

2 Fundamentação Teórica

2.1 Robótica Móvel

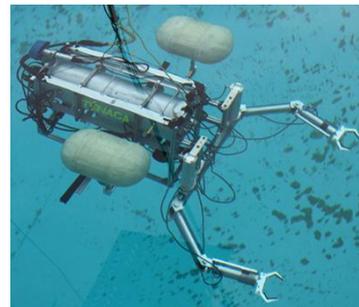
A robótica móvel tem ganhado bastante destaque nos últimos anos. Cada vez mais presentes em ambientes residenciais ou em locais de alto risco para o ser humano: robôs que desarmam bombas, espões em missões aéreas não tripuladas ou ainda submarinos exploradores, que chegam a locais antes inexplorados pelo ser humano. Além disso, certamente serão utensílios comuns em nossas residências em forma humana, como é o caso do ASIMO da Honda, como animais de estimação como o AIBO da Sony ou ainda para realizar tarefas específicas, como é o caso do robô Roomba da iRobot, que navega no ambiente residencial aspirando o pó. As possibilidades são inesgotáveis e essa é uma tendência que dificilmente será revertida.



(a)



(b)



(c)



(d)



(e)



(f)

Figura 4 - Exemplo de robôs. (a) AIBO - Robô cachorro da Sony, (b) ASIMO - humanóide da honda, (c) Submarino explorador desenvolvido pelo departamento de robótica da Universidade japonesa Ritsumeikan, (d) Predator – Veículo aéreo não tripulado utilizado pela força aérea dos Estados Unidos, (e) SUGV – Robô militar da iRobot especializado em situações arriscadas como desarmamento de bombas, (f) Roomba – Robô aspirador de pó, também da iRobt.

Segundo a *RIA (Robotics Institute of America)* apud (Ulrich Nehmzow, 2003):

“Um robô é um dispositivo reprogramável, multitarefa desenvolvido para manipular materiais, peças ou ferramentas ou um dispositivo especializado que através de movimentos programados e variados realizam uma variedade de tarefas.”

Para um robô móvel, além da descrição dada acima também é imprescindível que este se locomova no meio, seja por meio de rodas (carros), pernas (humanóides) ou veículos voadores (Veículos Aéreos não Tripulados – VANTs). Entretanto, nem todos os robôs móveis são autônomos, pois não são capazes de tomar decisões em situações imprevistas. A grande maioria são teleguiados e muitos outros são apenas programados a realizar tarefas mecânicas e rotineiras

Alguns pesquisadores acreditam que a robótica deve ser puramente reativa (Brooks, 1991). Isso significa que o robô não teria, necessariamente, que tomar decisões complexas ou manipular e manter grande quantidade de informação, ao invés disso apenas reagiria ao meio. Reagindo de forma adequada, o robô completaria sua tarefa sem que, necessariamente, tenha sido explicitamente programado para fazê-la. O grande desafio para tal arquitetura é identificar quais tipos de reações seriam necessárias para completar tarefas específicas (Meyer, Husbands, & Harvey, 1998). Contudo, os robôs terrestres que têm tido mais sucesso em completar tarefas complexas são aqueles equipados com rodas e programados para uma missão específica.

Os robôs móveis terrestres equipados com duas rodas de comando diferencial são os mais amplamente utilizados, por sua simplicidade. Sua simplicidade não reside apenas em sua forma construtiva, mas também na facilidade do seu controle. Robôs de comando diferencial são equipados com no mínimo duas rodas comandadas independentemente, usualmente paralelas e com eixos colineares. Uma possível terceira roda, chamada castor, pode ser utilizada

para manter o equilíbrio sem adicionar graus de liberdade ao robô. Um tanque de guerra é um típico veículo de comando diferencial que não possui roda castor e é controlado pela diferença entre as velocidades das esteiras. Os tanques com esteiras não possuem rodas castores pois apenas as suas esteiras são suficientes para manter sua estabilidade.

Assumindo um robô de comando diferencial, como mostrado na Figura 5, é possível obter, de forma aproximada, qual será seu estado no tempo t , $P_t(x, y, \theta)$, sabendo quais foram as velocidades aplicadas na roda direita (v_R) e esquerda (v_L) e a seu estado anterior, P_{t-1} , utilizando as Equações (1), (2) e (3), a este processo dá-se o nome de odometria.

$$x_t = x_{t-1} + v_t \cdot \cos(\theta_{t-1}) \cdot \Delta t \quad (1)$$

$$y_t = y_{t-1} + v_t \cdot \sin(\theta_{t-1}) \cdot \Delta t \quad (2)$$

$$\theta_t = \theta_{t-1} + w_t \cdot \Delta t \quad (3)$$

Onde:

$$v = \frac{v_R + v_L}{2} \quad (4)$$

$$w = \frac{v_L - v_R}{2 \cdot l} \quad (5)$$

Onde l é a distância entre as rodas e Δt o tempo decorrido. As Equações (1) a (3) são chamadas de função de transição de estado do robô.

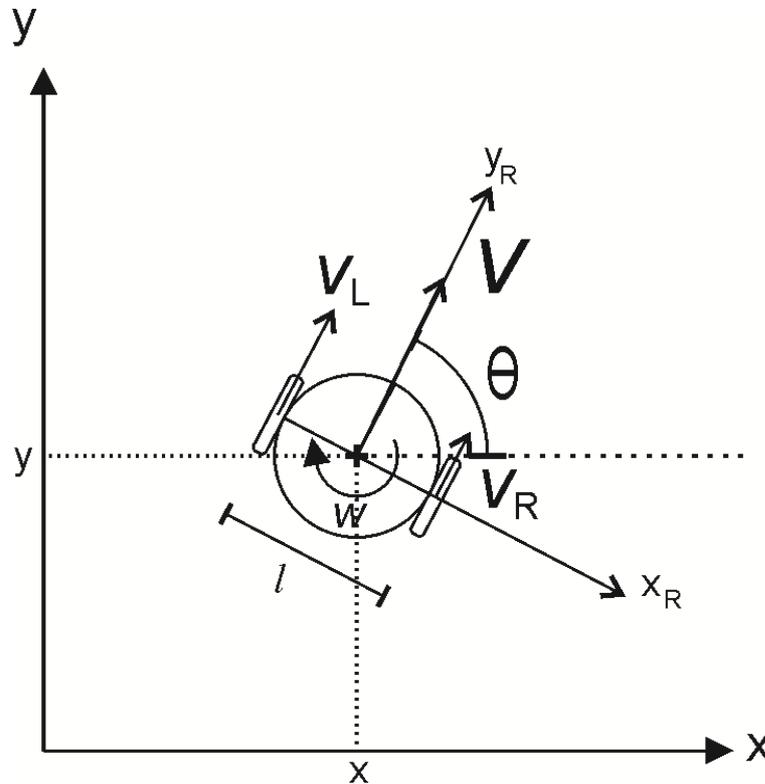


Figura 5 – Cinemática Direta de um robô móvel com drive diferencial.

2.2 Localização e Mapeamento Simultâneos,

O problema da localização e mapeamento simultâneos, também conhecido como SLAM (*Simultaneous Localization and Mapping*), é um dos problemas fundamentais na robótica móvel. A problemática ocorre quando o robô não tem acesso a um mapa do ambiente em que está inserido e também não conhece a sua posição. As informações disponíveis são apenas: medições dos sensores e sinais de comando. Usualmente considera-se que o robô conhece sua posição inicial, ou que a sua posição inicial está no centro do sistema de coordenadas do mundo.

A complexidade do SLAM reside na imprecisão dos sensores e dos componentes mecânicos que formam o robô. Idealmente é possível conhecer a posição exata do robô sabendo apenas a sua posição inicial e as leituras dos *encoders* através da odometria, como foi mostrado na seção 2.1.

