SCE5880
Algoritmos de Estimação para Robótica Móvel

Localização II

Filtro de Kalman
Monte Carlo

1

Filtro de Kalman

- Uma das primeiras implementações práticas do filtro de Bayes (1960).
- Hipóteses para utilização do filtro:
  - Erro médio de cada variável igual a zero;
  - Erro de cada variável independente;
  - Modelo linear de evolução do sistema;
  - Relacionamento linear entre variáveis de estado e variáveis medidas.
- Se as hipóteses acima não forem cumpridas, a optimalidade não é assegurada.

2

Filtro de Kalman

- Estima o estado *x* de um processo controlado em tempo discreto que é governado por uma equação estocástica

$$\boxed{x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t}$$

- Com medição

$$\boxed{z_t = C_t x_t + \delta_t}$$

3

Componentes do filtro de Kalman

A_t Matriz (nxn) que descreve a evolução do estado de *t* a *t-1* sem ações de controle ou ruído.

B_t Matriz (nxl) que descreve como as ações de controle u_t mudam o estado de *t-1* para *t*.

C_t Matriz (kxn) que mapeia o estado x_t em uma observação z_t.

ε_t δ_t Variáveis aleatórias que representam os ruídos de processo e de medição, assumidos independentes e com distribuição normal com covariância R_t e Q_t respectivamente.

4

Filtro de Kalman: dinâmica

- A dinâmica do sistema é uma função linear do estado e das ações de controle, mais o ruído do sistema:

$$\boxed{x_t = A_t x_{t-1} + B_t u_t + \varepsilon_t}$$

$$\boxed{p(x_t \mid u_t, x_{t-1}) = N\left(x_t; A_t x_{t-1} + B_t u_t, R_t\right)}$$

$$\overline{bel}(x_t) = \int \underbrace{p(x_t \mid u_t, x_{t-1})}_{\sim N\left(x_t; A_t x_{t-1} + B_t u_t, R_t\right)} \; \underbrace{bel(x_{t-1})}_{\sim N\left(x_{t-1}; \mu_{t-1}, \Sigma_{t-1}\right)} \, dx_{t-1}$$

5

Filtro de Kalman: observações

- As observações são funções lineares do estado, mais o ruído associado:

$$\boxed{z_t = C_t x_t + \delta_t}$$

$$\boxed{p(z_t \mid x_t) = N\left(z_t; C_t x_t, Q_t\right)}$$

$$bel(x_t) = \eta \; \underbrace{p(z_t \mid x_t)}_{\sim N\left(z_t; C_t x_t, Q_t\right)} \; \underbrace{\overline{bel}(x_t)}_{\sim N\left(x_t; \bar{\mu}_t, \bar{\Sigma}_t\right)}$$

6

Filtro de Kalman

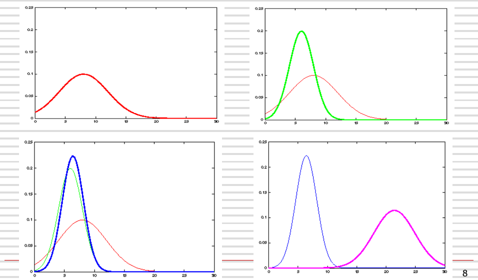
1. Algoritmo **Kalman_filter**($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$):

2. Predição:
3. $\bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$
4. $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$

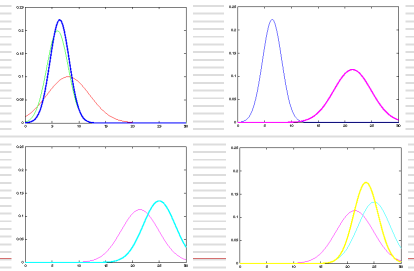
5. Correção:
6. $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$
7. $\mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$
8. $\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$

