

## 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

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

- Com medição

$$z_t = C_t x_t + \delta_t$$

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  $z_t$ .

$\varepsilon_t$  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.

$\delta_t$

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:

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

$$p(x_t | u_t, x_{t-1}) = N(x_t; A_t x_{t-1} + B_t u_t, R_t)$$

$$\overline{bel}(x_t) = \int p(x_t | u_t, x_{t-1}) \overline{bel}(x_{t-1}) dx_{t-1}$$

↓

$$\sim N(x_t; A_t x_{t-1} + B_t u_t, R_t)$$

↓

$$\sim N(x_{t-1}; \overline{\mu}_{t-1}, \overline{\Sigma}_{t-1})$$

## Filtro de Kalman: observações

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

$$z_t = C_t x_t + \delta_t$$

$$p(z_t | x_t) = N(z_t; C_t x_t, Q_t)$$

$$\overline{bel}(x_t) = \eta \int p(z_t | x_t) \overline{bel}(x_t) dx_t$$

↓

$$\sim N(z_t; C_t \overline{x}_t, Q_t)$$

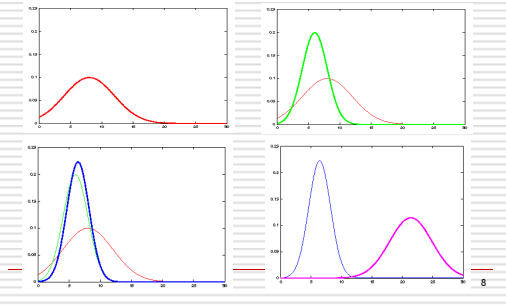
↓

$$\sim N(\overline{x}_t; \overline{\mu}_t, \overline{\Sigma}_t)$$

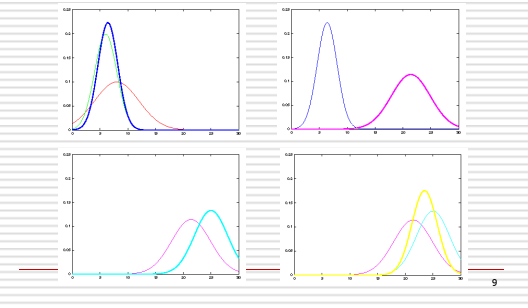
## Filtro de Kalman

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**  $\mu_t, \Sigma_t$

## Ciclo correção-predição



## Ciclo correção-predição

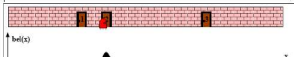


## Ciclo correção-predição

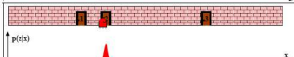
Estimação do estado inicial



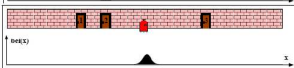
Predição:  
associação à  
ação de controle



Atualização:  
incorporação da  
medição



Predição:  
associação à  
ação de controle



10

## 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!

11

## Não linearidade

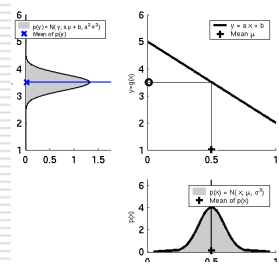
- A maioria dos problemas de robótica envolve funções não-lineares!

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

$$z_t = h(x_t)$$

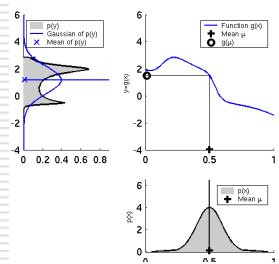
12

## Não linearidade



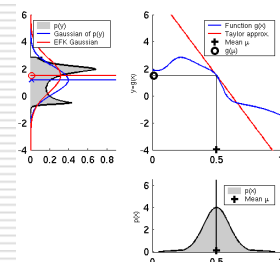
13

## 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:

$$\bar{\mu}_t = g(u_t, \mu_{t-1}) \quad \leftarrow \quad \bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$$

$$\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t \quad \leftarrow \quad \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t$$

#### 3. Correction:

$$K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1} \quad \leftarrow \quad K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + Q_t)^{-1}$$

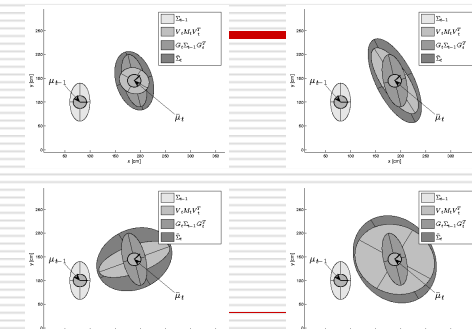
$$\mu_t = \bar{\mu}_t + K_t (z_t - h(\bar{\mu}_t)) \quad \leftarrow \quad \mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$$

$$\Sigma_t = (I - K_t C_t) \bar{\Sigma}_t \quad \leftarrow \quad \Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$$

