

Conteúdo

Lista de Figuras	iii
Lista de Tabelas	v
1 Localização usando representação do ambiente	1
1.1 Representação do ambiente e da crença de localização	3
1.1.1 Mapas métricos	5
1.1.2 Mapas topológicos	6
1.1.3 Representação da estimativa de localização	7
1.2 Técnicas de localização	9
1.3 Estimação Bayesiana	10
1.4 Filtro de Kalman	14
1.4.1 Sequência de passos do EKF	17
1.4.2 Associação de dados	24
1.5 Localização Monte Carlo	26
1.5.1 Sequência de passos da Localização Monte Carlo	28
1.6 Considerações Finais	30

Lista de Figuras

1.1	Classificação dos tipos mais comuns de mapas utilizados em robótica, de acordo com seu nível de abstração. (a) Mapa grade de ocupação, representando mapas métricos, no mais baixo nível. As áreas em cinza representam as células vazias da grade, a cor preta representa as células ocupadas e as áreas em branco são desconhecidas. (b) Mapa topológico, que interliga marcos terrestres com os comandos de controle utilizados para se deslocar de um para qualquer outro, extraído de [36].	4
1.2	Crença do robô em sua posição 1D em mapas contínuos e discretos. a) Uma única hipótese em um mapa contínuo, b) múltiplas hipóteses em um mapa contínuo, c) múltiplas hipóteses em um mapa discretizado por uma grade regular e d) múltiplas hipóteses em um mapa topológico. Extraído de [32].	8
1.3	Arquitetura geral do sistema de localização de um robô móvel. . .	10
1.4	Ilustração do modelo determinista de observação comumente utilizado para a localização com o EKF. (a) Exemplo de dados obtidos por um robô a partir de sensores de sonar. O robô é representado pelo círculo cinza, com um traço indicando sua orientação. As marcas em "x" representam os pontos de interesse do mapa, enquanto os segmentos de círculo indicam a área de atuação de cada sonar. (b) As linhas tracejadas representam os dados de distância e orientação do robô em relação aos pontos de interesse calculados a partir da associação dos dados do sonar com o mapa. (c) No caso do robô não conhecer sua localização corretamente, pode realizar associações erradas. As associações entre medições (representadas por quadrados) e ponto de interesse no mapa estão circuladas. Observe que a associação com o ponto em l_3 deveria ter sido feita com o ponto em l_4	25

1.5 Ilustração do funcionamento da Localização Monte Carlo. Os pontos negros representam as amostras da localização do robô em cada instante, embora as amostras das orientações não estejam graficamente representadas. A flecha cinza representa a posição e a orientação corretas do robô em cada instante. (a) Inicialmente, a crença sobre a localização do robô está uniformemente distribuída em todo o ambiente. (b) Após mover-se um pouco a localização do robô ainda é incerta, mas as medições realizadas já permitem definir regiões com maior probabilidade. (c) Após o robô se mover até a próxima quina, ainda não é possível resolver com certeza sua localização, pois qualquer uma das quatro quinas ainda são localizações plausíveis. (d) Após mover-se até a entrada do corredor, as ambiguidades sensoriais são resolvidas e as partículas concentram-se em uma única localidade. Inspirada no experimento relatado em [34]. 29

Lista de Tabelas

Localização usando representação do ambiente

Anna Helena Reali Costa and Antonio Henrique Pinto Selvatici

A tarefa de navegação de um robô móvel envolve a competência fundamental de se movimentar por um ambiente, evitando colisões e visando atingir localidades específicas para o cumprimento de uma determinada tarefa. Para isso, o robô deve fazer uso de seus atuadores, para que a locomoção seja executada, e de seus sensores, para que tenha indicações, por exemplo, do sentido e velocidade de seu movimento, da aproximação do alvo e dos obstáculos do ambiente.

A navegação do robô pode, muitas vezes, ser executada com sucesso sem o uso de representações explícitas do ambiente, do alvo e do próprio robô neste ambiente. Neste caso, não há necessidade de um sistema de localização, e arquiteturas baseadas em comportamentos reativos são utilizadas no projeto do robô, como, por exemplo, a arquitetura de subsunção [8] e a arquitetura baseada em campos potenciais [3]. Esta abordagem para a navegação apresenta a vantagem de permitir uma implementação rápida e simples, onde diversos comportamentos primitivos são projetados, atuando de forma coordenada em ambientes não muito extensos e com poucas posições como alvos. A dificuldade desta abordagem recai justamente na escolha dos comportamentos primitivos e na adequada sintonia entre eles, para que, em conjunto, executem a tarefa com sucesso. Ainda, a inserção de um novo comportamento na arquitetura pode obrigar o projetista a reprojeter outros comportamentos e a coordenação entre eles.

Devido a estas restrições, quando o robô deve interagir com um ambiente extenso e complexo, executando tarefas mais elaboradas, as arquiteturas mais frequentemente usadas utilizam representações internas como mapas, um sistema de localização e um sistema de cognição e de planejamento de trajetórias para que o robô possa, utilizando sua localização e uma posição como alvo, determinar a rota a seguir e a sequência de comandos de controle que deve executar para atingir o alvo, evitando colisões e outras eventuais restrições. A vantagem do uso de uma abordagem baseada em mapas é que estas representações são meios adequados de comunicação entre robôs e humanos, tornando disponível a informação a respeito da crença de localização do robô neste mapa. Entretanto, sua desvantagem reside justamente no uso de uma representação interna na qual o sistema se fia; assim, caso haja alguma divergência entre o modelo e o mundo real propriamente dito, como erros no mapa, o robô pode apresentar comportamentos inadequados.

O sistema de referência usado pelo robô é de suma importância no estabelecimento da relação entre as competências envolvidas na navegação. Define-se um ponto de referência no ambiente real e fixa-se, neste ponto, a origem do sistema de referência utilizado pelo robô. Neste sistema de referência são anotadas não só as posições de itens, marcos e objetos de interesse representados no mapa, mas também a própria crença do robô em relação à sua localização neste mapa, em cada instante. É neste sistema que o robô deverá executar um raciocínio espacial, de forma a estipular sua navegação para executar com sucesso suas tarefas e atingir suas metas.

Entretanto, os sensores do robô efetuam medidas que possuem grande dose de incerteza: as medidas são imprecisas, há ruídos. Da mesma forma, seus atuadores são limitados e ruidosos. Assim, as medidas que o robô efetua a respeito do ambiente e de sua movimentação carregam imprecisões que fazem com que o sistema de referência utilizado internamente pelo robô seja deslocado em relação às reais localizações inicialmente estabelecidas no mundo. Considere, por exemplo, que o robô execute um movimento que deveria colocá-lo alguns centímetros adiante. Ele mede sua velocidade e aceleração e calcula onde deverá estar após a execução de tal movimento. Porém, suponha que ele derrape no solo. Mesmo que suas medidas sejam exatas, ele “acreditará” que está em uma posição diferente daquela em que ele realmente se encontra após a execução do movimento. Quanto mais ele se movimentar, maior será o erro acumulado, pois seu sistema de referência será cada vez mais deslocado em relação ao sistema inicialmente considerado no mundo real. Este problema é bastante complexo e é sobre ele que as técnicas de navegação se debruçam, geralmente representando de forma explícita a incerteza envolvida no processo e usando recursos da teoria da probabilidade para lidar com esta incerteza e

melhorar as estimativas do robô.

Neste capítulo são abordadas formas de representação do ambiente e da crença do robô acerca de sua própria localização, assim como são descritas as técnicas de localização comumente usadas, que são aquelas baseadas na estimação Bayesiana, por apresentarem os melhores resultados na prática, já que permitem explicitar e tratar as incertezas envolvidas no processo.

1.1 Representação do ambiente e da crença de localização

Na navegação baseada em mapas, o robô tenta explicitamente se localizar por meio da coleta e tratamento de dados sensoriais e da atualização da crença de sua localização em relação ao mapa do ambiente onde navega.

Mapas são representações de um ambiente e são instrumentos comuns em várias atividades humanas, como a arquitetura, construção civil, cartografia, aviação, geografia, geologia, atividades militares e outras. Eles expressam sua informação atribuindo valores a certos elementos ali contidos. A natureza desses elementos varia dependendo do tipo de mapa, que pode corresponder a uma em um amplo espectro de possíveis estruturas. Os mapas, na robótica, são utilizados pelo robô para melhorar sua estimativa de localização, confrontando os dados dos sensores com os elementos registrados no mapa em determinadas posições.

Vários tipos de mapas foram propostos na literatura de robótica. O tipo ideal de mapa a ser utilizado por um robô móvel depende da tarefa e do ambiente onde está inserido. Também depende das características do robô, tais como os tipos de sensores que ele tem, bem como a forma com que ele se move e das capacidades computacionais para armazenamento e manipulação da informação.

Mapas utilizados em robótica podem receber nomes diferentes de acordo com os elementos que usam para descrever o ambiente. É possível classificar os tipos básicos de mapas de acordo com o nível de abstração dos elementos utilizados na sua construção. A Figura 1.1 ilustra esta classificação, tendo por base a taxonomia de mapas proposta por [14]. Mapas métricos são aqueles no mais baixo nível de abstração, e seus elementos correspondem a células indicando o valor de alguma característica local. Mapas topológicos ou relacionais estão no outro extremo, apresentando o mais alto nível abstração. Representam localidades conectadas por algum tipo de relacionamento, em geral a adjacência e conectividade entre elas. As localidades, no entanto, não são posições geográficas definidas, mas sim regiões no espaço representadas por um mesmo conceito, como um cômodo de uma casa ou o pátio de uma

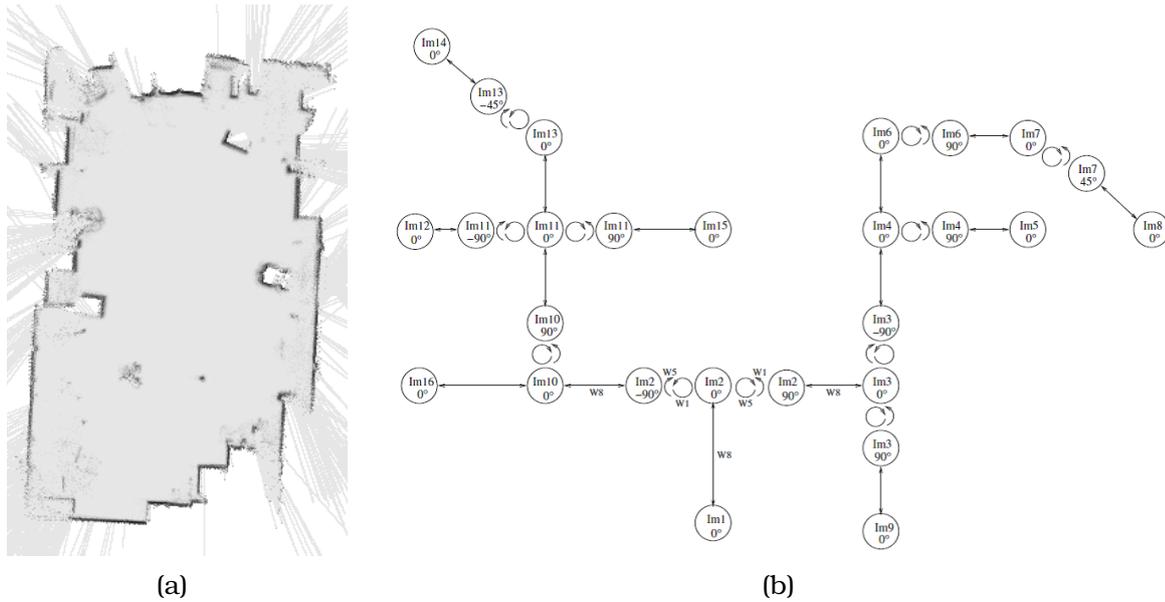


Figura 1.1: Classificação dos tipos mais comuns de mapas utilizados em robótica, de acordo com seu nível de abstração. (a) Mapa grade de ocupação, representando mapas métricos, no mais baixo nível. As áreas em cinza representam as células vazias da grade, a cor preta representa as células ocupadas e as áreas em branco são desconhecidas. (b) Mapa topológico, que interliga marcos terrestres com os comandos de controle utilizados para se deslocar de um para qualquer outro, extraído de [36].

escola. Vale ressaltar, entretanto, que há diversas abordagens intermediárias, que combinam as informações no domínio espacial com informações no nível conceitual.

Apesar de existir uma grande variedade de mapas, cada um adequado a uma determinada tarefa ou ambiente, há uma distinção muito clara entre a finalidade dos mapas métricos e topológicos. No caso de mapas métricos, o objetivo é fornecer informações espaciais detalhadas do ambiente, de onde podem ser inferidos a forma e tamanho dos objetos e os limites das áreas livres para a navegação, tais como corredores, quartos, trilhas e estradas. Para representar essa complexa e detalhada informação, o mapa é geralmente dividido em uma densa rede de modo que cada célula contenha informações sobre a ocupação desse espaço por um objeto e, possivelmente, outras características ambientais que estão sendo mapeadas.

No caso de mapas topológicos, o objetivo é fornecer uma estrutura relacional, geralmente na forma de um grafo, informando que locais o robô pode acessar dado que se encontra em algum outro local. Aqui, entende-se por local uma região no espaço que compartilha uma mesma propriedade que interesse para a tarefa que o robô deve executar. É fácil perceber que as informações contidas no mapa topológico são menos detalhadas e distribuem-se de forma dispersa, concentrando-se apenas em alguns pontos de interesse. Assim, o

mapa topológico é muito seletivo com relação à informação que ele registra. A seguir são descritos os tipos de mapas métricos e topológicos mais comuns.

