

# Robots Móviles

Tema 3. Localización y mapeado de un robot móvil

Parte III. Localización con filtros Discretos y filtro de Kalman Extendido

Sergio Orts Escolano  
Otto Colomina Pardo

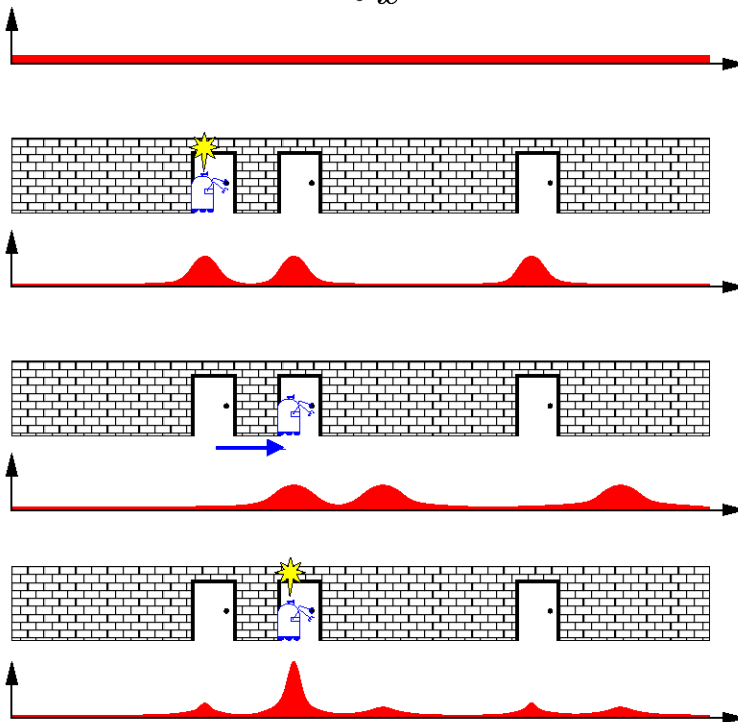
# Índice

## **Filtros discretos**

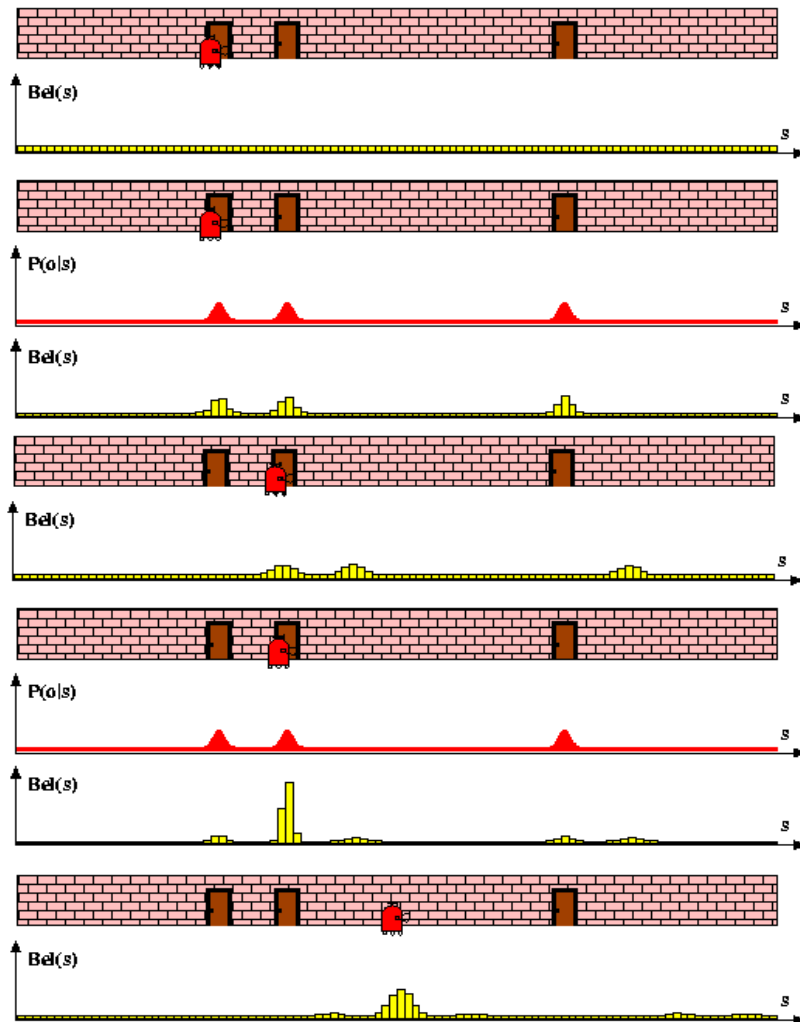
Filtro de Kalman (KF) y Filtro de Kalman Extendido (EKF)