No entanto os sensores possuem limitações físicas e estão sujeitos a ruídos que acabam por adicionar erros em suas medições. Um das limitações físicas dos *encoders* é a sua resolução, a resolução dos *encoders* do LEGO *Mindstorms* é de 1° , sendo impossível para o robô, por exemplo, detectar um movimento de suas

rodas de apenas 0,5°. Como outros fatores podem-se citar: o derrapar das rodas, deformação das rodas do robô e as pequenas diferenças construtivas entre dois motores, que mesmo sendo aparentemente similares sempre terão respostas ligeiramente diferentes para um mesmo sinal de entrada.

Devido a esses e outros fatores a posição do robô só pode ser estimada de forma aproximada, sendo necessária uma segunda forma de localização, uma forma que não leve em consideração apenas as informações oriundas do próprio robô e sim informações estáticas do ambiente ou marcos do ambiente. Uma vez que o robô seja capaz de detectar marcos no ambiente ele pode usá-los para filtrar os erros da odometria acumulados. Existem diversas técnicas na literatura capaz de resolver este problema, evidentemente dependendo dos tipos de sensores disponíveis na plataforma robótica um método será mais indicado que outro. Dentre os métodos clássicos dois se destacam: EKF-SLAM e o Filtro de partícula, também conhecido como *Rao-Blackwellized particle filter*. O EKF-SLAM utiliza o filtro de kalman estendido para atualizar a posição do robô sempre que um marco é detectado. O EKF-SLAM tornou-se bastante popular porque foi uma das técnicas pioneiras na resolução deste problema, apresentando bons resultados e mostrando-se bastante eficiente (Thrun, 2005). O filtro de partículas ficou popular por causa de duas grandes vantagens: facilidade de implementação e a capacidade de resolver o problema do sequestro do robô, onde o robô é reposicionado de forma a perder completamente suas referências, problema este que o FKE-SLAM não é capaz de resolver.

Para detectar marcos o robô faz uso de seus sensores exteroceptivos. A plataforma robótica utilizada neste trabalho utiliza apenas sonares como sensores exteroceptivos.

Assim, a próxima seção discorrerá a respeito de sonares, mostrando seus pontos fortes e seus pontos fracos, bem como as possíveis formas de interpretar a informação proveniente do mesmo.

2.3 O Sonar

O SONAR (*SONd Navigation And Ranging*) é um dispositivo formado por um transmissor e um receptor de ondas sonoras de alta frequência, entre 40 e 180kHz (Ulrich Nehmzow, 2003).

O sonar, por ser um sensor amplamente utilizado e bastante robusto, principalmente em uso subaquático, possui vasta literatura e grande aplicabilidade nas mais diversas áreas: engenharia, medicina e militar. O sonar pode ser utilizado para tarefas simples, como medir a distância a um obstáculo (como é o caso dos sensores de estacionamento, que está se popularizando em automóveis) ou para tarefas mais complicadas, como a ultrassonografia (exame médico capaz de mostrar as estruturas dos órgãos internos de um paciente através do eco gerado pelo ultrassom).

O princípio de funcionamento do sonar é baseado no tempo que o som leva para percorrer o espaço entre o emissor do pulso sonoro e o obstáculo. Conhecendo a velocidade do som no meio e o tempo percorrido para o som retornar, é possível calcular a distância entre o sonar e o obstáculo, como mostrado na Figura 6.

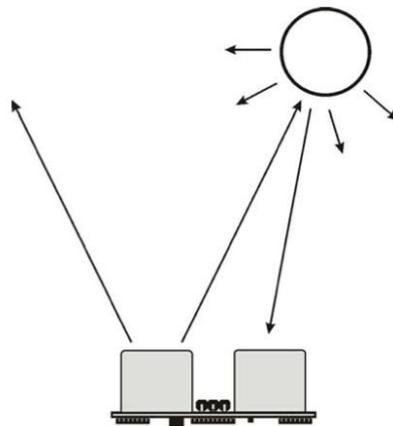


Figura 6 - Propagação da onda sonora até o obstáculo, (Specifications, 2009).

Este cálculo é conhecido como *Time of Flight – TOF*, literalmente traduzindo é o “Tempo de Voo” do pulso originado no sonar até o obstáculo e é calculada segundo a Equação (6).

$$d = \frac{1}{2} \cdot v_{som} \cdot \Delta t \quad (6)$$

Onde v_{som} representa a velocidade do som no meio e Δt o tempo decorrido da emissão do pulso de onda até o retorno do eco ao receptor.

Usualmente é feito um chaveamento do estado de transmissão para recepção do som (Ulrich Nehmzow, 2003). O tempo para emitir o som é dado por t_{burst} e durante este tempo nada é captado, assim o alcance mínimo do sonar pode ser calculado segundo a Equação (7):

$$d_{min} = \frac{1}{2} \cdot v_{som} \cdot t_{busrt} \quad (7)$$

Após emitir o pulso sonoro o sonar é chaveado para a função de recepção e fica neste estado também por apenas alguns milissegundos, (t_{wait}), até que o som atinja o obstáculo e retorne. A máxima distância que o sonar pode detectar é função do tempo de espera e é calculado pela Equação (8):

$$d_{max} = \frac{1}{2} \cdot v_{som} \cdot t_{wait} \quad (8)$$

Por fim, a resolução do sonar é calculado pela Equação (9):

$$d_{res} = \frac{d_{max}}{q} \quad (9)$$

Onde q é o número de quantizações possível. Este número está intimamente ligado ao conversor Analógico Digital utilizado. Um conversor com 8 bits é capaz de representar até 256 níveis de energia. Se cada unidade representar um centímetro pode-se representar até 256 centímetros de distância em uma leitura do sonar.

Tomando o sonar PING))) (Specifications, 2009) como exemplo e suas velocidade sendo $t_{busrt} = 115\mu s$ e $t_{wait} = 18,5ms$ e a velocidade do som a $346m/s$ a uma temperatura de $25^{\circ}C$, temos $d_{min} = 19,895cm$ e $d_{max} = 3200,5cm$. Contudo, estes valores são teóricos e usualmente usa-se um intervalo de confiança.

Devida à propagação omnidirecional do som a área útil de medição do sonar não é pontual, consistindo de um lóbulo principal e outros lóbulos menores laterais em formato de um taco de baseball.

Entretanto, considera-se apenas a área de um cone inscrito dentro do lóbulo principal, como mostrado na Figura 7.

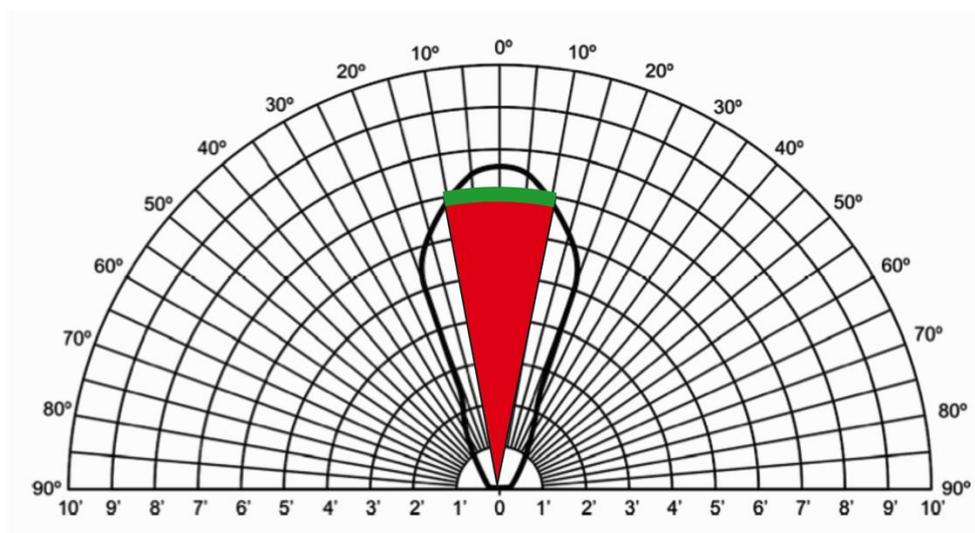
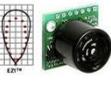
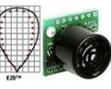


Figura 7 - Cone inscrito dentro do lóbulo principal do sonar, adaptado de (Siegwart & Nourbakhsh, 2004).