1.1.1 Mapas métricos

Existem vários tipos de mapas que se mantêm na abordagem métrica, embora envolvam alguma representação de conceitos, em geral com baixo nível de abstração.

Um exemplo bastante difundido de mapa métrico é a grade de ocupação, que divide o espaço 2D ou 3D em células regulares que representam o estado da ocupação do local de interesse, funcionando, por exemplo, como uma planta de uma sala ou pavimento de edifício, indicando os objetos e limitando as áreas onde o robô pode navegar. Um exemplo clássico desse tipo de mapa é a grade de evidência [15], que representa o ambiente como uma matriz de forma que cada célula contém a probabilidade de que esse espaço seja ocupado por um obstáculo para o robô. Este conceito foi estendido para mapas de evidência 3D, que dividem o espaço 3D em células representando a probabilidade de ocupação [24].

Um tipo menos comum de mapa métrico é o mapa sensorial, no qual são registradas as informações dos sensores, brutas ou processadas, que seriam obtidas em diversas localizações do robô. O principal exemplo desse tipo de mapa métrico é o mapa de aspecto visual, o qual representa a imagem capturada pelo robô para cada postura que este possa assumir. Neste caso, cada célula do mapa representa diretamente a informação sensorial que o robô percebe, em vez de uma característica do local do ambiente que não depende da percepção do robô. Um exemplo desse mapa é o proposto por [27], onde as imagens são codificadas pelo Transformada Discreta do Cosseno.

Em geral, mapas de grades necessitam de muito espaço em memória para serem armazenados. Um modo de diminuir essa exigência é condensar informações espaciais em forma de atributos, elementos geométricos que representam características especiais do ambiente. Diferentemente dos mapas de grades, o espaço ambiental não é plenamente representado pelos mapas de características, mas apenas algumas estruturas especiais, tais como quinas, paredes ou objetos.

Os atributos mais comuns utilizados nos mapas para navegação de robô são pontos especiais detectados no ambiente. Atributos pontuais 2D ou 3D foram utilizados em algumas trabalhos seminais [33, 22, 12, 30, 11] e desde então têm sido amplamente adotados. Eles contêm pouca quantidade de informação, especialmente quando detectados por sensores de alcance como os sonares ou varreduras de laser, uma vez que informam somente as posições pontuais no plano de movimentação do robô ou no espaço. Assim, muitos

pontos característicos são necessários para descrever o ambiente com detalhes suficientes para tarefas como a navegação do robô. Se forem detectados pontos nas imagens capturadas pelo robô, alguns descritores locais de imagem podem também ser atribuídos às características mapeadas.

Linhas e planos são outras características geométricas comuns utilizadas no mapeamento robótico [26, 35, 16, 5, 38]. Elas descrevem estruturas mais complexas do que pontos, exigindo menos espaço de memória para representar a mesma área. Por exemplo, uma parede num corredor pode ser descrita utilizando apenas um segmento de reta em mapas 2D, ou um pedaço de plano em mapas 3D, enquanto vários pontos seriam necessários para fazer o mesmo. No entanto, utilizar linhas ou planos para descrever o ambiente requer a detecção de estruturas mais complexas e correspondê-las entre diferentes varreduras ou imagens. Assim, há um compromisso entre utilizar características mais simples ou mais complexas, de modo análogo ao compromisso existente entre mapas métricos e topológicos.

Recentemente, os objetos também foram considerados como características em mapas de navegação de robô. Apesar do reconhecimento e reconstrução de objetos geométricos no ambiente já terem sido tópicos de investigação em alguns trabalhos isolados — por exemplo, [1] —, esse tópico não entrou no cerne da pesquisa robótica até o desenvolvimento de técnicas robustas de reconhecimento de objeto por visão computacional. Em contraste com elementos geométricos simples, os objetos incorporam informação semântica, aumentando consideravelmente as aplicações potenciais desses mapas.

Os trabalhos de [37, 29, 31] são exemplos de mapas de características que modelam os objetos no ambiente. Enquanto o primeiro trabalho representa a posição dos objetos detectados em grandes espaços interiores com relação a um sistema de coordenadas local, os dois últimos representam-nos no sistema global de coordenadas, incrementando a representação dos objetos com o parâmetro tamanho. Em [29], apenas os objetos capturados em uma única imagem são modelados, enquanto [31] apresenta um arcabouço para o mapeamento de todos os objetos que o robô conseguir identificar no ambiente enquanto o percorre.

1.1.2 Mapas topológicos

Diferentemente dos mapas métricos, os mapas topológicos não se propõem a fornecer informações geométricas a respeito do ambiente. Também chamados de mapas relacionais, fornecem as relações entre as diversas localidades ou marcos existentes no ambiente, geralmente na forma de um grafo. A proposta do mapa topológico para robôs móveis surgiu a partir da obra seminal de [19] sobre o modelo TOUR, que consiste em um grafo onde os nós representam

localidades no ambiente e as arestas ligam localidades adjacentes, podendo também representar rótulos semânticos indicando a estratégia de alto nível para uma pessoa ou robô ir de uma localidade a outra. Assim, localidades e suas relações são os elementos dos mapas topológicos.

As relações no mapa topológico podem ser de vários tipos. Elas podem indicar o deslocamento relativo entre dois marcos, a existência de um caminho entre eles, os comandos de controle que movem o robô de um marco para outro, rótulos semânticos mostrando instruções de alto nível para ir de um ao outro, e assim por diante. Exemplos de mapas topológicos incluem os Mapas Topológicos Probabilísticos (PTMs) [28], que elencam diferentes topologias que mostram marcos métricos e os caminhos entre eles; e mapas visomotores [36], que descrevem marcos por uma característica panorâmica da imagem e os ligam entre si por arestas anotadas com comandos de controle de orientação que leva o robô de um para o outro marco.

Mapas topológicos exigem muito menos memória quando comparados aos mapas métricos. Por esta razão, eles são os mais apropriados para o mapeamento de grandes áreas. Por outro lado, mesmo que uma iteração de qualquer algoritmo que constrói mapas topológicos processe alguns nós de um grafo, o processo de mapeamento é um pouco mais complexo. Este fato é devido à necessidade de processar os dados do sensor do robô de uma forma mais abstrata, a fim de extrair informação sobre as localidades mais relevantes que devem ser consideradas como marcos. Além disso, deve ser realizado um processamento extra para determinar as relações entre os marcos. Por isso, muitas vezes é vantajoso desenvolver soluções práticas que tentam conciliar elementos métricos em mapas relacionais, ou vice-versa, numa tentativa de obter vantagens das diferentes abordagens, quer através da construção de mapas híbridos [25, 6] ou utilizando múltiplos níveis de abstração para representar diferentes aspectos do ambiente, como proposto por [13].

1.1.3 Representação da estimativa de localização

Na navegação baseada em mapas, o robô também precisa representar a crença que possui a respeito de sua própria localização no mapa. O tipo de mapa utilizado afeta a forma de representação da crença, a qual também está relacionada com o fato do robô identificar uma ou várias posições como sua estimativa atual. Assim, duas dimensões devem ser observadas: uma, se o mapa é contínuo ou discreto; outra, se há uma única hipótese de localização ou se existem várias hipóteses, nas quais o robô expressa claramente sua incerteza em relação às diversas localizações no mapa.

A Figura 1.2 mostra representações unimodal e multimodal da crença do robô em sua posição 1D, em mapas contínuos e discretos.

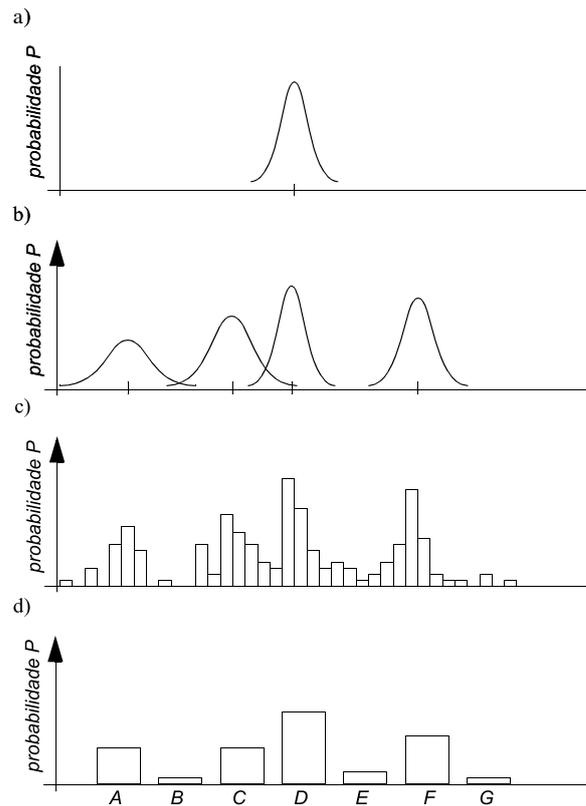


Figura 1.2: Crença do robô em sua posição 1D em mapas contínuos e discretos. a) Uma única hipótese em um mapa contínuo, b) múltiplas hipóteses em um mapa contínuo, c) múltiplas hipóteses em um mapa discretizado por uma grade regular e d) múltiplas hipóteses em um mapa topológico. Extraído de [32].

A hipótese única é a representação de crença mais simples e direta: dado um mapa do ambiente, o robô expressa sua crença por meio de um único ponto neste mapa. Por exemplo, se o mapa for 2D e contínuo, sua crença pode ser dada por uma posição (x, y) ou uma postura (x, y, θ) neste mapa; se for 2D e discreto, pode ser dada por uma célula, no caso de grades de evidência, ou por um vértice no grafo, no caso de mapas topológicos. A vantagem da hipótese única de localização é a de não possibilitar ambiguidades, o que facilita não só o planejamento de trajetórias, mas também o processo de atualização da crença de localização. Por outro lado, muitas vezes o uso de uma única hipótese de localização torna-se impossível devido à incerteza e aos ruídos associados aos sensores e atuadores do robô.

Crenças descritas por múltiplas hipóteses de localização permitem o uso de um conjunto possivelmente infinito de posições, dentre as quais pode ou não ser estabelecida uma ordem de preferência. Em [20], a localização do robô é descrita em termos de um polígono convexo definido em um mapa 2D do ambiente, sendo igualmente prováveis as hipóteses inseridas no interior do polígono. Já [23] e [10] utilizam uma distribuição gaussiana para representar a crença do robô em sua localização, onde a média corresponde à hipótese mais provável de localização. Pode-se ainda expressar crenças multimodais, como em [9] and [17]. Obviamente, a grande vantagem no uso de múltiplas hipóteses para representar a crença na localização é a possibilidade de expressar explicitamente sua incerteza na localização. Entretanto, esta representação torna o processo de planejamento e tomada de decisão mais custoso, já que fica difícil decidir para onde ir quando a trajetória adequada para uma possível localização for inconsistente ou inadequada para outra possível localização. Nestes casos, geralmente guia-se o robô de forma a ativamente sensoriar melhor o ambiente e, assim, reduzir sua incerteza em relação à localização.

1.2 Técnicas de localização

Para responder à questão de onde se encontra à medida em que se movimenta em um determinado ambiente, um robô basicamente executa ciclos que intercalam movimento com percepção, conforme ilustra o esquema geral mostrado na figura 1.3. Assim, dada uma estimativa corrente da sua localização, o robô se movimenta e, com base nas medidas de seus sensores internos, como a odometria, e da informação do mapa do ambiente ou de um modelo de seu movimento, efetua a predição de onde deveria estar após a execução de tal movimento. O robô então efetua a percepção do ambiente com seus sensores externos, como visão ou sonares, visando a captura de dados que expressem as evidências de sua localização atual no ambiente. Estes dados são traduzi-

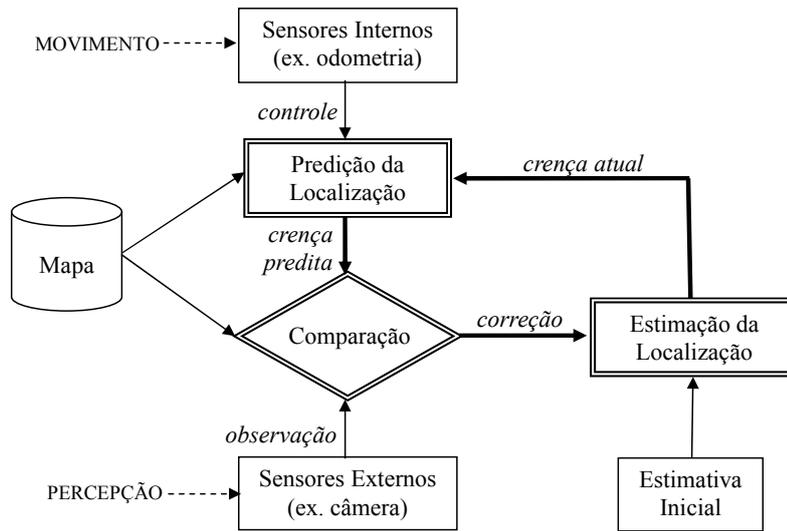


Figura 1.3: Arquitetura geral do sistema de localização de um robô móvel.

dos, com o auxílio do mapa, para uma localização observada pelos sensores, a qual é comparada com a anteriormente predita, e a diferença entre elas é finalmente utilizada para corrigir a estimativa atual, reiniciando o ciclo de movimento e percepção. Assim, a crença do robô a respeito de sua localização varia, no tempo, consistentemente com suas saídas motoras e suas entradas perceptuais.

