INTRODUÇÃO À ROBÓTICA MÓVEL

Aula 25

Edson Prestes
Departamento de Informática Teórica
http://www.inf.ufrgs.br/~prestes
prestes@inf.ufrgs.br

Odometria and Métodos de Dead-Reckoning

- Odometria permite boa precisão em curtas distâncias e possui baixo custo.

- Porém, devido à integração incremental do movimento do robô, ela leva ao acumulo de erros proporcional à distância navegada.

- Ela é amplamente usada por sistemas de navegação em : Algoritmos de localização, mapeamento, exploração, etc.;

Odometria and Métodos de Dead-Reckoning

- Erros sistemáticos e não sistemáticos
  - Odometria é baseada na idéia de que as revoluções da rodas do robô podem ser convertidas em deslocamento linear relativo ao chão.
  - Esta idéia é limitada, pois considere o caso onde o robô está navegando por um chão sujo de óleo.
  - Além disso, existem outras fontes de erros que podem gerar imprecisões na conversão das revolução em deslocamento linear: erros sistemáticos e não sistemáticos.

Odometria and Métodos de Dead-Reckoning

- Fontes de Erros Sistemáticos
  - Diâmetro desigual das rodas;
  - Medida do diâmetro modelado difere da medida do diâmetro real;
  - Desalinhamento das rodas;
  - Resolução finita do encoder;
  - Taxa de amostragem das leituras do encoder finita,etc.

Odometria and Métodos de Dead-Reckoning

- Fontes de Erros Não-Sistemáticos
  - Navegação em pisos irregulares;

  - Derrapagem das rodas devido à:
    - Presença de óleo no chão;
    - Acelerações bruscas;
    - Manobras rápidas;
    - Forças externas (interação com objetos), etc.

Odometria and Métodos de Dead-Reckoning

- Erros sistemáticos e não-sistemáticos
  - Na maioria das superfícies de ambientes internos erros sistemáticos contribuem mais para erros de odometria que erros não sistemáticos.
  - Em superfícies irregulares, erros-não sistemáticos são dominantes.
  - Erros não-sistemáticos podem aparecer inesperadamente. Portanto são mais difíceis de serem modelados.

Odometria and Métodos de Dead-Reckoning

- Erros sistemáticos e não-sistemáticos

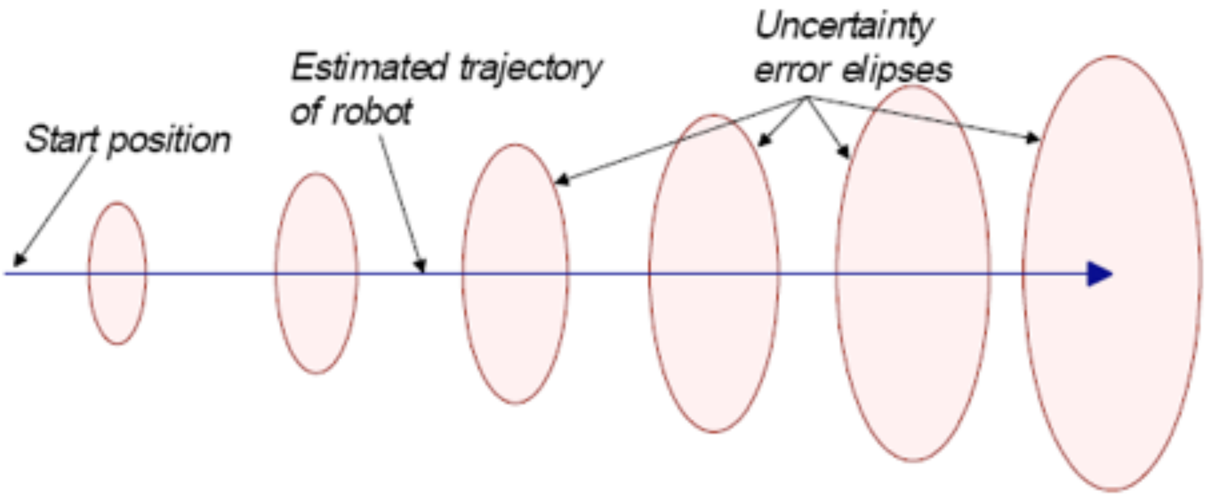


Figura extraída de [4]

Localização

- É um componente essencial para um robô ser completamente autônomo.
- Se o robô não sabe sua posição no ambiente ele não pode realizar nenhuma tarefa de navegação
- A odometria por si só não é suficiente para ser usada como estimativa da posição do robô, pois, em geral, ela é contaminada por erros sistemáticos e não sistemáticos.
- Os principais algoritmos são baseados em processo de Filtragem Bayesiana.

INSTITUTO DE INFORMÁTICA UFRGS

Localização

Figuras extraídas de [3]

Localização – Estimação Probabilística

- Ao invés de manter uma simples hipótese de onde o robô está no mundo, mantém uma distribuição de probabilidade sobre o espaço de todas as hipóteses.
- Este tipo de representação leva em consideração às incertezas presentes na movimentação e no sensoriamento do robô
- Esta distribuição pode ser multimodal para indicar situações ambíguas ou unimodal quando a certeza sobre a postura do robô é alta.

