



PROPOSTA DE TRABALHO DE DISSERTAÇÃO

Técnicas Probabilísticas de Localização e Mapeamento aplicadas à Robótica
Móvel

Autor:

Claudia Patricia Ochoa Díaz

Orientador:

Alberto José Álvares, Dr. Eng.

UNIVERSIDADE DE BRASÍLIA

Faculdade de Engenharia Mecânica

Pós-Graduação em Sistemas Mecatrônicos

Brasília,DF

2008

Sumário

1	Introdução	1
2	Descrição do trabalho de pesquisa	3
2.1	Título	3
2.2	Orientador do projeto	3
2.3	Autor do projeto	3
2.4	Entidades interessadas na realização do projeto	3
3	Justificativa	4
4	Exposição do problema	6
5	Objetivos do trabalho de pesquisa	9
5.1	Objetivo Geral	9
5.2	Objetivo Específicos	9
6	Revisão da Literatura	10
6.1	Localização	10
6.1.1	Classificação do problema de localização	10
6.1.2	Método de localização de Markov	11
6.1.3	Filtro Estendido de Kalman (EKF)	12
6.1.4	Filtro de Partículas	12
6.2	Mapeamento	12
6.2.1	Extração de Features	13
6.2.2	Formulação e estrutura do problema SLAM	14
6.3	Filtro de Kalman Estendido (EKF)	17
6.3.1	Custo computacional	19
6.3.2	Associação de dados	19
6.3.3	Não-Linearidade	19
6.4	FastSLAM	19
6.4.1	Custo computacional	20
6.4.2	Associação de dados	21
6.4.3	Não-Linearidade	21
7	Arquitetura Proposta	22
7.1	Aquisição de Dados	22

7.2	Extração de Features	23
7.3	Geração de Mapa	23
7.4	Estimação de Estado	23
7.5	Corrêspndencia	23
7.6	Atualização do Mapa	23
8	Cronograma de atividades	24

1 Introdução

São bem sabidos todos os avanços relacionados com área de robótica desde os princípios dos anos 60 quando a revolução industrial colocou os manipuladores industriais a realizar tarefas que até esse momento só eram feitas por os humanos. Posteriormente, à medida que os robôs incorporaram se em outros campos de aplicação surgiram outros tipos de requerimentos para melhorar aspectos como flexibilidade, mobilidade e autonomia.

Robótica Móvel é um termo que descreve um sistema robótico movimentado por elementos locomotivos que provêem a capacidade para realizar diferentes tipos de tarefas em diversas áreas de aplicação [1].

Um robô móvel pode ser considerado como um sistema constituído por diferentes etapas que em conjunto garantem o bom desempenho das tarefas a serem realizadas. Na literatura geralmente são consideradas três etapas constitutivas: *percepção*, *localização e mapeamento*, e *navegação*. Cada uma destas etapas são áreas de estudo bem definidas que tem produzido consideráveis aportes na evolução de técnicas que tentam resolver cada um dos problemas relacionados.

Um dos aspectos onde têm sido concentrados maiores esforços nas ultimas duas décadas de pesquisa é o problema da autonomia do robô móvel. Um robô autônomo é aquele que seja capaz de tomar uma serie de decisões (com base na informação fornecida previamente) com relação a uma tarefa específica, sem necessidade de instruções de controle externas.

Para garantir a autonomia na etapa de navegação do robô é necessário que este conheça bastante bem seu ambiente de trabalho assim como a localização nele em todo momento. Portanto, a localização e o mapeamento convertem-se em uma parte fundamental para resolver o problema de autonomia.

O laboratório de robótica móvel do Grupo de Automação e Controle (GRACO) da UnB possui uma plataforma móvel XR4000 construída pela Nomadic Technologies que tem sido utilizada para a realização de diferentes trabalhos tanto de conclusão de curso de formatura quanto dissertações de mestrado.

Com o fim de realizar projetos de atualidade científica, o GRACO está interessado em desenvolver projetos relacionados com mapeamento, localização e navegação autônoma de robôs móveis que possam ser validados na plataforma móvel XR4000.

O objetivo deste documento é apresentar uma metodologia para abordar o problema de localização e mapeamento de um robô móvel dentro de um ambiente interno ou "indoor". A abordagem apresentado está enfocado na aplicação de algoritmos probabilísticos que resolvem o problema de localização e mapeamento mediante a representação das incerteza do sistema em distribuições de probabilidade.

Também será abordado o problema de mapeamento e localização simultâneo ou SLAM, por suas siglas em inglês (Simultaneous Localization and mapping), onde o robô sem informação inicial nenhuma faz o mapeamento do ambiente, ao mesmo tempo em que se localiza dentro dele.

2 Descrição do trabalho de pesquisa

2.1 Título

Técnicas Probabilísticas de Localização e Mapamento aplicadas à Robótica Móvel

2.2 Orientador do projeto

Alberto José Alvares, Dr. Eng.

2.3 Autor do projeto

Claudia Patricia Ochoa Díaz

2.4 Entidades interessadas na realização do projeto

Universidade de Brasília - Grupo de Automação e Controle, GRACO

3 Justificativa

O interesse neste tema nasce da necessidade de brindar autonomia aos robôs móveis à medida que estes atingem uma tarefa específica.

O problema de mapeamento e localização tem sido uma das principais preocupações dos pesquisadores na área de robótica móvel nas últimas duas décadas, e tem sido formulado e resolvido de múltiplas maneiras, algumas destas estão baseadas em técnicas probabilísticas.

A *Robótica Probabilística* é uma subárea da robótica que tem como base teórica as técnicas estatísticas para representar informação e tomar decisões. Tem como objetivo procurar a melhor representação das incertezas próprias do ambiente de trabalho onde o robô está inserido.

