uma representação do mapa, previamente providenciado ao robô ou que, em alternativa, pode ser construída em tempo real. A tarefa do robô construir o mapa e em simultâneo localizar-se em relação a esse mapa é designada em Robótica por SLAM (do Inglês *Simultaneous Localization and Mapping*), sendo esta o principal foco deste trabalho.

Os métodos de localização mais utilizados actualmente recorrem a leituras de odometria, sensores especializados como o GPS e outros. No processamento da informação gerada por esses métodos são usados processos probabilísticos como o filtro de Kalman [K60] e filtro de partículas [TFB+01], também conhecido como método sequencial de Monte Carlo. As abordagens clássicas para o problema de localização e mapeamento simultâneo são: EKF-SLAM (*Extended Kalman Filter - Simultaneous Localization and Mapping*) [CKD+10] e FastSLAM [BW06] que seguem o diagrama da figura 2 e serão explicadas no capítulo seguinte.

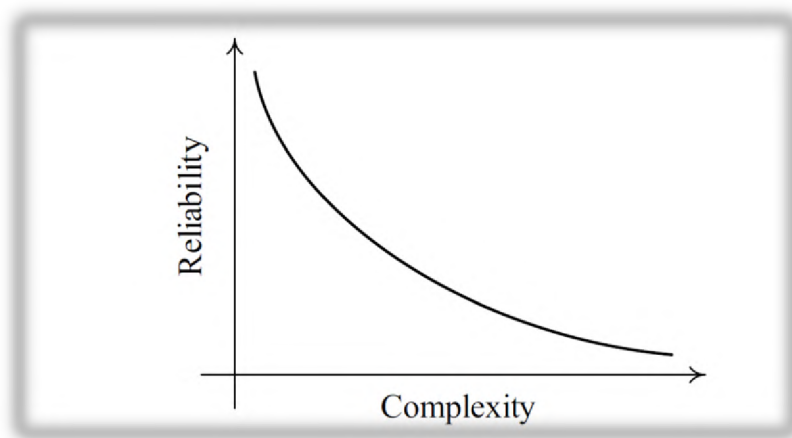


Figura 3. Relação entre Fiabilidade e Complexidade em sistemas robóticos [R06].

O número de trabalhos que explora a coordenação de uma equipa de robôs para a concretização deste tipo de tarefas tem crescido a um ritmo elevado. Algumas razões para este facto são as seguintes [R06]:

- O custo total de uma equipa de robôs simples pode ser menor do que um único robô mais complexo.
- A capacidade de trocar informação entre os robôs permite diminuir a incerteza no processo de estimação.
- Distribuição no tempo – vários robôs podem realizar tarefas, diferentes ou não, ao mesmo tempo.
- Distribuição no espaço – vários robôs podem estar em diferentes lugares ao mesmo tempo.
- Decomposição do problema em partes – Utilização literal da expressão *Divide and Conquer*. Certos problemas são facilmente resolvidos se divididos em pequenos problemas e distribuídos pelos vários robôs.
- Fiabilidade e Robustez a falhas.

A distribuição no tempo e no espaço contribuem ambas para que o tempo de realização da tarefa ou da missão seja menor devido ao paralelismo obtido com a utilização de múltiplos robôs. Essas distribuições estão relacionadas com operações em simultâneo, no espaço e no tempo, respectivamente. Estas podem ser requeridas por uma tarefa como exploração de uma vasta área ou detecção de alvos móveis, melhorando o desempenho em tarefas intrinsecamente distribuídas como limpeza de uma determinada área ou em missões em larga escala no menor tempo possível. Apesar de algumas tarefas não necessitarem de uma solução multi-robô, é desaconselhada a implementação de um único robô complexo e

robusto em simultâneo, devido à relação entre o desempenho e a fiabilidade, como se procura representar na Figura 3.

Soluções multi-robô oferecem maior flexibilidade na gestão da complexidade e distribuição de risco. Por exemplo, em vez da construção de um robô monolítico concebido para ter diversos sensores e capacidade de percepção e raciocínio necessários para uma tarefa específica, um sistema multi-robô pode ser uma solução muito mais fiável e flexível. Se houver uma sobreposição nas capacidades individuais de cada robô, o sistema tem maior robustez, pois caso haja uma falha, esta não significará necessariamente o fracasso de todo o sistema [R06]. Os sistemas de múltiplos robôs são por estas razões adequados a um vasto conjunto de aplicações, tais como a realização de tarefas de busca e salvamento e mapeamento do ambiente [CHOPIN]. Por este motivo, o problema de *SLAM* multi-robô (*MRSLAM*) tem despertado um crescente interesse na comunidade de investigação em Robótica Móvel.

A dificuldade deste desafio consiste em conseguir uma estratégia eficiente para combinar a informação recolhida pelos sensores de todos os robôs. O *SLAM* multi-robô, pode ser implementado de duas formas diferentes: considerando um só mapa global que é actualizado por todos os robôs que executam a tarefa, ou cada robô constrói um mapa parcial do ambiente, que vai sendo combinado com os mapas dos outros robôs da equipa [AN09].

Em relação ao processo de combinação de mapas na segunda metodologia, este pode ser dividido em duas fases:

- A primeira é chamada de alinhamento, no qual se determina a transformação de coordenadas que permitirá colocar a pose de um robô e os seus respectivos *landmarks* no referencial de coordenadas de outro robô.
- Na segunda fase, após o alinhamento, as estimativas dos *landmarks* comuns precisam de ser integradas com o objectivo de se gerar um mapa global.

2 Tipos de Representação de Ambientes

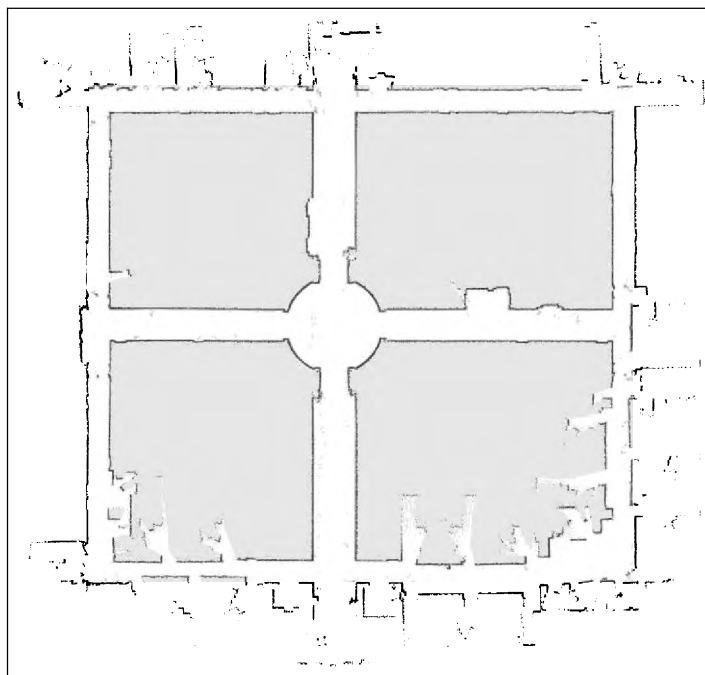


Figura 4. Exemplo de Mapa Métrico criado por robôs [GSB07].

Nesta secção iremos abordar o problema de modelação de ambientes. O *output* do algoritmo de *MRSLAM* é juntamente a construção de um mapa do ambiente real em questão. A actividade de construir um mapa é a criação de um modelo de ambiente através de dados recolhidos pelos sensores, estes modelos são classificados em dois tipos: Mapas Métricos e Mapas Topológicos.

Os Mapas Métricos modelam o ambiente real através de uma representação contendo as informações cartesianas do ambiente, daí poderem ser chamados de mapas cartesianos. Estes mapas podem ser criados pelo próprio robô ou fornecido por um servidor externo, como a Figura 4 ilustra. Esta modelagem conta com a vantagem de ser uma representação simples e facilmente inteligível para os seres humanos. Contudo, devido à quantidade de detalhes que estes mapas normalmente possuem, demandam um consumo de tempo habitualmente grande para ser construídos e muitas das vezes alguns destes detalhes são pouco relevantes para a tarefa de mapeamento.