Localização – Estimação Probabilística

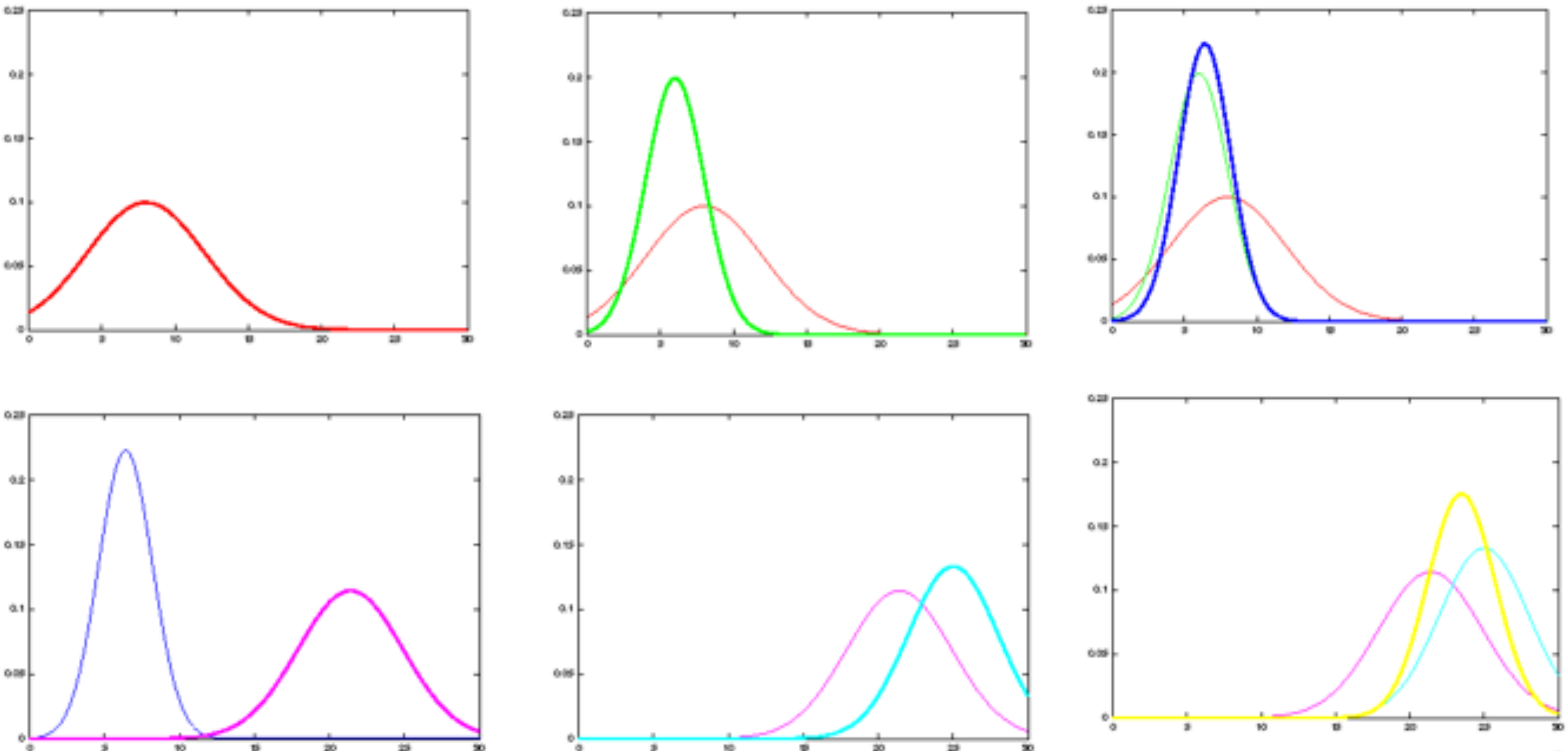
- Considere que o robô conheça sua posição inicial. Logo,a PDF terá um pico estreito(sharp) no local correspondente.

- Quanto mais estreito e alto maior a certeza sobre a estimativa da posição do robô.

- Quando o robô se desloca, o local do pico se desloca e devido a incerteza na movimentação, ele se tornará menor e mais largo. Quanto maior a velocidade do robô e o ruído, mais rápido o pico ficará largo.

- Para evitar isto, é necessário integrar nesta PDF informação sensorial. Isto compensa a perda de informação, fazendo com que o pico se torne mais estreito.

Localização – Estimação Probabilística



Figuras extraídas de [4]

Localização – Filtragem de Kalman

- Filtragem de kalman fornece uma abordagem recursiva para a estimação de estados de um sistema dinâmico na presença de ruídos.
- Ele é caracterizado por manter uma estimativa do estado corrente do sistema x e uma matriz de covariância P do erro de estimativa.
- Podemos visualizar o Filtro de Kalman como sendo aquele que fornece como saída uma função de densidade probabilística Gaussiana com média x e covariancia P

Localização – Filtragem de Kalman Linear

Considere o vetor $x(k) \in R^n$ como o estado do robô no instante $k \in \mathbf{N}$.
A evolução do estado do robô e das medidas sensoriais é dada por

$$x(k+1) = F(k)x(k) + G(k)u(k) + v(k)$$
$$y(k) = H(k)x(k) + w(k)$$

Onde

- $u(k) \in R^m$ é a entrada para o sistema (velocidades, torque, etc);
- $y(k) \in R^p$ é a saída do sistema e contém os valores dados pelos sensores do robô
- $F(k) \in R^{n\times n}$ codifica a dinâmica do sistema
- $G(k) \in R^{n\times m}$ descreve como as entradas afetam a dinâmica.
- $v(k) \in R^n$ é o ruído gaussiano com média 0 e matriz de covariância V(k).
- $H(k) \in R^{p\times n}$ descreve como o vetor de estado é mapeado para a saída
- $w(k) \in R^p$ é o ruído gaussiano com média 0 e matriz de covariância W(k).

Localização – Filtragem de Kalman Linear

Para determinar a melhor estimativa existem dois problemas :

- presença de ruídos desconhecidos v(k) e w(k), que devem ser eliminados pelo Filtro.

- em geral o estado x(k) não é observável diretamente a partir das saídas pois H(k) pode não ser invertível

O estado estimado deve ser reconstruído usando o histórico temporal dos sinais y(k) e u(k) com os parâmetros F(k), G(k), H(k) , V(k) e W(k). Esta reconstrução é feita por um dispositivo chamado observador.

O filtro de Kalman é tanto observador quanto filtro.

Localização – Filtragem de Kalman Linear Simples Observador

Considere um sistema discreto sem ruído

$$x(k+1) = F(k)x(k) + G(k)\,u(k)$$

$$y(k) = H(k)x(k)$$

H é assumido ser uma matriz full rank a cada instante k. Logo, o determinante de H não é 0.

O observador executará um processo de dois passos.

