Tema 7: Razonamiento Probabilístico en el Tiempo

Razonamiento y Representación del Conocimiento

Índice

- Introducción
- Filtros de Kalman
- El filtro de Kalman extendido
- Localización mediante EKF

Introducción

- Razonamiento probabilista en el tiempo
  - En ocasiones tenemos que mantener el conocimiento sobre el entorno a lo largo del tiempo
  - Intuitivamente:
    - Sin nuevas evidencias nuestro conocimiento del entorno se vuelve más incierto con el paso del tiempo
    - Al obtener una evidencia reduciremos la incertidumbre sobre alguna variable del entorno

Introducción

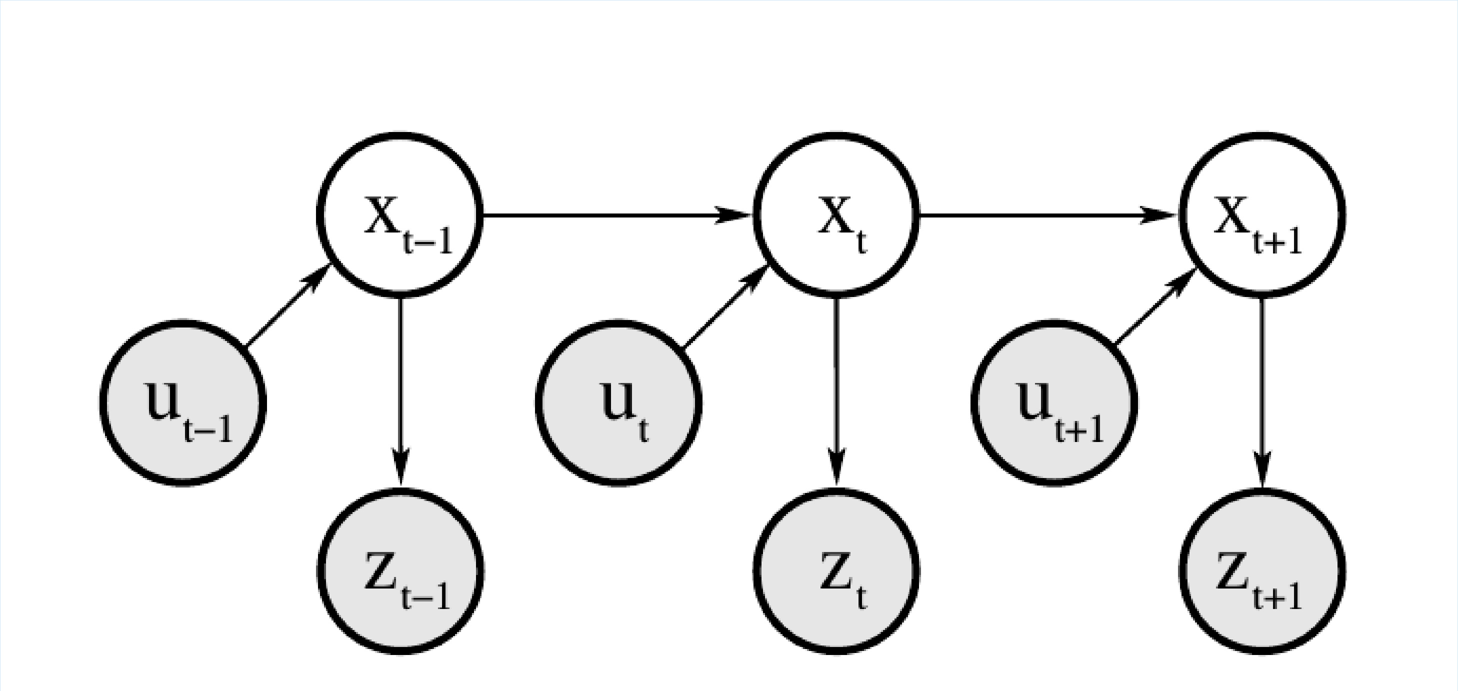
- Razonamiento probabilista en el tiempo
  - Aplicaciones: CUALQUIER sistema que evolucione temporalmente y sobre el que se desee hacer predicciones:
    - Modelos físicos: corrientes marinas, meteorología, etc.
    - Seguimiento de objetos → localización de robots
    - Sistemas económicos
    - Y casi cualquier cosa que se os ocurra

Introducción

- Razonamiento probabilista en el tiempo
  - Encontramos dos enfoques para resolverlo:
    - Modelos de Markov (y Modelos ocultos de Markov)
    - Filtros de Kalman (y filtros extendidos de Kalman)
  - Ambos enfoques son casos particulares de Redes Bayesianas Dinámicas
- Vamos a estudiar el Filtro de Kalman
  - Muy utilizado en robótica para tareas de localización de robots móviles y para resolver el problema de **SLAM**