A forma mais usual de implementar mapas métricos é através de mapas baseados em grelhas. A abordagem baseada em grelhas de ocupação ("*occupancy grids*") foi proposta em [Elf87]. Nesta abordagem, o mundo é representado numa forma bidimensional, utilizando uma matriz, onde cada célula representa uma região do ambiente. Desta forma, a área onde o robô actua é dividida em pequenas regiões, cuja dimensão pode variar de centímetros até metros quadrados. Quanto maior for o tamanho da região, menor será a precisão, isto é, a resolução do mapa. No modelo de grelhas, cada célula da matriz bidimensional representa um ponto correspondente do ambiente real que contém um valor de ocupação, denotado por

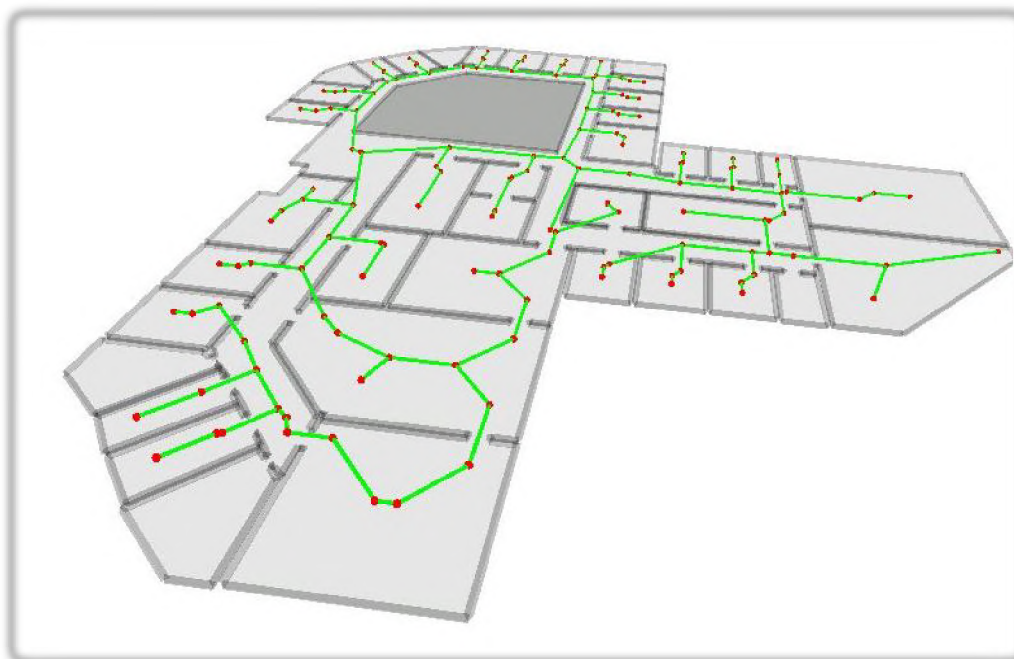


Figura 5. Exemplo de Mapa Topológico [Spa12].

$\text{ProbOc}(x,y)$, que mede a probabilidade da célula $\langle x,y \rangle$ estar ocupada. Por exemplo na Fig. 6, uma sala poderia ser representada pela grelha da Fig. 7.

Neste exemplo o ambiente está perfeitamente representado, isto é, não há dúvidas quanto à presença ou não de um obstáculo. As células correspondente à área livre possuem um valor de ocupação 0 e células correspondente a áreas ocupadas são colocadas com valor de 1 representando 0% e 100% de probabilidade da célula estar ocupada respectivamente. Em aplicações no mundo real é impossível se obter um grelhas de ocupação com este nível de perfeição.

. Abordagem métrica tem a vantagem que o mapa é inteligível para o operador humano. Esta permite que o funcionamento do mecanismo de navegação seja facilmente acompanhado pelos operadores do sistema. O grande inconveniente desta abordagem é o facto de ter um grande consumo de memória e esforço computacional. Para atenuar o problema do consumo de memória, alguns sistemas mapeiam uma grande região do ambiente real em cada uma das células da grelha. O problema desta solução é que ela leva a uma perda de qualidade na representação do mundo uma vez que reduz a resolução do mapa. Em alguns sistemas o tamanho das células não é fixo, em regiões onde há a necessidade de se mapear uma grande quantidade de detalhe, como por exemplo uma sala, são utilizadas grelhas com alta resolução, isto é, com células que correspondem a pequenas regiões do mundo. Em regiões onde não há muita informação para ser armazenada são utilizadas células que mapeiam grandes áreas.

Os Mapas Topológicos consistem em mapa sob a forma de nós e arcos, à semelhança de um grafo onde cada nó representa um ponto relevante do mundo, como por exemplo uma sala, uma porta ou um objeto e cada arco representa a existência de um caminho direto entre estes pontos, como se pode ver a partir da Fig. 5. Esta representação é compacta uma vez que codifica unicamente lugares distintos dentro do meio. Adicionalmente, este tipo de mapa é

Em alguns casos é utilizada uma abordagem híbrida, que combina características das abordagens em grelha e topológica. Esta abordagem híbrida utiliza um mapa topológico para representar a existência de caminhos entre pontos relevantes para a actividade do robô, tais pontos podem corresponder:

- A salas onde o robô irá actuar;
- A pontos relevantes para a localização dos respectivos *Landmarks*;
- A cruzamentos de caminhos;

Os nós que representam as salas são implementados com um grelha de tamanho variável que contém a descrição da mesma. Nos outros nós são utilizadas estruturas mais simplificadas contendo apenas as informações necessárias para caracterizar o ponto permitindo sua localização no futuro. De seguida apresentamos a Tabela 1 com as principais características dos mapas métricos e topológicos.

Mapas Métricos	Mapas Topológicos
O reconhecimento de lugares parecidos não é um problema, uma vez que a posição é mantida em relação ao ponto de partida.	Reconhecimento de lugares parecidos torna-se complexo, sobretudo se não é feito o uso de <i>Landmarks</i> ou outra técnica de apoio.
Requer um conhecimento preciso da localização do robô, que devida à falta de eficiência dos sistemas odométricos, constitui uma tarefa complexa.	Não requer precisão na determinação da posição do robô.
É fácil de construir, mas não de se manter.	É difícil de se construir e de se manter.
Dependendo do tamanho do ambiente, da configuração das células e da capacidade de memória, planejar trajectórias nestas categorias de mapas torna-se numa tarefa complexa.	O planeamento de trajectórias é mais simples, uma vez que é possível adoptar algoritmos de busca conhecidos em grafos.

Tabela 1. Tabela comparativa entre o Mapas Métricos e Mapas Topológicos.

Neste documento, é conduzida uma revisão da literatura onde é discutido o problema de *SLAM*, tanto no caso de um único robô como no caso multi-robô, apresentando as diferentes abordagens descritas na literatura.

3 Revisão da literatura

Os principais temas que serão abordados neste estado da arte são sucintamente descritas ao longo deste capítulo. Inicia-se com um resumo da literatura sobre algoritmos de

SLAM e de seguida é feita uma descrição das metodologias de combinação de mapas mais utilizadas em *Multi-Robot Simultaneous Localization and Mapping* (MRSLAM).

3.1 Algoritmos de SLAM populares

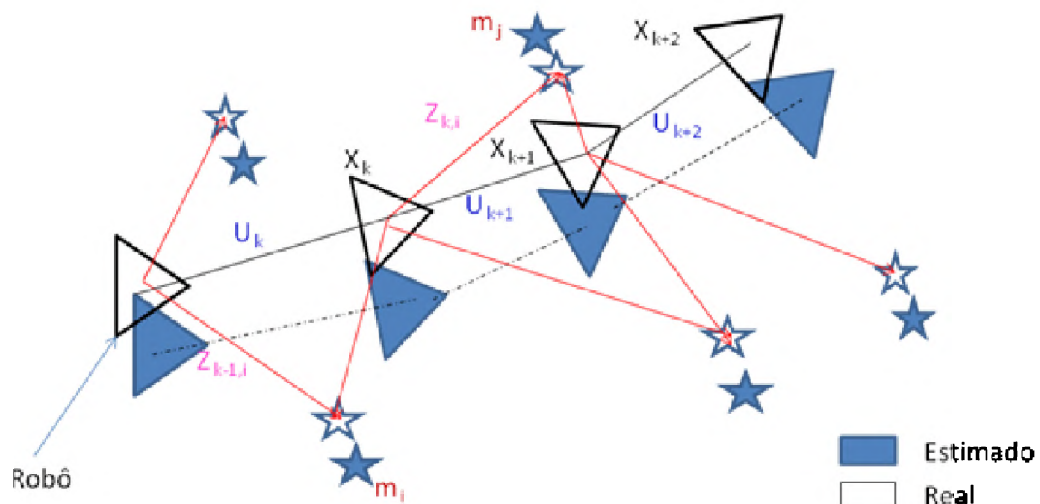


Figura 9. Representação do problema SLAM.