9. Return μ_t, Σ_t

Ciclo correção-predição



Ciclo correção-predição



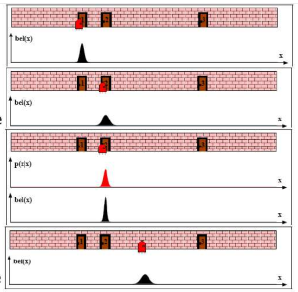
Ciclo correção-predição

Estimação do estado inicial

Predição: associação à ação de controle

Atualização: incorporação da medição

Predição: associação à ação de controle



Filtro de Kalman na robótica

- Altamente Eficiente: complexidade polinomial com as dimensões de medição k e de estado n: $O(k^{2.376} + n^2)$
- Ótimo para sistemas lineares Gaussianos
- Maioria dos sistemas robóticos é não-linear!

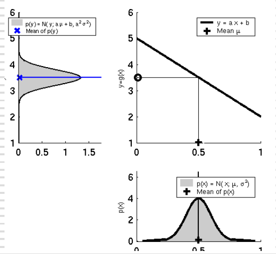
Não linearidade

- A maioria dos problemas de robótica envolve funções não-lineares!

$$x_t = g(u_t, x_{t-1})$$

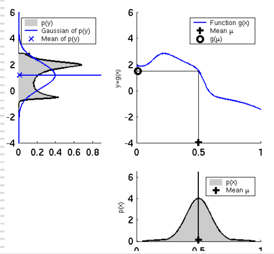
$$z_t = h(x_t)$$

Não linearidade



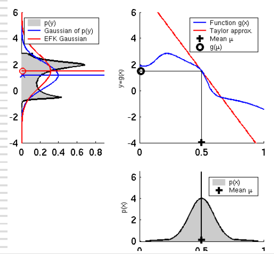
13

Não linearidade



14

Não linearidade



15

Linearização do KF: Série de Taylor – primeira ordem

- Predição:

$$g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}}(x_{t-1} - \mu_{t-1})$$

$$g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + G_t\,(x_{t-1} - \mu_{t-1})$$

- Correção:

$$h(x_t) \approx h(\bar{\mu}_t) + \frac{\partial h(\bar{\mu}_t)}{\partial x_t}(x_t - \bar{\mu}_t)$$

$$h(x_t) \approx h(\bar{\mu}_t) + H_t\,(x_t - \bar{\mu}_t)$$

16

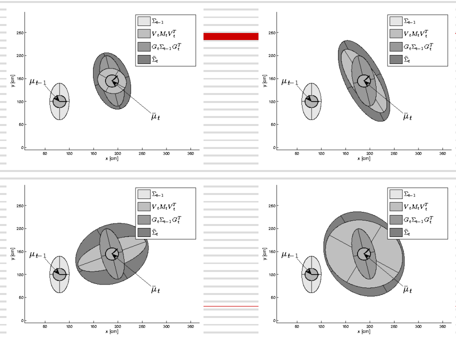
Algoritmo: filtro de Kalman estendido

1. **Extended_Kalman_filter**($\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$):
2. Prediction:
3. $\bar{\mu}_t = g(u_t, \mu_{t-1})$ $\longleftarrow$ $\bar{\mu}_t = A_t\mu_{t-1} + B_t u_t$
4. $\bar{\Sigma}_t = G_t\Sigma_{t-1}G_t^T + R_t$ $\longleftarrow$ $\bar{\Sigma}_t = A_t\Sigma_{t-1}A_t^T + R_t$
5. Correction:
6.
7. $K_t = \bar{\Sigma}_t H_t^T(H_t\bar{\Sigma}_t H_t^T + Q_t)^{-1}$ $\longleftarrow$ $K_t = \bar{\Sigma}_t C_t^T(C_t\bar{\Sigma}_t C_t^T + Q_t)^{-1}$
8. $\mu_t = \bar{\mu}_t + K_t(z_t - h(\bar{\mu}_t))$ $\longleftarrow$ $\mu_t = \bar{\mu}_t + K_t(z_t - C_t\bar{\mu}_t)$