Introducción

- Filtro de Kalman como una Red Bayesiana Dinámica



Introducción

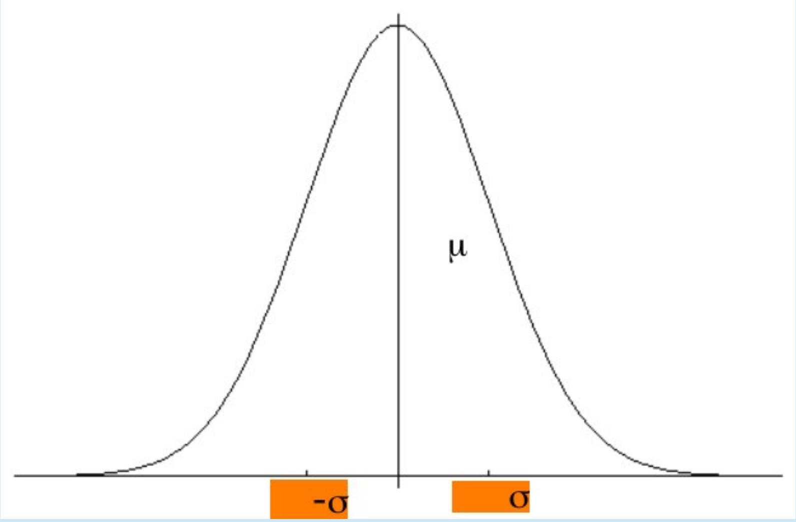
- Revisión de Gaussianas
  - Univariada

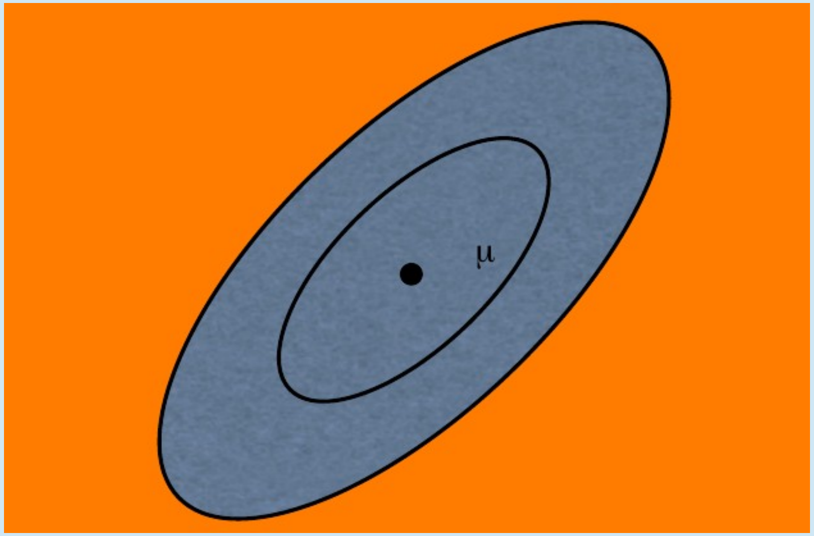
$$g(x) \sim N(\mu, \sigma^2):$$

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

  - Multivariada

$$g(x) = \frac{1}{\sqrt{Det(2\pi\Sigma)}} e^{\frac{-1}{2}(x-\mu)^T \Sigma (x-\mu)}$$