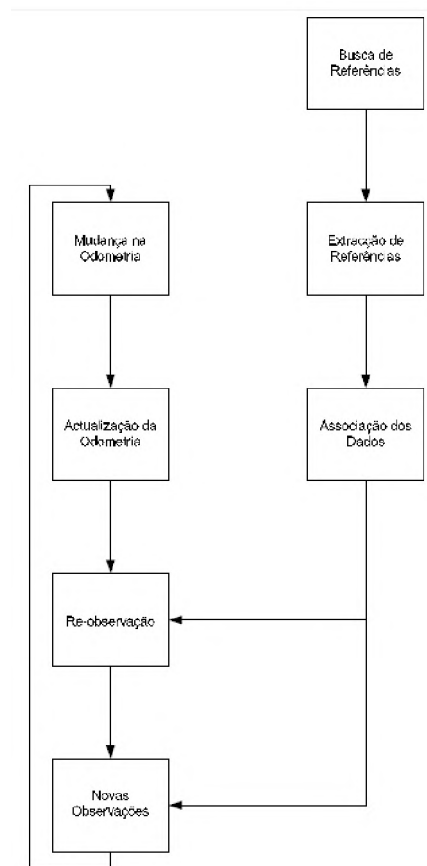


Figura 8. Representação do processo SLAM [SC86].

O *SLAM* aborda o problema de construir um mapa num ambiente desconhecido através de um robô móvel enquanto ao mesmo tempo o utiliza para navegar. Este consiste em múltiplos processos: extracção de referências espaciais, associação dos dados, estimação do estado (posição) e actualização dos estados das *landmarks* (Fig. 8). O robô regista observações de um número desconhecido de *landmarks* utilizando o seu sistema sensorial.

Na Fig. 8, o robô é representado pelos triângulos e as referências espaciais (*landmarks*) por estrelas. A figura ilustra em detalhe o processo de estimação de posição.

Em síntese, o robô movimenta-se e considera inicialmente a sua posição como sendo a resultante da distância obtida por odometria. De seguida, efectua medições em relação às *landmarks* utilizando os seus sensores. Cruza essa informação com a informação calculada por odometria e conclui que não está exactamente onde julga estar (representado pelo triângulo de cor azul). O triângulo branco representa a posição verdadeira do robô. Observamos que embora não seja perfeita a informação baseada na extracção das *landmarks*, geralmente é bem mais viável que a proveniente dos dados de odometria, desde que calculada correctamente.

Uma representação esquemática das diversas fases do processo de *SLAM* é visível na Fig. 9. O problema de *SLAM* começou por ser discutido por Smith *et al.* [SC86], onde foi apresentado um método para estimar as relações entre sistemas de coordenadas mantendo uma estimativa do erro que se propaga aquando das transformações de coordenadas, fornecendo uma modelação explícita da incerteza no processo de estimação.

Quando a odometria fornece novos dados devido à movimentação do robô, a incerteza associada à sua nova posição é actualizada pelo filtro. As referências espaciais são

então extraídas do ambiente através da sua nova posição. O robô tenta associar as novas referências às referências obtidas em observações anteriores e a re-observação das referências é utilizada para actualizar a posição do robô no filtro. As referências que não foram anteriormente identificadas são agora adicionadas ao filtro como novas observações para que possam ser re-observadas posteriormente. De seguida, passamos à descrição do funcionamento do Filtro de Kalman e a sua importância na Robótica Móvel.

3.1.1. Filtro de Kalman

O filtro de Kalman é um estimador linear recursivo que calcula uma estimativa de variância mínima para um estado que evolui no tempo a partir de observações relacionadas linearmente com este estado. Este é satisfatório assumindo determinados pressupostos lineares relativos ao ruído do processo e ao ruído nas observações. O filtro de Kalman possui bastante aplicação em problemas de navegação aeroespacial, controlo e Robótica em geral. É reconhecido como sendo um dos melhores estimadores para sistemas lineares com ruído Gaussiano [Lag08].

Para sistemas não lineares, o filtro de Kalman não é aplicável com rigor, pois a hipótese de linearidade é importante para a dedução do filtro de Kalman como um filtro óptimo. O filtro de Kalman estendido (*EKF*) procura transpor esta dificuldade utilizando uma

linearização em torno da estimativa do estado. O *EKF* é utilizado para estimar a posição do robô através de dados provenientes da odometria e da observação de referências espaciais. Este filtro é normalmente descrito em termos de determinação da posição do robô por si só, ou seja, geralmente é fornecido ao robô um mapa. Não tendo a actualização do mapa, como acontece no caso de um problema de *SLAM*, as matrizes do *EKF* são alteradas e surge uma vertente que é o *EKF-SLAM*, cuja teoria é descrita de seguida de acordo com [WB06] e utilizando as seguintes variáveis:

- x_k - Vector de estado que contém a localização e orientação do robô, isto é, a pose do robô.
- u_k - Vector que contém os valores de saída resultantes do comando aplicado no instante $k-1$ para mover o robô para a posição x_k no instante k .
- m_k - Vector das referências que contém a posição estimada das N referências no instante k .
- z_k - Vector de observações que contém a observação relativa de $n \leq N$ referências no instante k .

O método *EKF-SLAM* descreve a movimentação do veículo na seguinte forma:

$$P(x_k | x_{k-1}, u_k) \leftrightarrow x_k = f(x_{k-1}, u_k) + w_k \quad (1)$$

Na Eq. (1), f é a função que modela a cinemática do veículo e w_k corresponde às perturbações adicionais ao movimento, assumidas Gaussianas, não correlacionadas, de média nula e com covariância Q_k . O modelo de observação é descrito na forma:

$$P(z_k | x_k, m_k) \leftrightarrow z_k = h(x_k, m_k) + o_k, \quad (2)$$

onde h é a função que modela a geometria da observação e o_k corresponde aos erros de observação adicionais, considerados Gaussianos, não correlacionados, de média nula e com covariância R_k que afecta as observações. Com estas definições o método do *EKF* pode ser aplicado para calcular a média e a covariância da distribuição posterior $P(x_k, m_k | Z_{0:k}, U_{0:k}, x_0)$ através de uma predição iterativa e de um algoritmo de correcção:

$$\begin{bmatrix} \hat{x}_{k|k} \\ \hat{m}_k \end{bmatrix} = E \begin{bmatrix} x_k \\ m_k \end{bmatrix} | Z_{0:k} \quad (3)$$

$$P_{k|k} = E \left[\begin{pmatrix} x_k - \hat{x}_{k|k} \\ m_k - \hat{m}_k \end{pmatrix} \begin{pmatrix} x_k - \hat{x}_{k|k} \\ m_k - \hat{m}_k \end{pmatrix}^T | Z_{0:k} \right] \quad (4)$$

A etapa de actualização do tempo (Predição) para a média e covariância é dada por:

$$\hat{x}_{k|k-1} = f(\hat{x}_{k-1|k-1}, u_k) \quad (5)$$

$$P_{k|k-1} = \nabla f P_{k-1|k-1} \nabla f^T + Q_k \quad (6)$$

Em que as estimações são baseadas na actualização da predição no instante i e da actualização da observação no instante j . Para tal, $\hat{x}_{k|k}$ representa a estimativa no instante k após a actualização da predição da observação, e $\hat{x}_{k|k-1}$ representa a estimativa no instante k depois da actualização da predição, mas antes da actualização da observação. Na Eq. (6), ∇f é o Jacobiano de f avaliado para a estimativa $\hat{x}_{k-1|k-1}$. Uma matriz Jacobiana contém todas as derivadas parciais de primeira ordem de um função e se a função for diferenciável representa a melhor aproximação linear à função num ponto específico. Assim a covariância é actualizada utilizando uma aproximação linear do modelo não linear em torno do ponto de operação estimado, que é a diferença fundamental entre o filtro de Kalman e o filtro de Kalman estendido. Seguindo este passo de predição, é realizada uma actualização à observação:

$$\begin{bmatrix} \hat{x}_{k|k} \\ \hat{m}_k \end{bmatrix} = \begin{bmatrix} \hat{x}_{k|k-1} & \hat{m}_{k-1} \end{bmatrix} + W_k [Z_k - h(\hat{x}_{k|k-1}, \hat{m}_{k-1})] \quad (7)$$

$$P_{k|k} = P_{k|k-1} - W_k S_k W_k^T \quad (8)$$

Em que:

$$S_k = \nabla h P_{k|k-1} \nabla h^T + R_k \quad (9)$$

$$W_k = P_{k|k-1} \nabla h^T S_k^{-1} \quad (10)$$

E onde ∇h representa o jacobiano de h avaliado para $\hat{x}_{k|k-1}$ e para \hat{m}_{k-1} .