Na Figura 7, a área em vermelho é considerada uma região vazia e o limite do cone, região em verde, é a região onde o obstáculo se encontra.

Devido às limitações citadas acima, o sonar vem sendo gradativamente substituído por sensores mais precisos e mais robustos, como é o caso do *Lazer Range Finder - LRF* e pela visão computacional. Entretanto, o sonar ainda é um grande atrativo, principalmente pelo baixo custo. Para exemplificar, tomemos a tabela abaixo como referencia, nela encontram-se alguns modelos de sonares e LRF comerciais.

Figura	Modelo	Range (cm)	Valor (\$)
	Devantech SRF02	15 a 600	24,22
	Maxbotix LV-MaxSonar-EZ1	15 a 645	24,95
	Maxbotix LV-MaxSonar-EZ0	15 a 645	26,08

	Maxbotix LV-MaxSonar-EZ2	15 a 645	27,95
	Hagisonic HG-B40C	15 a 700	31,31
	Parallax PING	3,3 a 330	29,99
	Hokuyo URG-04LX-UG01	2 a 560	1245,50
	Hokuyo UBG-05LN	10 a 500	1855,00
	Microsoft Kinect	120 a 350	140,00

Tabela 1 - Comparativo entre sonares e LRF comerciais. Preços e características retirados da loja virtual robotshop.com.

O LRF mais acessível é cerca de 50 vezes o preço do sonar de mais baixo custo. Evidentemente não é possível utilizar apenas um sonar. Geralmente utiliza-se de 7 a 20 sonares montados em forma circular ao redor do corpo do robô. Entretanto, mesmo utilizando-se 20 sonares, o que daria um total de \$484,40, este ainda seria 3 vezes mais barato que o LRF e, teoricamente, cobriria a mesma região.

O uso da visão computacional com câmeras estereoscópicas é bastante difundido na robótica. O Kinect, acessório do videogame Xbox360 da Microsoft, é uma das mais poderosas câmeras estereoscópicas do mercado. Devido ao baixo custo e grande capacidade o Kinect vem sendo amplamente utilizado em pesquisas relacionadas a robótica e visão computacional (Hayashibe & Poignet,

2011), (Waibel, 2011). Entretanto, este deve estar conectado a um computador com um bom processamento e as condições do ambiente podem ser desfavoráveis para o seu uso, por exemplo, ambientes com pouca luminosidade, um típico problema da visão computacional.

Alguns sensores além de caros podem ser de difícil acesso. Seja pela dificuldade de importa-los ou por construírem peças integrantes em armamentos militares, como é o caso da central de navegação inercial.

2.3.1

Limitações do Sonar

Existem diversas incertezas associadas às leituras do sonar. A primeira, e mais significativa, é a incerteza da localização do obstáculo. Em uma leitura típica do sonar, como mostrado na Figura 7, o obstáculo que refletiu o som pode estar posicionado em qualquer ponto dentro da região em verde.

Este problema é agravado à medida que a distância para o obstáculo cresce, já que o semicírculo da extremidade do sonar cresce na proporção de $\frac{1}{2}\beta \cdot d$. Tomando um valor típico para β igual a $0,3491 \text{ rad}$ (aproximadamente 20 graus), o perímetro do semicírculo passa de $8,7275 \text{ cm}$ para um objeto a 50 cm de distância, para $43,6375 \text{ cm}$ para um objeto a 250 cm de distância, aumentando em até 500% o perímetro a ser buscado.

As chamadas *specular reflections* ocorrem quando o ângulo de incidência do som em superfície plana é muito grande, geralmente acima de 45° , o que faz com que o eco seja refletido para longe do sonar, inviabilizando a medida da distância, como é ilustrado na Figura 8.

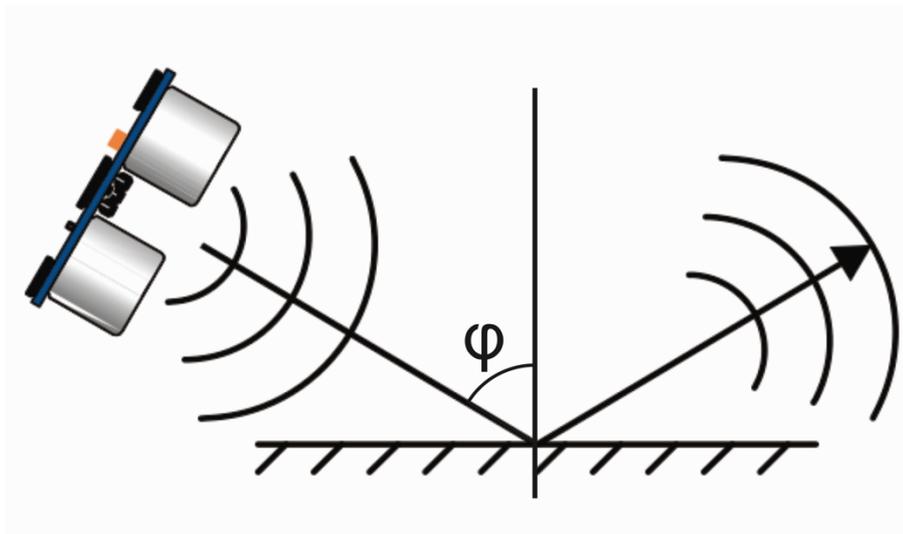


Figura 8 - Specular Reflection (Specifications, 2009)

O problema do *specular reflections* é a falsa interpretação de uma região livre, onde, na realidade, há um obstáculo. Em casos críticos pode haver a danificação do sistema embarcado ou do objeto atingido (por exemplo, uma porta de vidro).

Specular reflections também são frequentes quando o sonar está posicionado de frente para um canto côncavo, como mostrado na Figura 9. Estas situações costumam ser bastante problemáticas, isso porque a leitura do sonar será uma leitura válida³, mas o obstáculo se encontrará mais próximo do que a medida realizada.

³ Diz-se como uma leitura válida, todas as leituras que são maiores ou iguais ao alcance mínimo e menores ou iguais ao alcance máximo do sonar.

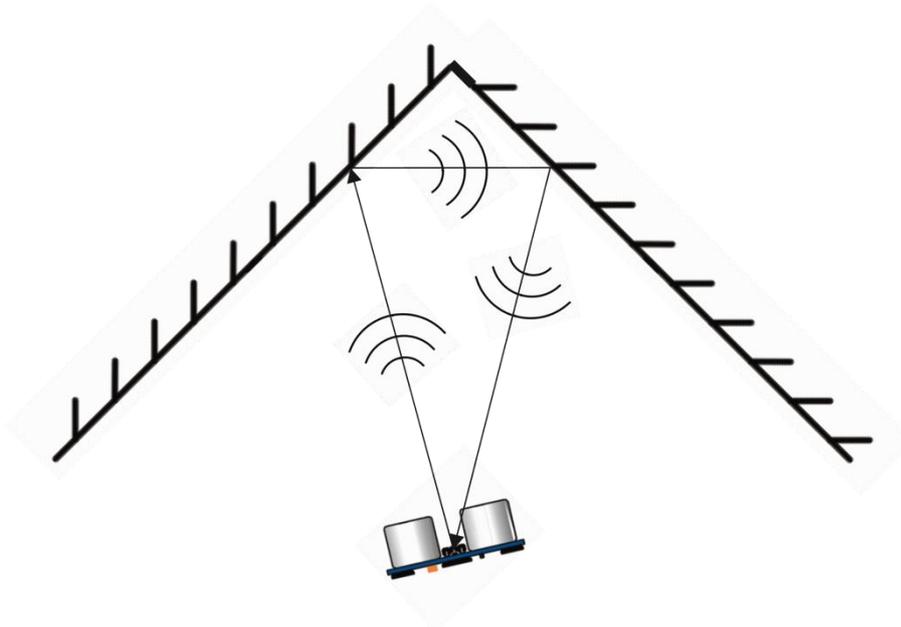


Figura 9 - Specular Reflections em cantos côncavos.