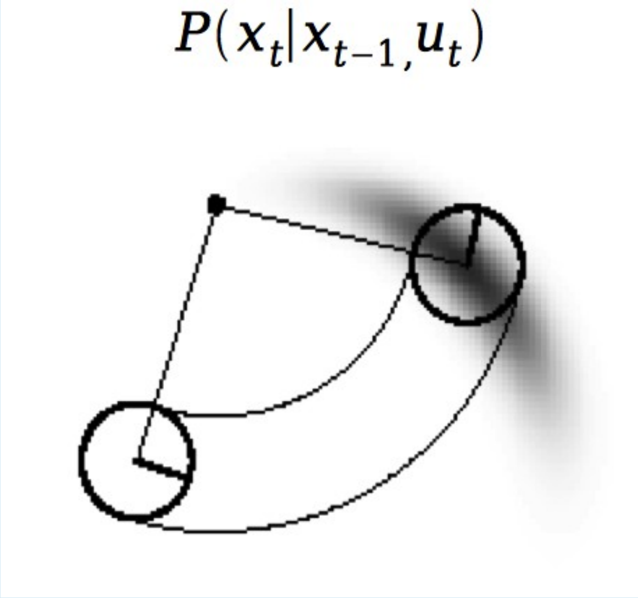


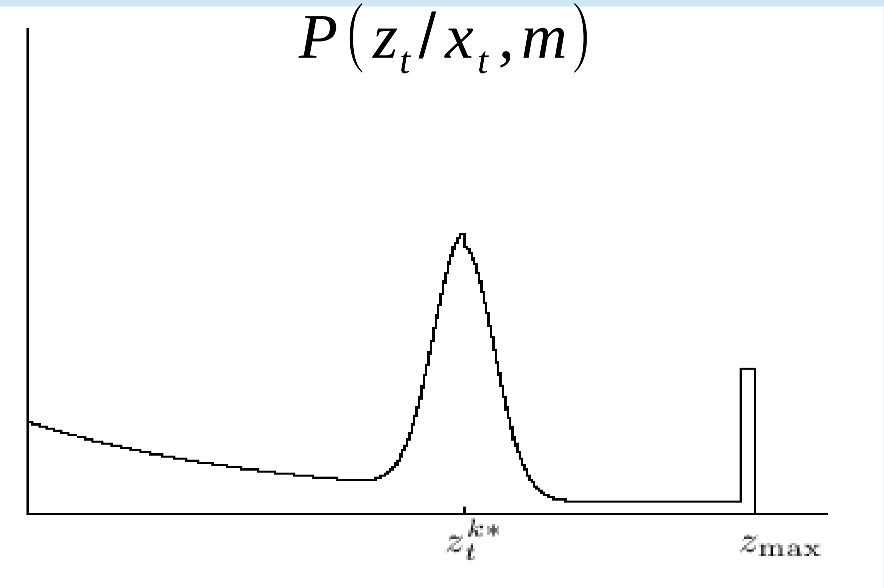
Introducción

- Ejemplo en IA: el problema de la localización de robots móviles
  - Imprescindible: mapa del entorno por el que se va a mover el robot
  - ¿Conocemos la posición inicial del robot?
    - Sí: tracking→ localización local
    - No: localización global
  - Sensores: el robot dispone de, al menos, un sensor para hacer observaciones del entorno
  - Asumimos ruido (errores) en el movimiento del robot (dado un comando) y en la lectura de los sensores

Filtro de Kalman

- El problema de la localización de robots móviles
- Vamos a trabajar con variables reales → pose de un robot en un plano: (X, Y, Φ)
- La incertidumbre la representamos por la varianza de una gaussiana centrada en cada variable
- Asumimos un modelo gaussiano para el movimiento del robot y para los sensores





Filtro de Kalman

- Componentes del filtro de Kalman
  - A_t: matriz (nxn) que describe cómo evoluciona el estado desde t a t-1 sin comandos de control o ruido
  - B_t: matriz (nxl) que describe cómo los comandos de control cambian el estado desde t a t-1
  - C_t: Matriz (kxn) que describe cómo mapear el estado x_t a una observación z_t
  - ε_t y δ_t: variables aleatorias que representan el ruido de procesado y medida. Se asumen independientes y normalmente distribuídas con covarianzas R_t y Q_t respectivamente

Filtro de Kalman

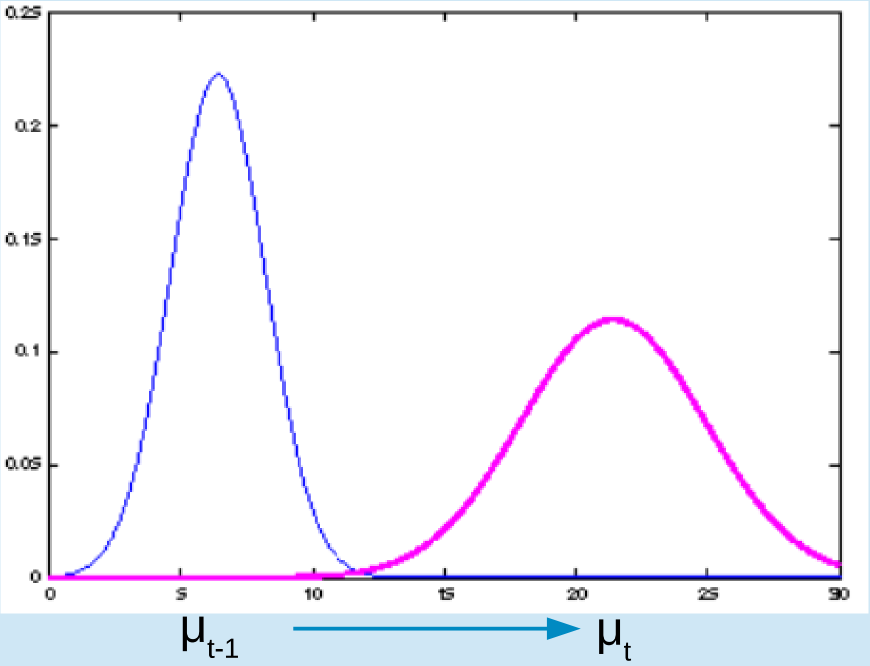
- Ciclo predicción-corrección
  - El filtro de kalman realiza iterativamente dos funciones:
    - Predicción del siguiente estado dado el comando de movimiento
    - Corrección del siguiente estado dadas las observaciones
  - Intiutivamente:
    - El paso del tiempo/movimiento incrementa la incertidumbre en el estado del sistema (pose del robot)
    - Las observaciones reducen la incertidumbre del sistema