No caso específico do problema de localização as técnicas probabilísticas utilizadas baseiam-se na regra de Bayes e sua extensão conhecida como *filtros Bayesianos*. As técnicas diferenciam-se basicamente na representação da distribuição de probabilidade da crença inicial que o robô tem acerca de sua posição no espaço. Estas representações podem ser *paramétricas*, onde a função de distribuição do conhecimento é modelada a partir de parâmetros que a definem. As representações *não paramétricas*, por outro lado, utilizam amostras da distribuição para construir a probabilidade posterior ou crença.

O *filtro estendido de Kalman* (EKF), é uma técnica paramétrica bastante utilizada para resolver o problema de localização. Os parâmetros que definem a distribuição posterior (ou crença da posição) são a média e a variância.

Entretanto, uma técnica não-paramétrica bastante utilizada devido à facilidade em sua implementação é a técnica de *localização de Monte Carlo* baseada em filtro de partículas. Aqui, a distribuição posterior é modelada como um conjunto de amostras ponderadas (partículas).

Já no problema de localização e mapeamento simultâneo as soluções propostas são o *filtro de Kalman estendido* ou EKFSlam e ou *Filtro "Rao-Blackwellized"* ou FastSLAM, baseado em filtro de partículas. Vistos como implementação em algoritmos, cada uma destas técnicas

apresentam vantagens e desvantagens relacionadas com seu desempenho, especificamente em aspectos como convergência, custo computacional, correspondência entre dados, entre outros.

Os trabalhos feitos anteriormente com a plataforma NOMAD XR4000 no laboratório do GRACO abordaram os problemas de localização, mapeamento e navegação com outros tipos de técnicas diferentes às probabilísticas, dando um ponto de partida para a implementação deste tipo de abordagem como solução alternativa tanto ao mapeamento e localização quanto a navegação; com a diferença que estas técnicas tentam brindar autonomia no desempenho do robô móvel.

4 Exposição do problema

Como já foi mencionado antes, a etapa de localização e mapeamento é talvez um dos componentes no esquema geral de robótica móvel que tem recebido maior interesse dentro da comunidade pesquisadora nas últimas duas décadas. Como resultado, avanços significativos tem-se apresentado nesta área.

A etapa de localização e mapeamento é uma ponte entre a percepção e a navegação e seu principal objetivo é a determinação da posição do robô dentro do ambiente, o qual é possível através das informações coletadas pelos sensores (proprioceptivos e exteroceptivos) e convertidas em modelos e mapas.

Na figura (1) apresenta-se um esquema geral do problema de localização. Como entradas têm-se os dados fornecidos pelo sensor proprioceptivo (e.g encoders e giroscópio), o mapa do ambiente (este pode ser construído previamente ou simultaneamente à localização) e as observações tomadas pelo sensor exteroceptivo (e.g sonar, radar laser, câmera). Os dados do sensor proprioceptivo e do mapa são usados para prever a posição do robô relativa ao ambiente (odometria), que posteriormente é comparada com as observações feitas pelo sensor exteroceptivo. Se as observações correspondem com a predição da posição, então a posição do robô é atualizada. Na próxima iteração, o dado calculado corresponde ao dado no instante anterior utilizado para calcular a estimativa de posição atual.

Por outro lado, a navegação baseada em mapeamento (*maped-based navigation*) é uma abordagem que integra em uma arquitetura as etapas de localização (e mapeamento) e navegação, onde o robô constrói seu próprio mapa e navega dentro dele até atingir seu objetivo. O sucesso desta arquitetura se concentra na representação interna do ambiente feita pelo robô, se esta representação diverge muito da realidade, o comportamento será indesejado mesmo assim que as leituras dos sensores não estejam afetadas por erros sistemáticos nem aleatórios (no caso hipotético).

Além da representação do mapa, o robô deve ter uma representação de incerteza ao redor da sua posição, pois ele não é capaz de identificar uma única posição com sua posição atual, em cambio, identifica um numero de possíveis posições. Uma das alternativas para esta representação é por meio de funções de distribuição de probabilidade, onde, por exemplo, a

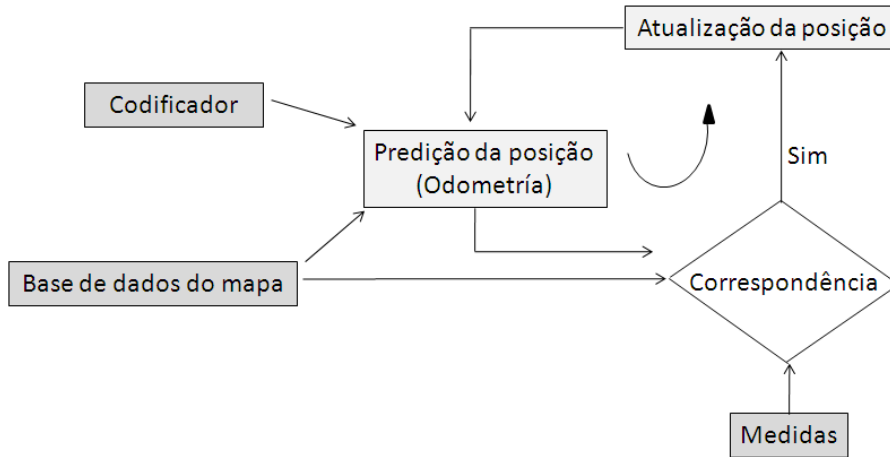


Figura 1: Esquema geral do problema de Localização(Adaptado de [1])

posição de robô pode estar representada por uma coordenada (x,y) que representa a média das possíveis posições acompanhada de uma incerteza representada pelo desvio padrão σ , definindo assim uma distribuição de probabilidade normal, onde em cada posição é representada a probabilidade que o robô realmente este localizado nessa área.