Uma técnica de localização com abordagem geométrica e utilizando uma representação de crença com múltiplas hipóteses estima possíveis localizações do robô por meio da especificação de um polígono na representação do ambiente [21]. Esta técnica não oferece indicações a respeito das chances relativas entre as possíveis localizações do robô. Por outro lado, técnicas probabilísticas permitem explicitar as probabilidades de cada possível localização, além de oferecerem um ferramental adequado para lidar com toda incerteza envolvida no processo. Por estas razões, estas técnicas têm sido o foco de interesse das pesquisas recentes na área e a estimativa Bayesiana constitui sua base. Nas seções subsequentes são descritos a estimativa Bayesiana e os principais métodos explorados na robótica móvel, o Filtro de Kalman e a Localização Monte Carlo.

1.3 Estimativa Bayesiana

Para que o problema geral da autolocalização de um robô seja descrito, bem como uma estratégia básica para sua solução, considere um robô navegando em um ambiente desconhecido, de posse de seu mapa. A cada instante t , ele executa um comando de movimento u_t e move-se até a postura X_t , desconhecida, onde adquire os dados de sensores z_t . Para resolver o problema de autolocalização sob a perspectiva Bayesiana, seu objetivo é encontrar o valor

esperado da postura a cada instante, ou seja:

$$\hat{X}_t = E[X_t | \mathbf{z}^t, \mathbf{u}^t] = \int_x x p(X_t = x | \mathbf{z}^t, \mathbf{u}^t) dx, \quad (1.1)$$

onde $\mathbf{z}^t = \{z_\tau\}_{\tau=0}^t$ e $\mathbf{u}^t = \{u_\tau\}_{\tau=1}^t$ são, respectivamente, o conjunto total de medições sensoriais realizadas e os comandos de movimento recebidos desde o instante 0 até o momento corrente. O movimento relativo medido por odometria entre X_{t-1} e X_t pode ser usado no lugar do comando de movimento, com a vantagem de ser uma medição direta do deslocamento do robô.

A densidade de probabilidade $p(X_t = x | \mathbf{z}^t, \mathbf{u}^t)$ traduz a crença do robô sobre o conjunto de posturas que possa estar assumindo no instante t com base em todas as medições e comandos de controle realizados. Assim, o uso da abordagem Bayesiana para localização exige que essa crença seja representada de alguma forma, o que depende do mapa sendo utilizado e, em geral, corresponde a uma das formas ilustradas na Figura 1.2. Como ela muda a cada instante, os métodos Bayesianos procuram manter atualizada sua representação.

A evolução da crença sobre a localização do robô é ditada pelo *modelo probabilístico de movimento* do robô, que corresponde à densidade condicional $p(X_t = x | X_{t-1} = x', u_t)$, onde u_t é a informação do deslocamento entre os instantes $t-1$ e t . Se num primeiro momento o robô conhecer precisamente sua postura $X_0 = x_0$, no instante seguinte sua postura X_1 será conhecida de modo incerto. Isso significa que o robô terá uma crença a respeito de sua postura traduzida por uma densidade de probabilidade, $p(X_1 = x | X_0 = x_0, u_1)$. Caso a postura do robô não seja corrigida com a informação de sensores externos, a incerteza associada aumentará, ou seja, a região de suporte da densidade de probabilidade irá aumentar.

Da mesma forma que ocorre com a medida de deslocamento do robô, seus sensores externos também fornecem medições incertas. Assim, a crença sobre o verdadeiro valor medido também é expressa por uma densidade de probabilidade condicional, que corresponde ao *modelo probabilístico de observação*. Esse modelo prevê a crença sobre o valor medido z_t caso seja conhecida a localização do robô X_t no mapa, sendo expressa por $p(Z_t = z_t | X_t = x)$. Note que Z_t , em letra maiúscula, corresponde à variável aleatória que representa as possíveis medições no instante t , e z_t , em letra minúscula, corresponde à medição efetivamente realizada.

Suponha que no instante anterior $t-1$ a crença sobre a postura do robô seja conhecida, sendo denotada pela densidade condicional $p(X_{t-1} = x' | \mathbf{z}^{t-1}, \mathbf{u}^{t-1})$. Após o robô executar o comando de movimento u_t e capturar os dados sensoriais z_t , a nova crença é obtida pelo processo de marginalização da postura

anterior X_{t-1} na densidade condicional conjunta das variáveis X_t e X_{t-1} :

$$p(X_t = x | \mathbf{z}^t, \mathbf{u}^t) = \int_{x'} p(X_t = x, X_{t-1} = x' | Z^t = \mathbf{z}^t, \mathbf{u}^t) dx', \quad (1.2)$$

onde $Z^t = \{Z_\tau\}_{\tau=0}^t$ é o conjunto das variáveis aleatórias que representam as medições sensoriais obtidas entre os instantes inicial 0 e corrente t . Observe que o fato de se conhecer uma medição sensorial z_t torna conhecido o valor da variável aleatória $Z_t = z_t$. Uma vez que se pode desmembrar $Z^t = \{Z_t\} \cup Z^{t-1}$, a Lei de Bayes é usada para explicitar o papel da nova medição na crença sobre a postura corrente:

$$p(X_t = x, X_{t-1} = x' | z_t, \mathbf{z}^{t-1}, \mathbf{u}^t) dx' = \frac{p(Z_t = z_t | X_t = x, X_{t-1} = x', \mathbf{z}^{t-1}, \mathbf{u}^t) p(X_t = x, X_{t-1} = x' | \mathbf{z}^{t-1}, \mathbf{u}^t)}{p(Z_t = z_t | \mathbf{z}^{t-1}, \mathbf{u}^t)}. \quad (1.3)$$

O termo $p(Z_t = z_t | \mathbf{z}^{t-1}, \mathbf{u}^t)$ não depende de nenhuma variável desconhecida, X_t ou X_{t-1} , e portanto trata-se apenas de uma constante de normalização da probabilidade posterior.

Uma vez que a medição realizada pelos sensores pode ser totalmente explicada pelo modelo de observação, ou seja, a crença sobre seu valor não depende das medições anteriores nem da postura do robô nos instantes anteriores, tem-se:

$$p(Z_t = z_t | X_t = x, X_{t-1} = x', \mathbf{z}^{t-1}, \mathbf{u}^t) = p(Z_t = z_t | X_t = x). \quad (1.4)$$

Após aplicar a identidade (1.4) em (1.3) e usar o resultado em (1.2), observa-se que:

$$p(X_t = x | \mathbf{z}^t, \mathbf{u}^t) \propto p(Z_t = z_t | X_t = x) \int_{x'} p(X_t = x, X_{t-1} = x' | \mathbf{z}^{t-1}, \mathbf{u}^t) dx'. \quad (1.5)$$

O segundo fator de (1.5) corresponde à seguinte densidade marginal:

$$\int_{x'} p(X_t = x, X_{t-1} = x' | \mathbf{z}^{t-1}, \mathbf{u}^t) dx' = p(X_t = x | \mathbf{z}^{t-1}, \mathbf{u}^t), \quad (1.6)$$

que reflete a crença sobre a postura do robô após o último comando de movimento u_t , mas sem considerar as medições sensoriais correntes z_t . A formulação em (1.5) sugere que a crença sobre a postura do robô seja atualizada em duas etapas: uma que use (1.6) para estimar a crença sobre postura do robô após o movimento, e outra que use as medições sensoriais mais recentes

através de (1.5) para corrigir essa crença.

Para calcular a densidade em (1.6), considera-se que o movimento do robô seja *Markoviano*. Isso implica que a postura corrente do robô X_t seja independente de qualquer informação antiga caso se conheça a sua postura anterior X_{t-1} e último comando de movimento, sendo regida apenas pelo modelo probabilístico de movimento. Assim, decompondo-se a densidade conjunta de (1.6) tem-se:

$$p(X_t = x, X_{t-1} = x' | \mathbf{z}^{t-1}, \mathbf{u}^t) = p(X_t = x | X_{t-1} = x', \mathbf{z}^{t-1}, \mathbf{u}^t) p(X_{t-1} = x' | \mathbf{z}^{t-1}, \mathbf{u}^t), \quad (1.7)$$

Como as observações até o instante $t - 1$ são informações antigas, elas não influenciam a crença sobre a postura corrente X_t dados o valor da postura anterior X_{t-1} e o último comando de movimento u_t . Da mesma forma, desmembrando-se $\mathbf{u}^t = \{u_t\} \cup \mathbf{u}^{t-1}$, nota-se que \mathbf{u}^{t-1} é também informação antiga, de forma que

$$p(X_t = x | X_{t-1} = x', \mathbf{z}^{t-1}, u_t, \mathbf{u}^{t-1}) = p(X_t = x | X_{t-1} = x', u_t) \quad (1.8)$$

Por fim, observe que o comando de movimento mais recente u_t não acrescenta nenhuma informação sobre X_{t-1} caso não se conheça nada a respeito de X_t ou das medições z_t realizadas com o robô estando nessa postura. Por isso, pode-se assumir

$$p(X_{t-1} = x' | \mathbf{z}^{t-1}, u_t, \mathbf{u}^{t-1}) = p(X_{t-1} = x' | \mathbf{z}^{t-1}, \mathbf{u}^{t-1}), \quad (1.9)$$

que é a crença sobre a postura anterior, assumida como conhecida. Substituindo-se as identidades (1.9) e (1.8) em (1.7), e aplicando o resultado em (1.6), chega-se a:

$$p(X_t = x | \mathbf{z}^{t-1}, \mathbf{u}^t) = \int_{x'} p(X_t = x | X_{t-1} = x', u_t) p(X_{t-1} = x' | \mathbf{z}^{t-1}, \mathbf{u}^{t-1}) dx'. \quad (1.10)$$

Dessa forma, a partir de (1.5), a crença $Bel(X_t = x)$ que se quer calcular pode ser representada de forma recursiva:

$$Bel(X_t = x) \propto p(Z_t = z_t | X_t = x) \int_{x'} p(X_t = x | X_{t-1} = x', u_t) Bel(X_{t-1} = x') dx', \quad (1.11)$$

onde $Bel(X_t = x) = p(X_t = x | \mathbf{z}^t, \mathbf{u}^t)$ é a crença sobre a postura do robô no instante t dadas todas as informações disponíveis até esse instante.

Esse formato recursivo proporciona um algoritmo genérico para a manutenção da crença, conhecido como *Filtro de Bayes*, descrito no Algoritmo 1. Ele possui duas etapas principais: *predição* e *correção*. Na etapa de predição,

a crença sobre a postura anterior e a informação de movimento são utilizadas para calcular uma crença incompleta sobre a postura no instante corrente através de (1.10). Como é realizada apenas uma predição, há um aumento na incerteza sobre a localização, refletido no aumento do suporte da função densidade de probabilidade. Na etapa de correção, as leituras dos sensores são usadas para corrigir essa crença incompleta, diminuindo sua incerteza e gerando a crença desejada, $Bel(X_t = x)$. Finalmente, a crença é normalizada e a estimativa corrente da postura do robô é calculada.

Algoritmo 1 Etapas da localização pelo Filtro de Bayes

Entradas: observações sensoriais z_t , dado de controle u_t , crença sobre a localização anterior $Bel(X_{t-1} = x)$.

Saída: representação da crença sobre a localização corrente $Bel(X_t = x)$ e o valor esperado da postura \hat{X}_t .

Para toda possível localização x , faça:

1. Predição: $p(X_t = x | \mathbf{z}^{t-1}, \mathbf{u}^t) \leftarrow \int_{x'} p(X_t = x | X_{t-1} = x', u_t) Bel(X_{t-1} = x') dx'$
2. Correção: $Bel(X_t = x) \leftarrow p(Z_t = z_t | X_t = x) p(X_t = x | \mathbf{z}^{t-1}, \mathbf{u}^t)$

Fim Para.

Normalize a crença para que $\int_x Bel(X_t = x) dx = 1$

Retorne $Bel(X_t = x)$ e $\hat{X}_t = \int_x x Bel(X_t = x) dx$

Observe que são muito poucas as situações em que o Filtro de Bayes pode ser usado diretamente: apenas quando (1.11) pode ser manipulada diretamente, como no caso do Filtro de Kalman, explicado na Seção 1.4. Nos demais casos, as funções de densidade de probabilidade envolvidas devem ser discretizadas em intervalos regulares ou amostradas, levando, respectivamente, às técnicas de Localização Markoviana e, de forma aproximada, Localização Monte Carlo [34].

A crença $Bel(X_0 = x)$, necessária para alimentar inicialmente o Filtro de Bayes, é uma função que depende do conhecimento do projetista a respeito da localização inicial do robô. No caso de sua localização inicial ser completamente desconhecida, ela deve ser representada por uma densidade uniforme abrangendo toda área livre do mapa e todas as orientações. Se ela for conhecida com alto nível de certeza, pode ser representada por uma densidade gaussiana com um valor baixo para a norma da covariância.