A solução do *EKF-SLAM* traz muitos benefícios para o problema da navegação, mas também tem limitações, como problemas de seguimento, complexidade computacional, associação de dados e não linearidades. Sob condições ideais para o *EKF-SLAM*, a covariância da estimativa da localização do robô e as posições individuais das *landmarks* convergiram para zero. Contudo, a complexidade computacional da etapa de correcção cresce quadraticamente com o número de *landmarks*, que é usualmente um problema em aplicações práticas. Ainda assim, a solução de *EKF-SLAM* é extremamente sensível à incorrecta associação das referências com as observações. Para além disso, o problema da associação de dados é muito difícil para o *SLAM* quando as referências são re-observadas por pontos de observação muito distintos. Finalmente, a linearização do modelo não linear do movimento e ou a observação dos modelos pode potenciar resultados com soluções inconsistentes [TBF05].

Em [SSC90] apresenta-se pela primeira vez o EKF como solução do problema de SLAM, tornando-se ao longo dos anos uma abordagem clássica na solução deste problema. Muitos trabalhos foram desenvolvidos com base nesta abordagem, entre os quais podem ser mencionados [GN01] e [DNC+02]. Em [GN01] é apresentado um algoritmo otimizado que

considera a forma especial das matrizes e um novo filtro comprimido que reduz significativamente os requisitos a nível computacional quando este trabalha em áreas locais ou com sensores externos de alta frequência. O algoritmo apresentado é testado com êxito em ambientes reais. Em [DNC+02] é apresentado um algoritmo de *SLAM* em agentes móveis operando em ambientes externos usando radares *MMW* para fornecer observações relativas do mapa. Este algoritmo é usado para demonstrar que assuntos chave como gestão do mapa e associação de dados podem ser tratados em ambientes reais.

3.1.2. FastSLAM (Rao-Blackwellized Particle Filter)

	Pose	Feature	Feature 2	Feature M
Partc. #1	x, y, θ	$\mu_1^{[1]}, \Sigma_1^{[1]}$	$\mu_2^{[1]}, \Sigma_2^{[1]}$	$\mu_M^{[1]}, \Sigma_M^{[1]}$
Partc. #2	x, y, θ	$\mu_1^{[2]}, \Sigma_1^{[2]}$	$\mu_2^{[2]}, \Sigma_2^{[2]}$	$\mu_M^{[2]}, \Sigma_M^{[2]}$
⋮				
Partc. #N	x, y, θ	$\mu_N^{[1]}, \Sigma_{1N}^{[1]}$	$\mu_N^{[2]}, \Sigma_N^{[2]}$	$\mu_N^{[M]}, \Sigma_N^{[M]}$

Figura 10. Ilustração do processo de FastSLAM [Díaz08].

Devido a limitações na abordagem de *EKF* (distribuições Gaussianas, isto é, unimodais, e modelos lineares) quanto à sua complexidade quadrática e à sensibilidade às falhas na associação de dados, surge como alternativa o algoritmo *FastSLAM*. Este algoritmo foi apresentado em [MTK+02,M03], sendo o primeiro a considerar modelos não lineares e distribuições multimodais na solução do problema de *SLAM*, com a vantagem de ter maior robustez na associação de dados.

O algoritmo *FastSLAM* é baseado numa característica muito importante do problema *SLAM*, que é a independência condicional que existe entre dois conjuntos diferentes de *landmarks* no mapa, dada a pose do robô. Por outras palavras, se a trajectória real do robô for conhecida, a estimação da posição de todas as *landmarks* do mapa poderia ser feita de modo independente entre cada uma. Isto permite que seja possível a implementação de uma versão do filtro de partículas no problema de *SLAM*, o filtro de partículas *Rao-Blackwellized*. [CKD+10].

A técnica *FastSLAM* utiliza o filtro de partículas para estimar a trajectória do robô, onde cada partícula $x_k^{[i]}$ contém a estimativa da pose do robô, assim como a estimativa de cada *feature* presente no mapa, que são modeladas por distribuições Gaussianas de média $\mu_{j,k}^{[i]}$ e covariância $\Sigma_{j,k}^{[i]}$. Aqui k representa uma partícula de um total de M . Deste modo, o *FastSLAM* pode ser visto como uma combinação entre o filtro de partículas (usado na etapa de predição) e o *EKF*, onde cada medida é actualizada mediante o cálculo dos seus parâmetros μ e Σ .

A distribuição posterior que define o problema de *SLAM* pode ser dividida em duas componentes, uma relacionada com o robô e outra com o mapa do ambiente, desta forma:

$$P(X_{0:k}|Z_{0:k}, U_{0:k}, x_0) = P(X_{0:k}|Z_{0:k}, U_{0:k}, x_0) \prod_{m=1}^M P(m_m|X_{0:k}, Z_{0:k}) \quad (11)$$

Observe-se que no caso do *FastSLAM* o cálculo é sobre a trajetória completa. Por isso, as posições dos objectos convertem-se independentemente e podem ser representadas como funções Gaussianas individuais. A Fig. 6 representa o processo de *FastSLAM*.

O algoritmo original efectua a amostragem sobre o estado do robô e a associação de dados para cada *landmark*. O *FastSLAM 2.0* é uma solução mais eficiente, pois melhora a amostragem tendo em conta as medições enquanto as efectua para um novo estado [M03]. Através da utilização de estruturas de dados eficientes o *FastSLAM* necessita de um tempo de actualização do mapa de $O(M \log N)$, onde M é o número de partículas e N é o número de *landmarks* no mapa. Há trabalhos desenvolvidos que usam esta abordagem e existe também versões melhoradas de *FastSLAM* que podem ser encontrados em [KSC08,ZMC09].

A Tabela 2 apresenta as vantagens e desvantagens dos filtros estudados anteriormente.

Filtro	Vantagens	Desvantagens
EKF	<ul style="list-style-type: none"> Actualização linear das observações utilizando actualizações fraccionadas; Através do submapeamento é possível obter uma solução em tempo constante. 	<ul style="list-style-type: none"> Assume erro Gaussiano o que nem sempre se verifica; A linearização de modelos não lineares pode causar divergência.
FastSLAM	<ul style="list-style-type: none"> Mantém simultaneamente várias hipóteses; Com um grande número de amostras pode representar eficientemente modelos não lineares ou não Gaussianos; Permite diferentes associações de dados em paralelo; Estima toda a trajetória do robô on-line. 	<ul style="list-style-type: none"> O número de partículas aumenta exponencialmente com a dimensão do estado.

Tabela 2. Vantagens e desvantagens das soluções EKF e FastSLAM.

3.1.3. Generalização do problema de SLAM para múltiplos robôs

Tanto o *EKF* como o *FastSLAM* foram generalizados para o problema de *MRSLAM*, porém outros algoritmos também são utilizados na literatura. Nesta secção é feita uma revisão do estado de arte em *MRSLAM*.

Em [TL03], o problema de *MRSLAM* é solucionado utilizando *Sparse Extended Information Filters (SEIF)*, nos quais os mapas e as poses dos robôs são representados através de campos aleatórios de Markov Gaussianos. Estas técnicas focam as actualizações num subconjunto de todas as *landmarks* de maneira a ganhar eficiência computacional. No entanto,

os *SEIF* criam submapas dinamicamente enquanto que versões anteriores dependiam da definição de submapas estáticos. Assume-se que a rede dispersa que é utilizada, captura as dependências naturais em problemas de *SLAM* mais eficientemente do que em decomposições de submapas estáticos. Este motivo origina a obtenção de resultados precisos. O algoritmo evita problemas frequentes na região de fronteira dos submapas onde a estimativa pode tornar-se instável. O algoritmo foi testado com êxito com oito robôs com dados obtidos no *Victoria Park, Sydney*.