# Localización probabilística

$$Bel(x \mid z, u) = \alpha p(z \mid x) \int_{x'} p(x \mid u, x') Bel(x') dx'$$

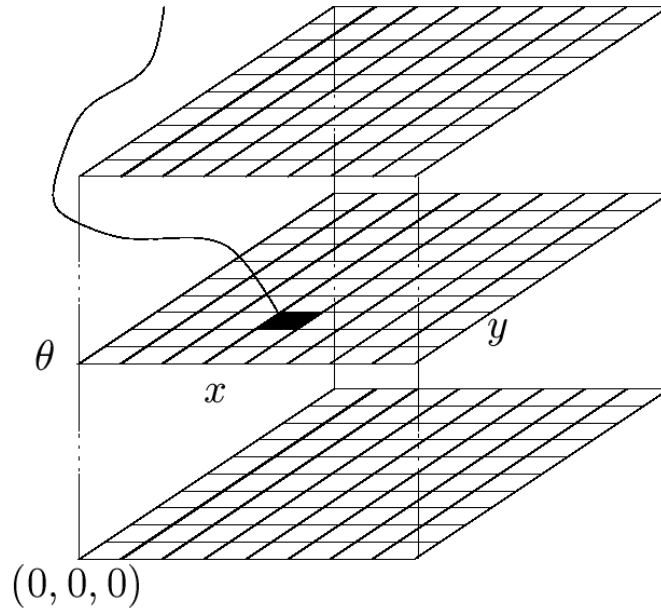


Los filtros discretos usan una representación discretizada (histogramas)



# Representación en un robot móvil real

$Bel(x_t = \langle x, y, \theta \rangle)$



# Algoritmo Filtro Discreto Bayes

1. Algorithm **Discrete\_Bayes\_filter**(  $Bel(x), d$  ):
2.  $\eta = 0$
3. If  $d$  is a **perceptual** data item  $z$  then
4.     For all  $x$  do
5.          $Bel'(x) = P(z | x) Bel(x)$
6.          $\eta = \eta + Bel'(x)$
7.     For all  $x$  do
8.          $Bel'(x) = \eta^{-1} Bel'(x)$
9. Else if  $d$  is an **action** data item  $u$  then
10.     For all  $x$  do
11.          $Bel'(x) = \sum_{x'} P(x | u, x') Bel(x')$
12. Return  $Bel'(x)$

# Optimizar la fase de medida

- Para actualizar la creencia sobre la medida del sensor y llevar a cabo la normalización se tiene que iterar sobre todas las células de la rejilla (*línea 4 algoritmo anterior*)
- Cuando la creencia está concentrada en un pico (habitual en localización cuando se conoce la pos. inicial), se quiere evitar actualizar partes irrelevantes del espacio de estados.
- Un enfoque es no actualizar subespacios enteros del espacio de estados.
- Esto, sin embargo, requiere monitorizar si el robot está localizado o no.
- Para lograr esto, se puede considerar la probabilidad de las observaciones para las partes “activas” del espacio de estado.

# Optimizar la fase de movimiento

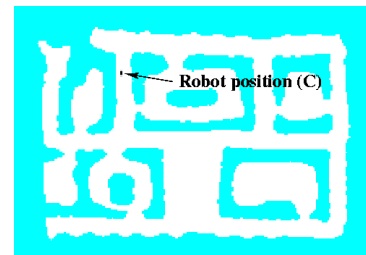
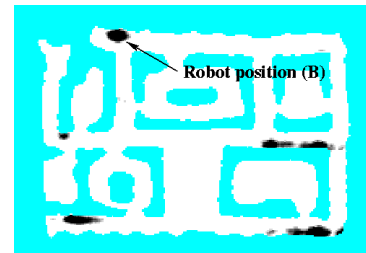
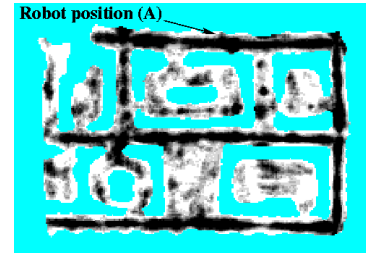
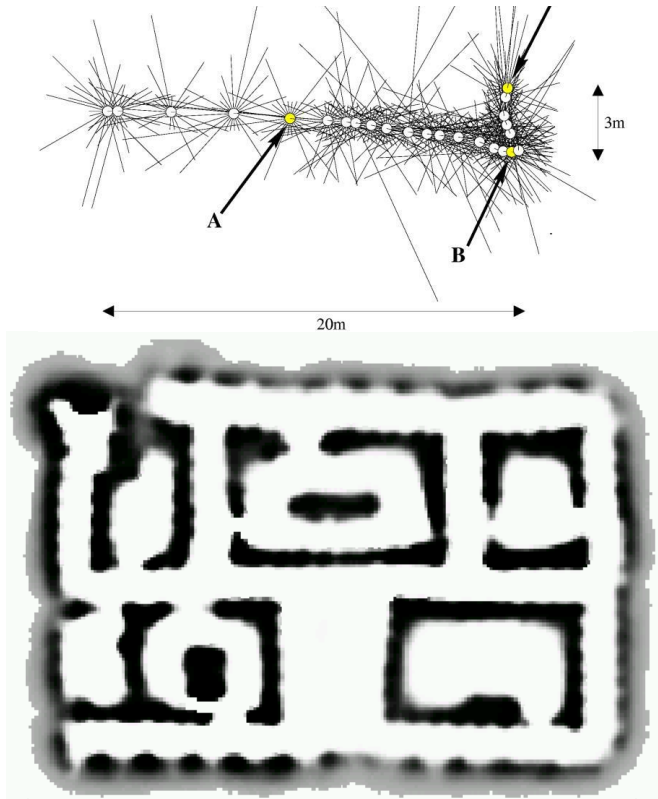
- El coste es de  $O(n^2)$  donde  $n$  es el número de estados (líneas 10 y 11 del algoritmo)
- Posible optimización
  - Para la incertidumbre del movimiento asumir un modelo gaussiano **limitado**
  - Esto reduce el coste de actualización de  $O(n^2)$  a  $O(n)$
- Otra optimización
  - Paso 1: desplazar los datos en la rejilla según el movimiento medido.
  - Paso 2: convolucionar la rejilla usando un filtro Gaussiano separable.
- Ejemplo en 2 dimensiones:

$$\begin{array}{|c|c|c|} \hline 1/16 & 1/8 & 1/16 \\ \hline 1/8 & 1/4 & 1/8 \\ \hline 1/16 & 1/8 & 1/16 \\ \hline \end{array} \cong \begin{array}{|c|} \hline 1/4 \\ \hline 1/2 \\ \hline 1/4 \\ \hline \end{array} + \begin{array}{|c|c|c|} \hline 1/4 & 1/2 & 1/4 \\ \hline \end{array}$$

- Unas pocas operaciones aritméticas
- Fácil de implementar

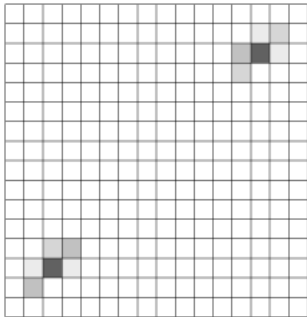


# Localización con sonar y rejilla de ocupación

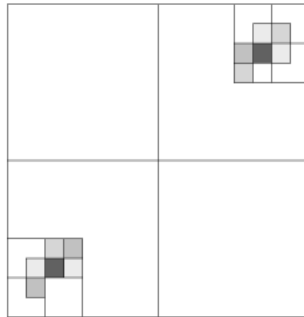


# Representaciones basadas en árboles

## Uniforme

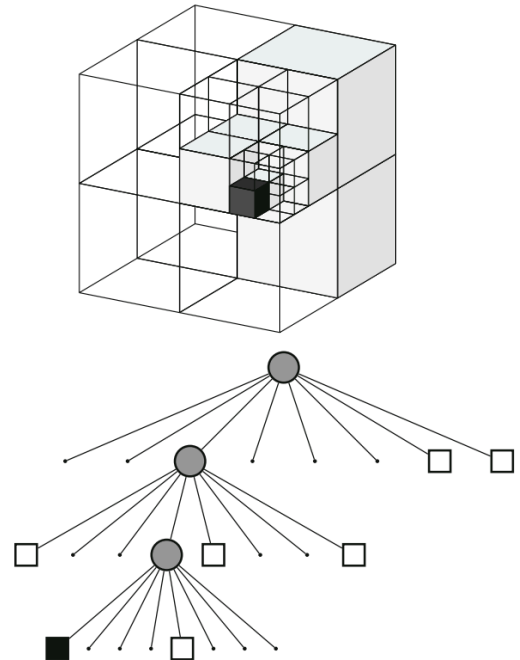


## Quadtree



2D

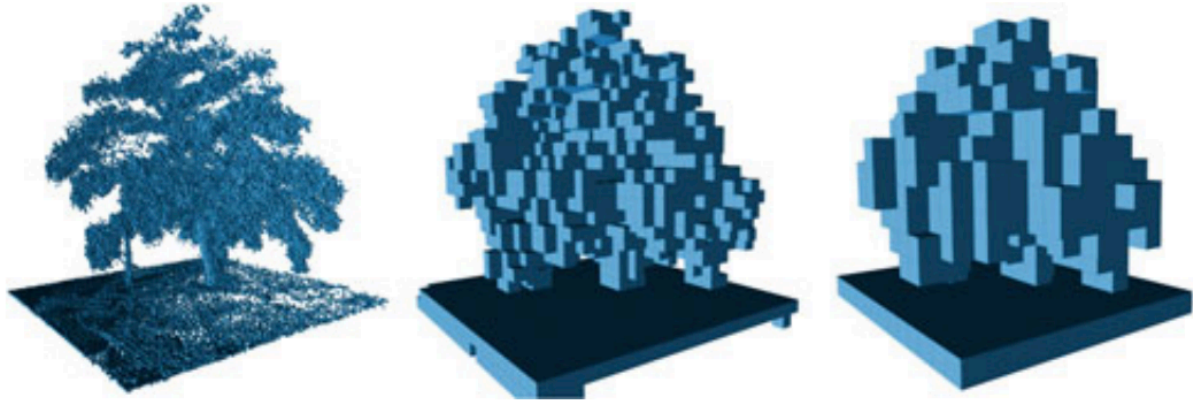
## Octree



3D

# Representaciones basadas en árboles

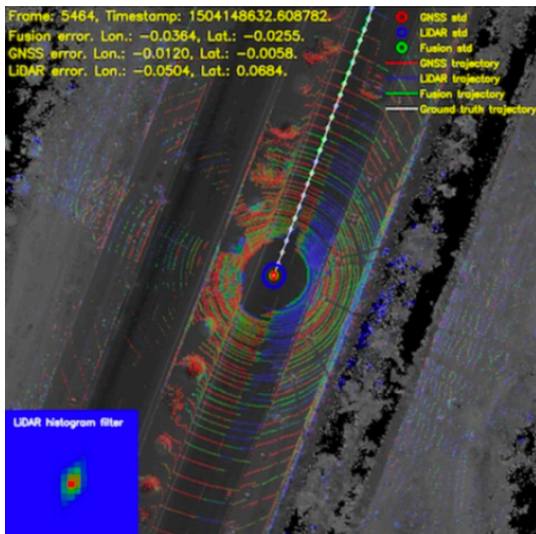
- Eficientes en espacio y tiempo
- Multi-resolución



OctoMap: an efficient probabilistic 3D mapping framework based on octrees, Armin Hornung · Kai M. Wurm · Maren Bennewitz · Cyrill Stachniss · Wolfram Burgard, *Autonomous Robots*, 34, 2013

# Localización en vehículos autónomos

- Uso habitual de **filtros bayesianos discretos**
  - El GPS proporciona una estimación inicial razonable (error de algunos metros)
  - Basta con una rejilla “pequeña”



Robust and Precise Vehicle Localization based on Multi-sensor Fusion in Diverse City Scenes