- Dada uma estimativa do estado corrente

$$\hat{x}(k \mid k)$$

Irá gerar uma predição de acordo com a dinâmica do sistema

$$\hat{x}(k+1|k) = F(k)\hat{x}(k|k) + G(k)u(k)$$

Localização – Filtragem de Kalman Linear – Simples Observador

O segundo passo é corrigir a predição baseada na saída y(k+1) para gerar a próxima estimativa

$$\hat{x}(k+1 \mid k+1)$$

É importante considerar que dada a saída y(k+1) o estado do sistema é restrito ao seguinte hiperplano

$$\Omega = \{x \in R^n | H(k+1)x = y(k+1)\}$$

Portanto, a melhor estimativa $\hat{x}(k+1 \mid k+1)$ será o ponto no hiperplano mais próximo de $\hat{x}(k+1 \mid k)$

Localização – Filtragem de Kalman Linear – Simples Observador

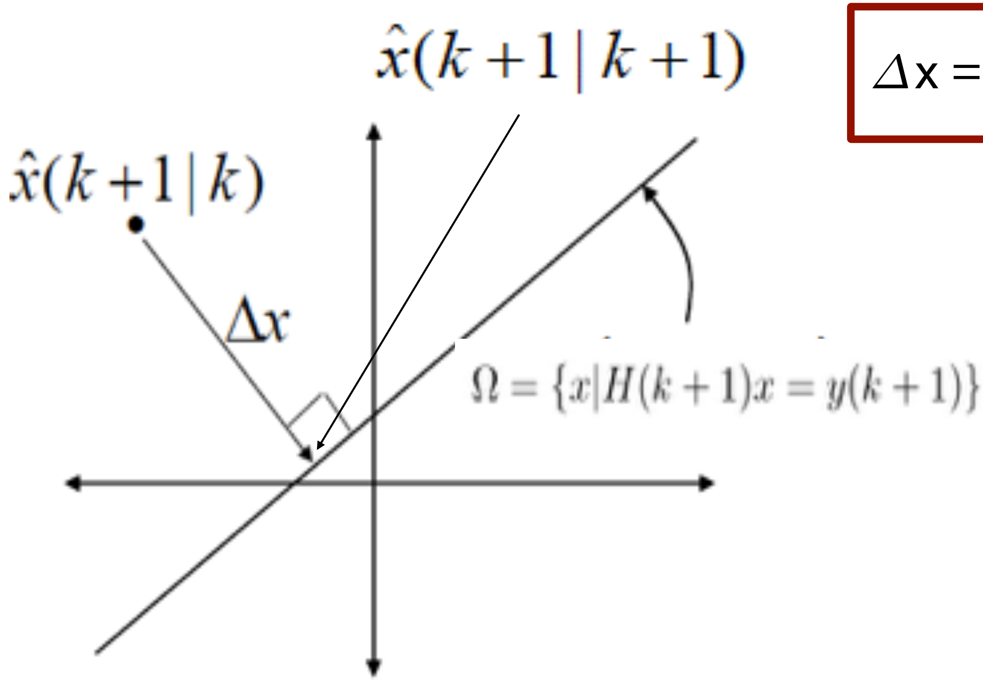


Figura extraída de [1]

$$\boxed{\Delta x = \hat{x}(k+1|k+1) - \hat{x}(k+1|k)}$$

Δx é perpendicular à Ω

$$\Delta x \in column(H(k+1)^T)$$

column(H(k+1)) é o espaço de colunas de H(k+1), ou seja, o espaço que corresponde a todas as combinações lineares das colunas de H.

Logo,

$$\Delta x = H(k+1)^T \gamma$$

para $\gamma \in R^p$

Localização – Filtragem de Kalman Linear – Simples Observador

Considere a inovação

$$v = y(k+1) - H(k+1)\hat{x}(k+1|k)$$

Como sendo a diferença entre o que os sensores reportam e o que reportariam se a predição estivesse certa.

Sabemos que $\Delta x = H(k+1)^T \gamma$

Logo γ, pode ser escrito como uma função linear de v, i.e.,γ = Kv, para K$\in R^{p\times p}$. Portanto temos,

$$\Delta x = H(k+1)^T K v$$

Lembrando que

$$\hat{x}(k+1|k+1) = \hat{x}(k+1|k) + \Delta x$$

Localização – Filtragem de Kalman Linear – Simples Observador

$$\Delta x = H(k+1)^T K v$$

K deve ser escolhido de forma que

$$H(k+1)(\hat{x}(k+1|k) + \Delta x) = y(k+1)$$

$$H(k+1)\Delta x = y(k+1) - H(k+1)\hat{x}(k+1|k) = v$$

Substituindo Δx ,

$$H(k+1)H(k+1)^T K v = v$$

encontramos

$$K = (H(k+1)H(k+1)^T)^{-1}$$

.inf INSTITUTO DE INFORMÁTICA UFRGS

Localização – Filtragem de Kalman Linear – Simples Observador

Sistema: $$x(k+1) = Fx(k) + Gu(k)$$
$$y(k) = Hx(k)$$

Predição: $$\hat{x}(k+1 \mid k) = F\hat{x}(k \mid k) + Gu(k)$$

Correção : $$\Delta x = H(k+1)^T K v$$

onde $$K = (H(k+1)H(k+1)^T)^{-1}$$
$$v = y(k+1) - H(k+1)\hat{x}(k+1|k)$$

Atualização: $$\hat{x}(k+1 \mid k+1) = \hat{x}(k+1 \mid k) + \Delta x$$

.inf INSTITUTO DE INFORMÁTICA UFRGS

Localização – Filtragem de Kalman Linear – Simples Observador

- Na atualização, o ajuste no estado do robô é sempre perpendicular ao hiperplano
- Erros na direção paralela à Ω nunca serão corrigidos.
- Como resultado, o estado estimado não convergirá para o estado real do sistema na maioria dos casos[1].
- A vantagem deste estimador é que as idéias usadas são as mesma do filtro de Kalman.

Localização – Filtragem de Kalman Observando com uma PDF

- O observador anterior produz uma estimativa que é um vetor.
- A estimativa produzida por um Filtro de Kalman é uma distribuição gaussiana multivariada.
- Ou seja, o filtro de kalman além de fornecer um vetor de estado estimado $\hat{x}(k \mid k)$ fornece também uma matriz de covariância do erro da estimativa P(k|k).

.inf INSTITUTO DE INFORMÁTICA UFRGS

Localização – Filtragem de Kalman Observando com uma PDF

Considere o seguinte sistema com ruído na dinâmica, mas sem ruído nas medidas

$$x(k+1) = F(k)x(k) + G(k)\,u(k) + v(k)$$
$$y(k) = H(k)x(k)$$

Ruído

Onde $v(k) \in R^n$ é um ruído gaussiano de média zero e matriz de covariância V(k).