Em [BMC+05] são escolhidos pontos objectivos apropriados para cada robô de modo a que se explore ao mesmo tempo áreas diferentes do mapa-mundo. A abordagem escolhida tem em conta o custo de alcance de um determinado ponto objectivo e a sua utilidade para a finalização da tarefa. Esta abordagem também discute o problema do alcance limitado nas comunicações entre robôs e é capaz de coordenar uma equipa de robôs de forma eficiente usando o algoritmo descrito inicialmente. A sua técnica de coordenação revelou diminuição significativa no tempo de execução da tarefa relativamente a outras abordagens de exploração que não coordenam explicitamente os robôs. Tarefas experimentais demonstram que o rendimento do seu algoritmo é escalável com comunicação limitada. Um rumo interessante na investigação é considerar uma situação em que não é conhecida a posição inicial dos robôs mas que estes podem comunicar entre si. Neste caso, o problema de exploração torna-se mais difícil, visto que os robôs têm que resolver dois problemas. Primeiro necessitam de estender o mapa e em segundo lugar, necessitam de descobrir como se relacionam uns com os outros em termos de transformação de coordenadas.

Em [H06], o filtro de partículas apresentado por Hähnel *et al.* [H03] é generalizado para o caso de *SLAM* multi-robô. Este trabalho segue uma nova direcção no projecto de métodos baseados em partículas. O trabalho aborda o problema de *MRSLAM* assumindo em primeiro lugar que a posição inicial dos robôs é conhecida e, em seguida, apresenta uma solução para o problema quando esta não é conhecida. Nessa abordagem, as sequências das observações de cada robô previamente armazenadas são combinadas num único mapa do ambiente quando ocorrem encontros entre eles. Finalmente, é apresentado um método com o objectivo de integrar no mapa os dados recolhidos pelos companheiros antes do primeiro encontro usando a noção de robô virtual capaz de navegar para trás no tempo. Na execução da tarefa de exploração e mapeamento, é assumido que os robôs permutam os dados recolhidos entre si através de uma ligação fiável via *wireless* e que são capazes de detectar outros robôs tendo em conta alguma particularidade que permita identificar inequivocamente um robô móvel de um objecto. Este algoritmo foi testado num ambiente fechado usando quatro robôs homogêneos e *occupancy grids* como representação do ambiente.

Em [CKD+10] é apresentada uma abordagem similar à de Howard [H06], considerando comunicação limitada entre robôs com base na distância entre eles e que não é conhecida a posição inicial. Das considerações anteriores, quando ocorre a troca de informação entre todos os robôs, existe uma pequena latência no processo, que depois tende a estabilizar. O atraso máximo registado usando um computador portátil comum foi de 36 segundos. Esta solução apresentada mostrou ser eficiente e robusta para *MRSLAM* e executou com êxito tarefas em ambientes reais.

Em [FKL+06], é apresentada uma abordagem distribuída de exploração e mapeamento em robôs móveis. O sistema permite que equipas de robôs explorem ambientes de forma eficiente não conhecendo as suas posições iniciais. O robô inicialmente explora autonomamente até que consiga comunicar com outros robôs. Quando os robôs conseguem trocar informação sensorial com os outros robôs, estimam a sua localização relativa utilizando um filtro de partículas adaptado. Com o objectivo de estimar mapas parciais sobrepostos, o filtro tem incorporado um modelo de Markov que prevê observações da área explorada. Durante a exploração, os robôs actualizam os seus modelos de predição com base nas observações do novo ambiente. Se os robôs se encontrarem durante a exploração, isto é, quando se dá o *rendezvous*, sabem as suas posições relativas e consequentemente podem combinar as suas informações num mapa partilhado. Esse mapa global partilhado é usado para coordenar os robôs e estimar a localizações dos outros robôs. Os testes realizados com esta abordagem apresentaram melhorias significativas em relação à fusão de mapas de vários robôs. A comunicação entre robôs é feita através de ligação *wireless* e todos os mapas gerados durante os testes foram virtualmente idênticos indicando uma grande precisão e robustez do sistema.

Em [AN09], aborda-se o problema de alinhar e combinar mapas construídos por vários robôs usando observações feitas entre eles durante os seus trajectos. A solução passa por aplicar *Collaborative Smoothing and Mapping (C-SAM)* que é uma abordagem de *smoothing* usada para combinar mapas criados por diferentes robôs. A principal contribuição do trabalho é justamente o algoritmo para solucionar o problema da associação de dados e eliminar falsas observações quando o alinhamento dos mapas é feito durante o *rendezvous*. Este trabalho é semelhante ao desenvolvido previamente em [ZR06], contudo para solucionar o problema de localização é usado o *EKF* (filtragem).

Em [LBL12] é examinado o problema de SLAM em sistema cooperativos descentralizados em que cada robô precisa de estimar o mapa e os estados de todos os outros robôs sabendo que a comunicação entre eles é limitada e a ligação dinâmica. Prova-se matematicamente como uma estimação equivalente a um sistema centralizado pode ser obtida por todos os robôs na rede, de forma descentralizada. Além disso, o robô só precisa de considerar a sua própria informação da rede topológica de modo a detectar quando o sistema centralizado equivalente estimado é obtido. Este trabalho apresenta mais de 250 minutos de tarefas experimentais utilizando uma equipa de robôs reais. As estimativas (sujeitas a erro) são comparadas com a sua posição real, para todas as poses do robot e das *landmarks*. Por fim, examina-se os efeitos da restrição da comunicação no rendimento do algoritmo apresentado onde se conclui que a memória usada é limitada devido às propriedades de modelo de Markov usado no algoritmo. Os resultados mostraram também que o sistema centralizado equivalente estimado pode ser sempre recuperado depois de um período de fraca conectividade da rede. Em termos de precisão, os resultados gerais mostram que a estimativa da pose e da posição das landmarks dos robôs são semelhantes a um sistema centralizado estimado mesmo com uma percentagem baixa de tempo em que a rede está completamente conectada.

Em [FKK+06] é apresentada uma abordagem distribuída para múltiplos robôs numa tarefa de exploração e mapeamento. O sistema permite equipas de robôs explorarem de forma eficiente o ambiente partindo de locais diferentes e desconhecidos. De modo a

assegurar consistência quando ocorre o processo de combinação de dados para os mapas partilhados, os robôs procuram verificar activamente as suas posições relativas. Usando mapas partilhados, os robôs coordenam a sua estratégia de exploração de modo a maximizar a eficiência da exploração. O sistema foi testado em condições no mundo real extremamente realísticas. Conclui-se num dos testes realizados, que o sistema é bastante eficiente e robusto. Os mapas gerados por esta abordagem são consistentemente mais precisos do que os gerados manualmente medindo as posições e comprimento das salas e dos objectos. Esta abordagem foi testada em ambiente de simulação *Saphira* com três robôs.

Em [CLH+07] é apresentado um algoritmo de SLAM para múltiplos robôs, que utiliza mapas topológicos com vértices dotados de informação métrica local e arestas, que descrevem a posição relativa dos mapas locais adjacentes. Neste trabalho, a fusão dos mapas entre robôs pode ser realizada naturalmente através da adição de uma aresta que estabelece ligação entre mapas topológicos, e a estimação das poses relativas dos robôs é realizada pela optimização dessa aresta. Os autores mostram que é possível desta forma desacoplar o problema de SLAM em problemas mais reduzidos de mapeamento e localização. O algoritmo em estudo foi simulado computacionalmente e foram realizados testes experimentais com robôs Pioneer 3-DX que validaram a actuação do algoritmo. A principal limitação do algoritmo é a necessidade de optimizar de forma offline a informação de mapas topológicos, sendo isto principalmente crítico em tarefas de exploração em ambientes grandes.

A Tabela 2 apresenta as principais características de cada algoritmo abordado.