# Resumen

- Los filtros discretos son una forma alternativa de implementar los filtros Bayes
- Se basan en histogramas para representar la densidad.
- Su precisión depende de la resolución de la cuadrícula.
- Tienen grandes requerimientos de memoria y cómputo
- Puede recuperarse fácilmente de errores de localización
- Es necesario realizar aproximaciones especiales para que este enfoque use memoria dinámica y sea computacionalmente eficiente.

# Índice

Filtros discretos  
**Filtro de Kalman (KF) y Filtro de Kalman Extendido (EKF)**

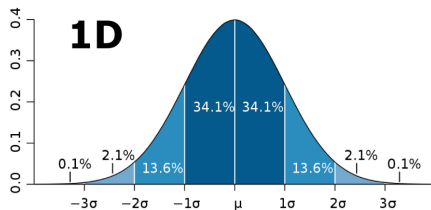
# Suposición básica

- Vamos a representar **Bel** con una distribución gaussiana
  - Ventaja: simplicidad, solo necesitamos dos parámetros: media y varianza
  - Problema: unimodal, no hay “hipótesis alternativas”

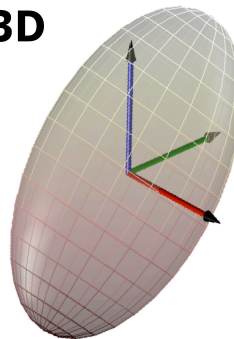
$$p(x) \sim N(\mu, \sigma^2):$$

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}(x-\mu)^2}$$

Univariate



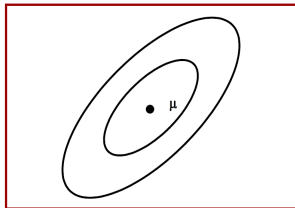
**3D**



$$p(\mathbf{x}) \sim N(\mu, \Sigma):$$

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1} (\mathbf{x}-\mu)}$$

Multivariate



# Filtro de Kalman

- Estima el estado  $x$  *de* un proceso discretizado en el tiempo

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

- Dada una observación (medida)