   $\Sigma_t = (I - K_t H_t)\bar{\Sigma}_t$ $\longleftarrow$ $\Sigma_t = (I - K_t C_t)\bar{\Sigma}_t$
9. Return μ_t, Σ_t

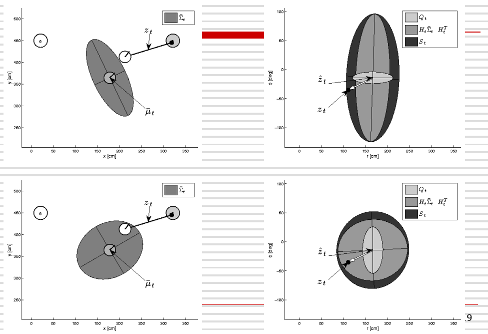
$$H_t = \frac{\partial h(\bar{\mu}_t)}{\partial x_t} \qquad G_t = \frac{\partial g(u_t, \mu_{t-1})}{\partial x_{t-1}}$$

EKF Predição



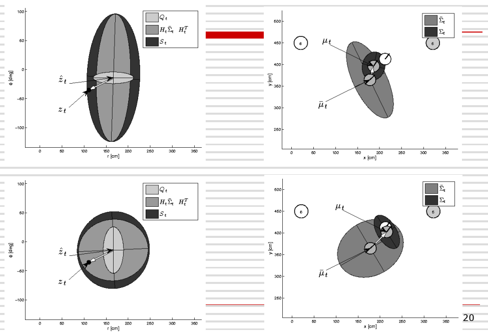
18

EKF Observação



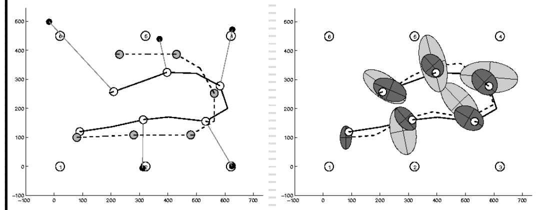
9

EKF Correção



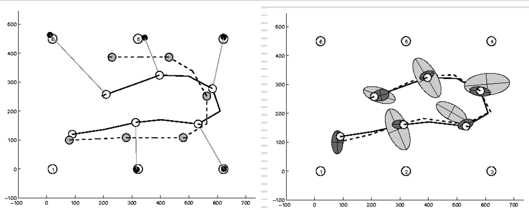
20

Seqüência de estimação



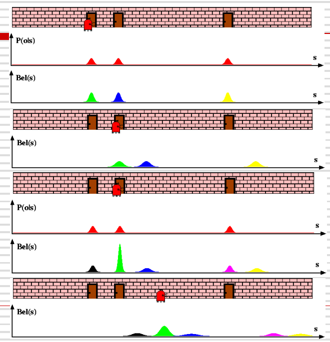
21

Seqüência de estimação



22

EKF multi-hipóteses



23

Filtro de Kalman estendido na robótica

- Altamente Eficiente: complexidade polinomial com as dimensões de medição k e de estado n: $O(k^{2.376} + n^2)$
- Não é ótimo.
- Pode divergir se a não-linearidade for forte!
- Funciona surpreendentemente bem, mesmo quando as suposições básicas são violadas!

24

Localização – resumo

	EKF	MHT	Coarse (topo-logical) grid	fine (metric) grid	MCL
Measurements	landmarks	landmarks	landmarks	raw measurements	raw measurements
Measurement noise	Gaussian	Gaussian	any	any	any
Posterior	Gaussian	mixture of Gaussians	histogram	histogram	particles
Efficiency (memory)	++	++	+	−	+
Efficiency (time)	++	+	+	−	+
Ease of implementation	+	−	+	−	++
Resolution	++	++	−	+	+
Robustness	−	+	+	++	++
Global localization	no	no	yes	yes	yes