Usualmente um sonar apenas não é suficiente para resolver problemas básicos de navegação, como desvio de obstáculos ou localização. Por esta razão usa-se uma grande quantidade de sonares, usualmente montados em forma circular sobre o robô, ou um sonar em uma plataforma girante. Em um sistema robótico equipado com vários sonares, facilmente o sinal emitido por um sonar pode, por acidente, ser captado por outro. *Cross-Talk*, como é conhecido esse tipo de ruído, pode ser resolvido por pelo menos duas formas diferente: ou espera-se um tempo de segurança entre duas leituras consecutivas, para que o som emitido seja completamente dissipado, ou codifica-se o pulso de onda, de forma que cada sonar capte apenas ondas sonoras com codificações específicas. Usualmente esperar um intervalo até emitir a segunda rajada sonora já é suficiente para evitar este problema. Mas quando o tempo é um fator crucial ou é necessário fazer a leitura de todos os sonares simultaneamente ou ainda quando mais de um robô equipado com sonares é utilizado simultaneamente no mesmo ambiente, é de fundamental importância que o sinal seja codificado.

2.4 Representação do Ambiente

A representação do ambiente é a forma como o robô modela o mundo a seu redor e como ele “enxerga” os marcos que irá utilizar para sua navegação.

A escolha da representação do ambiente deve considerar diversos fatores, entre eles os principais são (Siegwart & Nourbakhsh, 2004):

- A precisão do mapa deve estar de acordo com a precisão necessária para o robô atingir o seu objetivo;
- O robô deve possuir sensores capazes de monitorar o ambiente com a precisão exigida;
- A complexidade e o detalhamento do mapa devem estar de acordo com o hardware escolhido e com a tarefa a ser executada.

Como já foi mostrado, usar o sonar como sensor exteroceptivos não é uma tarefa simples. Felizmente, vários trabalhos evoluíram de forma significativa de modo que tornou-se possível a um robô ter apenas sonares embarcados (A Elfes, 1989), (Moravec & Elfes, 1985), (J L Crowley, 1989), (Lee & Song, 2010). A forma com que cada autor tratou a descrição do ambiente está ligada ao tipo de representação adotada para representar o mapa.

Segundo (Siegwart & Nourbakhsh, 2004) existem duas grandes possibilidades em relação à representação do ambiente: Contínua e em Grade.

2.4.1

Representação Contínua

Na representação contínua o ambiente é descrito por primitivas geométricas como: retas, círculos, quadrados ou outros polígonos. Este tipo de abordagem é bastante comum em robôs equipados com *LRF*, uma vez que esse é capaz de fornecer uma grande quantidade de pontos em uma única leitura, como pode ser visto na Figura 10.

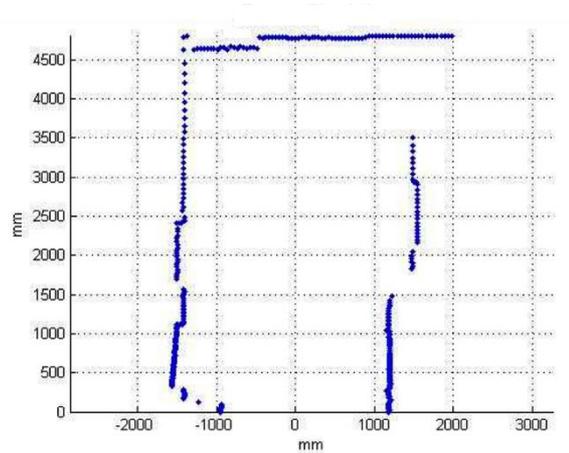


Figura 10 - Leitura típica de um LRF. O sensor encontra-se na posição (0,0) orientado em 90° e possui ângulo de varredura de 180° .

O LRF é o mais difundido sensor na robótica móvel, pois possui uma grande capacidade de varrer o ambiente e é de extrema utilidade para Localização e Mapeamento, pois é capaz de detectar formas geométricas com bastante facilidade.

A quantidade de trabalhos que utilizam LRF para localização ou mapeamento é extremamente extensa. Como exemplos tem-se (Lee & Song, 2010) e (Arras & Siegwart, 1997) que utilizaram um robô equipado com LRF para localização em um ambiente descrito por retas, (Zhang & Ghosh, 2000) descrevem retas e círculos como primitivas para resolver o problema do SLAM. (Graovac, Jurić-Kavelj, & Petrović, 2010) busca por retas através da Transformada de Hough para localização.

Contudo, diferente do LRF, cada leitura do sonar diz muito pouco sobre o obstáculo captado. Usualmente, extrai-se apenas um ponto de cada leitura e dificilmente este ponto estará precisamente localizado no local onde o obstáculo foi sentido pelo sonar. Frequentemente, para que uma reta seja extraída através de leituras oriundas de sonares, é necessário que várias medições sejam feitas e de ângulos diferentes. Alguns dos trabalhos que utilizam a representação contínua fazendo o uso apenas de sonares serão brevemente descritos a seguir.

(J L Crowley, 1989) utilizando um robô com 24 sonares distribuídos em torno de todo o robô, extrai de cada leitura do sonar um ponto parametrizado por: x , y , σ_w e σ_D . Onde x e y são as coordenadas do ponto, σ_w a incerteza da localização do ponto ao longo do semicírculo do cone do sonar e σ_D é a incerteza correspondente a profundidade da leitura. Os parâmetros σ_w e σ_D correspondem aos eixos de uma elipse centrada no ponto (x, y) , como mostrada na Figura 11.

Entretanto, para facilitar os cálculos, foi considerado que σ_W e σ_D possuem o mesmo valor, extraído de forma experimental, e igual a:

$$\sigma_W = 0.10 + d \cdot \tan^{-1}(5^\circ) \quad (10)$$

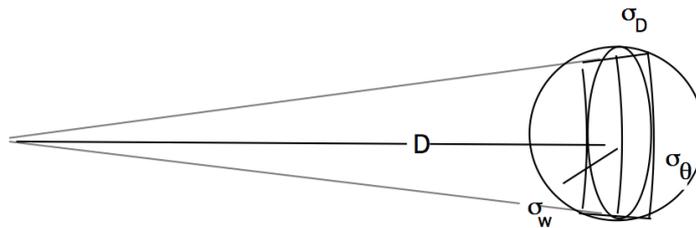


Figura 11 - Modelo da leitura de um sonar e sua incerteza (J L Crowley, 1989).

Utilizando regras bem definidas, três pontos são unidos para criar um seguimento de reta. A reta apenas é criada se os pontos estiverem suficientemente próximos (de forma que não seja possível que o robô passe por entre eles) e se a distância a um dos pontos for menor ou igual a um valor pré-definido ϵ . Um total de onze parâmetros são utilizados para modelar um segmento de reta. A grande quantidade e a redundância dos parâmetros são propositais e utilizados para facilitar a comparação entre as retas.

(Meng, Sun, & Cao, 2000) utilizam quatro padrões no ambiente para localizar um robô equipado com oito sonares em um ambiente estruturado. Os quatro padrões possíveis são: planos, cantos convexos, cantos côncavos e cilindros. Essas características são traduzidas nas primitivas retas, pontos, pontos e círculos respectivamente e um Filtro de Kalman Estendido é utilizado para localizar um robô em um ambiente estruturado.

(Lee & Song, 2010) desenvolveram uma forma bastante eficiente de mapear regiões salientes do ambiente, como: cantos convexos e cilindros. Através de um modelo denominado *Footprint Association (FPA)*, com duas ou mais leituras do sonar busca-se por uma circunferência que tangencia todos os semicírculos dos sonares, como mostrado na Figura 12. Um sistema de equações encontram os valores de q_o , q_r e ϕ_2 a partir do valor de ϕ_1 . Sendo que o valor de ϕ_1 pode variar de ϕ_{1min} a $\phi_{1min} + \beta$, onde β é a abertura do sonar. Como não é possível saber exatamente qual o valor de ϕ_1 , é feita uma busca exaustiva, subdividindo o ângulo do cone do sonar. Um círculo válido é aquele cujo valor do seu raio esteja

em um intervalo de confiança $p_{min} \leq q_r \leq p_{max}$. Nos experimentos feitos os valores adotados foram 3 e 30cm para p_{min} e p_{max} respectivamente.