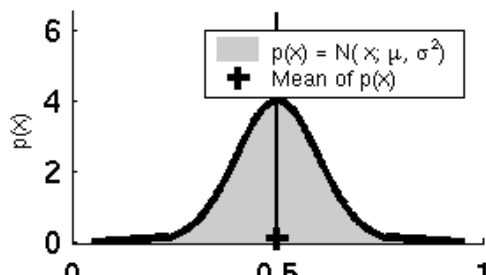
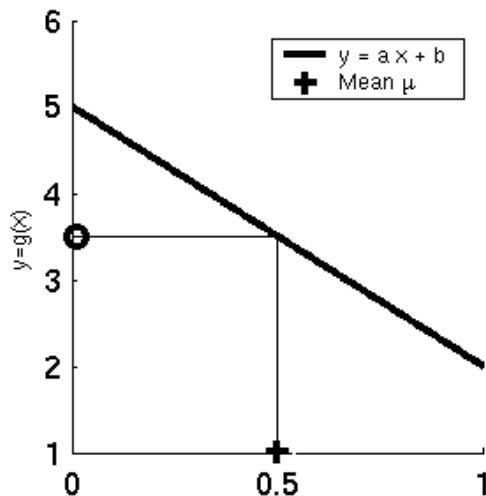
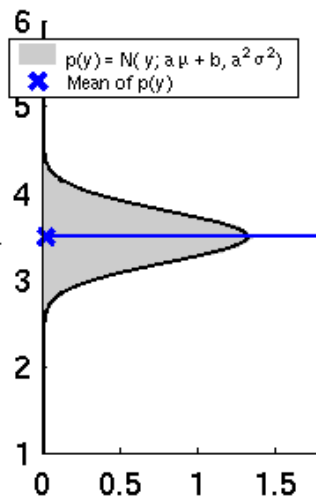
$$z_t = C_t x_t + \delta_t$$



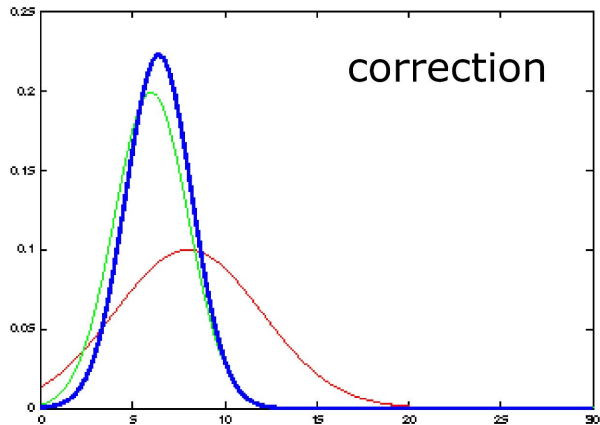
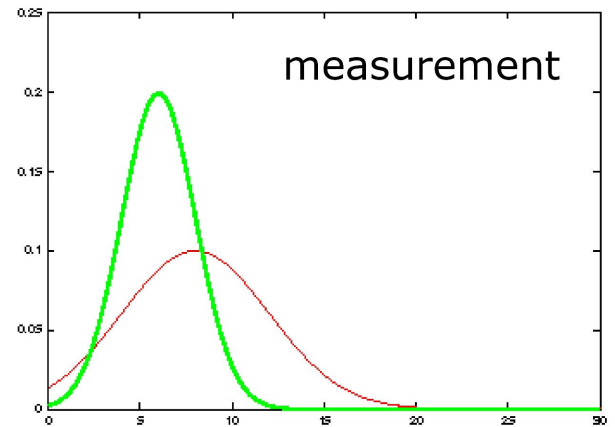
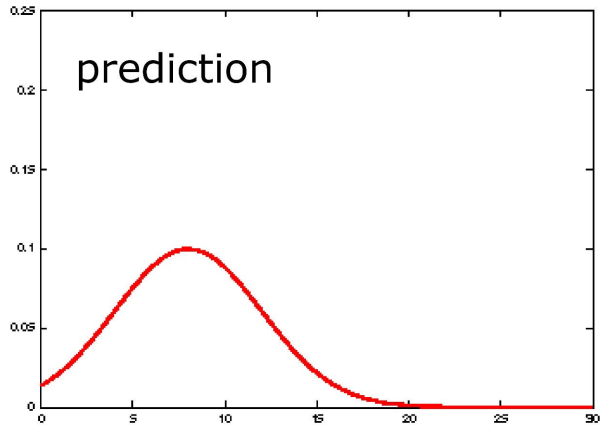
# Componentes del Filtro de Kalman

- $A_t$  Matriz ( $n \times n$ ) que describe cómo evoluciona el estado de  $t-1$  a  $t$  sin controles ni ruido.
- $B_t$  Matriz ( $n \times 1$ ) que describe como el control  $u_t$  cambia el estado de  $t-1$  a  $t$ .
- $C_t$  Matriz ( $k \times n$ ) que describe como mapear el estado  $x_t$  a una observación  $z_t$ .
- $\varepsilon_t$  Variables aleatorias que representan el ruido de movimiento y de medición. Son independientes y normalmente distribuidas con covarianza  $Q_t$  y  $R_t$  respectivamente
- $\delta_t$