Filtro de Kalman

- Ciclo predicción-corrección
  - Predicción

$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = a_t \mu_{t-1} + b_t u_t \\ \bar{\sigma}_t^2 = a_t^2 \sigma_t^2 + \sigma_{act,t}^2 \end{cases}$$

$$\overline{bel}(x_t) = \begin{cases} \bar{\mu}_t = A_t \mu_{t-1} + B_t u_t \\ \bar{\Sigma}_t = A_t \Sigma_{t-1} A_t^T + R_t \end{cases}$$

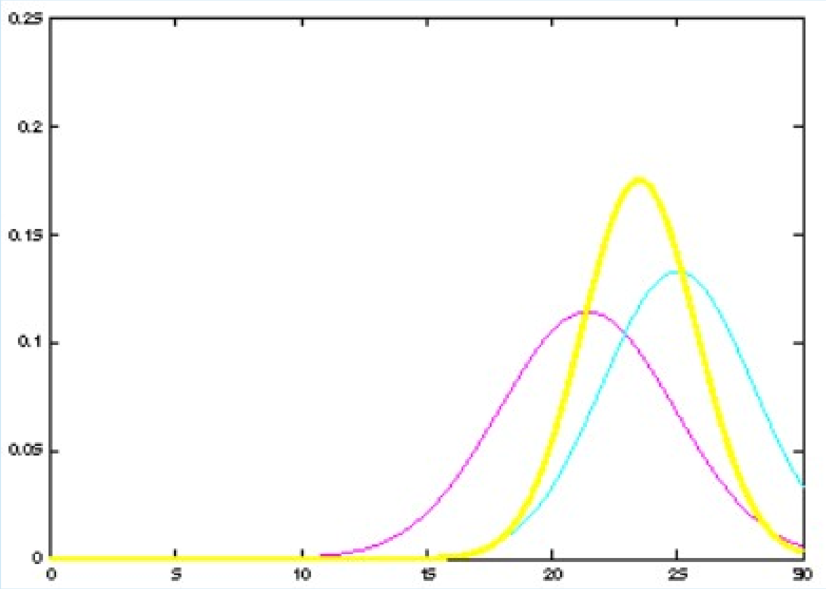


Predicción de la posición dado un comando de movimiento

Filtro de Kalman

- Ciclo predicción-corrección
  - Corrección

$$bel(x_t) = \begin{cases} K_t = \overline{\sigma}_t^2 / (\overline{\sigma}_t^2 + \sigma_{obs,t}^2) \\ \mu_t = \overline{\mu_{t-1}} + K_t(z_t - \overline{\mu}_t) \\ \sigma_t^2 = (1 - K_t)\overline{\sigma}_t^2 \end{cases}$$



μ_t z_t

Corrección de la pose tras la observación z N(z_t, $\sigma_{obs,t}$)

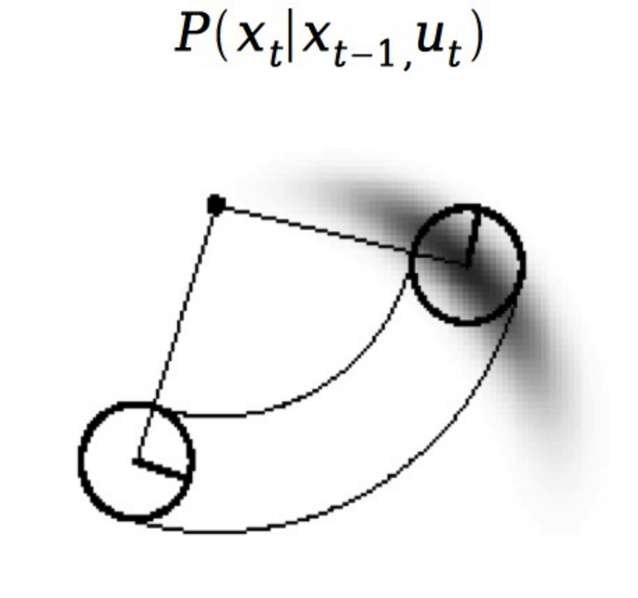
$$bel(x_t) = \begin{cases} K_t = \overline{\Sigma}_t C_t^T (C_t \overline{\Sigma}_t C_t^T + Q_t)^{-1} \\ \mu_t = \overline{\mu}_t + K_t(z_t - C_t \overline{\mu}_t) \\ \Sigma_t = (I - K_t C_t)\overline{\Sigma}_t \end{cases}$$