A vantagem deste tipo de representação é que o robô só precisa de informação parcial relacionada com sua posição e a partir deste conhecimento "*a priori*"pode prever, com certo nível de incerteza, sua localização atual. Esta representação permite que um robô com um sistema sensorial limitado possa navegar "robustamente"em diferentes ambientes.

Mas o problema da localização não é só questão de determinar a posição do robô no entorno, também inclui a correta localização dos obstáculos e demais componentes do ambiente de maneira que o robô (ou programador) possa calcular a melhor trajetória para chegar até a posição objetivo.

Tudo o anteriormente exposto tem como fim apresentar o abordagem do problema de localização e mapeamento através das técnicas baseadas em probabilidade. A aplicação de técnicas de estimação no problema de localização e mapeamento foi introduzida no final da década dos 80 princípios dos 90. A partir deste momento, a produção de pesquisas rela-

cionadas com este tema produziu avanços significativos permitindo estabelecer uma base de conhecimento que foi incrementando se até o dia de hoje.

Considerando o problema de mapeamento e localização com uma unidade, os esforços estiveram concentrados na solução do problema de localização e mapeamento simultâneo (*simultaneous localization and mapping*) ou SLAM. O problema de SLAM se origina da situação onde o robô não possui um mapa prévio do ambiente e também não conhece sua própria postura, toda a informação que ele possui são as leituras dos sensores, $z_{1:t}$, e as entradas de controle, $u_{1:t}$ (velocidade translacional e velocidade angular). Basicamente, o processo do SLAM tenta construir um mapa do ambiente e simultaneamente deduzir a postura da plataforma móvel por meio de informação sensorial.

Neste trabalho de pesquisa pretende-se desenvolver uma arquitetura de localização e mapeamento, considerando na etapa inicial os dois problemas por separado, para fusionar os dois conhecimentos para resolver o problema de SLAM. A arquitetura resultante será validada na plataforma XR4000 do laboratório do GRACO.

5 Objetivos do trabalho de pesquisa

5.1 Objetivo Geral

Desenvolver uma arquitetura de localização e mapeamento mediante a aplicação de algoritmos probabilísticos de estimação, validando estes na plataforma móvel Nomad XR4000.

5.2 Objetivo Específicos

- Implementar uma técnica de mapeamento para um ambiente interno, fazendo a escolha de uma representação adequada segundo as características do espaço de trabalho de robô.
- Simular os métodos de localização baseados em filtro de Kalman e filtro de partículas mediante dados extraídos do robô.
- Analisar comparativamente os desempenhos dos algoritmos localização estudados e simulados. Avaliar aspectos como convergência, custo computacional e implementação.
- Validar a técnica de localização escolhida no Nomad XR4000 inserido dentro do ambiente utilizado para a etapa de testes.
- Estudar o problema de localização e mapeamento simultâneo, SLAM, a partir do conhecimento adquirido no estudo dos problemas de localização e mapeamento considerados separadamente na etapa anterior.

6 Revisão da Literatura

6.1 Localização

O problema de localização consiste em determinar a postura do robô relativa ao um mapa do ambiente. Localização também é conhecida como um problema de estimação de postura [2].

Basicamente o problema de localização consiste em inferir a postura do robô a partir de um mapa fornecido previamente (em alguns casos é construído simultaneamente) e de percepções sensoriais do ambiente. Uma vez conhecida a localização do robô dentro do mapa ele executara uma ação de controle segundo a tarefa a ser atingida.

A localização também pode ser vista como um problema de *transformação de coordenadas*. Os mapas são descritos em um sistema global de coordenadas, que é independente à postura do robô. Desde este enfoque, o problema de localização resulta em estabelecer correspondência entre o sistema de coordenadas do mapa e o sistema local do robô.

6.1.1 Classificação do problema de localização

O problema de localização pode ser dividido segundo:

1. Natureza do ambiente.

- Ambiente estático: a única variável (ou vetor de variáveis) é a postura do robô. Todos demais objetos do ambiente são estáticos. Os ambientes estáticos possuem características matemáticas que permitem utilizar métodos de estimação bastante eficientes.
- Ambientes dinâmicos: possui objetos além do robô que se movimentam dentro do ambiente, quer dizer, suas posições variam constantemente. Os ambientes reais são dinâmicos com câmbios de estados que ocorrem a diferentes velocidades.

2. Conhecimento inicial que o robô possui.

- Seguimento de posição: assume que o robô conhece sua postura inicial. A incerteza é aproximada a uma distribuição unimodal. O problema de seguimento de posição é um problema local porque a incerteza é local e limitada a uma região perto da localização real do robô.
- Localização Global: a posição inicial do robô é desconhecida. A incerteza da posição do robô não pode ser modelada mediante uma função de distribuição multimodal.

3. Controle o não do movimento do robô

- Localização passiva: o modulo de medição só observa a operação do robô. O robô pode, por exemplo, movimentar-se aleatoriamente.
- Localização ativa: o modulo de localização controla os movimentos do robô para minimizar os erros de localização.

6.1.2 Método de localização de Markov

As diferentes técnicas probabilísticas não são nada mais que variações do filtro bayesiano. A *localização de Markov* é o nome da aplicação do filtro bayesiano dentro do problema de localização.