Algoritmo	Principais Características
Abordagem bayesiana [TL03]	<ul style="list-style-type: none"> • Utiliza filtro SEIF, onde os mapas e poses dos agentes são representados através de campos aleatórios de Markov gaussianos. • Permite aumento da eficiência computacional; • Criação de submapas dinâmicos contribui para obtenção de resultados mais precisos; • Previne instabilidade da estimativa na região de fronteira dos submapas; • Testado com sucesso em ambiente real utilizando 8 robôs.
Abordagem de coordenação com restrições de comunicação [BMC+05]	<ul style="list-style-type: none"> • Utiliza pontos objectivos para exploração de áreas diferentes de modo a maximizar os recursos e tempo da tarefa. • Tem em conta um custo de alcance associado a cada ponto objectivo assim como a sua utilidade para a conclusão da tarefa; • Comunicação limitada entre agentes; • Técnica de coordenação revelou diminuição significativa no tempo de execução da tarefa; • Escalável considerando comunicação limitada.
Abordagem RBPF [H06; CKD+10]	<ul style="list-style-type: none"> • Uso de filtro de partículas generalizado para <i>MRSLAM</i>; • Solução para as situações de posição inicial conhecida e desconhecida; • Combinação das observações dos robôs num único mapa quando se dá um encontro; • Comunicação via <i>wireless</i>; • Robôs capazes de Detecção mútua; • Representação do ambiente através de grelhas de ocupação; • Testado com êxito em ambientes reais usando 4 robôs homogêneos • Latência no processo quando ocorre a troca de informação entre agentes, contudo o resultado final mostrou ser eficiente e robusto.
Abordagem Filtro com modelo de Markov [FKL+06; LBL12]	<ul style="list-style-type: none"> • Utiliza filtro de partículas com modelo de Markov incorporado; • Robôs com Posição relativa desconhecida; • Criação de mapa partilhado após encontro entre robôs; • Abordagem com melhorias no que diz respeito a fusão de mapas; • Comunicação <i>Wireless</i>;
Abordagem C-AM [AN09]	<ul style="list-style-type: none"> • Algoritmo de C-SAM ; • Eliminação de falsas observações e solução do problema de associações de dados durante o alinhamento dos mapas. • Solução para as situações de posição inicial conhecida e desconhecida; • Abordagem suave para fusão de mapas criados por diferentes robôs.
Abordagem distribuída utilizando mapas partilhados [FKK+06]	<ul style="list-style-type: none"> • Abordagem Distribuída; • Solução para as situações de posição inicial conhecida e desconhecida; • Representação do ambiente através de mapas métricos partilhados; • Utiliza uma estratégia de exploração com objectivo de maximizar a eficiência da exploração; • Sistema eficiente e robusto; • Algoritmo testado em ambiente de simulação “Saphira” com três robôs;
[CLH+07]	<ul style="list-style-type: none"> • Representação do ambiente através de mapas topológico; • Trabalho testado em ambiente de simulação e em ambiente real utilizando robôs Pioneer 3-DX; • Limitação do algoritmo é a necessidade de otimizar a informação de mapas topológicos de forma <i>offline</i>

Tabela 3. Principais características entre os algoritmos estudados.

3.2 Classificação de abordagens para o problema de *MRSLAM*

No problema de *MRSLAM* é necessário responder às seguintes questões:

- Qual o número de mapas usado no processo?
- Qual o algoritmo de *SLAM* utilizado a partir do qual o algoritmo para múltiplos robôs é generalizado?
- Pressupõe-se que os robôs conhecem as suas posições iniciais?
- Qual a metodologia usada para o alinhamento de mapas?

A resposta dada a cada uma das perguntas permite classificar as diferentes abordagens de *MRSLAM*. Quando um único mapa global é usado, a abordagem diz-se centralizada. No caso em que cada robô constrói o seu próprio mapa e num determinado momento existe um processo de combinação de mapas, a abordagem diz-se distribuída ou descentralizada. É apresentada de seguida uma revisão da literatura sobre ambos os casos, aplicados ao problema de *MRSLAM*.

3.2.1. Abordagens Centralizadas

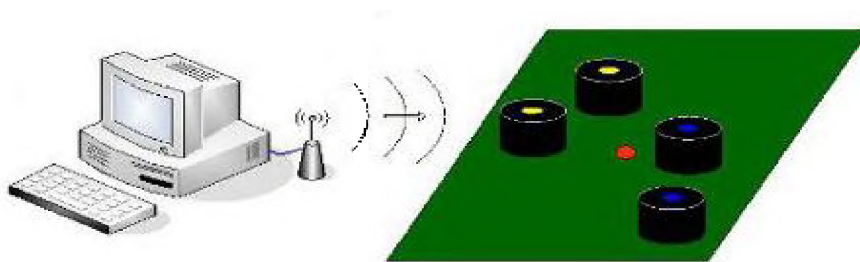


Figura 11. Representação de um sistema centralizado coordenando uma equipa de robôs [LWW+10].

As arquitecturas centralizadas são caracterizadas por terem um único controlador que é individualmente responsável pela tomada de decisões. Pressupõe-se que o processo central tem um modelo global do "mundo" que lhe permite produzir teoricamente soluções óptimas para problemas multi-agente, como se pode verificar na Fig. 7. Podemos considerar duas classes diferentes de sistemas centralizados: totalmente centralizados ou parcialmente centralizados [R06].

Num grupo de robôs totalmente centralizado, cada robô recebe a sua próxima acção de um processador central e transmite-lhe informação local. Nesta arquitectura, os agentes não têm qualquer autonomia de decisão.

Tipo de Sistema	Principais Características
Totalmente Centralizado	<ul style="list-style-type: none"> • Cada agente recebe a próxima acção de um servidor central e transfere-lhe informação local. • Os agentes não possuem qualquer autonomia de decisão.
Parcialmente Centralizado	<ul style="list-style-type: none"> • Existe um agente central contudo, outros agentes têm a autonomia de tomar decisões locais; • Todos os agentes agem localmente como agentes centrais que gerem planos locais e parciais de modo a reduzir a complexidade ; • O agente central concilia os planos locais para maximizar a interactividade para uma maior utilidade global;

Tabela 4. Características principais dos diferentes tipos de sistemas centralizados conhecidos.

Num sistema parcialmente centralizado também existe um agente central mas os outros agentes podem tomar decisões locais autónomas. Estes agem localmente como agentes centrais e geram planos locais e planos parciais que reduzem a complexidade da tarefa. De seguida, o agente central tenta conciliar os planos locais para maximizar a sua interactividade para uma maior utilidade global.

Um sistema centralizado é intrinsecamente coordenado e pode levar a soluções satisfatórias, coerentes e compreensivas mas possui várias limitações. Dependendo da dimensão do grupo, é muito difícil ou até impraticável possuir e manter um modelo global do "mundo" num único agente baseado em vistas locais e potencialmente inconsistentes entre os agentes locais. Além disso, tende a ser uma solução cara em termos de recursos e tempo por usar imensa comunicação entre os agentes locais e o agente central, o que pode levar a um congestionamento grave de comunicações. Tem também uma arquitectura que pode ser muito pouco fiável porque toda a informação está geralmente concentrada nesse agente central, que não pode falhar.

Em [Thr08] é feito um estudo sobre o problema de localização e mapeamento simultâneo (*SLAM*). É feita uma revisão de três algoritmos populares. Em primeiro, o tradicional *EKF* que representa a melhor estimativa da posição do robô. O segundo algoritmo aborda *SLAM* em mapas topológicos de restrições onde se pode aplicar um método de optimização não-linear para recuperar o mapa e as localizações do robô. Por fim, o filtro de partículas que aplica uma estimação de densidade não-paramétrica e um método eficiente de factorização para problemas de *SLAM*. Este trabalho aborda também a extensão destes métodos clássicos para o caso de *MRSLAM* num sistema centralizado.

Em [NGN03] é apresentada uma extensão centralizada de algoritmo de *FastSLAM* para o problema de associação de dados usando uma técnica chamada "vizinho mais próximo" (*nearest neighbour*). As informações recolhidas pelos robôs são enviadas para um agente central que executa o filtro centralizado para construção do mapa. O resultados experimentais realizados em vários ambientes exteriores demonstraram robustez do algoritmo e capacidade

para lidar com múltiplas hipóteses de maneira eficiente e elegante. Uma melhoria do trabalho seria a extensão do actual algoritmo para um sistema *MRSLAM* distribuído.

3.2.2. Abordagens Descentralizadas

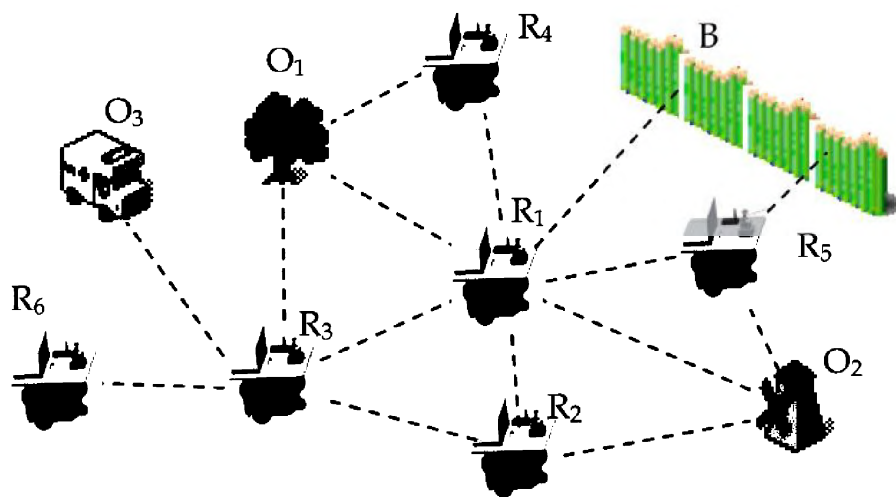


Figura 12. Exemplo de uma tarefa de exploração de um ambiente limitado com fronteiras (*Boundary*) 2D usando múltiplos robôs (R_i) para detectar objectos móveis e/ou estáticos (O_i) [Int2].