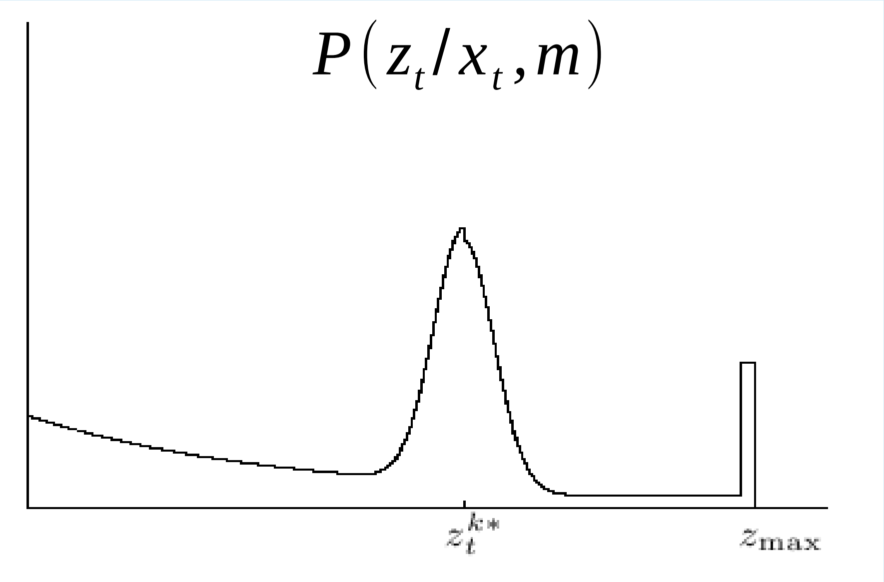
Filtro de Kalman

- Características
  - Óptimo: estima de forma exacta los cambios en el estado de un sistema → siempre que sea un sistema Gaussiano lineal
  - Forma cerrada: la entrada es una gaussiana y la salida también
  - Eficiente: coste polinomial dada la dimensionalidad de las medidas (k) y del estado (n) → $O(k^{2.376} + n^2)$

Filtro de Kalman Extendido

- Problema: No linealidad → cuando los sistemas no se pueden modelar mediante gaussianas lineales el filtro de Kalman no se puede utilizar
- La mayoría de sistemas robóticos se modelan de forma no-lineal





Filtro de Kalman Extendido

- No linealidad → El Filtro de Kalman Extendido (EKF) surge para dar una solución (aproximada, no óptima) a este problema
  - Ahora, la modelización del sistema se lleva a cabo por funciones no lineales
  - g: obtiene la pose del robot dados la pose anterior y el comando de movimiento
  - h: obtiene la medida del sensor

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

$$z_t = h(x_t) + \delta_t$$

Filtro de Kalman Extendido

- Linearización
  - EKF realiza una linealización de g y h utilizando una expansión de Taylor (primer orden) → aproximación de g y h mediante una función lineal