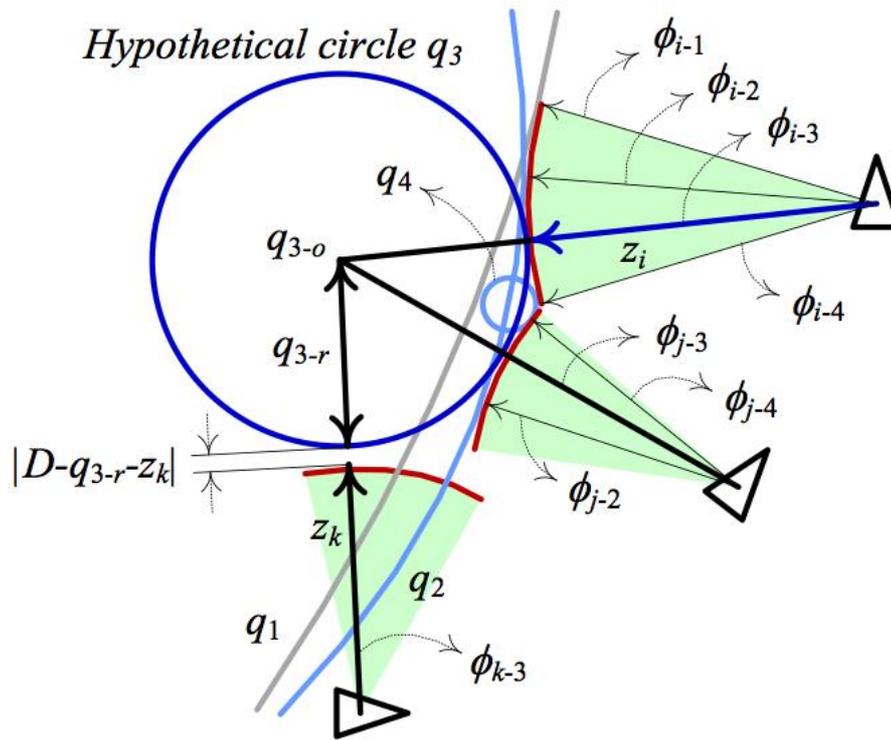


Figura 12 - Exemplo do modelo FPA, (Lee & Song, 2010).

Então, uma terceira leitura é usada e se a distância entre o semicírculo do sonar z e a extremidade do círculo de raio q_r , dado por $(|D - q_r - z_k|)$, for menor que um limiar (2cm por exemplo), então este círculo é dito como um marco e é gravado no mapa para referência. No exemplo da Figura 12 apenas o círculo de raio q_{3-r} satisfaz todas as restrições. Através destes marcos um Filtro de Kalman Estendido é usado para localização do robô.

Armazenar primitivas como: retas, pontos e círculos consomem pouca memória. Além disso, estas primitivas permitem o uso de filtros probabilísticos, como o Filtro de Kalman, para resolver o problema de Localização e do SLAM.

A despeito da pouca memória consumida, se a quantidade de marcos no ambiente for muito grande, outros problemas deverão ser enfrentados. Quando o robô detectar algum marco é imprescindível que o mesmo saiba a qual marco ele está se referindo e, caso a quantidade de primitivas contidas no mapa seja muito grande, fazer a correspondência entre elas será uma tarefa computacionalmente

custosa. Ameniza-se o problema diminuindo o espaço de busca de acordo com a pose estimada e os possíveis marcos ao alcance do robô.

2.4.2 Representação em Grade

A representação em Grade pode ser visto como um caso particular da estratégia de decomposição do ambiente. (Siegwart & Nourbakhsh, 2004) mostra quatro formas possíveis de decompor o ambiente.

Uma das possibilidades é a decomposição do ambiente em forma de células exatas (*exact cell decomposition*) selecionando limites entre as células baseado em uma regra geométrica, como mostrado na Figura 13.

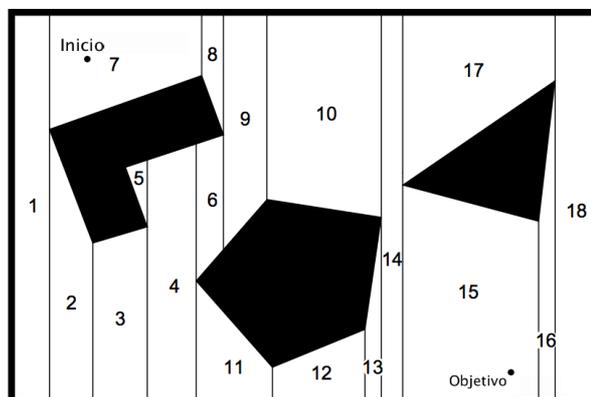


Figura 13 – Exemplo da decomposição exata (*exact cell decomposition*). Adaptado de (Siegwart & Nourbakhsh, 2004)

Outra possibilidade é decompor o ambiente de forma topológica (*topologic decomposition*), preocupando-se apenas com as informações relevantes para a sua localização ao invés de tentar retratá-lo em formas geométricas, como é mostrado na Figura 14.

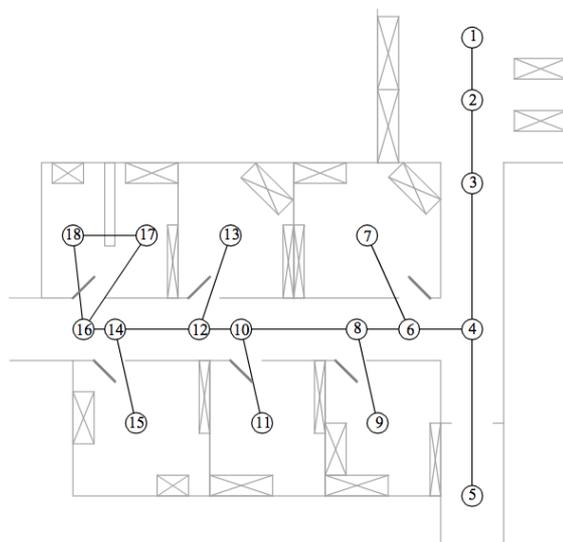


Figura 14 - Representação Topológica, (Siegwart & Nourbakhsh, 2004).

Essa representação gera um grafo onde os nós são partes do ambiente e as arestas denotam a transição de uma região do ambiente para outra.

A representação em grade ou decomposição fixa (*fixed decomposition*) divide o espaço em pequenas regiões de tamanho fixo, como é mostrado na Figura 15, de forma que cada célula (cada posição da grade) seja descrita como ocupada ou vazia. Usualmente usa-se uma matriz com zeros e uns, “0” significa uma célula vazia e “1” ocupada, por esta razão a representação em grade também é conhecida como grade de ocupação (*occupancy grid maps*). Na Figura 15, as células em branco estão vazias, logo possui valor igual a “0”, e as células em preto estão ocupadas, logo possuem valor igual a “1”.

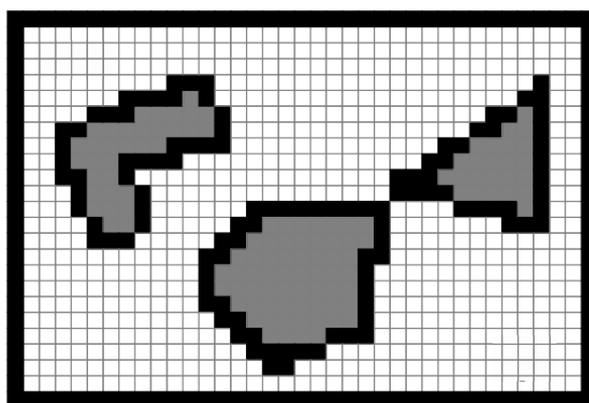


Figura 15 - Representação em grid, adaptado de (Siegwart & Nourbakhsh, 2004).