1.4 Filtro de Kalman

Uma situação favorável no problema de localização existe quando o robô já conhece sua posição inicial. Neste caso, mesmo que possuísse apenas os sensores internos, o erro de localização cresceria aos poucos, o que permite

o uso dos sensores externos apenas para limitar a incerteza. Nessa situação, o problema de localização é chamado *localização local* ou manutenção de postura [14]. Uma das facilidades desse problema é que a crença sobre a localização do robô concentra-se em uma região relativamente pequena, podendo ser aproximada por uma distribuição gaussiana de probabilidade. No caso do modelo de observação dos sensores externos do robô possuir também uma incerteza pequena e concentrada, ela também pode ser aproximada por uma distribuição de probabilidade gaussiana. Para esses casos existe um ferramental bem conhecido: o Filtro de Kalman, um filtro Bayesiano linear ótimo, e a sua versão para o caso não linear, o Filtro de Kalman Estendido. Ele é muitas vezes representado por seu acrônimo EKF, indicando as iniciais do seu nome em inglês: *Extended Kalman Filter*.

Sob o ponto de vista Bayesiano, o Filtro de Kalman e sua versão estendida estimam o estado de uma variável aleatória dinâmica, cuja densidade de probabilidade é gaussiana, a partir da sequência de observações ruidosas de grandezas afetadas por essa variável. No caso da localização de um robô, o intuito é estimar a sua localização a cada instante, X_t , que por sua vez influencia diretamente na observação dos sensores naquele momento, Z_t . Seguindo a metodologia mostrada na seção 1.3, isso é feito iterativamente, de modo que a cada nova informação proveniente dos sensores aumenta a confiança a respeito da estimativa sobre a localização do robô.

Para garantir que a densidade de probabilidade $p(X_t = x | \mathbf{z}^t, \mathbf{u}^t)$ seja consistentemente tratada como gaussiana, é preciso fazer algumas aproximações no método apresentado na seção 1.3. Primeiramente, tanto o modelo de movimento como o de observação devem ser aproximados por densidades gaussianas multivariáveis. Dependendo das características dos sensores e atuadores do robô, essa aproximação pode ser razoavelmente boa. Além disso, o conhecimento da localização inicial do robô é um fator que contribui para manter o erro de localização pequeno e unimodal, melhorando a qualidade da aproximação da crença por uma densidade gaussiana.

Seja A um vetor de variáveis aleatórias. Se $p(A = a)$ for uma densidade gaussiana multivariável, cuja média for μ e cuja matriz de covariância for Σ , então

$$p(A = a) = \frac{1}{\sqrt{|2\pi\Sigma|}} \exp \left\{ -\frac{1}{2} \|a - \mu\|_{\Sigma}^2 \right\}. \quad (1.12)$$

A notação $\|v\|_{\Sigma}^2$ é usada para indicar a norma de Mahalanobis ao quadrado do vetor v com respeito à matriz de covariância Σ , dada por $v^T \Sigma^{-1} v$.

No caso de um pequeno aumento da incerteza a cada iteração de controle do robô, podemos aproximar o seu modelo probabilístico de movimento por

uma densidade gaussiana da forma

$$p(X_t = x | X_{t-1} = x', u_t) \approx \frac{1}{\sqrt{|2\pi Q_t|}} \exp \left\{ -\frac{1}{2} \|x - f(x', u_t)\|_{Q_t}^2 \right\}. \quad (1.13)$$

A função $f(x', u) \mapsto x$ é o modelo determinista que tenta prever a localização x do robô a partir de uma dada localização anterior x' e do comando de movimento u . Q_t é a matriz simétrica de covariância relacionada com a incerteza acumulada sobre o resultado efetivo do comando u_t e sobre o modelo $f(x', u_t)$, indicando a amplitude e a orientação da distribuição gaussiana multivariável.

Além do mais, se assumido como uma densidade gaussiana, o modelo probabilístico de observação é aproximado por

$$p(Z_t = z_t | X_t = x) \approx \frac{1}{\sqrt{|2\pi R_t|}} \exp \left\{ -\frac{1}{2} \|z - h(x)\|_{R_t}^2 \right\} \quad (1.14)$$

onde t é o instante de observação, z_t é a leitura e a função $h(x) \mapsto z$ é o modelo determinista que prevê o resultado z da leitura dos sensores caso a localização do robô seja x . R_t é a matriz simétrica de covariância relacionada com a incerteza acumulada sobre a leitura realizada z_t e sobre o modelo $h(x)$.

Outra condição para que a densidade de probabilidade da variável sendo rastreada se mantenha gaussiana é que os modelos deterministas de observação e movimento sejam lineares, ou seja

$$f(x, u) = A \begin{bmatrix} x \\ u \end{bmatrix} + b \quad \text{e} \quad h(x) = Cx + d, \quad (1.15)$$

sendo A e C matrizes constantes, enquanto b e d são vetores constantes de dimensões compatíveis. O filtro de Kalman original assume essa condição para sua derivação matemática e prova de estabilidade. No entanto, no caso da localização de robôs, os modelos de movimento e observação raramente podem ser assim considerados, quando então o EKF deve ser utilizado.

O EKF trata os casos de modelos não lineares aproximando-os pelos termos de até primeira ordem de sua série de Taylor, e aplicando as mesmas regras do Filtro de Kalman. As aproximações lineares obtidas são dadas por

$$f(x, u) \approx f(x_0, u) + F(x_0, u)(x - x_0) \quad (1.16)$$

$$h(x) \approx h(x_1) + H(x_1)(x - x_1), \quad (1.17)$$

onde x_0 e x_1 são os valores da localização do robô em torno dos quais as aproximações das funções são calculadas. $F(x_0, u)$ e $H(x_1)$ são, respectivamente, os Jacobianos das funções $f(x, u)$ e $h(x)$, calculados nos pontos x_0 e x_1 , respectivamente. Observe que (1.16) considera que o valor de u é conhecido, e por isso

$F(x, u)$ é calculado apenas com as derivadas parciais com relação aos componentes de x . De fato, os valores do comando de controle a cada instante u_t são conhecidos, ou são calculados a partir da leitura dos odômetros. Observe também que, devido à aproximação dos modelos utilizados, a otimalidade e a convergência do EKF não são garantidas teoricamente. Além do mais, quanto menores forem as normas dos desvios $\|x - x_0\|$ e $\|x - x_1\|$, mais precisas serão as aproximações (1.16) e (1.17).

1.4.1 Sequência de passos do EKF

O Filtro de Kalman, e assim o EKF, seguem a mesma sequência de passos da filtragem Bayesiana, que alterna etapas de predição e correção do estado sendo rastreado, que, neste caso, é a postura do robô. Uma vez que a variável de estado dinâmica X_t é tratada como se tivesse densidade gaussiana, ela é representada apenas pelos parâmetros dessa densidade: seu valor esperado após considerar todas as leituras até o momento, $\hat{X}_t = E[X_t | \mathbf{z}^t, \mathbf{u}^t]$, e matriz de covariância associada, $P_t = E[(X_t - \hat{X}_t)(X_t - \hat{X}_t)^T | \mathbf{z}^t, \mathbf{u}^t]$.

Passo de predição

O objetivo do passo de predição do EKF é obter os parâmetros de média e covariância da densidade $p(X_t = x | \mathbf{z}^{t-1}, \mathbf{u}^t)$, respectivamente representados por \hat{X}_t^- e P_t^- . Considere que a estimativa da postura do robô até o instante $t-1$, \hat{X}_{t-1} , seja fornecida, bem como sua covariância associada, P_{t-1} . Assim, a crença sobre a postura do robô é dada por:

$$Bel(X_{t-1} = x) = \frac{1}{\sqrt{|2\pi P_{t-1}|}} \exp \left\{ -\frac{1}{2} \left\| x - \hat{X}_{t-1} \right\|_{P_{t-1}}^2 \right\} \quad (1.18)$$

Na etapa de predição da postura do robô, descrita no algoritmo 1, a crença corrente sem considerar as medições mais recentes é dada por

$$\begin{aligned} p(X_t = x | \mathbf{z}^{t-1}, \mathbf{u}^t) &= \int_{x'} \frac{\exp \left\{ -\frac{1}{2} \left\| x - f(x', u_t) \right\|_{Q_t}^2 \right\} \exp \left\{ -\frac{1}{2} \left\| x' - \hat{X}_{t-1} \right\|_{P_{t-1}}^2 \right\}}{\sqrt{|2\pi P_{t-1}| |2\pi Q_t|}} dx' \\ &\propto \int_{x'} \exp \left\{ -\frac{1}{2} \left(\left\| x - f(x', u_t) \right\|_{Q_t}^2 + \left\| x' - \hat{X}_{t-1} \right\|_{P_{t-1}}^2 \right) \right\} dx' \\ &\approx \int_{x'} \exp \left\{ -\frac{1}{2} \left(\left\| x - f(x_0, u_t) - F(x_0, u_t)(x' - x_0) \right\|_{Q_t}^2 + \left\| x' - \hat{X}_{t-1} \right\|_{P_{t-1}}^2 \right) \right\} dx' \\ &= \int_{x'} \exp \left\{ -\frac{1}{2} L(x, x') \right\} dx', \end{aligned}$$

considerando a aproximação dada em (1.16). O campo escalar

$$L(x, x') \triangleq \|x - f(x_0, u_t) - F(x_0, u_t)(x - x_0)\|_{Q_t}^2 + \|x' - \hat{X}_{t-1}\|_{P_{t-1}}^2 \quad (1.19)$$

é uma soma de termos quadráticos.

O ponto de linearização x_0 deve ser escolhido de modo a minimizar o erro de linearização da função de predição do movimento, diminuindo o desvio $\|x' - x_0\|$ para os valores de x' que concentram a maior massa de probabilidade de $Bel(X_{t-1} = x')$. Por isso, é razoável pensar que o último valor estimado, \hat{X}_{t-1} , seja o mais adequado, já que é o valor esperado dessa densidade de probabilidade. Assumindo $x_0 = \hat{X}_{t-1}$, tem-se:

$$L(x, x') = \|x - f(\hat{X}_{t-1}, u_t) - F(\hat{X}_{t-1}, u_t)(x - \hat{X}_{t-1})\|_{Q_t}^2 + \|x' - \hat{X}_{t-1}\|_{P_{t-1}}^2 \quad (1.20)$$

Em uma soma de termos quadráticos quaisquer $\|a\|_{\Sigma_a}^2$ e $\|b\|_{\Sigma_b}^2$, nota-se que

$$\begin{aligned} \|a\|_{\Sigma_a}^2 + \|b\|_{\Sigma_b}^2 &= a^T \Sigma_a^{-1} a + b^T \Sigma_b^{-1} b \\ &= \begin{bmatrix} a^T & b^T \end{bmatrix} \begin{bmatrix} \Sigma_a^{-1} & 0 \\ 0 & \Sigma_b^{-1} \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} \\ &= \left\| \begin{bmatrix} a \\ b \end{bmatrix} \right\|_{Cov}^2, \text{ onde } Cov = \begin{bmatrix} \Sigma_a & 0 \\ 0 & \Sigma_b \end{bmatrix}. \end{aligned} \quad (1.21)$$

Além disso, para um vetor real a e uma matriz inversível M é válida a relação:

$$\begin{aligned} \|Ma\|_{\Sigma}^2 &= a^T M^T \Sigma^{-1} M a \\ &= a^T ((M^T)^{-1})^{-1} \Sigma^{-1} (M^{-1})^{-1} \\ &= a^T (M^{-1} \Sigma (M^T)^{-1})^{-1} a \\ &= \|a\|_{M^{-1} \Sigma (M^T)^{-1}}^2. \end{aligned} \quad (1.22)$$