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

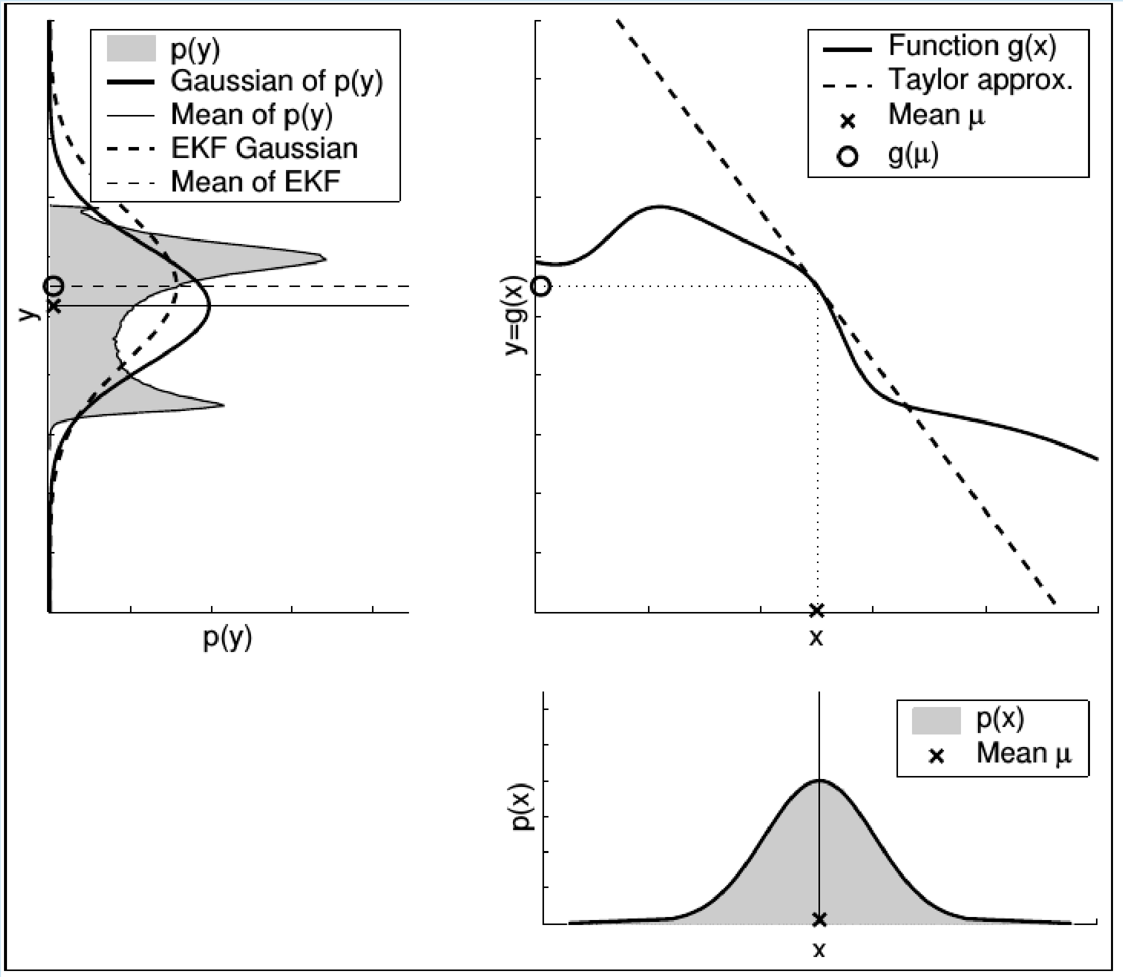
$$g(u_t, x_{t-1}) \approx g(u_t, \mu_{t-1}) + G_t \, (x_{t-1} - \mu_{t-1})$$

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

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

Filtro de Kalman Extendido

- Linearización: Explicación gráfica



Filtro de Kalman Extendido

- Algoritmo EKF

Filtro_Kalman_Extendido(μ_{t-1}, Σ_{t-1}, u_t, z_t)