A representação em grade é amplamente utilizada na robótica móvel pela reciprocidade que possui com sensores de distância, como o sonar e o LRF. De forma ideal, no local onde foi detectado um objeto, a célula que corresponde a

este local pode ser considerada ocupada, e o espaço que vai do sensor ao obstáculo está vazio.

Sabe-se, entretanto, que a afirmativa acima não é sempre verdadeira quando se trata de leituras feitas por sonares. Problemas como o *specular reflections* podem resultar em uma leitura completamente errada do ambiente, colocando em cheque todo o sistema. Numerosos trabalhos procuram formas eficientes de tratar as leituras do sonar afim de não comprometer o processo por ruídos indesejados.

(Moravec & Elfes, 1985) desenvolveram uma forma eficiente de transformar as leituras do sonar em um mapa em grade de ocupação, dividindo as regiões em vazias, ocupadas ou desconhecidas. Para cada leitura, eles dividem o cone do sonar em duas regiões: ocupada e vazia. A probabilidade de cada célula estar ocupada ou vazia é calculada segundo uma função quadrática que decai com o quadrado da distância. As células mais propensas a estarem, de fato, vazias são aquelas que estão mais próximas ao eixo principal do sonar e do próprio sonar. Já as células mais propícias a estarem ocupadas estão mais próximas ao local de reflexão do som sob o eixo principal do sonar, como mostrado na Figura 16. Para compor o mapa final, uma regra simples é aplicada, como mostrado na Equação (11).

$$Map(X, Y) = \begin{cases} Occ(X, Y) & se\ Occ(X, Y) \geq Emp(X, Y) \\ -Emp(X, Y) & se\ Occ(X, Y) < Emp(X, Y) \end{cases} \quad (11)$$

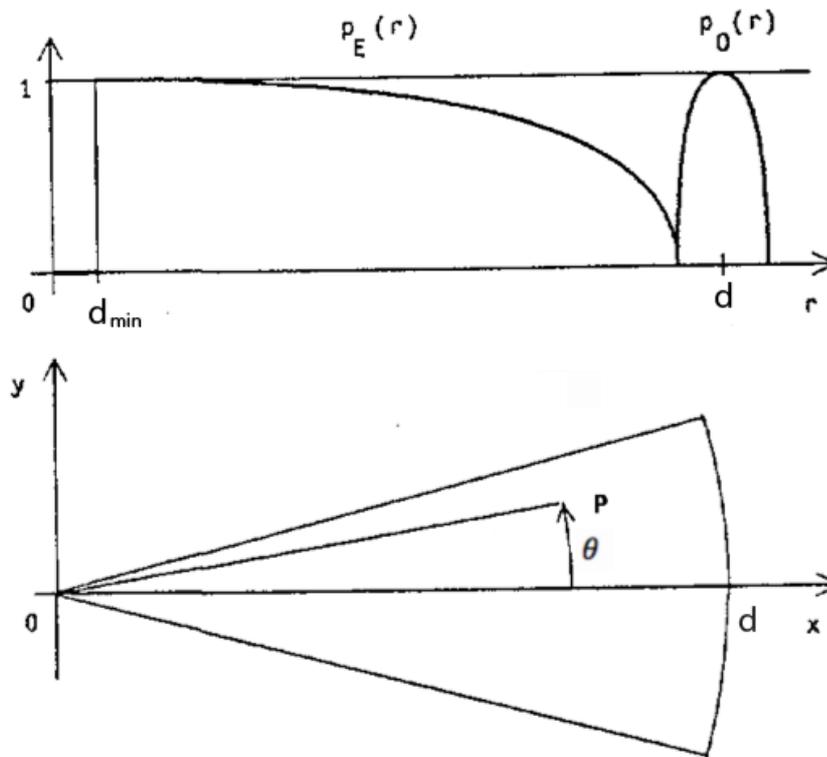


Figura 16 - Modelagem de uma leitura do sonar em um mapa em grid. Adaptado de (Moravec & Elfes, 1985).

Mais tarde (A Elfes, 1991) remodelou sua representação, tornando possível a representação do ambiente em apenas uma matriz. Considerando que cada célula é probabilisticamente independente das demais quando se trata de sua ocupação, Elfes considerou que a probabilidade inicial de cada célula está ocupada é $1/2$, e a leitura do sonar aproxima-se agora de uma gaussiana (e não mais de uma função quadrática), como mostrado na Figura 17.

Desta forma, o mapa final pode ser interpretado da seguinte forma: Se a probabilidade da célula for maior que $1/2$ ela está ocupada, se o valor for $1/2$ então nada se pode dizer a respeito, e caso seja menor que $1/2$ a célula encontra-se livre.

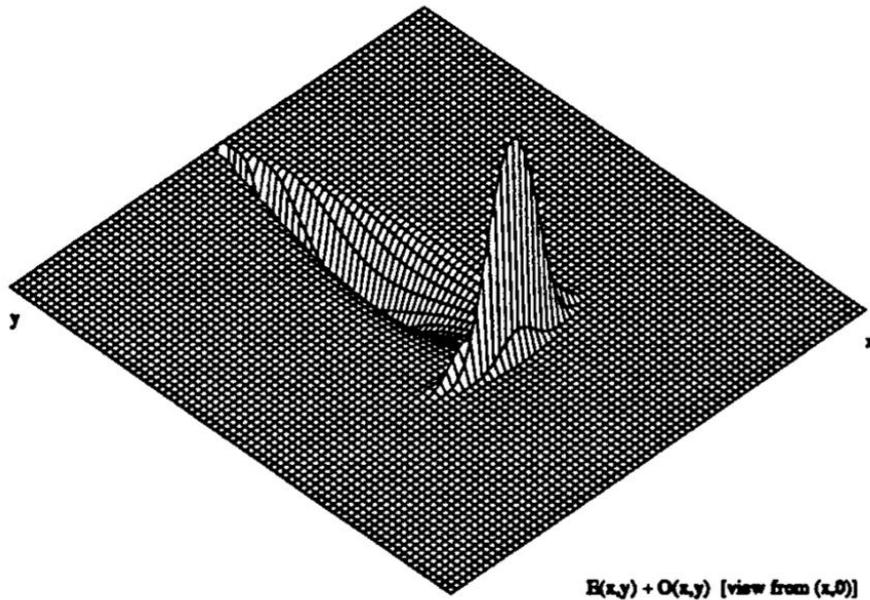


Figura 17 - Leitura típica do sonar modelado por (A Elfes, 1991) em 3D.

(Konolige, 1997) desenvolveu um sofisticado modelo para interpretação das leituras do sonar, apesar de à primeira vista parecer semelhante ao modelo de (Moravec & Elfes, 1985), este traz algumas características interessantes:

- O erro da leitura do sonar aumenta proporcionalmente à distância do obstáculo;
- A probabilidade de detecção torna-se menor com a distância.

O modelo ainda considera que todas as leituras do sonar possuem uma pequena probabilidade de serem *specular reflections*, tornando-o ainda mais realista. Outra grande mudança em relação ao modelo de (Moravec & Elfes, 1985) é que toda a análise é feita a partir do logaritmo da verossimilhança, fazendo com que células vazias tenham valor negativos e células ocupadas valores positivos.

A Figura 18 mostra uma leitura típica utilizando este modelo. Supondo um obstáculo nas posições 100, 200 e 300 cm de distância do sonar e que o modelo está parametrizado para um sonar com alcance máximo de 400 cm.

É importante notar como a probabilidade de detecção decai quando o obstáculo está mais distante do sonar.