Usando $\hat{x}(k \mid k)$, temos

$$\hat{x}(k+1|k) = F(k)\hat{x}(k|k) + G(k)u(k)$$

pois o valor esperado de v(k) é zero

Localização – Filtragem de Kalman Observando com uma PDF

A matriz de covariância é dada por

$$P(k+1\,|\,k) = E\left((x(k+1) - \hat{x}(k+1\,|\,k))(x(k+1) - \hat{x}(k+1\,|\,k))^T\right)$$

Substituindo o valor de $x(k+1)$ e $\hat{x}(k+1\,|\,k)$ encontramos

$$\begin{aligned} P(k+1|k) = \quad & E[(F(k)x(k) + G(k)u(k) + v(k) - (F(k)\hat{x}(k|k) + G(k)u(k))) \\ & (F(k)x(k) + G(k)u(k) + v(k) - (F(k)\hat{x}(k|k) + G(k)u(k)))^T] \end{aligned}$$

Após algumas manipulações temos

$$\begin{aligned} P(k+1|k) = \quad & E[F(k)(x(k) - \hat{x}(k|k))(x(k) - \hat{x}(k|k))^T F(k)^T \\ & +2F(k)(x(k) - \hat{x}(k|k))v(k)^T + v(k)v(k)^T] \end{aligned}$$

Localização – Filtragem de Kalman Observando com uma PDF

Usando a propriedade da soma do valor esperado, temos

$$\begin{aligned} P(k+1|k) = \quad & E[F(k)(x(k) - \hat{x}(k|k))(x(k) - \hat{x}(k|k))^T F(k)^T] \\ & + E[2F(k)(x(k) - \hat{x}(k|k))v(k)^T] + E[v(k)v(k)^T] \end{aligned}$$

Como v(k) é independente de x(k) e de $\hat{x}(k \mid k)$

$$E[(x(k) - \hat{x}(k|k))v(k)] = E[(x(k) - \hat{x}(k|k))]E[v(k)]$$

Como E[v(k)]=0,

$$\begin{aligned} P(k+1|k) = \quad & F(k)\boxed{E[(x(k) - \hat{x}(k|k))(x(k) - \hat{x}(k|k))^T]}F(k)^T \\ & + E[v(k)v(k)^T] \end{aligned}$$

Torna-se

$$P(k+1|k) = F(k)\boxed{P(k|k)}F(k)^T + V(k)$$

Localização – Filtragem de Kalman – Observando com uma PDF

Para encontrar a equação de atualização, escolhemos $\hat{x}(k+1|k+1)$ que seja o mais provável ponto no hiperplano

$$\Omega = \{x \in R^n | H(k+1)x = y(k+1)\}$$

ou seja, o que maximiza a distribuição gaussiana definida por $\hat{x}(k+1|k)$ e $P(k+1|k)$, i.e.,

$$P(x) = \frac{1}{\sqrt{(2\pi)^n |P(k+1|k)|}} e^{-\frac{1}{2}(x-\hat{x}(k+1|k))^T P(k+1|k)^{-1}(x-\hat{x}(k+1|k))}$$

A maximização de P(x), envolve a minimização do expoente de **e**, i.e.,

$$(x-\hat{x}(k+1|k))^T P(k+1|k)^{-1}(x-\hat{x}(k+1|k))$$

.inf INSTITUTO DE INFORMÁTICA UFRGS

Localização – Filtragem de Kalman – Observando com uma PDF

A distância e o produto interno de Mahalanobis[1] entre duas variáveis aleatórias x e y e matriz de covariância S são dados por

$$||x-y||_M^2 = (x-y)^T S^{-1} (x-y)$$

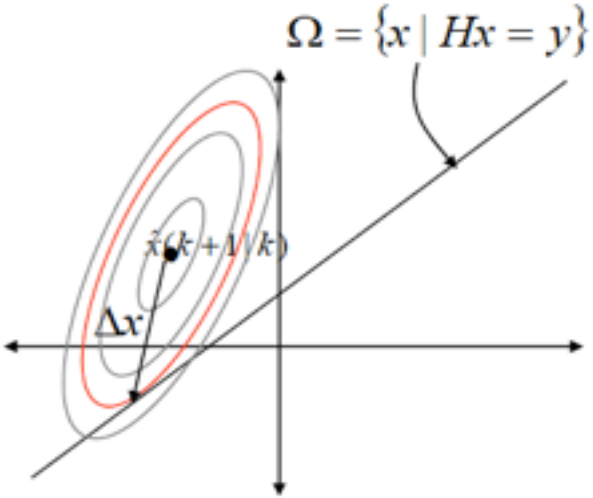
Logo, a minimização de $(x-\hat{x}(k+1|k))^T P(k+1|k)^{-1} (x-\hat{x}(k+1|k))$

Consiste em calcular $x \in \Omega$, ou seja, $\hat{x}(k+1 \mid k+1)$ de forma a minimizar

$$||x-\hat{x}(k+1|k)||_M = ||\Delta x||_M = ||\hat{x}(k+1 \mid k+1) - \hat{x}(k+1 \mid k)||_M$$

Localização – Filtragem de Kalman – Observando com uma PDF

Δx é ortogonal a Ω com relação ao produto interno de mahalanobis, ou seja, influenciada por P(k+1|k).