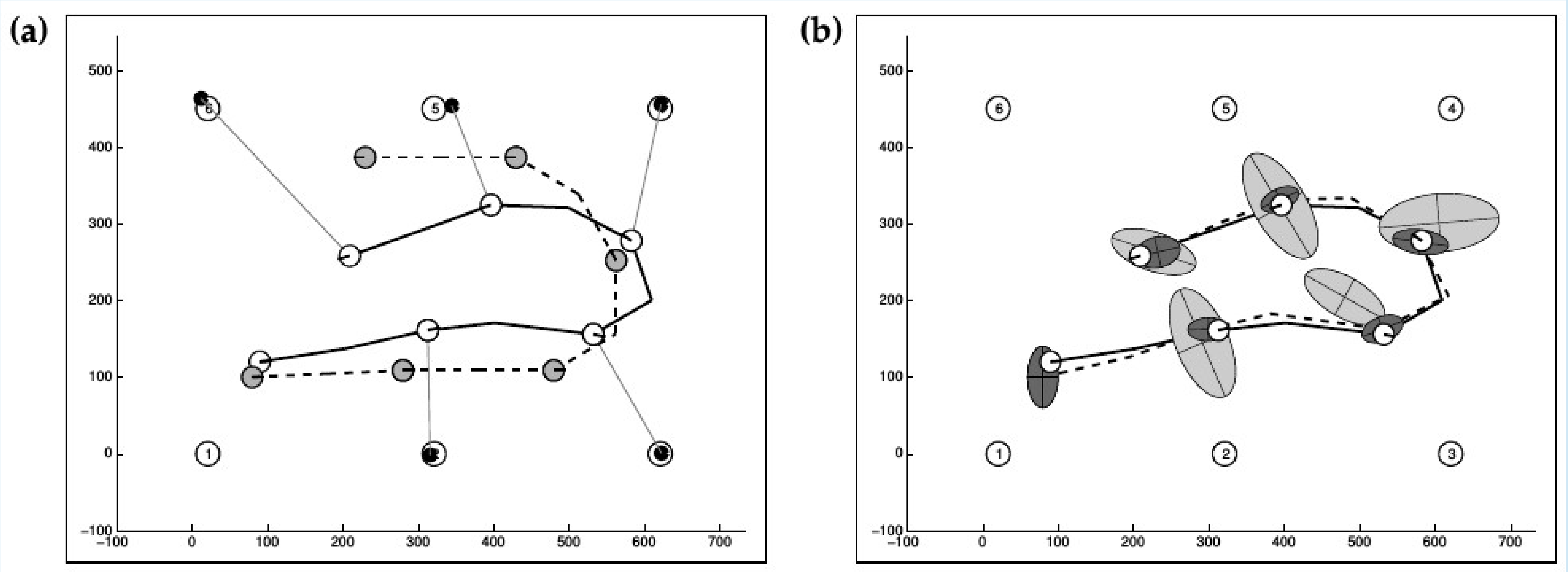
1. $\bar{\mu}_t = g(u_t, \mu_{t-1})$
2. $\bar{\Sigma}_t = G_t \Sigma_{t-1} G_t^T + R_t$
3. $K_t = \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1}$
4. $\mu_t = \bar{\mu}_t + K_t (z_t - h(\bar{\mu}_t))$
5. $\Sigma_t = (I - K_t H_t) \bar{\Sigma}_t$
6. Return μ_t, Σ_t

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

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

Filtro de Kalman Extendido

- Ejemplo

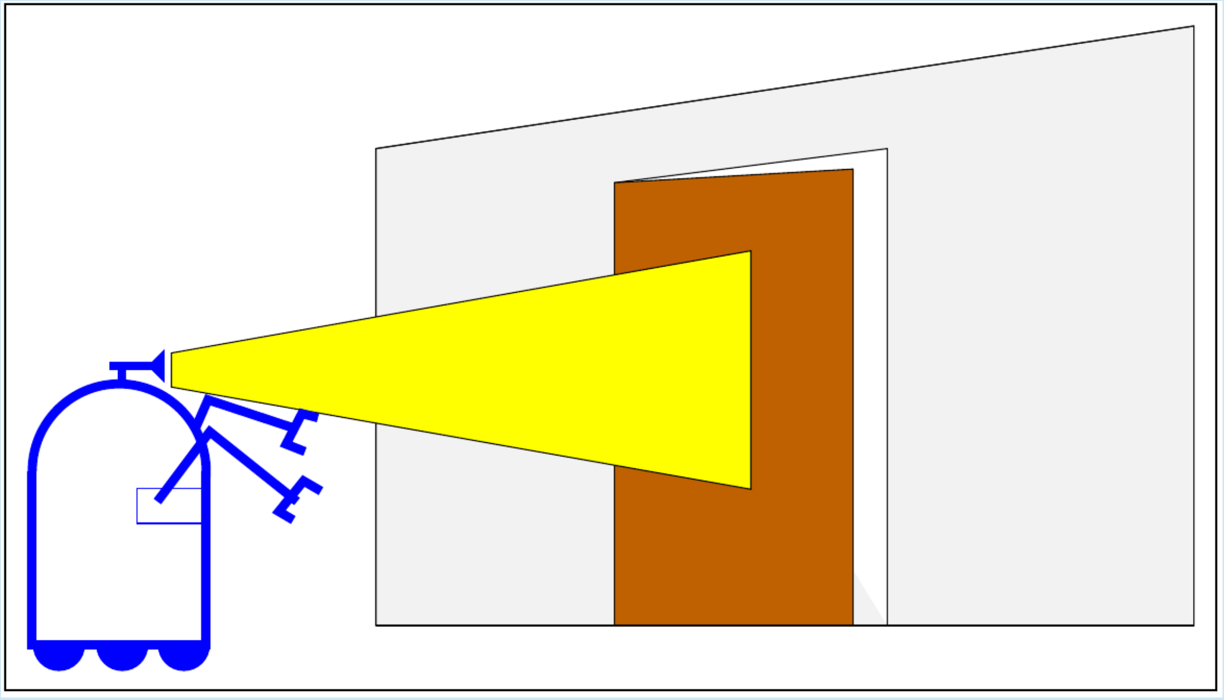


Localización mediante EKF

- La localización EKF nos sirve para mantener la información (con incertidumbre) de la pose del robot mediante una gaussiana
- La media de la gaussiana indica dónde es más probable que se encuentre el robot, y la covarianza cómo de seguros estamos de que esté ahí
- Esto es posible si partimos de una posición inicial conocida → Tracking