Ao igual que o filtro Bayesiano, a localização de Markov precisa de uma distribuição de probabilidade que represente o conhecimento inicial que o robô possui, também conhecida como crença $bel(x)$. Adicionalmente, o comando de controle u_t , a medida feita z_t e o mapa m , são consideradas como entradas.

A continuação serão explicadas as duas etapas constitutivas da localização de Markov:

- Etapa de predição: a partir do conhecimento inicial (no tempo $t - 1$) e por meio do modelo de movimento do robô (especifico para cada tipo de robô) é obtida a crença no tempo t . Expressado matematicamente como:

$$\overline{bel}(x_t) = \int \overbrace{p(x_t|u_t, x_{t-1})}^{\text{modelo de mov.}} \underbrace{bel(x_{t-1})}_{\substack{\text{crena} \\ \text{inicial}}} dx \quad (1)$$

- Etapa de atualização da medida: nesta etapa é incorporada a medida do sensor no tempo t e o $\overline{bel}(x_t)$ calculado na etapa anterior, obtindo assim a crença $bel(x_t)$:

$$bel(x_t) = \eta \underbrace{p(z_t|x_t, m)}_{\text{modelo de medida}} \overline{bel}(x_t) \quad (2)$$

6.1.3 Filtro Estendido de Kalman (EKF)

O filtro estendido de Kalman é uma variação do filtro de Kalman original. Aqui são consideradas as não linearidades próprias dos modelos de movimento e de medida.

O EKF utiliza técnicas de linearização como a aproximação por series de Taylor para modelar a crença $bel(x)$ por meio de dois parâmetros, a média μ é a covariância Σ .

Na formulação do problema SLAM, o EKF será explicado de forma mais detalhada.

6.1.4 Filtro de Partículas

O filtro de partículas, também conhecido como *localização de Monte Carlo* aproxima a probabilidade posterior ou crença por meio de um numero finito de partículas, que são um conjunto aleatório de amostras geradas precisamente da probabilidade posterior $bel(x)$. Cada partícula representa a hipótese do valor do estado no tempo t . Este conjunto é denotado matematicamente como:

$$\chi_t = x_t^1, x_t^2, \dots, x_t^M$$

Onde M é o numero de partículas gerado.

6.2 Mapeamento

Para processar a informação proveniente dos sensores é necessário especificar o ambiente onde são geradas as medições. O *mapa* do ambiente é uma lista dos objetos dentro do ambiente com suas respectivas posições, isto é:

$$m = \{m_1, m_2, \dots, m_N\} \quad (3)$$

O N é o número total de objetos dentro do ambiente, e cada m_n com $1 \leq n \leq N$ especifica uma propriedade de cada um.

Geralmente os mapas podem ser gerados baseados em features (*feature-based mapping*) e baseados em posição (*location-based mapping*). No primeiro caso, n é um indicador da *feature* e o valor de m_n contém a posição específica da *feature* em coordenadas cartesianas. Nos mapas baseados em posição o indicador n corresponde a uma posição específica, geralmente uma coordenada (x, y) .

Ambos tipos de mapas possuem vantagens e desvantagens relacionadas com sua aplicação final. Os mapas baseados em posição contêm informação de qualquer posição do espaço, isto é, não só contêm informação dos objetos dentro do ambiente, também contêm informação da área livre dentro dele, por isto são preferidos nas tarefas de navegação.

Entanto, os mapas baseados em *features* só especificam a aparência do ambiente, por esta razão são mais apropriados para as tarefas de localização. A continuação será explicada este tipo de técnica de mapeamento.

6.2.1 Extração de Features

Em uma tarefa de exploração de um ambiente, os sensores armazenam uma quantidade enorme de dados do ambiente. Este conjunto de informação é chamado de *dados brutos*.

Uma alternativa para aproveitar a informação útil destes dados brutos é extrair só os dados relacionados com as *features* presentes no ambiente. Isto representa uma redução importante na complexidade computacional na hora de fazer o processamento destes dados.

Os tipos de *features* variam de acordo com o sensor utilizado, por exemplo, no caso de sensores de ultra-som é comum extrair linhas e esquinas, que podem corresponder a paredes ou objetos bem definidos.

Em algumas aplicações de robótica, as features podem corresponder a objetos específicos dentro do ambiente, por exemplo, portas ou janelas, no caso de ambientes internos. Estes tipos de features são conhecidos como *landmarks*, utilizados geralmente nas tarefas de navegação.

Por outro lado, as leituras dos sensores de ultra-sôm provem a distância e a orientação dos objetos que são detectados com respeito ao sistema de referencia local do robô, o que os fazem apropriados para detectar objetos no ambiente. A função de extração $f(z_t)$ que seja definida para a detecção de landmarks pode aproveitar a informação de distância e orientação proveniente do sensor. Adicionalmente, a função de extração pode assinar uma identificação s_t a cada feature. Finalmente, o conjunto ou vetor de features quedaria conformado da seguinte maneira:

$$f(z_t) = \{f_t^1, f_t^2, \dots\} = \left\{ \begin{bmatrix} r_t^1 \\ \phi_t^1 \\ s_t^1 \end{bmatrix}, \begin{bmatrix} r_t^2 \\ \phi_t^2 \\ s_t^2 \end{bmatrix}, \dots \right\} \quad (4)$$

Onde r é a distância entre o sensor e o objeto detectado, ϕ é a orientação com respeito ao sistema local do robô e s é a identificação de cada *landmark*.

6.2.2 Formulação e estrutura do problema SLAM

Basicamente, o processo do SLAM consiste em construir um mapa do ambiente e simultaneamente deduzir a postura do robô dentro dele. O mapa está conformado por os objetos presentes no ambiente e que são detectados por o sistema sensorial do robô na medida em que este se movimentar. A localização pode ser feita pelas técnicas enunciadas nas seções anteriores.