Dessa forma, pode-se escrever (1.20) como:

$$\begin{aligned} L(x, x') &= \left\| \begin{bmatrix} x - f(\hat{X}_{t-1}, u_t) - F(\hat{X}_{t-1}, u_t)x' + F(\hat{X}_{t-1}, u_t)\hat{X}_{t-1} \\ x' - \hat{X}_{t-1} \end{bmatrix} \right\|_{\begin{bmatrix} Q_t & 0 \\ 0 & P_{t-1} \end{bmatrix}}^2 \\ &= \left\| \begin{bmatrix} I & -F(\hat{X}_{t-1}, u_t) \\ 0 & I \end{bmatrix} \left(\begin{bmatrix} x \\ x' \end{bmatrix} - \begin{bmatrix} f(\hat{X}_{t-1}, u_t) \\ \hat{X}_{t-1} \end{bmatrix} \right) \right\|_{\begin{bmatrix} Q_t & 0 \\ 0 & P_{t-1} \end{bmatrix}}^2 \\ &= \left\| \begin{bmatrix} x \\ x' \end{bmatrix} - \begin{bmatrix} f(\hat{X}_{t-1}, u_t) \\ \hat{X}_{t-1} \end{bmatrix} \right\|_{Cov1}^2, \end{aligned} \quad (1.23)$$

onde

$$Cov1 \triangleq \begin{bmatrix} I & -F(\hat{X}_{t-1}, u_t) \\ 0 & I \end{bmatrix}^{-1} \begin{bmatrix} Q_t & 0 \\ 0 & P_{t-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -F(\hat{X}_{t-1}, u_t)^T & I \end{bmatrix}^{-1} \quad (1.24)$$

e I é a matriz identidade de dimensões compatíveis com os vetores de postura x e x' . Sabendo-se que a inversão de uma matriz em blocos do tipo $\begin{bmatrix} I & X \\ 0 & I \end{bmatrix}$ é dada por $\begin{bmatrix} I & -X \\ 0 & I \end{bmatrix}$, com I sendo a matriz identidade, segue que:

$$\begin{aligned} Cov1 &= \begin{bmatrix} I & -F(\hat{X}_{t-1}, u_t) \\ 0 & I \end{bmatrix}^{-1} \begin{bmatrix} Q_t & 0 \\ 0 & P_{t-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -F(\hat{X}_{t-1}, u_t)^T & I \end{bmatrix}^{-1} \\ &= \begin{bmatrix} I & F(\hat{X}_{t-1}, u_t) \\ 0 & I \end{bmatrix} \begin{bmatrix} Q_t & 0 \\ 0 & P_{t-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ F(\hat{X}_{t-1}, u_t)^T & I \end{bmatrix} \\ &= \begin{bmatrix} Q_t + F(\hat{X}_{t-1}, u_t)P_{t-1}F(\hat{X}_{t-1}, u_t)^T & F(\hat{X}_{t-1}, u_t)P_{t-1} \\ P_{t-1}F(\hat{X}_{t-1}, u_t)^T & P_{t-1} \end{bmatrix}, \end{aligned} \quad (1.25)$$

resultando em

$$p(X_t = x | \mathbf{z}^{t-1}, \mathbf{u}^t) \approx \int_{x'} \frac{\exp \left\{ -\frac{1}{2} \left\| \begin{bmatrix} x \\ x' \end{bmatrix} - \begin{bmatrix} f(\hat{X}_{t-1}, u_t) \\ \hat{X}_{t-1} \end{bmatrix} \right\|_{Cov1}^2 \right\}}{\sqrt{|2\pi Q_t| |2\pi P_{t-1}|}} dx'. \quad (1.26)$$

Uma das propriedades de uma densidade gaussiana multivariável é que a marginalização de uma ou mais variáveis envolvidas resulta em outra densidade gaussiana. Além do mais, a média de cada variável remanescente corresponde à própria média dessa variável na densidade original, bem como a matriz de covariância resultante também mantém os mesmos elementos referentes às variáveis remanescentes da matriz de covariância original, removendo os elementos referentes às variáveis marginalizadas [4].

Por exemplo, seja a densidade gaussiana conjunta das variáveis aleatórias vetoriais A e B , com média $\mu = \begin{bmatrix} \mu_a \\ \mu_b \end{bmatrix}$ e covariância $\Sigma = \begin{bmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{bmatrix}$, sendo $\Sigma_{ab} = \Sigma_{ba}^T$. Então, a densidade de probabilidade marginal de A também é gaussiana, com média μ_a e matriz de covariância Σ_a , dada por

$$p(A = a) = \int_b p(A = a, B = b) db = \frac{1}{\sqrt{|2\pi \Sigma_a|}} e^{\left\{ -\frac{1}{2} \|a - \mu_a\|_{\Sigma_a}^2 \right\}}. \quad (1.27)$$

Essa propriedade permite escrever (1.26) como

$$p(X_t = x | \mathbf{z}^{t-1}, \mathbf{u}^t) \approx \frac{\exp \left\{ -\frac{1}{2} \left\| x - f(\hat{X}_{t-1}, u_t) \right\|_{Q_t + F(\hat{X}_{t-1}, u_t) P_{t-1} F(\hat{X}_{t-1}, u_t)}^2 \right\}}{\sqrt{|2\pi(Q_t + F(\hat{X}_{t-1}, u_t) P_{t-1} F(\hat{X}_{t-1}, u_t))|}}, \quad (1.28)$$

uma densidade de probabilidade gaussiana de média $f(\hat{X}_{t-1}, u)$ e matriz de covariância dada por $Q_t + F(\hat{X}_{t-1}, u_t) P_{t-1} F(\hat{X}_{t-1}, u_t)$. Como resultado, obtêm-se as equações de *predição* da postura do robô a partir do último comando de movimento u_t :

$$\hat{X}_t^- = f(\hat{X}_{t-1}, u_t) \quad (1.29)$$

$$P_t^- = F(\hat{X}_{t-1}, u_t) P_{t-1} F(\hat{X}_{t-1}, u_t)^T + Q_t \quad (1.30)$$

Observe que, sendo $F(\hat{X}_{t-1}, u_t)$ uma matriz real e P_{t-1} uma matriz positiva-definida, o valor da norma da covariância irá sempre aumentar no passo de predição, refletindo o aumento da incerteza sobre o estado sendo estimado.

Passo de correção

Na etapa de correção da localização Bayesiana, descrita no algoritmo 1, a crença sobre a postura corrente X_t é ajustada pelas últimas medições sensoriais z_t através do modelo de observação descrito em (1.14). Uma vez que tanto a crença sobre a postura do robô quanto o modelo de observação são considerados densidades gaussianas no EKF, e levando também em consideração a aproximação dada em (1.17), a equação de correção do algoritmo 1 é dada por:

$$\begin{aligned} Bel(X_t = x) &\propto p(Z_t = z_t | X_t = x) p(X_t = x | \mathbf{z}^{t-1}, \mathbf{u}^t) \\ &\propto \exp \left\{ -\frac{1}{2} \|z_t - h(x)\|_{R_t}^2 \right\} \exp \left\{ -\frac{1}{2} \left\| x - \hat{X}_t^- \right\|_{P_t^-}^2 \right\} \\ &= \exp \left\{ -\frac{1}{2} \left(\|z_t - h(x)\|_{R_t}^2 + \left\| x - \hat{X}_t^- \right\|_{P_t^-}^2 \right) \right\} \\ &\approx \exp \left\{ -\frac{1}{2} \left(\|z_t - h(x_1) - H(x_1)(x - x_1)\|_{R_t}^2 + \left\| x - \hat{X}_t^- \right\|_{P_t^-}^2 \right) \right\} \\ &= \exp \left\{ -\frac{1}{2} M(x) \right\}, \end{aligned} \quad (1.31)$$

onde

$$M(x) \triangleq \|z_t - h(x_1) - H(x_1)x + H(x_1)x_1\|_{R_t}^2 + \left\| x - \hat{X}_t^- \right\|_{P_t^-}^2 \quad (1.32)$$

corresponde a uma soma de termos quadráticos.

O ponto de linearização x_1 da função de predição da observação deve ser escolhido de modo a minimizar o seu erro de linearização, diminuindo o desvio $\|x - x_1\|$ para o valor de x correspondente à postura onde a medida z_t foi realizada. Por isso, é razoável pensar que a melhor estimativa que se tem da postura do robô até o momento, \hat{X}_t^- , seja o valor mais adequado. Assumindo $x_1 = \hat{X}_t^-$, tem-se:

$$M(x) = \left\| z_t - h(\hat{X}_t^-) - H(\hat{X}_t^-)x + H(\hat{X}_t^-)\hat{X}_t^- \right\|_{R_t}^2 + \left\| x - \hat{X}_t^- \right\|_{P_t^-}^2. \quad (1.33)$$

De acordo com (1.21) e (1.22), pode-se escrever

$$\begin{aligned} M(x) &= \left\| \begin{bmatrix} x - \hat{X}_t^- \\ z_t - h(\hat{X}_t^-) - H(\hat{X}_t^-)x + H(\hat{X}_t^-)\hat{X}_t^- \end{bmatrix} \right\|_{\begin{bmatrix} P_t^- & 0 \\ 0 & R_t \end{bmatrix}}^2 \\ &= \left\| \begin{bmatrix} I & 0 \\ -H(\hat{X}_t^-) & \bar{I} \end{bmatrix} \left(\begin{bmatrix} x \\ z_t \end{bmatrix} - \begin{bmatrix} \hat{X}_t^- \\ h(\hat{X}_t^-) \end{bmatrix} \right) \right\|_{\begin{bmatrix} P_t^- & 0 \\ 0 & R_t \end{bmatrix}}^2 \\ &= \left\| \begin{bmatrix} x \\ z_t \end{bmatrix} - \begin{bmatrix} \hat{X}_t^- \\ h(\hat{X}_t^-) \end{bmatrix} \right\|_{Cov2}^2, \end{aligned} \quad (1.34)$$

com I sendo a matriz identidade de dimensões compatíveis com o vetor x , \bar{I} sendo a matriz identidade com dimensões compatíveis com o vetor z_t e

$$\begin{aligned} Cov2 &\triangleq \begin{bmatrix} I & 0 \\ -H(\hat{X}_t^-) & \bar{I} \end{bmatrix}^{-1} \begin{bmatrix} P_t^- & 0 \\ 0 & R_t \end{bmatrix} \begin{bmatrix} I & -H(\hat{X}_t^-)^T \\ 0 & \bar{I} \end{bmatrix}^{-1} \\ &= \begin{bmatrix} I & 0 \\ -H(\hat{X}_t^-) & \bar{I} \end{bmatrix}^{-1} \begin{bmatrix} (P_t^-)^{-1} & 0 \\ 0 & R_t^{-1} \end{bmatrix}^{-1} \begin{bmatrix} I & -H(\hat{X}_t^-)^T \\ 0 & \bar{I} \end{bmatrix}^{-1} \\ &= \left(\begin{bmatrix} I & -H(\hat{X}_t^-)^T \\ 0 & \bar{I} \end{bmatrix} \begin{bmatrix} (P_t^-)^{-1} & 0 \\ 0 & R_t^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -H(\hat{X}_t^-) & \bar{I} \end{bmatrix} \right)^{-1} \\ &= \begin{bmatrix} (P_t^-)^{-1} + H(\hat{X}_t^-)^T R_t^{-1} H(\hat{X}_t^-) & -H(\hat{X}_t^-)^T R_t^{-1} \\ -R_t^{-1} H(\hat{X}_t^-) & R_t^{-1} \end{bmatrix}^{-1} = \Lambda^{-1}, \end{aligned} \quad (1.35)$$

onde $\Lambda = Cov2^{-1}$ é a chamada *matriz de informação*. A crença sobre a postura do robô fica:

$$Bel(X_t = x) \propto \exp \left\{ -\frac{1}{2} M(x) \right\} = \exp \left\{ -\frac{1}{2} \left\| \begin{bmatrix} x \\ z_t \end{bmatrix} - \begin{bmatrix} \hat{X}_t^- \\ h(\hat{X}_t^-) \end{bmatrix} \right\|_{\Lambda^{-1}}^2 \right\}. \quad (1.36)$$

Uma vez que o valor de z_t é conhecido, pois é obtido a partir das informa-

ções sensoriais, os parâmetros de média e covariância da densidade (1.36) são obtidos a partir de manipulação algébrica dos termos de $M(x)$. Considerando-se

$$\Lambda \triangleq \begin{bmatrix} (P_t^-)^{-1} + H(\hat{X}_t^-)^T R_t^{-1} H(\hat{X}_t^-) & H(\hat{X}_t^-)^T R_t^{-1} \\ R_t^{-1} H(\hat{X}_t^-) & R_t^{-1} \end{bmatrix} = \begin{bmatrix} \Lambda_{xx} & \Lambda_{xz} \\ \Lambda_{zx} & \Lambda_{zz} \end{bmatrix}, \quad (1.37)$$

com $\Lambda_{xx} = (P_t^-)^{-1} + H(\hat{X}_t^-)^T R_t^{-1} H(\hat{X}_t^-)$, $\Lambda_{zz} = R_t^{-1}$ e $\Lambda_{zx} = \Lambda_{xz}^T = -R_t^{-1} H(\hat{X}_t^-)$, pode-se escrever:

$$\begin{aligned} M(x) &= \left(\begin{bmatrix} x \\ z_t \end{bmatrix} - \begin{bmatrix} \hat{X}_t^- \\ h(\hat{X}_t^-) \end{bmatrix} \right)^T \begin{bmatrix} \Lambda_{xx} & \Lambda_{xz} \\ \Lambda_{zx} & \Lambda_{zz} \end{bmatrix} \left(\begin{bmatrix} x \\ z_t \end{bmatrix} - \begin{bmatrix} \hat{X}_t^- \\ h(\hat{X}_t^-) \end{bmatrix} \right) \\ &= (x - \hat{X}_t^-)^T \Lambda_{xx} (x - \hat{X}_t^-) + (x - \hat{X}_t^-)^T \Lambda_{xz} (z_t - h(\hat{X}_t^-)) + \\ &\quad (z_t - h(\hat{X}_t^-))^T \Lambda_{zx} (x - \hat{X}_t^-) + (z_t - h(\hat{X}_t^-))^T \Lambda_{zz} (z_t - h(\hat{X}_t^-)) \\ &= (x - \hat{X}_t^-)^T \Lambda_{xx} (x - \hat{X}_t^-) + (x - \hat{X}_t^-)^T \Lambda_{xx} \Lambda_{xx}^{-1} \Lambda_{xz} (z_t - h(\hat{X}_t^-)) + \\ &\quad (z_t - h(\hat{X}_t^-))^T \Lambda_{zx} \Lambda_{xx}^{-1} \Lambda_{xx} (x - \hat{X}_t^-) + (z_t - h(\hat{X}_t^-))^T \Lambda_{zz} (z_t - h(\hat{X}_t^-)). \end{aligned} \quad (1.38)$$

Seja também o vetor δ_t , de valor conhecido, com $\delta_t \triangleq -\Lambda_{xx}^{-1} \Lambda_{xz} (z_t - h(\hat{X}_t^-)) = -((z_t - h(\hat{X}_t^-))^T \Lambda_{zx} \Lambda_{xx}^{-1})^T$. Ele permite reescrever $M(x)$ como

$$\begin{aligned} M(x) &= (x - \hat{X}_t^-)^T \Lambda_{xx} (x - \hat{X}_t^-) - (x - \hat{X}_t^-)^T \Lambda_{xx} \delta_t - \delta_t^T \Lambda_{xx} (x - \hat{X}_t^-) + \\ &\quad (z_t - h(\hat{X}_t^-))^T \Lambda_{zz} (z_t - h(\hat{X}_t^-)) + \delta_t^T \Lambda_{xx} \delta_t - \delta_t^T \Lambda_{xx} \delta_t \\ &= (x - \hat{X}_t^- - \delta_t)^T \Lambda_{xx} (x - \hat{X}_t^- - \delta_t) + (z_t - h(\hat{X}_t^-))^T \Lambda_{zz} (z_t - h(\hat{X}_t^-)) - \delta_t^T \Lambda_{xx} \delta_t \\ &= \|x - \hat{X}_t^- - \delta_t\|_{\Lambda_{xx}^{-1}}^2 + (z_t - h(\hat{X}_t^-))^T (\Lambda_{zz} - \Lambda_{zx} \Lambda_{xx}^{-1} \Lambda_{xz}) (z_t - h(\hat{X}_t^-)) \\ &= \|x - \hat{X}_t^- - \delta_t\|_{\Lambda_{xx}^{-1}}^2 + r, \end{aligned} \quad (1.39)$$

com $r \triangleq \|z_t - h(\hat{X}_t^-)\|_{(\Lambda_{zz} - \Lambda_{zx} \Lambda_{xx}^{-1} \Lambda_{xz})^{-1}}^2$ sendo um valor constante, pois não depende de x .