As abordagens descentralizadas são compostas por uma rede de agentes lógicos e fisicamente independentes. Cada agente é capaz de "pensar" em planos e decidir as suas próprias acções e vê as dinâmicas do sistema como sendo determinadas pelas interacções com outros agentes. Abordagens descentralizadas podem ser classificadas, pelo grau de autonomia dado a cada agente, como Hierárquicas ou Distribuídas [R06].

Os Sistemas Descentralizados Hierárquicos podem ser vistos como sistemas centralizados localmente onde existe uma hierarquia entre "agentes centrais". As decisões são distribuídas entre camadas a níveis diferentes e existe um agente coordenador em cada nível, o que pode ser visto como um agente central, que produz planos para os agentes em níveis hierárquicos inferiores. Estes planos são conciliados através de uma discussão entre os agentes coordenadores.

Os Sistemas Descentralizados Distribuídos dão a todos os agentes o mesmo poder de decisão e autonomia. Cada agente produz autonomamente os seus próprios planos como função para cumprir os seus próprios objectivos e assegurar um comportamento coerente num modelo de coordenação. Em sistemas descentralizados deste género, cada agente tem capacidades que lhe permite entender o ambiente, debater-se sobre uma tarefa complexa, tomar decisões para cumprir essa tarefa e executar planos.

Tipo de Sistema	Principais Características
Descentralizado Hierárquico	<ul style="list-style-type: none"> • Visto como vários sistemas centralizados localmente; • Decisões distribuídas entre níveis hierárquicos diferentes; • Para cada nível existe um agente central que distribui planos a agentes de níveis inferiores;
Descentralizado Distribuído	<ul style="list-style-type: none"> • Agentes com mesmo poder de decisão e autonomia; • Agentes com comportamento coerente em termos de coordenação; • Agentes com capacidade de interpretar as observações do ambiente e tomar decisões em relação à complexidade da tarefa que estão a realizar.

Tabela 5. Características principais dos diferentes tipos de sistemas distribuídos conhecidos.

As Arquitecturas descentralizadas têm várias vantagens reconhecidas em relação a arquitecturas centralizadas, tais como tolerância a falhas, garantia de fiabilidade, robustez, capacidade de paralelismo, flexibilidade e escalabilidade. Os sistemas baseados em controlo distribuído e dados distribuídos mostram uma melhor *performance* a nível de recursos e de comunicação e uma melhor robustez em relação a sistemas centralizados, já que não existe um controlador central. Assim, estes podem ser bastante flexíveis já que o papel de cada agente pode variar dependendo do contexto.

Em tarefas mais simples, um sistema descentralizado distribuído é preferível a um sistema descentralizado hierárquico, cuja complexidade seria desadequada neste contexto. No entanto, distribuir controlo e dados significa que o conhecimento do estado actual do sistema fica disperso por várias entidades e cada indivíduo tem apenas uma perspectiva parcial, incompleta e imprecisa. Como consequência, os sistemas descentralizados apresentam um grau de incerteza elevado, tornando mais difícil obter um comportamento global optimizado. Além disso, se não existir uma coordenação eficiente, as dinâmicas destes sistemas podem tornar-se extremamente complexas, dando origem a oscilações não lineares e situações de caos [R06].

Ao contrário das abordagens centralizadas, os robôs poderão criar e manter mapas locais com as suas observações. Num certo momento da navegação, os robôs poderão encontrar-se e efectuar um processo de alinhamento e combinação desses mapas. Para efectuar tal combinação, os robôs necessitam da matriz de transformação de coordenadas. Essa matriz pode ser obtida através de duas fontes de informação. Das associações entre *landmarks* mapeadas pelos robôs [TL03] ou das detecções mútuas [ZR06, H06].

A associação das correspondências entre *landmarks* pode ser realizada através de diversos métodos de estimação como: *RANSAC*, *SVD*, *ICP* e *ImplCP*. Em [B08] é feito um estudo comparativo onde se avalia o desempenho de cada uma destas técnicas. Conclui-se que os melhores resultados obtidos nos cenários testados foram: *RANSAC* e *ImplCP*. A primeira caracterizou-se por valores de erro menores e a segunda por ter melhor estabilidade, isto é, ter um menor número de falhas no cálculo da matriz de transformação de coordenadas. Nos trabalhos onde são usadas as observações entre robôs para fazer o alinhamento, o critério

Tipo de Sistema	Vantagens	Desvantagens
Sistema Centralizado	<ul style="list-style-type: none"> • Intrinsecamente coordenado; • Permite obter soluções satisfatórias, coerentes e compreensivas. 	<ul style="list-style-type: none"> • Soluções dispendiosas em termos de recurso e tempo devido ao grande fluxo de comunicação entre os agentes locais e o agente central. • Tarefas com um grau elevado de complexidade torna o sistema impraticável devido à dimensão do espaço de pesquisa. • Para um elevado numero de agentes torna-se impraticável armazenar toda a informação recolhida do ambiente num único agente. • Exige conhecimento da posição inicial dos robôs;
Sistema Descentralizado	<ul style="list-style-type: none"> • Constituído por agentes lógicos e fisicamente independentes; • Maior tolerância a erros e falhas; • Maior robustez, flexibilidade e escalabilidade. • Paralelismo de tarefas; • Maior garantia de sucesso; 	<ul style="list-style-type: none"> • Grau de incerteza elevado que dificulta obtenção de um comportamento global coerente do sistema. • Coordenação pouco eficiente do sistema resulta em dinâmicas complexas e consequentemente, origina oscilações não-lineares e situações de caos. • Problema Detecção mútua;

Tabela 6. Vantagens e desvantagens entre Sistemas Centralizados e Sistemas Distribuídos.

para definir se duas *landmarks* são correspondentes é a chamada distância de Mahalanobis para o caso de abordagens com *EKF*; e a distância Euclidiana no caso de *FastSLAM* [MTK+02].

Apresenta-se a tabela 5 que compara os dois tipos de arquitecturas estudados de modo a tirar conclusões para abordagem a seguir nesta dissertação.

4 Conclusão

Existem diversas abordagens para o problema de SLAM, cada uma delas tem as suas vantagens e desvantagens. O principal objectivo deste trabalho é avaliar as diferentes abordagens de modo a decidir como enfrentar o problema de MRSLAM, dadas as limitações de comunicação, armazenamento e processamento. Assim sendo, a solução final deverá ser computacionalmente eficiente, garantindo sucesso, flexibilidade, escalabilidade e tolerância a falhas no sistema.

Pelos motivos apresentados anteriormente e da interpretação dos dados apresentados prevê-se seguir uma abordagem semelhante à apresentada em [H06], ou seja, um sistema distribuído com um algoritmo robusto e coordenação eficiente dos vários agentes para a tarefa de exploração. Esta abordagem permite a continuação da tarefa caso o sistema falhe que é fulcral para sucesso da tarefa e este algoritmo é bastante completo tendo em conta os algoritmos analisados para as tarefas de coordenação e mapeamento que pretendemos desempenhar. Em relação ao filtro a usar nesta abordagem, será necessário fazer testes experimentais de modo a verificar a eficiência computacional dos diversos filtros nos agentes que correrão em computadores portáteis de processamento limitado. A abordagem terá em conta situações onde será conhecida ou não, a posição inicial dos robôs. Este facto permite a introdução de novos agentes a qualquer momento e lugar na execução da tarefa em questão. Nessa abordagem, os robôs deverão ser capazes de detectar-se mutuamente, comunicar via uma rede WI-FI ad-hoc e representar o ambiente através de grelhas de ocupação.