Antes de expor formalmente o problema de SLAM, é importante familiarizar-se com a notação que será utilizada durante toda a análise. Esta notação é a mesma que foi utilizada em [3], por tanto, de aqui na frente o tempo será discretizado, representado pela variável k

Considere-se um robô que se está movimentando dentro de determinado ambiente e que está tomando observações de um numero desconhecido de landmarks utilizando seu sistema sensorial. A figura 2 ilustra a situação.

No instante k , as seguintes variáveis são definidas:

- x_k : o vetor de estado do robô, que é a postura definida pela posição (x, y) e rotação θ .

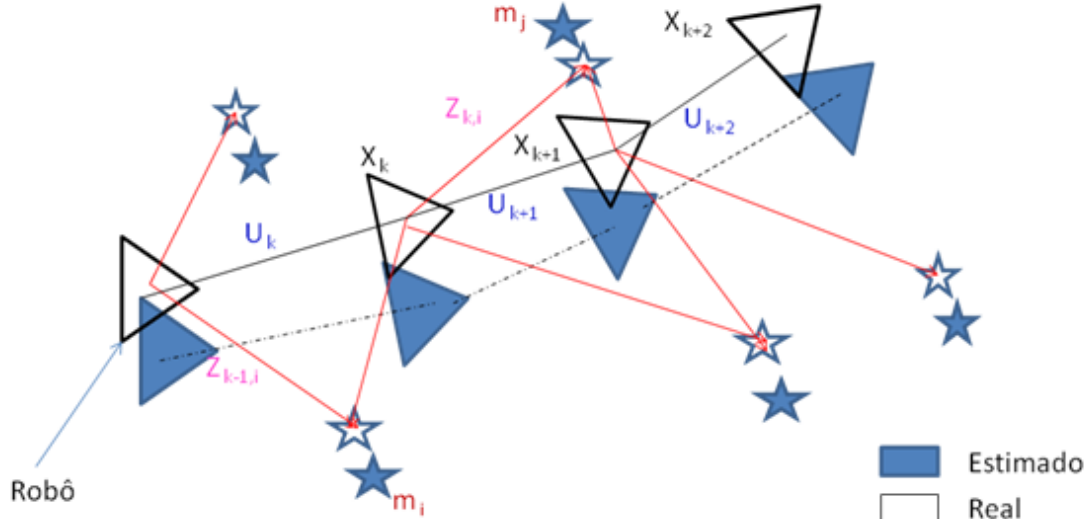


Figura 2: Ilustração do problema SLAM

- u_k : vetor de controle. Para maioria dos robôs este vetor está conformado por a velocidade translacional (v_k) e a velocidade rotacional (ω_k).
- m_i : vetor que contem a posição da i -ésima landmark observada pelo robô. Estas posições são assumidas invariantes no tempo.
- $z_{k,i}$: medição da i -ésima landmark tomada desde o robô no instante k. Quando varias medições de vários objetos são feitas no mesmo instante de tempo o vetor só é representado como z_k .

O problema do SLAM consiste em estimar a probabilidade conjunta da postura do robô e a posição das landmarks no instante k, dadas as observações tomadas, as entradas de controle e o estado inicial x_0 . Isto pode-se expressar matematicamente como:

$$P(x_k, m | Z_{0:k}, U_{0:k}, x_0) \quad (5)$$

A solução do problema SLAM é recursiva, isto é, a estimativa do estado no tempo k é calculada a partir da estimativa calculada no tempo k-1. A seguir serão apresentados os modelos de movimento e o modelo de medida.

O *modelo de movimento* é a probabilidade da transição do estado k ao estado $k - 1$ dada uma entrada de controle, assim:

$$P(\mathbf{x}_k | \mathbf{x}_{k-1}, \mathbf{u}_k) \quad (6)$$

Entretanto, o *modelo de medida* descreve a probabilidade de fazer uma medição z_k a partir do conhecimento da postura do robô e o mapa, isto é:

$$P(\mathbf{z}_k | \mathbf{x}_k, \mathbf{m}) \quad (7)$$

Desta forma o algoritmo SLAM pode ser estruturado, baseado no filtro Bayesiano, em duas etapas, uma etapa de predição e uma etapa de atualização de medidas.

Etapa de Predição

$$P(x_k, m | Z_{0:k-1}, U_{0:k}, x_0) = \int P(x_k | x_{k-1}, u_k) P(x_{k-1}, m | Z_{0:k-1}, U_{0:k-1}, x_0) dx_{k-1} \quad (8)$$

Etapa de Atualização da Medida

$$P(x_k, m | Z_{0:k}, U_{0:k}, x_0) = \frac{P(z_k | x_k, m) P(x_k, m | Z_{0:k-1}, U_{0:k}, x_0)}{P(z_k | Z_{0:k-1}, U_{0:k-1})} \quad (9)$$

Deste modo o algoritmo calcula a postura do robô e a localização das landmarks simultaneamente de uma forma recursiva. A figura (3) apresenta um esquema que interpreta de maneira gráfica o funcionamento do algoritmo SLAM.

Voltando à figura 2, pode se observar que o erro de posição das landmarks detetadas é o mesmo para todas (i.e. a distância entre a posição estimada e a real é a mesma para todas as landmarks) devido a que a fonte de erro se mantém constante para cada cálculo do novo estado.

A relação entre os erros de posição de um par de landmarks implica que existe uma correlação bem forte entre suas posições respectivas, e que na medida em que novos estados sejam calculados a correlação vai ser maior. No limite, os erros nas estimativas de posição de um par qualquer de landmarks m_i, m_j serão completamente correlacionados, isto é se a posição de m_i é conhecida de modo exato, então a posição de m_j pode ser calculada com total certeza.