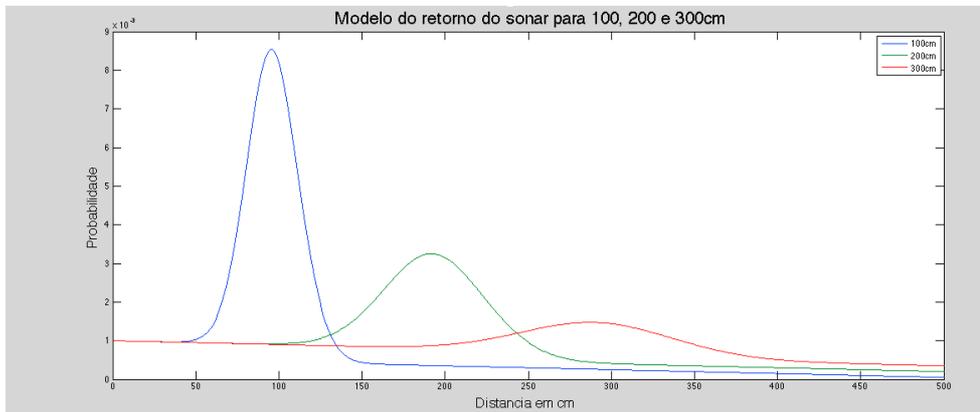


Figura 18 - Exemplo de leitura 2D do sonar utilizando o modelo de (Konolige, 1997).

O modelo de (Konolige, 1997), contudo, é computacionalmente mais custoso e consome mais memória em relação ao modelo proposto por (Moravec & Elfes, 1985).

2.5 Filtro de Kalman Estendido

O Filtro de Kalman é uma forma particular da implementação do Filtro de Bayes. O Filtro de Bayes calcula a distribuição de probabilidade da crença do estado do sistema com as informações das medições feitas, dos comandos tomados e do estado anterior do sistema. O algoritmo da Tabela 2 mostra os principais passos do Filtro de Bayes.

```

1  Algoritmo Filtro_de_Bayes( $bel(x_{t-1}), u_t, z_t$ ):
2  para todo  $x_t$  faça:
3       $\overline{bel}(x_t) = \int p(x_t | u_t, x_{t-1}) \cdot bel(x_{t-1}) \cdot dx_{t-1}$ 
4       $bel(x_t) = \eta \cdot p(z_t | x_t) \cdot \overline{bel}(x_t)$ 
5  Fim para
6  Retorne  $bel(x_t)$ 

```

Tabela 2 - Algoritmo geral para o Filtro de Bayes, adaptado de (Thrun, 2005)

O Filtro de Bayes é um processo recursivo, que usa como uma de suas entradas a crença do estado no tempo $t - 1$, o comando u no tempo t e a medida z feita no tempo t . O termo η é um fator de normalização, uma vez que $bel(x_t)$ é uma função densidade de probabilidade e sua integral deve ser igual a um. O passo referente a linha 3 da Tabela 2 corresponde a predição do estado no tempo t dado o estado anterior. O segundo passo, linha 4, é chamado de atualização da medida, ou seja, é incluída a informação das medidas feitas pelo sistema para atualizar a crença bel . E então o resultado é retornado, linha 6.

O Filtro de Kalman é considerado também um filtro gaussiano, pois se assume que as variáveis do problema são variáveis aleatórias gaussianas parametrizadas segundo uma média μ e variância σ , e, inicialmente, foi desenvolvido para resolver problemas lineares (Kalman, 1960).

Contudo, as equações de transição de estado de um robô e as equações de observações não são lineares. Por isso não é possível aplicar diretamente o Filtro de Kalman.

Para aplicar o Filtro de Kalman em sistemas não lineares é necessário linearizar o sistema no ponto em que o processo se encontra, aproximando-o a um sistema linear. A grande maioria dos trabalhos trata a linearização do sistema através da Expansão de Taylor de primeira ordem.

A Série de Taylor é uma técnica usada para aproximar funções e pode conter infinitos termos, como mostrado na Equação (12).

$$f(x) = f(a) \cdot (x - a)^0 + \frac{f'(a) \cdot (x - a)^1}{1!} + \frac{f''(a) \cdot (x - a)^2}{2!} + \dots + \frac{f^n(a) \cdot (x - a)^n}{n!} \quad (12)$$

Como se utiliza apenas a primeira ordem da série, a linearização do sistema ficará algo semelhante à Equação (13).

$$f(x) = f(a) + f'(a) \cdot (x - a)^1 \quad (13)$$

A Tabela 3 mostra os principais passos do Filtro de Kalman Estendido.

1	Algoritmo Filtro_de_Kalman_Estendido($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$)
2	$\bar{\mu}_t = g(u_t, \mu_{t-1})$
3	$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + V_t R_t V_t^T$
4	$K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$
5	$\mu_t = \bar{\mu}_t + K(z_t - h(\mu_t))$
6	$\Sigma_t = (I - K_t H_t) \bar{\Sigma}_t$
7	Retorne (μ_t, Σ)

Tabela 3 - Algoritmo Filtro de Kalman Estendido, adaptado de (Thrun, 2005)

Assim como o Filtro de Bayes, o Filtro de Kalman Estendido tem como entrada a crença do estado anterior, dado pelos parâmetros μ_{t-1} e Σ_{t-1} , da gaussiana que representa a variável aleatória do estado do sistema, o comando u_t e a medição z_t no tempo t .

As variáveis de estado do sistema (x , y e θ) são variáveis aleatórias gaussianas. Os parâmetros μ e Σ , então, são os parâmetros de uma gaussiana de média μ e covariância Σ .

As linhas 2 e 3 do algoritmo da Tabela 3 referem-se a atualização a priori, ou predição. Nesta etapa a transição de estado do sistema é feita sem que nenhuma medição seja levada em consideração, utilizando apenas o estado no passo anterior, μ_{t-1} , e o comando, u_t .

A função de transição de estado $g(u_t, \mu_{t-1})$ na linha 2 é função do comando u_t e da média da variável aleatória μ_{t-1} no estado anterior. As Equações (1) a (3) podem ser utilizadas como função de transição de estado para um robô de comando diferencial, como foi mostrado na Secção 2.1.

Para a atualização da covariância Σ_t é necessário calcular a derivada da função de transição de estado $g(u_t, \mu_{t-1})$ em relação ao estado do sistema, G_t , e em relação aos comandos, V_t , a fim de linearizar o sistema via Expansão de Taylor. Como a função de transição de estado é multivariada as derivadas de G_t e V_t tornam-se matrizes Jacobianas, dadas pelas Equações (14) e (15).

$$\nabla G_t = \begin{bmatrix} \frac{\partial \mu_{x_{t-1}}}{\partial x} & \frac{\partial \mu_{x_{t-1}}}{\partial y} & \frac{\partial \mu_{x_{t-1}}}{\partial \theta} \\ \frac{\partial \mu_{y_{t-1}}}{\partial x} & \frac{\partial \mu_{y_{t-1}}}{\partial y} & \frac{\partial \mu_{y_{t-1}}}{\partial \theta} \\ \frac{\partial \mu_{\theta_{t-1}}}{\partial x} & \frac{\partial \mu_{\theta_{t-1}}}{\partial y} & \frac{\partial \mu_{\theta_{t-1}}}{\partial \theta} \end{bmatrix} \quad (14)$$

$$\nabla V_t = \begin{bmatrix} \frac{\partial \mu_{x_{t-1}}}{\partial v} & \frac{\partial \mu_{x_{t-1}}}{\partial w} \\ \frac{\partial \mu_{y_{t-1}}}{\partial v} & \frac{\partial \mu_{y_{t-1}}}{\partial w} \\ \frac{\partial \mu_{\theta_{t-1}}}{\partial v} & \frac{\partial \mu_{\theta_{t-1}}}{\partial w} \end{bmatrix} \quad (15)$$

Na segunda parte do algoritmo, descrito na Tabela 3, as medições feitas são incorporadas a estimação do estado, linhas 4 a 6. A função $h(\bar{\mu}_t)$ é chamada de função de observação, pois através da crença do estado do sistema ela deve ser capaz de descrever a posição do marco em relação ao robô. Ao subtrairmos o valor medido pelo sistema, z_t do resultado da função $h(\bar{\mu}_t)$ têm-se a inovação do sistema, ou seja, a discrepância entre o estimado e o observando. A inovação é

multiplicada pelo ganho de Kalman, K_t , e então somado à média $\bar{\mu}_t$ para se obter a nova média do sistema: μ_t .