# Sistemas lineales

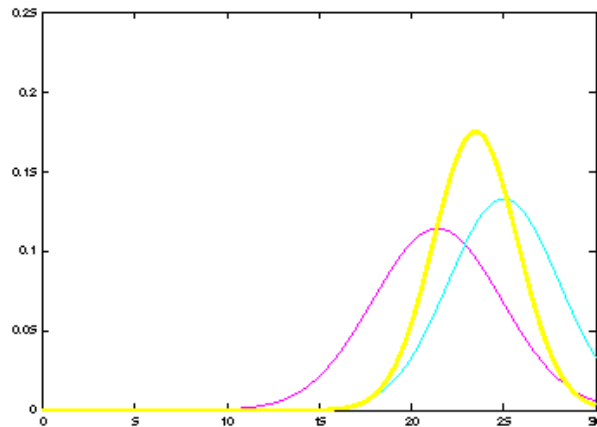
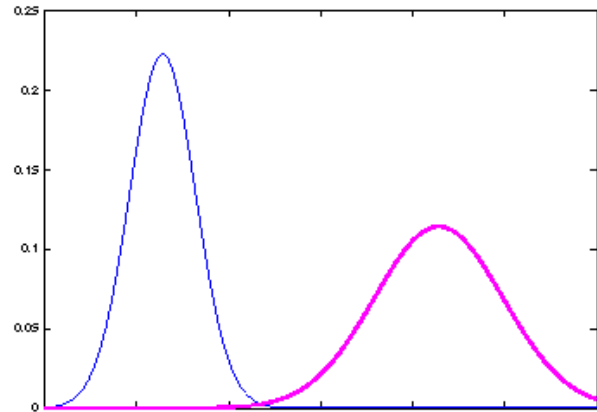
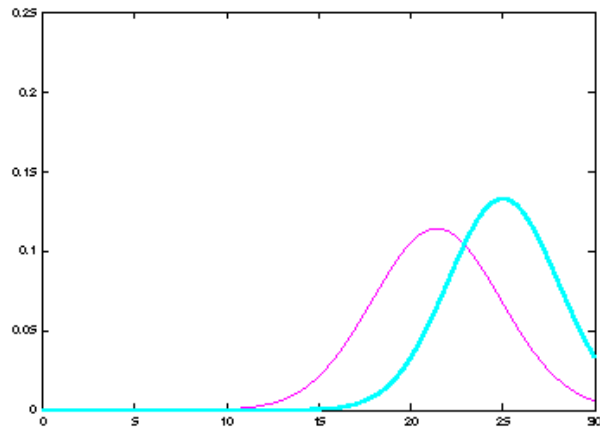
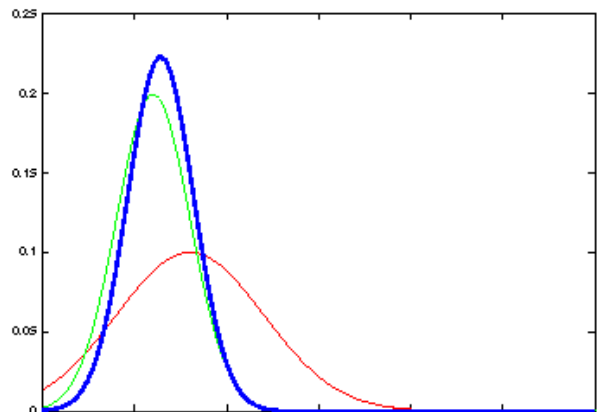


# Ejemplo actualización filtro de Kalman



It's a weighted mean!

# Ejemplo actualización filtro de Kalman (II)



# Algoritmo Filtro de Kalman

1. Algorithm **Kalman\_filter**(  $\mu_{t-1}$ ,  $\Sigma_{t-1}$ ,  $u_t$ ,  $z_t$ ):

2. Prediction:

3.  $\mu_t = A_t \mu_{t-1} + B_t u_t$

4.  $\bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + Q_t$

K: ganancia de Kalman, en qué grado influye la medida del sensor

5. Correction:

6.  $K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + R_t)^{-1}$

7.  $\mu_t = \mu_t + K_t (z_t - C_t \mu_t)$

8.  $\Sigma_t = (I - K_t C_t) \Sigma_t$  *Innovación: diferencia entre la medida real y la predicha*