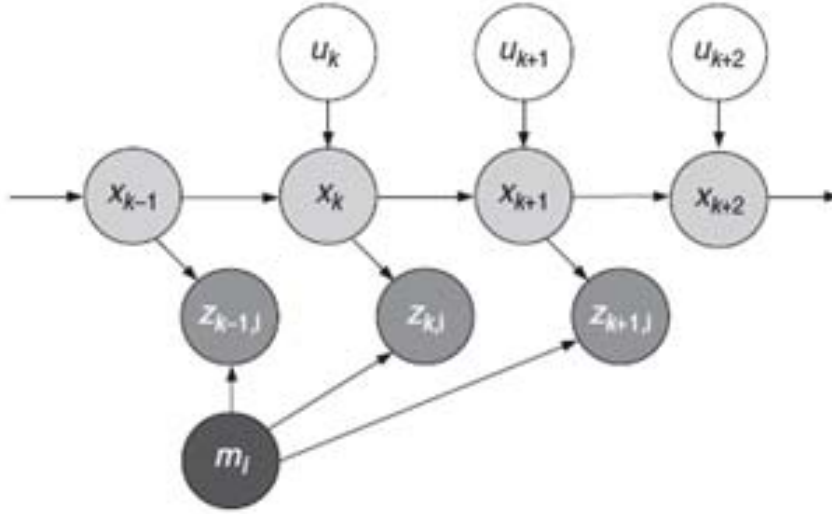


Figura 3: Ilustração do problema SLAM

Finalmente e a maneira de conclusão, a solução ao problema do SLAM desde o enfoque probabilístico é encontrar a representação mais apropriada para os modelos de movimento e de medida que permitam um cálculo eficiente das probabilidades que serão utilizadas nas etapas de predição e correção respectivamente.

6.3 Filtro de Kalman Estendido (EKF)

O algoritmo EKF- SLAM descreve o modelo de movimento da seguinte forma:

$$P(x_k|x_{k-1}, u_k) \leftrightarrow x_k = \mathbf{f}(x_{k-1}, u_k) + w_k \quad (10)$$

Onde $\mathbf{f}(\cdot)$ representa o modelo cinemático do veículo e w_k é o ruído gaussiano não correlacionado de média zero e covariância Q_k que representa as perturbações devidas ao movimento do robô.

Entretanto, o modelo de medida é:

$$P(z_k|x_k, m) \leftrightarrow z_k = \mathbf{h}(x_k, m) + v_k \quad (11)$$

Onde $\mathbf{h}(\cdot)$ descreve a geometria da medida (que relaciona a saída do sensor, z_k , com o vetor de estados, x_k) e v_k é o erro de medida gaussiano não correlacionado de média zero e

covariância R_k que representa as perturbações devidas ao movimento do robô.

O algoritmo aplica ambos modelos para calcular as médias do vetor de estados do robô e do vetor das posições dos objetos, assim como a matriz de covariância entre eles. As expressões destes parâmetros são:

$$\begin{bmatrix} \hat{x}_{k|k} \\ \hat{m}_k \end{bmatrix} = E \left[\begin{bmatrix} x_k \\ m \end{bmatrix} | Z_{0:k} \right] \quad (12)$$

e covariância igual a :

$$\mathbf{P}_{k|k} = \begin{bmatrix} \mathbf{P}_{xx} & \mathbf{P}_{xm} \\ \mathbf{P}_{xm} & \mathbf{P}_{mm} \end{bmatrix}_{k|k} \quad (13)$$

O passo seguinte consiste na execução das etapas de predição e de atualização de medida.

Predição

$$\begin{aligned} \hat{\mathbf{x}}_{k|k-1} &= \mathbf{f}(\hat{\mathbf{x}}_{k-1|k-1}, \mathbf{u}_k) \\ \mathbf{P}_{xx,k|k-1} &= \nabla \mathbf{f} \mathbf{P}_{xx,k-1|k-1} \nabla \mathbf{f}^T + \mathbf{Q}_k \end{aligned} \quad (14)$$

Onde $\nabla \mathbf{f}$ é o jacobiano de \mathbf{f} avaliado no estimador $\hat{\mathbf{x}}_{k-1|k-1}$.

Atualização de medida

$$\begin{aligned} \begin{bmatrix} \hat{x}_{k|k} \\ \hat{m}_k \end{bmatrix} &= \begin{bmatrix} \hat{x}_{k|k-1} \\ \hat{m}_{k-1} \end{bmatrix} + \mathbf{W}_k \left[\mathbf{z}_k - \mathbf{h}(\hat{x}_{k|k-1}, \hat{m}_{k-1}) \right] \\ \mathbf{P}_{k|k} &= \mathbf{P}_{k|k-1} - \mathbf{W}_k \mathbf{S}_k \mathbf{W}_k^T \end{aligned} \quad (15)$$

onde

$$\begin{aligned} \mathbf{S}_k &= \nabla \mathbf{h} \mathbf{P}_{k|k-1} \nabla \mathbf{h}^T + \mathbf{R}_k \\ \mathbf{W}_k &= \mathbf{P}_{k|k-1} \nabla \mathbf{h}^T \mathbf{S}_k^{-1} \end{aligned} \quad (16)$$

Onde $\nabla \mathbf{h}$ é o jacobiano de \mathbf{h} avaliado em $\hat{\mathbf{x}}_{k|k-1}$ e $\hat{\mathbf{m}}_{k-1}$.