Logo, $\Delta x = P(k+1|k)H(k+1)^T\gamma$

Considerando a inovação como

$$i = y(k+1) - H(k+1)x(k+1|k)$$

Temos $\Delta x = P(k+1|k)H(k+1)^T Ki$, para $K \in R^p$. De forma similar ao apresentado no observador simples, temos que ter

$$H(k+1)(\hat{x}(k+1|k) + \Delta x) = y(k+1)$$

$$H(k+1)\Delta x = y(k+1) - H(k+1)\hat{x}(k+1|k) = i$$

Localização – Filtragem de Kalman Observando com uma PDF

Substituindo Δx ,

$$H(k+1)P(k+1|k)H(k+1)^T K i = i$$

Encontramos

$$K = (H(k+1)P(k+1|k)H(k+1)^T)^{-1}$$

Localização – Filtragem de Kalman Observando com uma PDF

Sistema: x(k+1) = F(k)x(k) + G(k) u(k)+v(k)

y(k) = H(k)x(k)

Predição: $$\hat{x}(k+1|k) = F(k)\hat{x}(k|k) + G(k)u(k)$$

$$P(k+1|k) = F(k)P(k|k)F(k)^T + V(k)$$

Correção: $$\Delta x = P(k+1|k)H(k+1)^T Ki$$

onde

$$K = (H(k+1)P(k+1|k)H(k+1)^T)^{-1}$$

$$i = y(k+1) - H(k+1)x(k+1|k)$$

Atualização: $$\hat{x}(k+1 \mid k+1) = \hat{x}(k+1 \mid k) + \Delta x$$

$$P(k+1|k+1) = P(k+1|k) - P(k+1|k)H(k+1)^T KHP(k+1|k)$$

.inf INSTITUTO DE INFORMÁTICA UFRGS

Localização – Filtragem de Kalman Observando com uma PDF

- Como o observador simples, este novo observador possui alguns problemas[1]:
  - Devido ao fato de não termos considerado ruído na leitura dos sensores, as equações de atualização farão com que a matriz de covariança se torne singular.
  - Uma matriz singular possui determinante igual a 0, o que é coerente com o fato de que as medidas sensoriais não possuem incertezas.
  - Isto prejudica a distribuição gaussiana que usa a inversa da matriz de covariância.

Localização – Filtragem de Kalman

- O filtro de Kalman usa os conceitos e mecanismos apresentados anteriormente, com a diferença de que o sistema é descrito por

  $x(k+1) = F(k)x(k) + G(k)\,u(k) + v(k)$

  $y(k) = H(k)x(k) + w(k)$

  Onde
  - $u(k) \in R^m$ é a entrada para o sistema (velocidades, torque, etc);
  - $y(k) \in R^p$ é a saída do sistema e contém os valores dados pelos sensores do robô
  - $F(k) \in R^{n \times n}$ codifica a dinâmica do sistema
  - $G(k) \in R^{n \times m}$ descreve como as entradas influênciam a dinâmica.
  - $v(k) \in R^n$ é o ruído gaussiano com média 0 e matriz de covariância V(k).
  - $H(k) \in R^{p \times n}$ descreve como o vetor de estado é mapeado para a saída
  - $w(k) \in R^p$ é o ruído gaussiano com média 0 e matriz de covariância W(k).

Localização – Filtragem de Kalman Observando com uma PDF

Sistema: x(k+1) = F(k)x(k) + G(k) u(k)+v(k)

y(k) = H(k)x(k) + w(k)

Predição:

$$\hat{x}(k+1|k) = F(k)\hat{x}(k|k) + G(k)u(k)$$

$$P(k+1|k) = F(k)P(k|k)F(k)^T + V(k)$$

Correção:

$$\Delta x = P(k+1|k)H(k+1)^T Ki$$

onde

$$K = (H(k+1)P(k+1|k)H(k+1)^T + W(k+1))^{-1}$$

Novo termo

$$i = y(k+1) - H(k+1)x(k+1|k)$$

Atualização:

$$\hat{x}(k+1\,|\,k+1) = \hat{x}(k+1\,|\,k) + \Delta x$$

$$P(k+1|k+1) = P(k+1|k) - P(k+1|k)H(k+1)^T KHP(k+1|k)$$

.inf INSTITUTO DE INFORMÁTICA UFRGS

Localização – Filtragem de Kalman – Exemplo – Dead Reckoning

Considere um robô que se move em linha reta com estado $x = [x_r, v_r]^T$ e u como sendo a força que se aplica ao robô. Sabemos pela segunda lei de newton que

$$\frac{dv_r}{dt} = \frac{u}{m} \quad \text{ou} \quad \frac{v_r(k+1) - v_r(k)}{T} = \frac{u(k)}{m}$$

Logo, o próximo estado do robô é dado por

$$x(k+1) = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ \frac{T}{m} \end{bmatrix} u(k) + v(k)$$

$$\triangleq Fx(k) + Gu(k) + v(k)$$

Ruído gaussiano com média zero

$$y(k+1) = [0\ 1]x(k+1) + w(k)$$

$$\triangleq Gx(k) + w(k)$$

O robô possui um sensor que mede velocidade.

Localização – Filtragem de Kalman – Exemplo – Dead Reckoning

$$x(k+1) = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} x(k) + \begin{bmatrix} 0 \\ \frac{T}{m} \end{bmatrix} u(k) + v(k)$$
$$\triangleq Fx(k) + Gu(k) + v(k),$$

$$y(k+1) = \begin{bmatrix} 0 & 1 \end{bmatrix} x(k) + w(k)$$
$$\triangleq Gx(k) + w(k),$$

Em [1], são assumidos os seguintes parâmetros m=1, T=0.5, W=0.5 (variância na medida) e matriz de covariância V