Através da nova formulação para $M(x)$, é possível reescrever a crença em (1.36) como:

$$\begin{aligned} Bel(X_t = x) &\propto \exp \left\{ -\frac{1}{2} M(x) \right\} \\ &= \exp \left\{ -\frac{1}{2} \left(\|x - \hat{X}_t^- - \delta\|_{\Lambda_{xx}^{-1}}^2 + r \right) \right\} \\ &\propto \exp \left\{ -\frac{1}{2} \|x - (\hat{X}_t^- + \delta)\|_{\Lambda_{xx}^{-1}}^2 \right\}, \end{aligned} \quad (1.40)$$

resultando em uma densidade gaussiana com média $\hat{X}_t^- + \delta$ e covariância Λ_{xx}^{-1} . Como o valor esperado de uma densidade gaussiana é igual à sua média, a nova estimativa da postura é dada por

$$\begin{aligned}
\hat{X}_t &= \hat{X}_t^- + \delta \\
&= \hat{X}_t^- - \Lambda_{xx}^{-1} \Lambda_{xz} (z_t - h(\hat{X}_t^-)) \\
&= \hat{X}_t^- + K_t \nu_t.
\end{aligned} \tag{1.41}$$

O termo $\nu_t \triangleq z_t - h(\hat{X}_t^-)$ é chamado de *inovação*, pois representa a diferença entre a leitura sensorial realizada, z_t , e o valor esperado para essa leitura dada a última estimativa de postura do robô, $h(\hat{X}_t^-)$. A matriz K_t é o chamado *ganho de Kalman*, que multiplica o valor da inovação para gerar o vetor δ_t que corrige a estimativa de postura. O valor do ganho de Kalman é dado por

$$\begin{aligned}
K_t &= -\Lambda_{xx}^{-1} \Lambda_{xz} \\
&= \Lambda_{xx}^{-1} H(\hat{X}_t^-)^T R_t^{-1} \\
&= \Lambda_{xx}^{-1} H(\hat{X}_t^-)^T R_t^{-1} (R_t + H(\hat{X}_t^-) P_t^- H(\hat{X}_t^-)^T) (R_t + H(\hat{X}_t^-) P_t^- H(\hat{X}_t^-)^T)^{-1} \\
&= \Lambda_{xx}^{-1} (H(\hat{X}_t^-)^T \widehat{R_t^{-1}} + H(\hat{X}_t^-)^T R_t^{-1} H(\hat{X}_t^-) P_t^- H(\hat{X}_t^-)^T) (R_t + H(\hat{X}_t^-) P_t^- H(\hat{X}_t^-)^T)^{-1} \\
&= \Lambda_{xx}^{-1} (H(\hat{X}_t^-)^T + \underbrace{H(\hat{X}_t^-)^T R_t^{-1} H(\hat{X}_t^-) P_t^- H(\hat{X}_t^-)^T}_{=\Lambda_{xx}}) (R_t + H(\hat{X}_t^-) P_t^- H(\hat{X}_t^-)^T)^{-1} \\
&= \Lambda_{xx}^{-1} ((P_t^-)^{-1} + H(\hat{X}_t^-)^T R_t^{-1} H(\hat{X}_t^-)) P_t^- H(\hat{X}_t^-)^T (R_t + H(\hat{X}_t^-) P_t^- H(\hat{X}_t^-)^T)^{-1} \\
&= \widehat{\Lambda_{xx}^{-1} \Lambda_{xz}} P_t^- H(\hat{X}_t^-)^T (R_t + H(\hat{X}_t^-) P_t^- H(\hat{X}_t^-)^T)^{-1} \\
&= P_t^- H(\hat{X}_t^-)^T (R_t + H(\hat{X}_t^-) P_t^- H(\hat{X}_t^-)^T)^{-1} \\
&= P_t^- H(\hat{X}_t^-)^T S_t^{-1},
\end{aligned} \tag{1.42}$$

onde

$$S_t \triangleq R_t + H(\hat{X}_t^-) P_t^- H(\hat{X}_t^-)^T \tag{1.43}$$

é a chamada *covariância da inovação*. É possível mostrar que S_t é a covariância da densidade marginal

$$\begin{aligned}
p(Z_t = z | \mathbf{z}^{t-1}, \mathbf{u}^t) &= \int_x p(Z_t = z | X_t = x, \mathbf{z}^{t-1}, \mathbf{u}^t) p(X_t = x, | \mathbf{z}^{t-1}, \mathbf{u}^t) dx \\
&= \frac{1}{\sqrt{|2\pi S_t|}} \exp \left\{ -\frac{1}{2} \|z - h(\hat{X}_t^-)\|_{S_t}^2 \right\}.
\end{aligned}$$

A matriz de covariância associada com a crença sobre a nova postura do robô é dada por

$$P_t = \Lambda_{xx}^{-1} = \left((P_t^-)^{-1} + H(\hat{X}_t^-)^T R_t^{-1} H(\hat{X}_t^-) \right)^{-1}. \tag{1.44}$$

No entanto, há um problema com a formulação em (1.44). Calcular a nova matriz de covariância requereria, por essa formulação, a inversão da matriz

de covariância P_t^- , obtida no passo de predição, seguida de uma nova inversão envolvendo a mesma matriz. A inversão de matrizes é uma operação computacionalmente custosa, e pode gerar instabilidade numérica no caso da matriz sendo invertida não ser bem condicionada. Por isso, o cálculo da covariância é realizado com o auxílio do *lema de inversão de matrizes*, que garante que

$$\begin{aligned}
P_t &= \left((P_t^-)^{-1} + H(\hat{X}_t^-)^T R_t^{-1} H(\hat{X}_t^-) \right)^{-1} \\
&= P_t^- - P_t^- H(\hat{X}_t^-)^T \overbrace{\left(R_t + H(\hat{X}_t^-) P_t^- H(\hat{X}_t^-)^T \right)^{-1}}^{=S_t} H(\hat{X}_t^-) P_t^- \\
&= \left(I - \overbrace{P_t^- H(\hat{X}_t^-)^T S_t^{-1} H(\hat{X}_t^-)}^{=K_t} \right) P_t^- \\
&= (I - K_t H(\hat{X}_t^-)) P_t^-. \tag{1.45}
\end{aligned}$$

A vantagem dessa nova formulação é que a inversão é aplicada apenas sobre a matriz S_t , que, tipicamente, possui dimensões menores e é melhor condicionada do que P_t^- . A demonstração do lema de inversão de matrizes para o caso particular do EKF encontra-se em [34].

Finalmente, no passo de correção, a nova estimativa da postura do robô bem como sua covariância associada são dadas por:

$$\hat{X}_t = \hat{X}_t^- + K_t \nu_t \tag{1.46}$$

$$P_t = (I - K_t H(\hat{X}_t^-)) P_t^- \tag{1.47}$$

Observe que a importância da inovação nessa estimativa é proporcional à norma do ganho de Kalman, dado por (1.42). Assim, quanto maior a incerteza da predição do estado, indicada pela norma de P_t^- , maior importância terá a inovação. Por outro lado, quanto maior a incerteza da inovação, indicada pela norma de S_t , maior será a importância da predição do estado na sua estimativa final.

1.4.2 Associação de dados

Embora o EKF permita a utilização de diversos modelos de sensores, geralmente essa técnica é utilizada com sensores que forneçam a distância e orientação até algum ponto de interesse no espaço, conforme ilustrado na Figura 1.4. Assim, o mapa é matematicamente representado por um conjunto $\{l_j\}_{j=1}^N$, onde l_j é a posição de um ponto de interesse no ambiente e N é o número de pontos de interesse do mapa. Assumindo esse tipo de sensor, a linearização do modelo determinista de observação não acarreta grandes

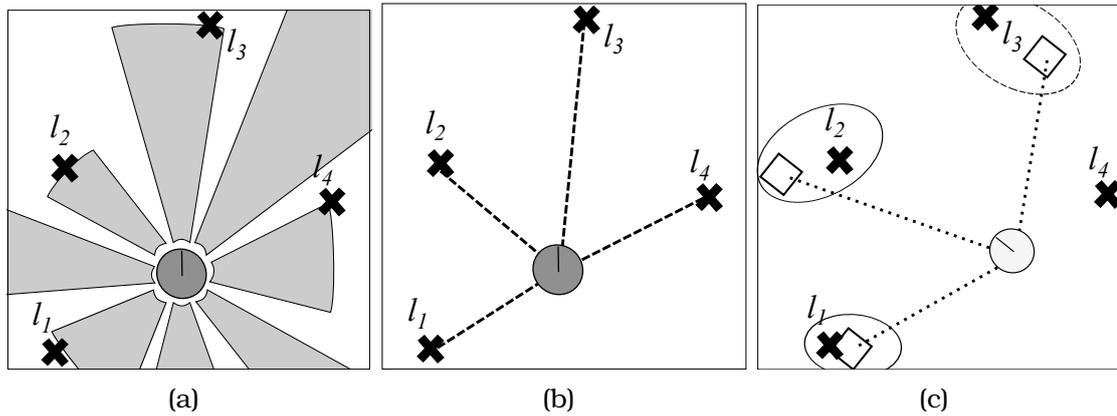


Figura 1.4: Ilustração do modelo determinista de observação comumente utilizado para a localização com o EKF. (a) Exemplo de dados obtidos por um robô a partir de sensores de sonar. O robô é representado pelo círculo cinza, com um traço indicando sua orientação. As marcas em "x" representam os pontos de interesse do mapa, enquanto os segmentos de círculo indicam a área de atuação de cada sonar. (b) As linhas tracejadas representam os dados de distância e orientação do robô em relação aos pontos de interesse calculados a partir da associação dos dados do sonar com o mapa. (c) No caso do robô não conhecer sua localização corretamente, pode realizar associações erradas. As associações entre medições (representadas por quadrados) e ponto de interesse no mapa estão circuladas. Observe que a associação com o ponto em l_3 deveria ter sido feita com o ponto em l_4 .

discrepâncias com relação ao modelo não-linear, mantendo, em geral, a capacidade de convergência do filtro. Por outro lado, esse modelo de observação exige que os dados brutos dos sensores sejam pré-processados para que se extraiam os valores das distâncias e orientações até os pontos do mapa sendo mensurados.

Esse pré-processamento dos dados brutos dos sensores deve fornecer as medidas (distância e orientação) de cada ponto de interesse no mapa que seja visível para o robô. No entanto, ele precisa combinar os dados obtidos pelos sensores com os diversos pontos do mapa, atribuindo uma medição a cada ponto observado. Para que isso seja feito, é necessário que o robô conheça com certa precisão sua localização com relação aos pontos do mapa, conforme ilustrado na Figura 1.4(c). A esse problema denomina-se associação de dados.

O modelo determinista de observação agora prevê a medição realizada pelo robô de um ponto específico, de modo que as medições realizadas em determinado instante tornam-se:

$$z_t = \{z_t^k\}_{k=1}^{M_t}, \quad (1.48)$$

onde z_t^k é a k -ésima medição realizada no instante t , sendo M_t o número de medições utilizadas. O modelo determinista de movimento agora recebe dois argumentos, a localização do robô e a posição do ponto no mapa, de modo que

o modelo de geração de cada medição torna-se:

$$p(Z_t^k = z | X_t = x) \approx \frac{1}{\sqrt{|2\pi R_t^k|}} \exp \left\{ -\frac{1}{2} \left\| z - h(x, l_{c_t^k}) \right\|_{R_t^k}^2 \right\}, \quad (1.49)$$

onde $1 \leq c_t^k \leq N$ é a variável de associação, que atribui um ponto do mapa a cada medição z_t^k . Os passos da localização por EKF para mapas de pontos de interesse estão descritos no Algoritmo 2.