6.3.1 Custo computacional

A etapa de atualização de medida requer que todos as landmarks detectadas e todas as matrizes de covariância conjuntas sejam atualizados cada vez que uma observação seja feita, isto significa que com cada observação o numero de cálculos se acrescenta quadraticamente dependendo do numero de landmarks.

6.3.2 Associação de dados

O algoritmo EKF-SLAM é bastante frágil na associação de dados, isto é, se o robô volta ao um ponto onde faz uma observação de uma landmark que já foi identificado previamente, é provável que esta medida não seja associada com esta landmark e que a identifique como um uma nova.

6.3.3 Não-Linearidade

O EKF-SLAM utiliza faz uso de técnicas de linearização para calcular os parâmetros que modelam a distribuição posterior, o que pode originar algumas inconsistências nas estimativas dos estados.

6.4 FastSLAM

O algoritmo FastSLAM está baseado numa característica muito importante do problema SLAM, que é independência condicional que existe entre dois conjuntos diferentes de features no mapa, dada a postura do robô. Em outras palavras, se a trajetória real do robô fosse conhecida, a estimação da posição de todas as features do mapa poderia ser feita de modo independente entre cada uma.

Isto faz possível que seja implementado uma versão do filtro de partículas ao problema SLAM, o filtro de partículas Rao-Blackwellized, o qual usa um conjunto de partículas para representar a probabilidade posterior dado algumas variáveis, ajudado de distribuições gaussianas para representar outras variáveis do problema.

A técnica FastSLAM usa o filtro de partículas para estimar a trajetória do robô, onde cada partícula $x_k^{[i]}$ contem na estimativa de um estado, assim como a estimativa de cada fea-

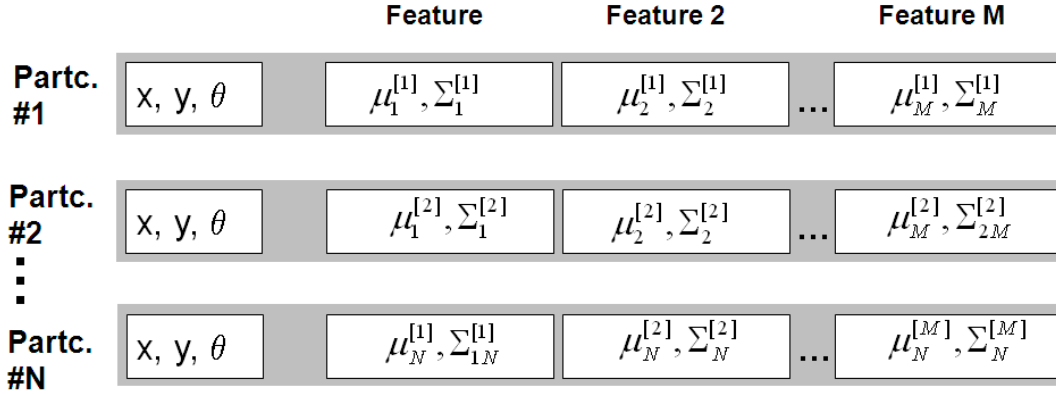


Figura 4: Ilustração do FastSLAM

ture presente no mapa, que são modeladas cada uma por distribuições gaussianas de média $\mu_{j,k}^{[z]}$ e covariância $\Sigma_{j,k}^{[z]}$. Aqui o k representa uma partícula de um total de M .

Deste modo o FastSlam pode ser visto com uma combinação entre o filtro de partículas (usada na etapa de predição) e o EKF, onde cada medida é atualizada mediante o calculo de seus parâmetros μ e Σ

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

$$P(X_{0:k} | Z_{0:k}, U_{0:k}, x_0) = P(X_{(0:k)} | Z_{0:k}, U_{0:k}, x_0) \prod_{m=1}^M P(m_m | X_{(0:k)}, Z_{(0:k)}) \quad (17)$$

Observe-se que no caso do FastSLAM o calculo é sobre a trajetória completa . Devido a isto, as posições dos objetos convertem se independentes e podem ser representadas como funções gaussianas individuais.

A figura (4) apresenta as partículas na técnica FastSLAM.

6.4.1 Custo computacional

O algoritmo básico do FastSLAM pode ser implementado em um tempo algoritmico no número de features, por tanto esta representa menor custo computacional em sua implementação comparada com o EKF.

6.4.2 Associação de dados

A vantagem chave do FastSLAM está no fato que as decisões relacionadas com a associação de dados podem ser feitas sobre cada partícula, mantendo as probabilidades posteriores sobre todas as associações feitas, não só para as mais parecidas.

6.4.3 Não-Linearidade

A principal diferença com respeito ao algoritmo EKF é a possibilidade de representar processos não lineares sem necessidade de aplicar técnicas de linearização (embora o modelo de medida seja ainda linearizado)

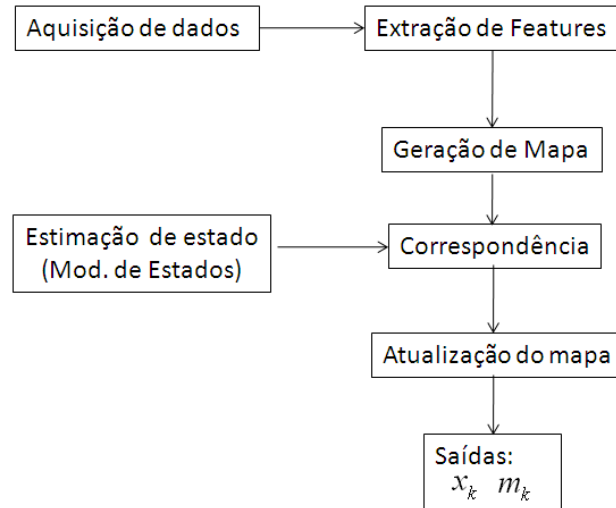


Figura 5: Esquema da arquitetura proposta

7 Arquitetura Proposta

A arquitetura proposta está baseada na revisão de trabalhos onde tratam os problema de localização e mapeamento por separado e a integração deles por meio do FastSLAM ([4], [5], [6]). Também são considerados aspectos diretamente relacionados com os sensores ([7], [8]), o ambiente de trabalho, e a representação de mapa, que é neste caso baseada em features.