9. Return  $\mu_t$ ,  $\Sigma_t$

# Sistemas no lineales

La mayoría de sistemas robóticos reales requieren funciones no lineales

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



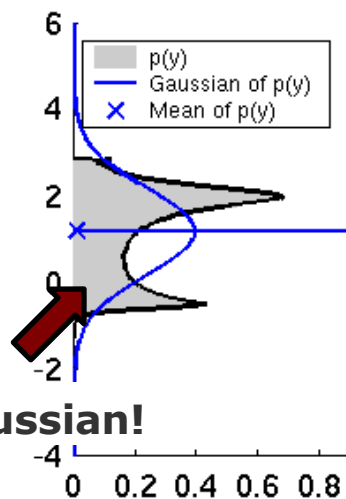
$$x_t = g(u_t, x_{t-1})$$

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

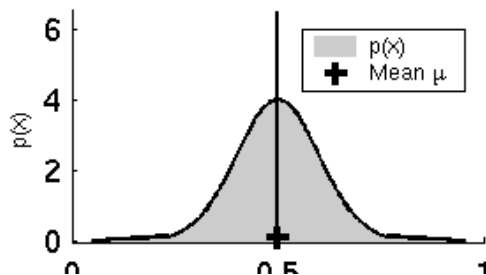
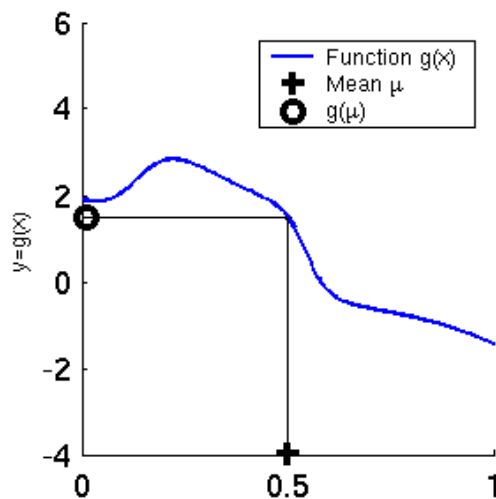


$$z_t = h(x_t)$$

# Sistemas no lineales



**Non-Gaussian!**



# Distribuciones no Gaussianas

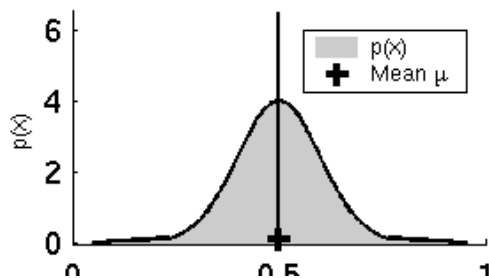
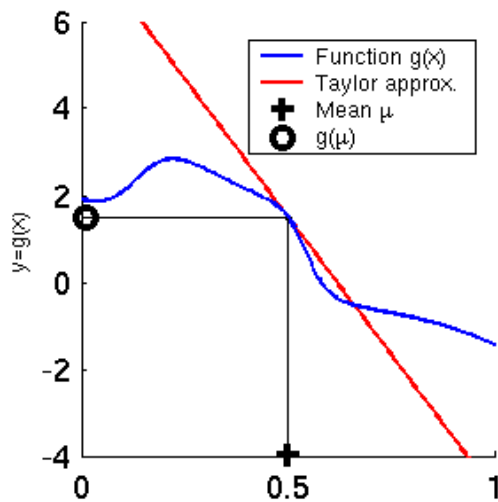
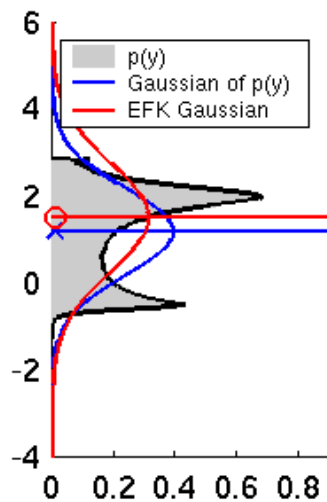
- Las funciones no lineales conducen a distribuciones no gaussianas
- El filtro de Kalman no es aplicable en estos casos!

## **Cómo podemos resolver este problema?**

Linealización: Teorema Expansión de Taylor



# EKF: Linealización



# Algoritmo EKF

1. **Extended\_Kalman\_filter**(  $\mu_{t-1}, \Sigma_{t-1}, u_t, z_t$ ):

2. Prediction:

$$3. \quad \bar{\mu}_t = \underline{g(u_t, \mu_{t-1})} \quad \longleftarrow \quad \bar{\mu}_t = A_t \mu_{t-1} + B_t u_t$$

$$4. \quad \bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + Q_t \quad \longleftarrow \quad \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + Q_t$$