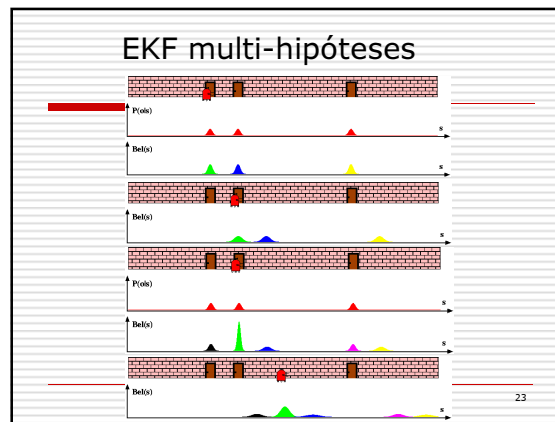
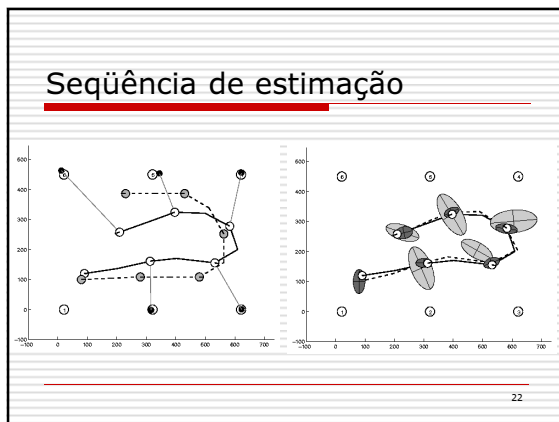
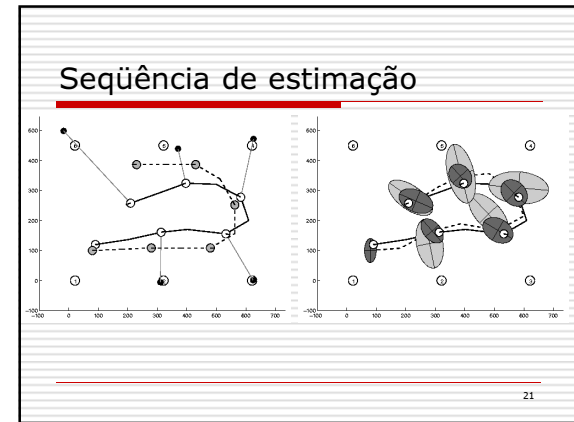
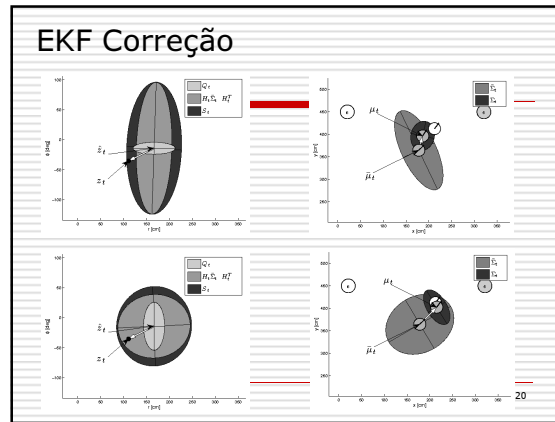
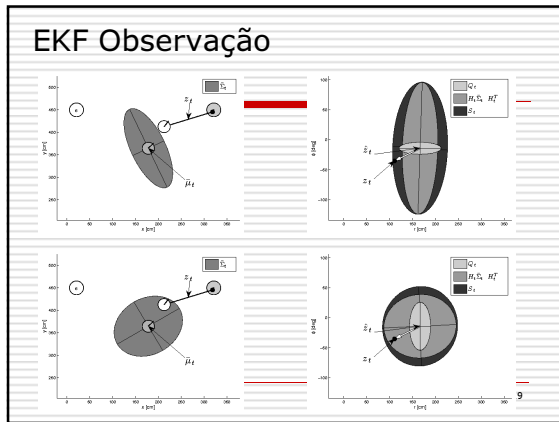
#### 4. Return $\mu_t, \Sigma_t$

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

## EKF Predição



18



### 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