$$V = \begin{bmatrix} 0.2 & 0.05 \\ 0.05 & 0.1 \end{bmatrix}$$

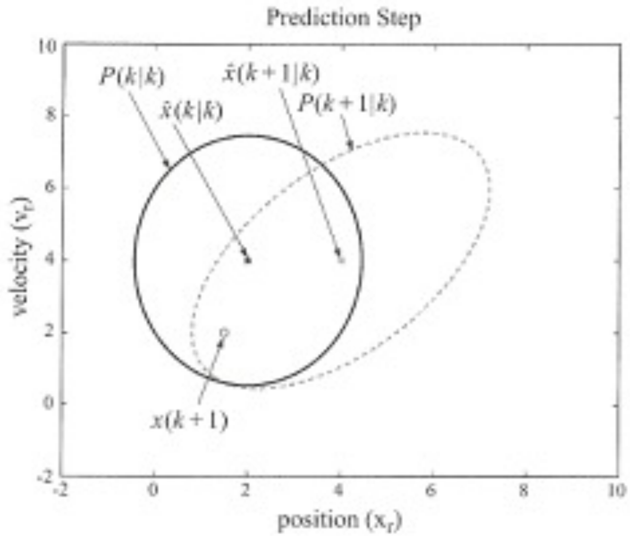
No instante k

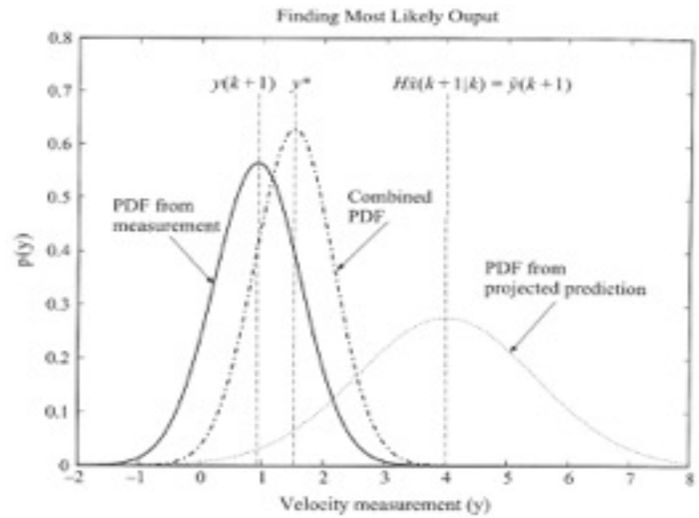
$$u(k) = 0$$

$$\hat{x}(k|k) = [2,\ 4]^T$$

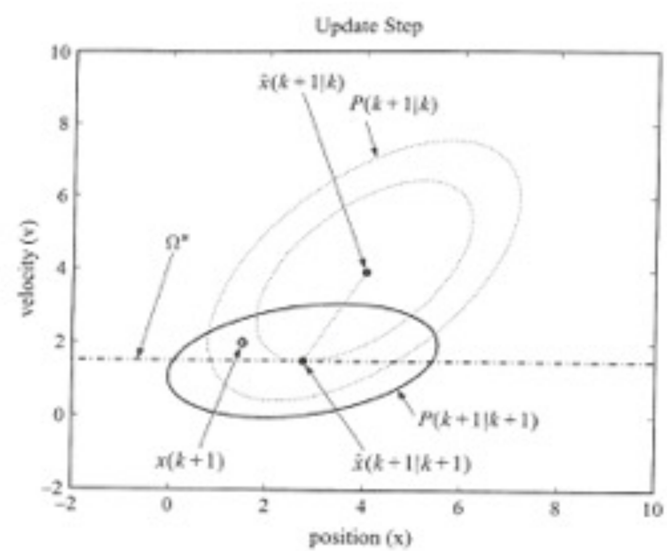
$$P(k|k) = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$$

Localização – Filtragem de Kalman – Exemplo – Dead Reckoning





Figuras extraídas de [1]



Localização – Filtragem de Kalman – Exemplo – Dead Reckoning

- A incerteza diminuiu na direção vertical devido à informação obtida pelo sensor de velocidade, porém existe perda de informação na direção da posição.

- Este problema não ocorre devido ao Filtro de Kalman, mas devido ao fato do sistema não ser observável, ou seja, o estado do sistema não pode ser recuperado a partir das saídas disponíveis. Logo, o erro esperado irá crescer ilimitadamente.

- Se o sistema é observável, a estimativa fornecida pelo Filtro de Kalman converge, ou seja, o erro esperado entre o estado atual e o estado estimado é limitado.

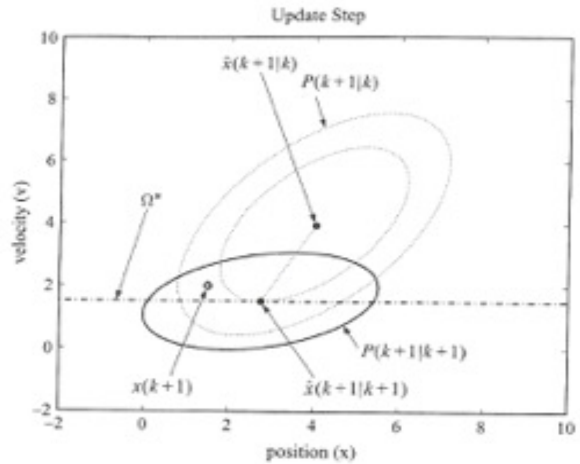


Figura extraída de [1]

Localização – Filtragem de Kalman – Filtro de Kalman Estendido

Em geral, os sistemas dinâmicos são não-lineares. Portanto, temos que estender o Filtro de Kalman para tratar a não-linearidade do sistema. Considere o seguinte sistema

$$
\begin{aligned}
x(k+1) &= f(x(k), u(k), k) + v(k) \\
y(k) &= h(x(k), k) + w(k),
\end{aligned}
\qquad
\begin{aligned}
f &: \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{Z}^+ \to \mathbb{R}^n \\
h &: \mathbb{R}^n \times \mathbb{Z}^+ \to \mathbb{R}^p
\end{aligned}
$$

A fase de predição é dada pelas equações

$$
\begin{aligned}
\hat{x}(k+1|k) &= f(\hat{x}(k|k), u(k), k) \\
P(k+1|k) &= F(k)P(k|k)F(k)^T + V(k)
\end{aligned}
$$

onde

$$
F(k) = \left.\frac{\partial f}{\partial x}\right|_{x=\hat{x}(k|k)} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}_{x=\hat{x}(k|k)}
$$

Localização – Filtragem de Kalman – Filtro de Kalman Estendido

Correção:
$$\begin{aligned}\hat{x}(k+1|k+1) &= \hat{x}(k+1|k) + R\nu \\ P(k+1|k+1) &= P(k+1|k) - RH(k+1)P(k+1|k)\end{aligned}$$

onde
$$\begin{aligned}\nu &= y(k+1) - h(x(k+1|k), k+1) \\ S &= H(k+1)P(k+1|k)H(k+1)^T + W(k+1) \\ R &= P(k+1|k)H(k+1)^T S^{-1}\end{aligned}$$

e

$$H(k+1) = \left.\frac{\partial h}{\partial x}\right|_{x=\hat{x}(k+1|k)} = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \cdots & \frac{\partial h_1}{\partial x_n} \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \cdots & \frac{\partial h_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial h_p}{\partial x_1} & \frac{\partial h_p}{\partial x_2} & \cdots & \frac{\partial h_p}{\partial x_n} \end{bmatrix}_{x=\hat{x}(k+1|k)}$$

.inf INSTITUTO DE INFORMÁTICA UFRGS

Localização – Filtragem de Kalman – Exemplo – Filtro de Kalman Estendido

Considere um robô capaz de detectar landmarks previamente mapeados no ambiente. O estado do robô no instante k é dado por

$$x(k) = [x_r(k), y_r(k), \theta_r(k)]^T$$

E a entrada de controle é dada por $u(k) = [u_1(k), u_2(k)]^T$ onde $u_1(k) e u_2(k)$ representam as velocidades linear e angular.