Na figura (??) é apresentado um esquema da arquitetura proposta. A seguir são explicadas cada uma delas.

7.1 Aquisição de Dados

Nesta etapa serão feitas as medidas do ambiente por meio dos sensores de ultra-som é infravermelho. Ainda não se tem considerado uma técnica para fusão de dados provenientes dos sensores.

7.2 Extração de Features

A partir dos dados brutos, mas fusionados, pretende-se extrair algumas features do ambiente como podem ser linhas e esquinas que representem algumas das estruturas presentes no ambiente. Como já foi mencionado em seções de localização e mapeamento, a extração de features é a técnica de mapeamento mais utilizada no problema de localização, porque o robô só precisa conhecer a aparência do ambiente e não uma informação detalhada de todos espaços livres, como é o caso de planejamento de trajetória.

7.3 Geração de Mapa

AA partir das medições das features feitas no estado atual, será elaborado um mapa atual que contem as informações de posição de cada uma das features detectadas por o sistema sensorial. Cada feature deve precisar de uma identificação que a diferença das demais. A identificação será útil na etapa de correspondência que posteriormente atualizará o mapa se a correspondência é correta.

7.4 Estimação de Estado

Nesta etapa a informação proveniente dos codificadores junto com o modelo cinemático do robô se juntam para estimar o estado do modelo atual. Esta estimativa leva em conta os erro propios dos cálculos por odometria que vão ser integrando em cada cálculo ao longo da trajetória feita por o robô.

7.5 Corrêspendencia

A postura do robô será comparada com as estruturas provenientes do mapa gerado a partir dos dados atuais de maneira de calcular uma postura atual definitiva. Esta etapa no momento não esta muito bem definida. É necessário fazer uma revisão mais aprofundada de correspondência de dados e estimação da postura definitiva a partir de mapeamento baseado em features.

7.6 Atualização do Mapa

Finalmente, se a etapa de correspondência dá um resultado certo, então o mapa gerado no estado anterior será atualizado com as features detectadas no estado atual. Se as features

detectadas já estiverem no mapa anterior, então estas serão atualizadas e as posições delas dentro do mapa serão corrigidas, se fosse o caso. A postura definitiva será utilizada para efetuar o processo de localização de robô dentro do ambiente.

8 Cronograma de atividades

Na figura (6) é descrito em forma detalhada o conjunto de atividades que serão desenvolvidas até o final do projeto. As tarefas em conjunto estão relacionadas com o modelagem sensorial a definição e a construção do ambiente de teste.

Referências

- [1] E. García, M. Jimenez, P. Gonzalez, and M. Armada, “The evolution of robotics research,” *IEEE Robotics and Automation Magazine*.
- [2] S. Thrun, W. Burgard, and D. Fox, *Probabilistics Robotics*. MIT Press, 2005.
- [3] H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping:parti,” *IEEE Robotics and Automation Magazine*.
- [4] J. Costa and F. Diaz, “Simultaneous localization and map building by integrating a cache of features,” *IEEE Robotics and Automation Magazine*.
- [5] G. Borges and M. J. Aldon, “Enviroment mapping and robust localization for mobile robots navigation in indoor enviroments.”
- [6] A. Álvares, G. Andriolli, P. Dutra, M. de Sousa, and J. Ferreira, “A navigation and path planning system for the nomad xr4000 mobile robot wiht remote web monitoring,” *ABCM Symposium Series in Mechatronics*, vol. 1.
- [7] G. G. de Souza, “Estudo e aplicação do sonar na robótica móvel,” Dissertação de Mestrado, Universidade de Montpellier II, Montpellier, França, 2002.
- [8] “Nomad xr4000 hardware manual release 1.0,” Nomadic Technologies Inc., Tech. Rep.

Atividade	Duração (Mês/Ano)															
	01/09	02/09	03/09	04/09	05/09	06/09	07/09	08/09	09/09	10/09	11/09	12/09	01/10	02/10	03/10	04/10
Fundamentação Teórica e Revisão Bibliográfica																
Disciplina Filtragem Estocástica																
Estudo por separado de:																
EKF e Filtro de Partículas																
Mapeamento baseado em features																
Fusão sensorial																
Modelagem Sensorial																
Sonares, Infravermelhos e Encoders. (Modelo de Medida)																
Modelo cinemático.																
Definição do Ambiente																
Definição (Tipos de features)																
Construção																
Modelagem da Arquitetura																
Modelagem das classes: Aquisição, estimação de estado, mapeamento, localização																
Análise do modelo proposto (Reuniões de acompanhamento)																
Desenvolvimento da Arquitetura																
Desenvolvimento da classe de localização e mapeamento.																
Teste das classes desenvolvidas.																
Documentação do código-fonte.																
Teste e Avaliação da Arquitetura																
Testes de integração.																
Análise dos resultados.																
Ajustes e correções na arquitetura.																
Documentação																
Redação da Dissertação																
Divulgação de resultados.																
Defesa da dissertação.																

Figura 6: Cronograma de Atividades