5. Correction:

$$6. \quad K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + R_t)^{-1} \quad \longleftarrow \quad K_t = \bar{\Sigma}_t C_t^T (C_t \bar{\Sigma}_t C_t^T + R_t)^{-1}$$

$$7. \quad \mu_t = \bar{\mu}_t + K_t (z_t - \underline{h(\bar{\mu}_t)}) \quad \longleftarrow \quad \mu_t = \bar{\mu}_t + K_t (z_t - C_t \bar{\mu}_t)$$

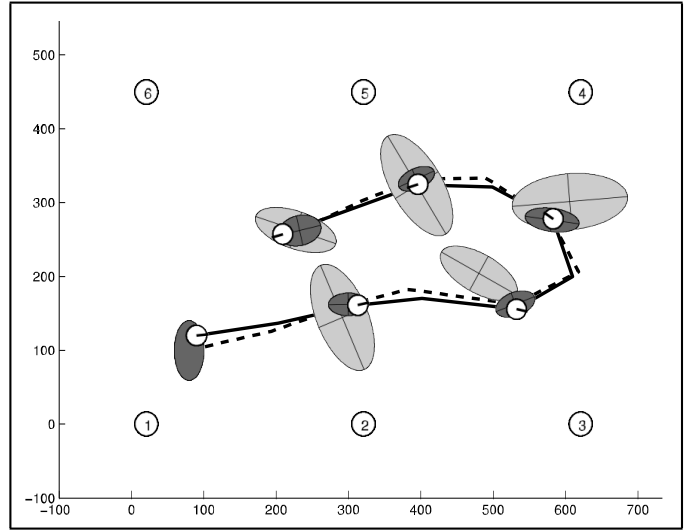
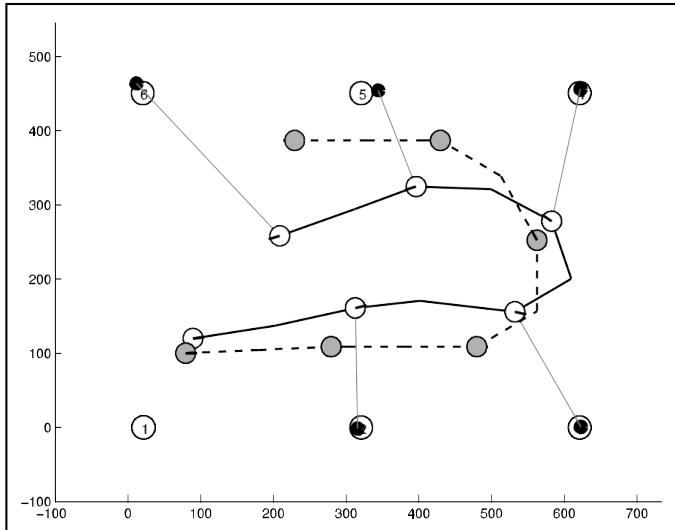
$$8. \quad \Sigma_t = (I - K_t H_t) \bar{\Sigma}_t \quad \longleftarrow \quad \Sigma_t = (I - K_t C_t) \bar{\Sigma}_t$$

9. Return  $\mu_t, \Sigma_t$

$$H_t = \frac{\partial h(\bar{\mu}_t)}{\partial x_t}$$

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

# Ejemplo EKF



# Resumen

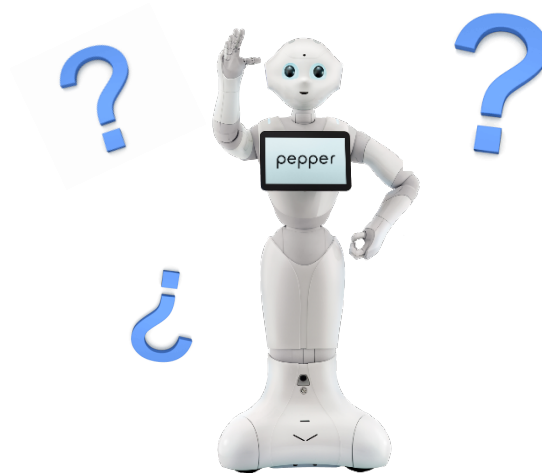
- El EKF es una solución ad-hoc para tratar las no linealidades.
- Realiza linealización local en cada paso
- Funciona bien en la práctica para no linealidades moderadas (ejemplo: localización de marcadores).
- Es óptimo si la medición y el modelo de movimiento son lineales, en cuyo caso el EKF se reduce a KF.
- EKF es una solución aproximada (no es una solución “closed-form”)

# Bibliografía

- Thrun, Burgard, Fox. Probabilistic Robotics. MIT Press, 2005.
  - Cap. 4 filtros discretos en general, Cap. 8 aplicación a la localización
  - Cap. 3. KF, EKF, cap. 7 aplicación a la localización

# Robots Móviles

## Filtros Discretos y Filtro de Kalman Extendido



Sergio Orts Escolano  
Otto Colomina Pardo

sorts @ ua.es