O ganho ótimo de Kalman, K , é calculado para minimizar de forma ótima o erro quadrático entre z_t e $h(\bar{\mu}_t)$ (ou seja, minimizar a diferença entre o medido e o estimado). Assim como foi necessário calcular a derivada da função de transição de estado, será também necessário calcular a derivada da função de observação em relação à média $\bar{\mu}_t$ do sistema. A função $h(\bar{\mu}_t)$ também costuma ser uma função multivariada, e nestes casos, a sua derivada também será uma matriz Jacobiana. A função de observação $h(\bar{\mu}_t)$ dependerá das variáveis do marco, sendo assim, ela será específica para cada problema. No Capítulo 3 ela será calculada para a representação de retas utilizada no modelo proposto.

Os valores de R_t e Q_t são as covariâncias de transição de estado e da medição respectivamente. Estes valores de covariâncias representam o grau de incerteza que é inserido no sistema à medida que é incorporada uma nova transição de estado ou uma nova medição.

Para exemplificar o papel da covariância de transição de estado, suponha um robô deslocando-se de forma linear, sem que nenhuma observação seja feita. Neste caso, apenas a parte a posteriori do filtro de Kalman pode ser utilizada e a atualização da média é feita pela função de transição de estado, acumulando o erro gerado pela covariância R_t . Como não há detecção de marcos, a incerteza relativa ao seu deslocamento tenderá apenas a aumentar. A Figura 19 mostra esse processo, onde a elipse ao redor do robô representa a região de confiança de 99%.

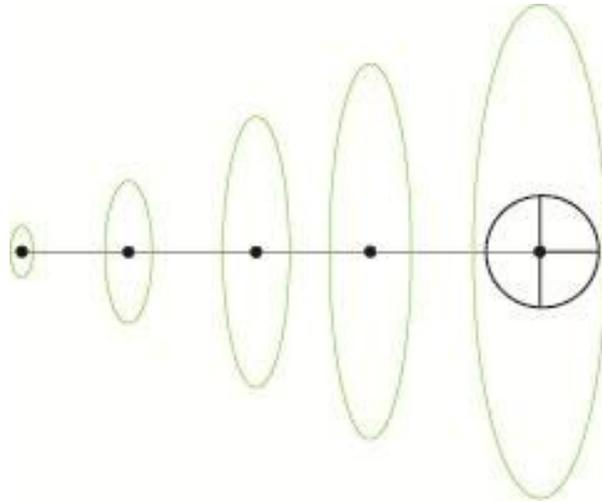


Figura 19 - Aumento da covariância ao longo do deslocamento linear.

Uma das grandes dificuldades quando se utiliza o FKE é saber quais valores de R_t e Q_t utilizar. Seus valores alteram-se dependendo do sistema. Usualmente as covariâncias de medição e de transição de estado são obtidas de forma experimental.

De modo geral o Filtro de Kalman Estendido procura minimizar o erro entre o medido e o estimado. Para exemplificar, suponha que um robô se desloca de um ponto A para um ponto B e que o mesmo tenha conhecimento e uma parede que se encontra à sua direita, como mostrado na Figura 20 (a). Através das leituras dos *encoders*, estima-se que o robô encontra-se na posição B, como mostrado na Figura 20 (a). Entretanto, um erro é encontrado entre a distância medida pelos sensores do robô e a distância estimada através da crença do estado do robô após a transição de estado A para B, como mostrado na Figura 20 (b).

Assim, o Filtro de Kalman Estendido muda a estimação do estado do robô de B para C, diminuindo o erro entre o estimado e o medido.

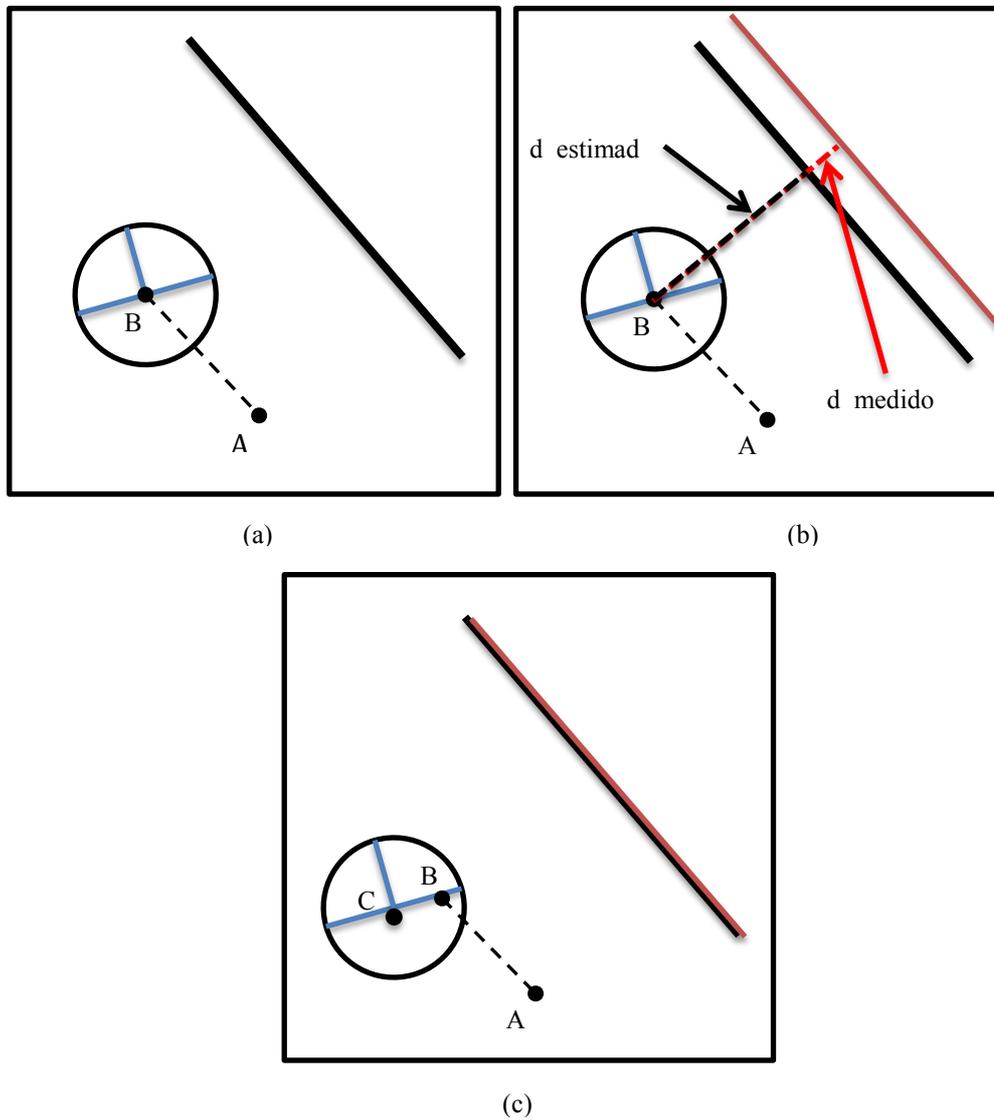


Figura 20 – Exemplo de correção de trajetória via Filtro de Kalman Estendido.

Neste capítulo foram apresentadas algumas das dificuldades e características da robótica móvel, e algumas das soluções propostas para torná-la viável. Foram mostradas também as dificuldades de utilizar sonares como sensores exteroceptivos, bem como alguns dos modelos mais tradicionais encontrados na literatura para descrevê-lo. No próximo capítulo será mostrado como as adversidades mostradas aqui foram tratadas, bem como uma descrição detalhada do método proposto.