Algoritmo 2 Etapas da localização por EKF para uso com mapas de pontos de interesse

Entradas: Sequência de observações z^t , dados de controle u^t , localização inicial X_0 , matrizes de covariância R_1 e Q_1 , conjunto de variáveis de associação $c^t = \{\{c_\tau^k\}_{k=1}^{M_\tau}\}_{\tau=1}^t$, mapa descrevendo as posições $\{l_j\}_{j=1}^N$ dos pontos de interesse do ambiente.

Saídas: localização estimada \hat{X}_t , covariância da localização P_t .

Variáveis: τ : Inteiro.

Para $\tau \leftarrow 1$ a t , faça

1. Predição: Calcule os parâmetros \hat{X}_τ^- e P_τ^- usando (1.29) e (1.30), respectivamente.
2. Correção:
 - (a) Calcule a inovação $\nu_\tau \triangleq z_\tau - h(\hat{X}_\tau^-)$, onde $z_\tau^T = [(z_\tau^1)^T \cdots (z_\tau^{M_\tau})^T]^T$ e $h(\hat{X}_\tau^-)^T = [h(\hat{X}_\tau^-, l_{c_\tau^1})^T \cdots h(\hat{X}_\tau^-, l_{c_\tau^{M_\tau}})^T]^T$
 - (b) Calcule a covariância da inovação fazendo $S_\tau = H(\hat{X}_\tau^-)P_\tau^-H(\hat{X}_\tau^-)^T + R_\tau$
 - (c) Calcule o ganho de Kalman fazendo $K_\tau = P_\tau^-H(\hat{X}_\tau^-)^T S_\tau^{-1}$
 - (d) Calcule os parâmetros \hat{X}_τ e P_τ através de (1.46) e (1.47)

Fim Para.

Retorne \hat{X}_t e P_t .

1.5 Localização Monte Carlo

Embora muito eficiente, o filtro de Kalman mantém uma representação muito restritiva da crença sobre a localização do robô, representada em uma única hipótese de localização. Se, por outro lado, for utilizada uma representação discreta da densidade de probabilidade da localização do robô através da amostragem uniforme do espaço de configurações, a representação não seria restritiva, mas possuiria a grande desvantagem de um excessivo custo computacional quando o número de células usadas para representar o ambiente crescesse consideravelmente. Essa é a abordagem adotada pela *Localização Markoviana*, que implementa essa representação genérica da crença por meio

da partição do espaço de configurações do robô em um número finito de possíveis posturas do robô em um mapa.

Uma característica poderosa desta abordagem é que ela permite que a localização do robô seja determinada sem o conhecimento de sua localização inicial. Abordagens deste tipo são conhecidas como *localização global*, e sugerem que um robô possa se localizar mesmo quando uma força externa o transporta para uma localização desconhecida; este problema é conhecido como o *problema do robô sequestrado*.

Entretanto, para melhorar o desempenho da Localização Markoviana, neste caso, é necessário levar a cabo uma amostragem mais inteligente do espaço de configurações. Isto é feito na *Localização Monte Carlo*. Intuitivamente, pode-se propor que essa amostragem não represente regiões de probabilidade zero ou muito pequena. Além disso, tenderia a representar com mais detalhes as regiões de maior densidade de probabilidade, pois há grande chance da localização do robô se encontrar de fato nessas regiões.

Já vimos que na abordagem Bayesiana importa encontrar \hat{X}_t , o valor esperado da localização do robô. No entanto, uma das possíveis definições do valor esperado de uma variável aleatória é justamente a média dos valores assumidos dado um número suficiente de amostras, o que, para uma variável aleatória A corresponde a:

$$\hat{A} = E_A[A] \triangleq \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{s=1}^n A^s, \quad (1.50)$$

com n sendo o número de amostras do valor da variável A , obtidos pela repetição das situações em que esse valor é gerado, como no caso de um experimento científico que mede uma grandeza física ou no caso de uma pesquisa de opinião.

No entanto, a localização de um determinado robô num certo instante de tempo é um fenômeno que ocorre apenas uma vez. Mesmo assim, é possível obter a sua localização esperada através da definição (1.50) se for considerada a geração de amostras através da simulação do modelo de movimento do robô e dos dados dos sensores. De modo geral, as técnicas que usam o expediente da simulação dos fenômenos para estimar parâmetros do modelo são denominadas técnicas Monte Carlo, uma referência ao cassino de mesmo nome em Mônaco [2].

Devido ao caráter sequencial e Markoviano do movimento do robô, o filtro de partículas é o modo natural de aplicar o princípio das técnicas Monte Carlo à localização de robôs, dando origem à técnica conhecida como *Monte Carlo Localization* (MCL) [18]. O filtro de partículas usa como base a *amostragem por importância*, que representa a densidade de probabilidade da localização

do robô através de amostras ponderadas, de forma que seu valor esperado seja $\hat{X}_t \approx \sum_{s=1}^n w_t^s X_t^s$, onde $\{w_t^s\}_{s=1}^n$ são os pesos que ponderam as n amostras no instante t . Os pesos devem ser normalizados, de forma que $\sum_{s=1}^n w_t^s = 1$. A diferença é que o cálculo das amostras é feito de modo sequencial pelo filtro de partículas, sendo que as amostras $\{X_{t-1}^s\}_{s=1}^n$ são usadas para gerar as amostras no instante posterior, $\{X_t^s\}_{s=1}^n$. Por isso, a versão mais básica desse filtro também recebe o nome, em inglês, de *Sequential Importance Sampling* (SIS), ou amostragem por importância sequencial.

A Figura 1.5 ilustra a evolução da crença sobre a localização do robô obtida através da MCL. Primeiramente, nota-se que não é necessário conhecer previamente a localização inicial do robô. No caso de ela ser completamente desconhecida, basta espalhar as amostras iniciais $\{X_0^s\}_{s=1}^n$ uniformemente sobre a área livre do mapa. Nota-se também que, conforme o robô move-se pelo ambiente e observa novas áreas, o número de localidades onde possivelmente se encontra vai diminuindo, aumentando a certeza sobre sua localização. Isso se deve ao fato de que as amostras num determinado instante carregam a informação de todo o histórico de observações do robô, de forma que, com o passar do tempo, haja informação suficiente para se determinar com alto nível de certeza onde o robô se encontra.

1.5.1 Sequência de passos da Localização Monte Carlo

Como todos os filtros Bayesianos, a MCL possui passos de predição e de correção (ou atualização). No passo de predição, as amostras da localização corrente, $\{X_t^s\}_{s=1}^n$, são geradas a partir de amostras $\{X_{t-1}^s\}_{s=1}^n$ do instante anterior, através da *simulação* do sistema dinâmico que rege a variável de estado X_t . Assim, cada nova amostra da localização é gerada amostrando-se uma vez o modelo probabilístico de movimento aplicado a cada amostra do instante anterior, de forma que

$$X_t^s \sim p(X_t = x | X_{t-1} = X_{t-1}^s, u_t), \quad (1.51)$$

para as amostras $s = 1, 2, \dots, n$. Observe que, na versão básica da MCL, cada amostra do instante anterior gera uma nova amostra, mantendo-se, assim, o seu número. Na teoria de filtro de partículas, a densidade usada para gerar amostras na fase de predição é denominada *proposta de distribuição*.

Então, no passo de correção, as observações instantâneas z_t são utilizadas para calcular os pesos w_t^s de forma a corrigir a distribuição das amostras para a distribuição alvo. Os novos pesos normalizados são calculados fazendo-se

$$w_t^s = \frac{w_{t-1}^s p(Z_t = z_t | X_{t-1} = X_{t-1}^s)}{\sum_{s=1}^n w_{t-1}^s p(Z_t = z_t | X_{t-1} = X_{t-1}^s)}. \quad (1.52)$$

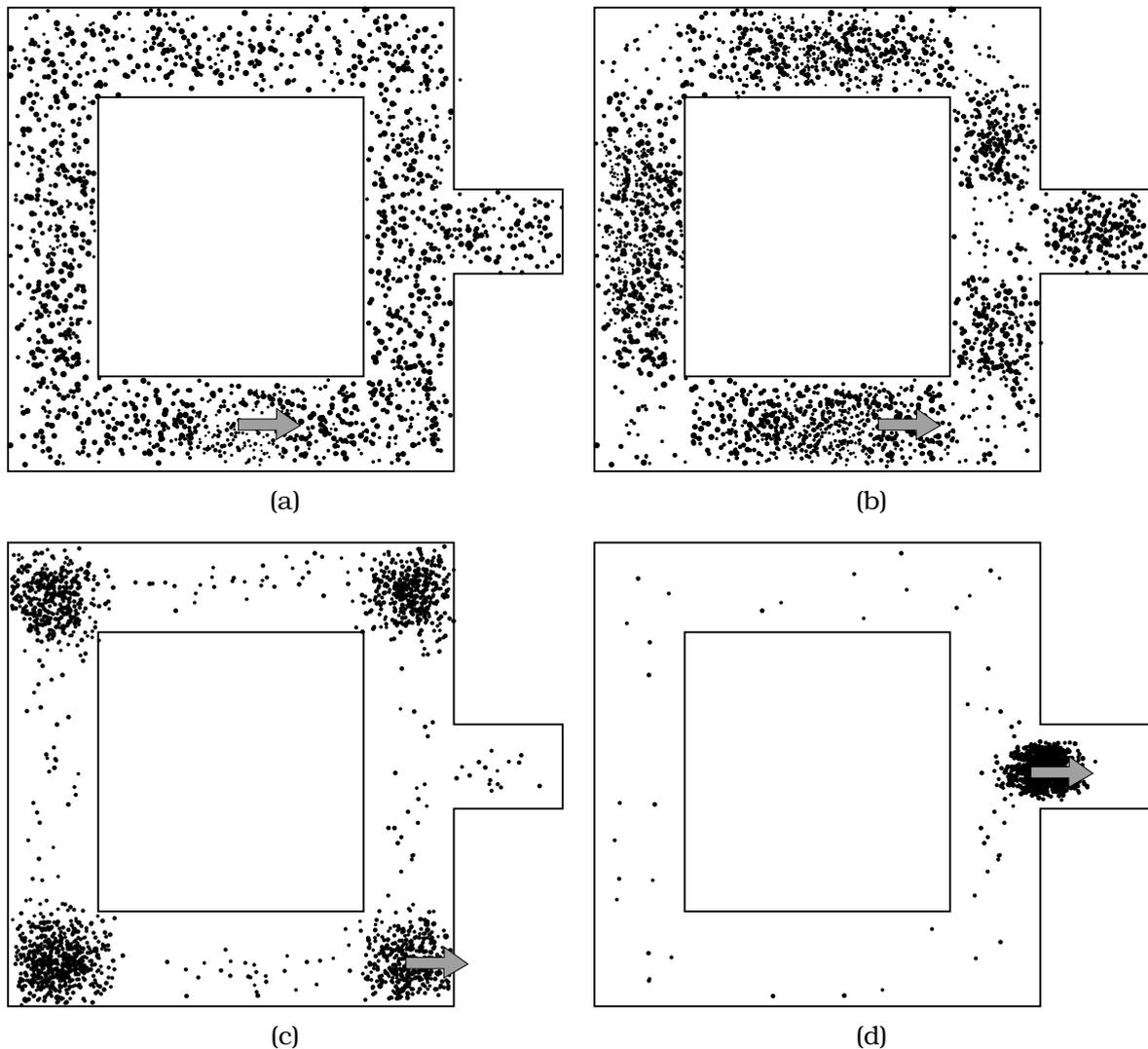


Figura 1.5: Ilustração do funcionamento da Localização Monte Carlo. Os pontos negros representam as amostras da localização do robô em cada instante, embora as amostras das orientações não estejam graficamente representadas. A flecha cinza representa a posição e a orientação corretas do robô em cada instante. (a) Inicialmente, a crença sobre a localização do robô está uniformemente distribuída em todo o ambiente. (b) Após mover-se um pouco a localização do robô ainda é incerta, mas as medições realizadas já permitem definir regiões com maior probabilidade. (c) Após o robô se mover até a próxima quina, ainda não é possível resolver com certeza sua localização, pois qualquer uma das quatro quinas ainda são localizações plausíveis. (d) Após mover-se até a entrada do corredor, as ambiguidades sensoriais são resolvidas e as partículas concentram-se em uma única localidade. Inspirada no experimento relatado em [34].

Além do filtro de partículas ser bastante flexível, permitindo a localização em áreas muito grandes e exigindo relativamente pouco espaço em memória, nota-se que sua implementação é simples e de rápida execução. No entanto, sua formulação não é exata, principalmente pelo fato de se dispor de um número finito de partículas. Por isso, uma parte das partículas tende a se degenerar com o tempo, amostrando regiões de probabilidade muito pequena, o que faz que seus pesos assumam valores cada vez menores. Com isso, essas partículas praticamente não mais influenciam no cálculo do valor esperado da localização, representando apenas custo computacional desnecessário.

Para atenuar o problema da degeneração das partículas, pode-se fazer, a cada passo, uma reamostragem delas, escolhendo-se aquelas que serão eliminadas e substituídas por cópias de partículas com maior peso. Essa reamostragem é feita construindo-se uma distribuição de probabilidade $P(s) = w_t^s$, posteriormente gerando-se n novas partículas $\{s_i\}_{i=1}^n \sim P(s)$ e escolhendo-se $X_t^i = X_t^{s_i}$, para $i = 1, 2, \dots, n$ como as novas amostras da localização do robô.