O sistema dinâmico é dado por

$$x(k+1) = \begin{bmatrix} \cos\theta_r(k)u_1(k) + x_r(k) \\ \sin\theta_r(k)u_1(k) + y_r(k) \\ u_2(k) + \theta_r(k) \end{bmatrix} + v(k)$$

O ambiente possui n_l landmarks com posição $(x_{\ell i}, y_{\ell i})$ porém o robô consegue apenas ver apenas uma quantidade p(k) a cada instante k. A função abaixo associa cada landmark visto com os conhecidos previamente.

$$a : \{1, 2, \ldots, p(k)\} \rightarrow \{1, 2, \ldots, n_\ell\}$$

Localização – Filtragem de Kalman – Exemplo – Filtro de Kalman Estendido

A saída do sistema é dada por

$$y(k) = \begin{bmatrix} h_1(x(k), a(1)) \\ h_2(x(k), a(2)) \\ \vdots \\ h_{p(k)}(x(k), a(p(k))) \end{bmatrix} + \begin{bmatrix} w_1(k) \\ w_2(k) \\ \vdots \\ w_{p(k)}(k) \end{bmatrix}$$

com

$$h_j(x(k), j) = \begin{bmatrix} \sqrt{(x_r(k) - x_{\ell j})^2 + (y_r(k) - y_{\ell j})^2} \\ \mathrm{atan2}(y_r(k) - y_{\ell j}, x_r(k) - x_{\ell j}) - \theta_r(k) \end{bmatrix}$$

Como a dinâmica do sistema é dada por

$$x(k+1) = \boxed{\begin{bmatrix} \cos\theta_r(k)u_1(k) + x_r(k) \\ \sin\theta_r(k)u_1(k) + y_r(k) \\ u_2(k) + \theta_r(k) \end{bmatrix}} + v(k)$$

$$f(x(k), u(k), k)$$

A Jacobiana de f é

$$F = \begin{bmatrix} 1 & 0 & -\sin\theta_r(k)u_1(k) \\ 0 & 1 & \cos\theta_r(k)u_1(k) \\ 0 & 0 & 1 \end{bmatrix}$$

Lembrando que

$$F(k) = \left.\frac{\partial f}{\partial x}\right|_{x=\hat{x}(k|k)} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}_{x=\hat{x}(k|k)}$$

Localização – Filtragem de Kalman – Exemplo – Filtro de Kalman Estendido

A matriz H que é usada para produzir a saída do Filtro é definida por

$$H(k+1) = \begin{bmatrix} H_1(k+1, a(1)) \\ H_2(k+1, a(2)) \\ \vdots \\ H_{p(k+1)}(k+1, a(p(k+1))) \end{bmatrix} = \begin{bmatrix} \left.\frac{\partial h_1}{\partial x}\right|_{x=\hat{x}(k+1|k)} \\ \left.\frac{\partial h_2}{\partial x}\right|_{x=\hat{x}(k+1|k)} \\ \vdots \\ \left.\frac{\partial h_{p(k+1)}}{\partial x}\right|_{x=\hat{x}(k+1|k)} \end{bmatrix}$$

com

$$H_i(k+1, j) = \begin{bmatrix} \frac{(\hat{x}_r(k+1|k) - x_{\ell j})}{\sqrt{(\hat{x}_r(k+1|k) - x_{\ell j})^2 + (\hat{y}_r(k+1|k) - y_{\ell j})^2}} & \frac{(\hat{y}_r(k+1|k) - y_{\ell j})}{\sqrt{(\hat{x}_r(k+1|k) - x_{\ell j})^2 + (\hat{y}_r(k+1|k) - y_{\ell j})^2}} & 0 \\ \frac{-(\hat{y}_r(k+1|k) - y_{\ell j})}{1 + \left(\frac{\hat{y}_r(k+1|k) - y_{\ell j}}{\hat{x}_r(k+1|k) - x_{\ell j}}\right)^2 (\hat{x}_r(k+1|k) - x_{\ell j})^2} & \frac{1}{1 + \left(\frac{\hat{y}_r(k+1|k) - y_{\ell j}}{\hat{x}_r(k+1|k) - x_{\ell j}}\right)^2 (\hat{x}_r(k+1|k) - x_{\ell j})} & -1 \end{bmatrix}$$

Lembrando que

$$h_j(x(k), j) = \begin{bmatrix} \sqrt{(x_r(k) - x_{\ell j})^2 + (y_r(k) - y_{\ell j})^2} \\ \operatorname{atan2}(y_r(k) - y_{\ell j}, x_r(k) - x_{\ell j}) - \theta_r(k) \end{bmatrix}$$

Localização – Filtragem de Kalman – Exemplo – Filtro de Kalman Estendido

A abordagem apresentada assume que a associação entre cada medida com o landmark já conhecido é feita corretamente, sem erros.

Na prática, esta associação pode não ser tão fácil!! já que os landmarks podem ser muito parecidos. Este problema é chamado **Associação de Dados, ou Data Association.**

A idéia básica para a associação de dados é considerar para cada medida i ($y_i(k$+1)) e landmark j um valor de inovação v_{ij}

$$\nu_{ij} \stackrel{\triangle}{=} y_i(k+1) - h_i(\hat{x}(k+1|k), j)$$

A medida $y_i(k$+1) é associada ao landmark j que produza a menor inovação de acordo com a norma de Mahalanobis. Esta norma incorpora as incertezas associadas à predição e às medidas.