5 Referências

- [AN09] Anderson, L. A; Nygard, J. "On multi-robot map fusion by inter-robot observations". In: *12th International Conference on Information Fusion*. [S.l.: s.n], pp. 1712-1721, 2009.
- [B08] 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)*, pp. 825-832, 2008.
- [BMC+05] Burgard, W.; Moors, M.; Cyrill, S.; Schneider, F. E. "Coordinated Multi-Robot Exploration". *IEEE Transactions on Robotics* 21(3):376-386, 2005.
- [BW06] Bailey, T.; Whyte, H. D. "Simultaneous localization and mapping: Part i" (tutorial). *IEEE Robotics and Automation Magazine*, pp. 99-108, 2006.
- [CHOPIN] CHOPIN R&D Project, "Cooperation between Human and robotic teams in catastroPhic INcidents", retrieved in April 2012 at <http://chopin.isr.uc.pt/>.
- [CKD+10] Carlone, L.; Kaouk, M.; Du, J.; Bona, B.; Indri, M. "Rao-Blackwellized particle filters multi robot slam with unknown initial correspondences and limited communication". *IEEE International Conference on Robotics and Automation*., pp. 243-249, 2010.
- [Día08] Díaz, C. P. O., "Técnicas Probabilísticas de Localização e Mapeamento aplicadas à Robótica Móvel", Universidade de Brasília Faculdade de Engenharia Mecânica, 2008.
- [DNC+02] Dissanayake, M.W.M.G. and Newman, P., Clark, S., Durrant-Whyte, H. F., and Csorba, M. "A solution to the simultaneous localization and map building (SLAM) problema". *IEEE Transactions on Robotics and Automation* 17(3):229–241, 2002.
- [Elf87] Elfes, A. E. . "Sonar-based Real-World Mapping and Navigation", *IEEE Journal of Robotics and Automation* , Vol. RA-3, No 3, June 1987, pp. 249-265, 1987.
- [FKK+06] Fox, D.; Ko, J.; Konoligey, K.; Limketkai, B.; Schulzz, D.; Stewart, B. , "Distributed Multi-robot Exploration and Mapping", In: *Proceedings of the IEEE*, VOL. 94, No. 7, July 2006.
- [FKL+06] Fox, D.; Ko, J.; Konolige, K.; Limketkai, B.; Schulz, D.; Stewart, B. "Distributed multirobot exploration and mapping". In: *Proceedings of the IEEE*, pp. 1325-1339, 2006.
- [FNL02] 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*. pp.1810-1817, 2002.
- [GN01] Guivant, J.; Nebot, E. "Optimization of the simultaneous localization and map building algorithm for real time implementation". *IEEE Transactions on Robotics and Automation*, 17(3), pp. 242-257, 2001.
- [GSB07] Grisetti, G.; Stachniss, C.; Burgard, W. , "Improved techniques for grid mapping with Rao-Blackwellized Particle Filters", *IEEE Robotics and Automation Society*, Feb 2007.

- [H06] Howard, A. "Multi-robot simultaneous localization and mapping using particle filters". *The international Journal of Robotics Research*, 25(12), pp. 1243-1256, 2006.
- [HBF+03] Hähnel, D., Burgard, W., Fox, D., and Thrun, S. "An efficient FastSLAM algorithm for generating maps of largescale cyclic environments from raw laser range measurements". *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 2121-2427, 2003.
- [HSM06] Howard, A., Sukhatme, G. S., and Mataric, M. J. "Multi-Robot Mapping using Manifold Representations". In: *Proceedings of the IEEE - Special Issue on Multi-robot Systems* 94(7):1360-1369, 2006.
- [K60] Kalman, R. E. "A new approach to linear filtering and prediction problems". In: *Journal of basic Engineering*. pp. 35-45, 1960.
- [KSC08] Kim, C.; Sakthivel, R; Chung, W. K. "Unscented fastslam: A robust and efficient solution to the slam problema". *IEEE Transactions on Robotics*, 24(4), pp. 808-820, 2008.
- [Lag08] Lages, W. F., "Filtro de Kalman", Universidade Federal do Rio Grande do Sul Escola de Engenharia, Departamento de Engenharia Elétrica, 2008.
- [LBL12] Leung, K.; Barfoot, T.; Liu, H. , "Decentralized Cooperative SLAM for Sparsely-Communicating Robot Networks: A Centralized-Equivalent Approach", In: *Journal of Intelligent and Robotic Systems*, pp. 66:321-342, 2012
- [Luc09] Lucas, A. E. S.; "Mapeamento e Localização Baseados em Mosaicos Visuais", Instituto Superior Técnico Universidade Técnica de Lisboa, 2009.
- [LWW+10] Li, c.; Watanabe, T.; Wu, Z.; Li, H.; Huangfu, Y. , "The Real-Time and Embedded Soccer Robot Control System", Robot Soccer, Vladan Papi (Ed.), ISBN: 978-953-307-036-0 (2010), InTech, Available from: <http://www.intechopen.com/books/robot-soccer/the-real-time-and-embedded-soccer-robot-controlsystem>.
- [M03] Montemerlo, M. et al. "Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges". In: *International Joint Conferences on Artificial Intelligence*, 2003.
- [MT03] 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*. pp. 1985-1991, 2003.
- [MTK+02] Montemerlo, M., Thrun, S., Koller, D., and Wegbreit, B. 2002. "FastSLAM: a factored solution to the simultaneous localization and mapping problema". *AAAI National Conference on Artificial Intelligence (AAAI-2002)*.
- [NGN03] 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.I.: s.n.], pp. 412-418, 2003.
- [Pra07] Prado, J.P.A., "Navegação de Robôs Móveis", Seleção Documental, Centro Universitário de Lins São Paulo v. 1, p. 5-13, 2007
- [R06] Rocha, R. "Building Volumetric Maps with Cooperative Mobile Robots and Useful Information Sharing - A Distributed Control Approach based on Entropy", PhD Thesis, Faculdade de Engenharia do Porto, 2006.

- [SC86] Smith, R. C.; Cheeseman, P. "On the representation and estimation of spatial uncertainty". *The international Journal of Robotics Research*, pp. 56-68, 1986.
- [SN04] Siegwart, R., Nourbakhsh I. R., "Introduction to Autonomous Mobile Robots", First Edition, Intelligent Robotics and Autonomous Agents series, The MIT Press, Massachusetts Institute of Technology, Cambridge, Massachusetts, 2004
- [Sou03] Sousa, A. J. M. , "Arquitecturas de Sistemas Robóticos e Localização em Tempo Real Através de Visão". PhD Thesis, Faculdade de Engenharia do Porto, 2003.
- [Spa12] "Spatial cognition – Project R3 – [Q-Shape]", Website: <http://www.sfbtr8.uni-bremen.de/project/r3/research.html> (última visita 1 Dezembro, 2012).
- [SSC90] Smith, R. C.; Self, M.; and Cheeseman, P. "Estimating uncertain spatial relationships in robotics". *Autonomous Robot Vehicles*, pp. 167–193, 1990.
- [TBF05] Thrun, S.; Burgard, W.; Fox, D. "Probabilistic Robotics". The MIT Press, 2005.
- [TFB+01] Thrun, S.; Fox, D.; Burgard, W.; Dellaert, F. "Robust Monte Carlos Localization for mobile robots". In: *Artificial Intelligence*. pp. 99-141, 2001.
- [Thr08] Thrun, S.; "Simultaneous Localization and Mapping", In: *Robotics and Cognitive Approaches to Spatial Mapping, Springer Tracts in Advanced Robotics Volume 38*, pp 13-41, 2008.
- [TL03] Thrun, S. and Liu, Y. "Multi-robot SLAM with sparse extended information filters". In: *the 11th International Symposium of Robotics Research (ISRR-03)*, 2003.
- [Zho10] Yu Zhou. "Distributed Control of Multi-Robot Deployment Motion", Motion Control, Federico Casolo (Ed.), ISBN: 978-953-7619-55-8 (2010), InTech, Available from: <http://www.intechopen.com/books/motioncontrol/distributed-control-of-multi-robot-deployment-motion>.
- [ZMC09] Zhang, L.; Meng, X. J.; Chen, Y. W. "A fastslam algorithm based on the auxiliary particle filter with stirling interpolation". In: *Proceedings of the 2009 IEEE International Conference on Information and Automation*. pp. 167-172, 2009.
- [ZR06] Zhou, X.; Roumeliotis, S. I. "Multi-robot slam with unknown initial correspondence: The robot rendezvous case". In: *Proceedings of the IEEE/RSJ International Conference of the Intelligent Robots and Systems*. pp. 1785-1792, 2006.