Caso a reamostragem ocorra a cada iteração, tem-se a versão *Sequential Importance Resampling* (SIR) do filtro de partículas. No entanto, a reamostragem às vezes possui um custo computacional que pode comprometer o desempenho da MCL. Por isso, algumas versões realizam o passo de reamostragem apenas se a degeneração das partículas for significativa. O Algoritmo 3 traz os passos da Localização Monte Carlo com um critério para reamostragem. Nesse algoritmo, é verificado o número efetivo de partículas $n_{eff} = 1 / (\sum_{s=1}^n (w_t^s)^2)$. Caso esse número caia abaixo de um valor limite n_{lim} , a reamostragem é realizada. Em geral, n_{lim} é escolhido como uma fração do número de partículas n .

1.6 Considerações Finais

Técnicas de localização baseadas no Filtro de Kalman e em Monte Carlo são extremamente populares nos projetos de robôs móveis que navegam em ambientes internos, como museus, hospitais e edifícios em geral. Estas são aplicações muito frequentes da robótica móvel e os sistemas de localização baseados nestas técnicas têm tido muito sucesso.

Entretanto, inúmeras outras técnicas também foram propostas [7] e aplicadas com sucesso, como a localização baseada em sinais ativos de balizas e a baseada em marcos.

A técnica de *localização baseada em sinais ativos de balizas* computa a localização absoluta do robô medindo a direção de incidência ou distância de três ou mais balizas transmissoras visíveis pelo receptor colocado no robô. As balizas são geralmente transmissoras de sinais de luz ou rádio frequência, e

Algoritmo 3 Etapas da Localização Monte Carlo com reamostragem.

Entradas: Sequência de observações \mathbf{z}^t , dados de controle \mathbf{u}^t , conjunto de amostras iniciais $\{X_0^s\}_{s=1}^n$, valor efetivo limite de partículas n_{lim} .

Saídas: Conjunto de amostras $\{X_t^s\}_{s=1}^n$, localização estimada \hat{X}_t .

Variáveis: τ, s, i : Inteiros.

Para $\tau \leftarrow 1$ a t , faça

1. Predição: Para $s \leftarrow 1$ a n , faça

 Simule a localização X_τ^s a partir da amostra $X_{\tau-1}^s$ e de u_τ usando o modelo de movimento, como em (1.51)

 Fim Para

2. Correção: Para $s \leftarrow 1$ a n , faça

 Calcule o peso w_τ^s a partir de (1.52)

 Fim Para

3. Calcule o número efetivo de partículas $n_{eff} = 1 / (\sum_{s=1}^n (w_\tau^s)^2)$

4. Se $n_{eff} < n_{lim}$, faça

 (a) Construa a distribuição de probabilidade $P(s) = w_\tau^s$

 (b) Amostre um conjunto de partículas $\{s_i\}_{i=1}^n \sim P(s)$

 (c) Construa um novo conjunto de amostras da localização do robô

$\{\bar{X}_\tau^i\}_{i=1}^n$:

 Para $i \leftarrow 1$ a n , faça $\bar{X}_\tau^i = X_\tau^{s_i}$, Fim Para

 Para $i \leftarrow 1$ a n , faça $w_\tau^i = 1/n$, Fim Para

 (d) Adote $\{\bar{X}_\tau^i\}_{i=1}^n$ como o novo conjunto de amostras da localização:

$\{X_\tau^s\}_{s=1}^n = \{\bar{X}_\tau^i\}_{i=1}^n$

Fim Para.

Calcule $\hat{X}_t = \sum_{s=1}^n X_t^s w_t^s$.

Retorne $\{X_t^s\}_{s=1}^n$ e \hat{X}_t .

devem ser colocadas em lugares conhecidos no ambiente. É um sistema bastante popular em navios e aeronaves (ex. GPS – *Global Positioning Systems*), oferecendo boa acurácia e alta confiabilidade nas medidas. Já a técnica de *localização baseada em marcos* utiliza estruturas distintas, passivas – chamadas *marcos* – que um robô pode detectar e identificar a partir de seus sensores externos. Os marcos geralmente possuem uma localização fixa e conhecida no ambiente, e a detecção de três ou mais deles permite que o robô se localize. Os marcos, naturais ou artificiais, são cuidadosamente escolhidos para que sejam de fácil detecção e identificação. Mesmo assim, a localização baseada em marcos exige mais processamento e pode ser menos precisa e acurada do que aquela que utiliza balizas ativas. Porém, possuem a vantagem de necessitar de pouca ou nenhuma modificação no ambiente em que o robô navega. No entanto, as técnicas de localização baseadas em mapas, aqui descritas, são mais robustas e flexíveis do que as baseadas em balizas ativas ou marcos passivos.

Algumas técnicas, ainda, utilizam abordagens híbridas, unindo a localização baseada em mapas com a baseada em marcos. Essas técnicas utilizam sensores externos diversos para sensoriar o ambiente e extrair, a partir da fusão de diferentes informações sensoriais, estruturas distintas que servem como marcos, os quais são inseridos no mapa, para navegações futuras. Este método é conhecido como *Localização e Mapeamento Simultâneos* – SLAM, indicando as iniciais do seu nome em inglês, *Simultaneous Localization and Mapping* – e possuem a vantagem de permitir que os robôs naveguem e explorem ambientes desconhecidos, dando-lhes maior autonomia e facilidade de aplicação.

Bibliografia

- [1] C. R. Allen and I. C. Leggett, “3D Scene Reconstruction and Object Recognition for Use with Autonomously Guided Vehicles (AGVs)”, in *Proc. IEEE IECON 21st International Conference on Industrial Electronics, Control, and Instrumentation*, 1995, pp.1219–1224. 6
- [2] H. L. Anderson, “Metropolis, Monte Carlo and the MANIAC”, Los Alamos Science, Tech. Rep. LA-UR-86-2600, 1986. 27
- [3] R. C. Arkin, “Motor Schema–Based Mobile Robot Navigation”, *International Journal of Robotics Research*, vol. 8, no. 4, pp.92–112, 1989. 1
- [4] N. Balakrishnan and V. B. Nevzorov, *A Primer on Statistical Distributions*. Hoboken, NJ: John Wiley & Sons, Inc., 2003. 19
- [5] R. J. G. Barra, C. H. C. Ribeiro, and A. H. R. Costa, “Fast Vertical Line Correspondence Between Images for Mobile Robot Localization”, in *Proc. 9th International IFAC Symposium on Robot Control (SYROCO2009)*, 2009, pp.153–158. 6
- [6] J.–L. Blanco, J.–A. Fernandez-Madrigal, and J. Gonzalez, “Toward a Unified Bayesian Approach to Hybrid Metric–Topological SLAM”, *IEEE Transactions on Robotics*, vol. 24, no. 2, pp.259–270, 2008. 7
- [7] J. Borenstein, H. R. Everett, and L. Feng, *Where am I? Sensors and Methods for Mobile Robot Positioning*. The University of Michigan, 1996. 30
- [8] R. A. Brooks, “A Robust Layered Control System for Mobile Robot”, *IEEE Transactions on Robotics and Automation*, vol. 2, no. 1, pp.14–23, 1986. 1
- [9] I. J. Cox and J.J. Leonard, “Modeling a Dynamic Environment Using a Bayesian Multiple Hypothesis Approach”, *Artificial Intelligence*, vol. 66, no. 2, pp.311–344, 1994. 9

- [10] J. L. Crowley, “World Modeling and Position Estimation for a Mobile Robot Using Ultrasonic Ranging”, in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 1989, vol. 2, pp.674–680. 9
- [11] A. J. Davison, “Real-time Simultaneous Localisation and Mapping with a Single Camera”, in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 2003, pp.1403–1410. 5
- [12] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, “A Solution to the Simultaneous Localization and Map Building (SLAM Problem)”, *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp.229–241, 2001. 5
- [13] G. L. Dudek, “Environment Representation Using Multiple Abstraction Levels”, *Proceedings of the IEEE*, vol. 84, no. 11, pp.1684–1704, 1996. 7
- [14] G. L. Dudek and M. Jenkin, *Computational Principles of Mobile Robotics*. Cambridge, MA: Cambridge University Press, 2000. 3, 15
- [15] A. Elfes, “Sonar-Based Real-World Mapping and Navigation”, *IEEE Journal of Robotics and Automation*, vol. 3, no. 3, pp.249–265, 1987. 5
- [16] J. Folkesson, P. Jensfelt, and H. I. Christensen, “Graphical SLAM Using Vision and the Measurement Subspace”, in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2005, pp.325–330. 6
- [17] D. Fox, W. Burgard, F. Dellaert, and S. Thrun, “Monte Carlo Localization: Efficient Position Estimation for Mobile Robots”, in *Proc. of the AAAI National Conference on Artificial Intelligence*, 1999, pp.343–349. 9
- [18] D. Fox, S. Thrun, W. Burgard, and F. Dellaert, “Particle Filters for Mobile Robot Localization”, in *Sequential Monte Carlo Methods in Practice*, Berlin: Springer, 2001, pp.401–428. 27
- [19] B. Kuipers, “Modeling Spatial Knowledge”, *Cognitive Science*, vol. 2, no. 2, pp.129–153, 1978. 6
- [20] J.-C. Latombe, *Robot Motion Planning*. Norwood, MA: Kluwer Academic Publishers, 1991. 9
- [21] A. Lazanas and J.-C. Latombe, “Landmark-Based Robot Navigation”, *Algorithmica*, vol. 13, no. 5, pp.472–501, 1995. 10
- [22] J. J. Leonard, I. J. Cox, and H. F. Durrant-Whyte, “Dynamic Map Building for an Autonomous Mobile Robot”, *International Journal of Robotics Research*, vol. 11, no. 4, pp.286–289, 1992. 5

- [23] J. J. Leonard and H. F. Durrant-Whyte, “Mobile Robot Localization by Tracking Geometric Beacons”, *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp.376–382, 1991. 9
- [24] H. P. Moravec, “Robot Spatial Perception by Stereoscopic Vision and 3D Evidence Grids”, Robotics Institute, Carnegie Mellon University, Tech. Rep. CMU-RI-TR-96-34, 1996. 5
- [25] J. I. Nieto, J. E. Guivant, and E. M. Nebot, “The HYbrid Metric Maps (HYMMs): A Novel Map Representation for Dense SLAM”, in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2004, vol. 1, pp.391–396. 7
- [26] S. T. Pfister, S. I. Roumeliotis, and J. W. Burdick, “Weighted Line Fitting Algorithms for Mobile Robot Map Building and Efficient Data Representation”, in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2003, vol. 1, pp.1304–1311. 6
- [27] J. M. Porta, and B. J. A. Kröse, “Appearance-Based Concurrent Map Building and Localization”, *Journal of Robotics and Autonomous Systems*, vol. 54, no. 2, pp.159–164, 2006. 5
- [28] A. Ranganathan, “*Probabilistic Topological Maps*”, PhD thesis, Georgia Institute of Technology, Atlanta, USA, 2008. 7
- [29] A. Ranganathan and F. Dellaert, “Semantic Modeling of Places using Objects”, in *Proc. of Robotics: Science and Systems*, 2007. 6
- [30] S. Se, D. Lowe, and J. Little, “Mobile Robot Localization and Mapping with Uncertainty using Scale-Invariant Visual Landmarks”, *International Journal of Robotics Research*, vol. 21, no. 8, pp.735–758, 2002. 5
- [31] A. H. P. Selvatici, “*Construção de Mapas de Objetos para Navegação de Robôs Móveis*”, Tese, Escola Politécnica da Universidade de São Paulo, São Paulo, Brasil, 2009. 6
- [32] R. Siegwart and I. Nourbakhsh, *Introduction to Autonomous Mobile Robots*. Cambridge, MA: The MIT Press, 2004. 8
- [33] P. Smith, M. Self, and P. Cheeseman, “Estimating Uncertain Spatial Relationships in Robotics”, in *Autonomous Robot Vehicles*, Berlin: Springer, 1990, pp.167–193. 5
- [34] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA: The MIT Press, 2005. 14, 24, 29

- [35] S. Thrun, C. Martin, Y. Liu, D. Hahnel, R. Emery-Montemerlo, D. Chakrabarti, and W. Burgard, “A Real-Time Expectation-Maximization Algorithm for Acquiring Multiplanar Maps of Indoor Environments with Mobile Robots”, *IEEE Transactions on Robotics and Automation*, vol. 20, no. 3, pp.433–443, 2004. 6
- [36] R. F. Vassallo, “*Uso de Mapeamentos Visuomotores com Imagens Omnidirecionais para Aprendizagem por Imitação em Robótica*”, Tese, Universidade Federal do Espírito Santo, Vitória, Brasil, 2004. 4, 7
- [37] S. Vasudevan, S. Gachter, M. Berger, and R. Siegwart, “Cognitive Maps for Mobile Robots – An Object Based Approach”, *Journal of Robotics and Autonomous Systems*, vol. 55, no. 5, pp.359–371, 2007. 6
- [38] N. L. Werneck and A. H. R. Costa, “Mapping with Monocular Vision in Two Dimensions”, *International Journal of Natural Computing Research*, 2011. 6