$$\chi^2_{ij} = \nu_{ij} S^{-1}_{ij} \nu^T_{ij}.$$

$$S_{ij} = H_i(k+1,j)P(k+1 \mid k)H_i(k+1,j)^T + W_i(k+1).$$

Localização – Filtragem de Kalman – Exemplo – Filtro de Kalman Estendido

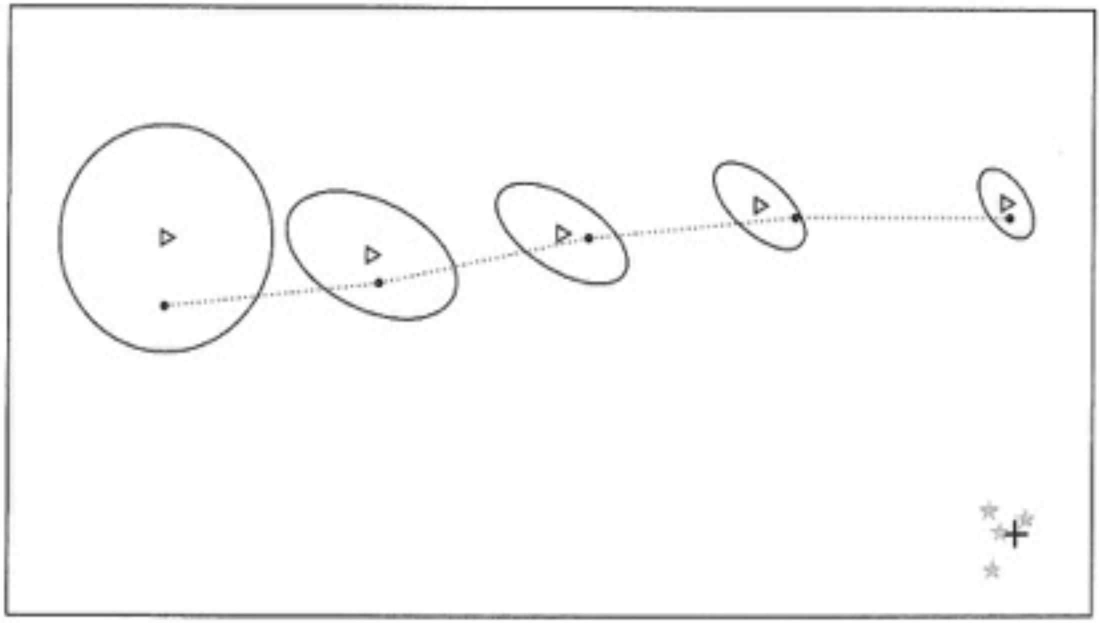


Figura extraída de [1]

Simple SLAM – Filtragem de Kalman

Estado do sistema

$$x = \left[\boxed{x_r \quad y_r} \; \boxed{x_{\ell 1} \quad y_{\ell 1} \quad x_{\ell 2} \quad y_{\ell 2} \quad \cdots \quad x_{\ell n_\ell} \quad y_{\ell n_\ell}} \right]^T$$

Posição do robô

Posições dos landmarks

Velocidade no eixo x e no eixo y

$$x(k+1) = x(k) + \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 0 \end{bmatrix} \begin{bmatrix} u_x(k) \\ u_y(k) \end{bmatrix} + \begin{bmatrix} \boxed{\begin{matrix} v_{rx}(k) \\ v_{ry}(k) \end{matrix}} \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

Erros associados ao deslocamento

$$x(k+1) = Fx(k) + Gu(k) + v(k)$$

$$V(k) = \begin{bmatrix} V_r(k) & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix}$$

A cada instante de tempo o robô consegue ver todos os landmarks, e pode distingui-los. Não temos o problema de associação de dados.

.inf INSTITUTO DE INFORMÁTICA UFRGS

Simple SLAM – Filtragem de Kalman

Estimativa da posição para o landmark i

$$y_i(k) = \begin{bmatrix} x_{\ell i}(k) - x_r(k) \\ y_{\ell i}(k) - y_r(k) \end{bmatrix} + w_i(k)$$

$$\downarrow$$

$$y_i(k) = H_i x(k) + w_i(k),$$

$$H_i = \begin{bmatrix} -1 & 0 & 0 & \cdots & 0 & 1 & 0 & 0 & \cdots & 0 \\ 0 & -1 & 0 & \cdots & 0 & 0 & 1 & 0 & \cdots & 0 \end{bmatrix}$$

Saída produzida pelo Filtro

$$y(k) = Hx(k) + w(k),$$

$$H = \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_{n_\ell} \end{bmatrix}, \quad \text{and} \quad w(k) = \begin{bmatrix} w_1(k) \\ w_2(k) \\ \vdots \\ w_{n_\ell}(k) \end{bmatrix},$$

Matriz de covariância associada ao ruído w(k)

$$W(k) = \begin{bmatrix} W_1(k) & 0 & \cdots & 0 \\ 0 & W_2(k) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & W_{n_\ell} \end{bmatrix},$$

Através deste conjunto de equações para o Filtro de Kalman Estendido, estamos estimando tanto a posição do robô quando dos landmarks

Simple SLAM – Filtragem de Kalman

Como transformar este exemplo em um SLAM efetivo,
capaz de fornecer uma estimativa sobre a posição do robô e dos landmarks desconhecidos presentes no ambiente ?

A chave é verificar se um landmark é novo ou é algum já armazenado no mapa. Isto é feito através norma de Mahalanobis aplicada ao termo de inovação

$$\chi^2_{ij} = (y(k)_i - h(k)_j)^T S_{ij} (y(k)_i - h(k)_j).$$

Se o valor mínimo encontrado é maior que um dado limiar, então o landmark é novo devendo ser inicializado e adicionado ao mapa.

BIBLIOGRAFIA

[1] H. Choset, K. Lynch, S. Hutchinson, G. Kantor, W. Burgard, L.Kavraki, S. Thrun. Principles of Robot Motion : Theory, Algorithms, and Implementations. MIT Press. 2005.

[2] D. Fox, W. Burgard and S. Thrun. Markov Localization for mobile robots in dynamic environments. Journal of Artificial Intelligence Research (JAIR), 11: 391-427, 1999.

[3] S.Thrun. Learning Metric-Topological Maps for Indoor Mobile Robot Navigation. Artificial Intelligence, v.99, n. 1, 21-71,1998.

[4]. S. Thrun, W. Burgard, and D. Fox. Probabilistic Robotics. MIT Press, Cambridge, MA, 2005.

.inf INSTITUTO DE INFORMÁTICA UFRGS