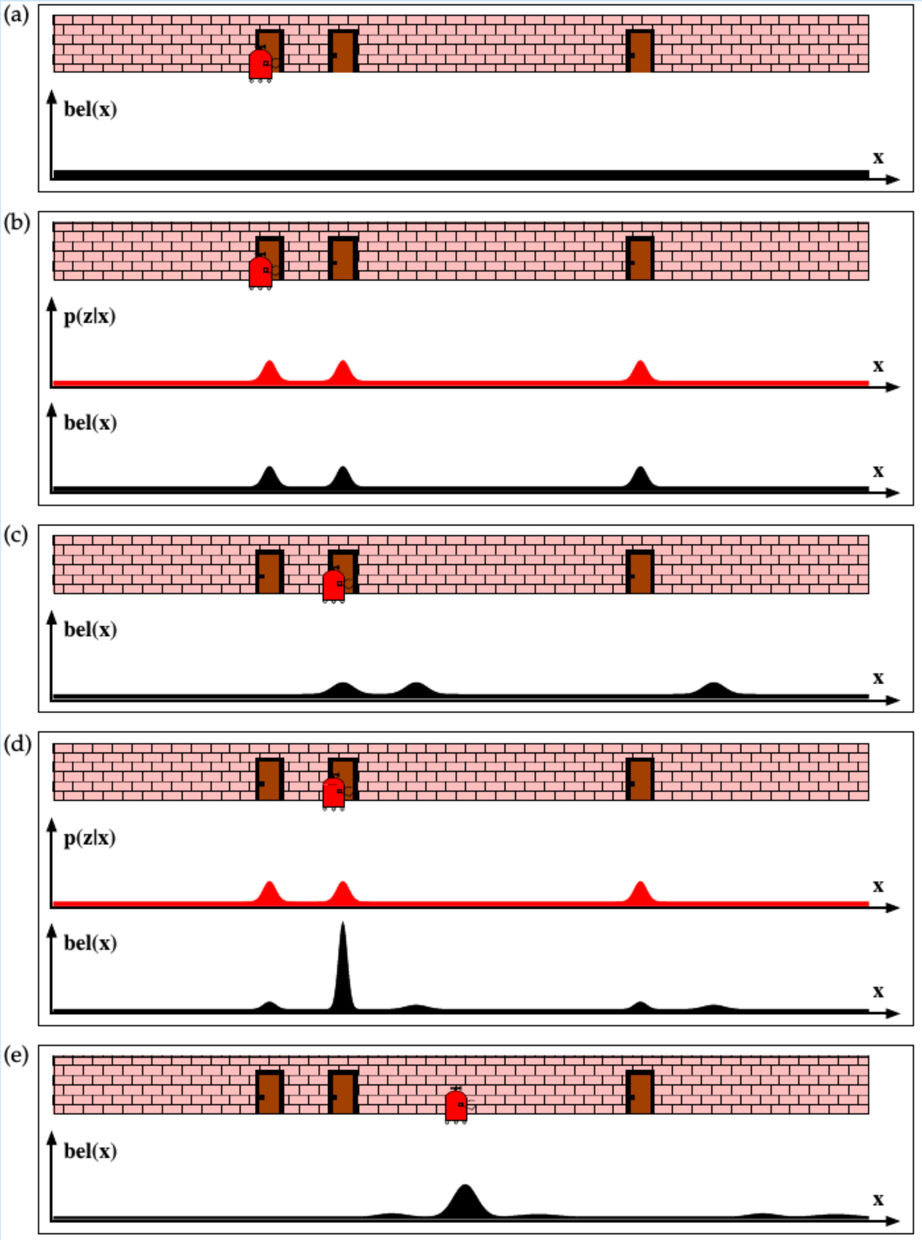
Localización mediante EKF

- Posición inicial desconocida → Algoritmo de localización de Markov
- Veamos un sencillo ejemplo 1-dimensional



Localización mediante EKF

- Algoritmo de localización de Markov
    - Mantenemos una función de distribución de probabilidad sobre cada posible pose del robot en el mapa → bel(x)



Bibliografía recomendada

- Inteligencia Artificial. Un enfoque Moderno. Stuart Russell, Peter Noving. Ed. Prentice Hall. 2004
- Pattern Recognition and Machine Learning. C. M. Bishop. Springer. 2006
- Probabilistic Robotics. S. Thrun, D. Fox, W. Burgard. 2006. MIT Press