Robots Móviles

Tema 3. Localización y mapeado de un robot móvil
Parte I. Modelo de error del movimiento y de los sensores

Sergio Orts Escolano
Otto Colomina Pardo

Índice

Incertidumbre en la robótica

- Los sensores obtienen lecturas imprecisas del entorno.
- De ellos queremos inferir (estimar) el estado más probable del mundo.
- El movimiento del robot es también una fuente de errores (odometría).
- **Tenemos que modelar el error y la imprecisión.**

Teoría de la probabilidad y robótica

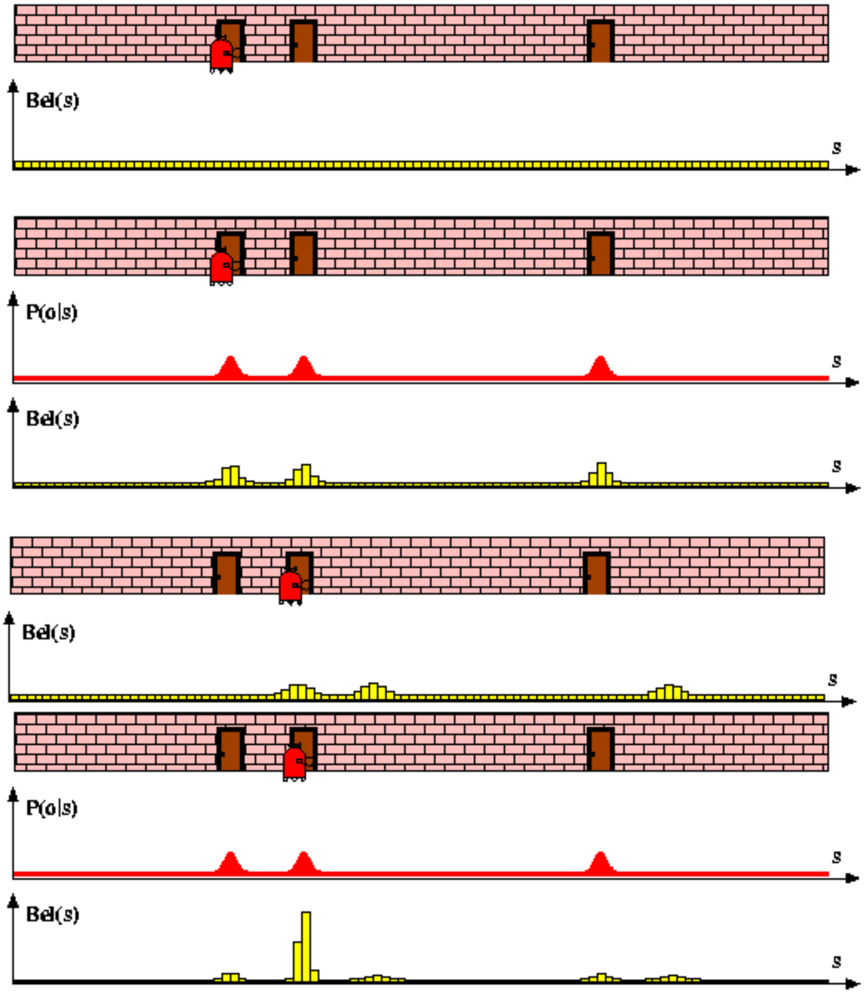
- **Variables aleatorias** y **funciones de probabilidad** para modelar las lecturas de los sensores.
- **Varianza** y covarianza para modelar el error.
- El **teorema de Bayes** para inferir datos desconocidos.
- **Filtros bayesianos** para integrar las lecturas.

$$\boxed{Bel(x_t) = \eta P(z_t \mid x_t) \int P(x_t \mid u_t, x_{t-1}) Bel(x_{t-1}) dx_{t-1}}$$

Modelo sensor

Modelo movimiento

Conocimiento estado anterior


Bel(s)
s
P(o|s)
s
Bel(s)
s
Bel(s)
s
P(o|s)
s
Bel(s)
s

Índice

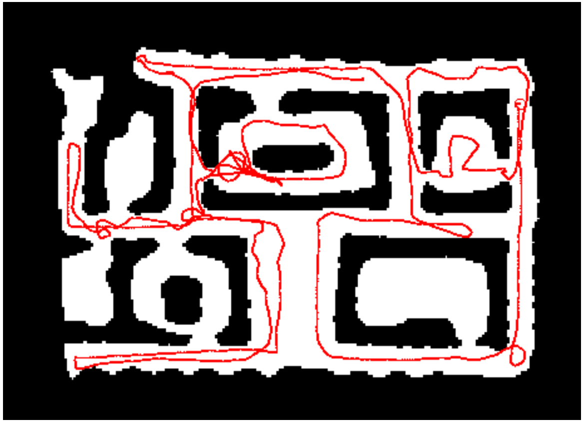
Incertidumbre en robótica móvil

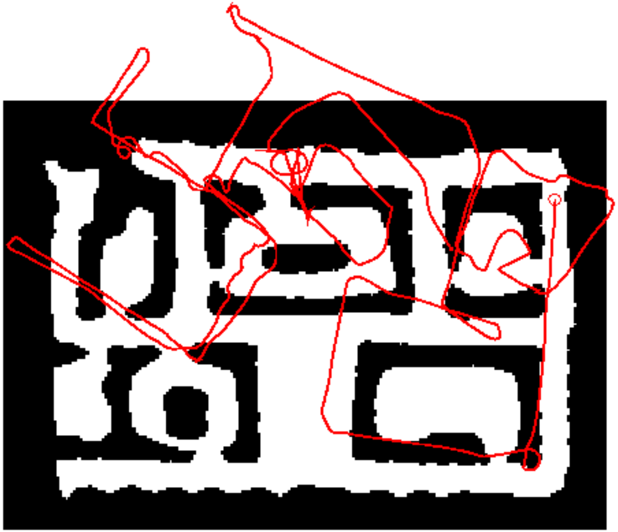
Modelo de error movimiento

Modelo de error de los sensores

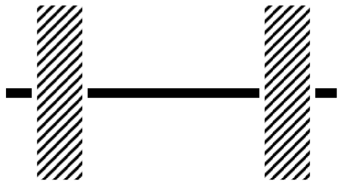
Modelo movimiento robot

- El movimiento del robot es intrínsecamente erróneo
- ¿Cómo podemos modelarlo?

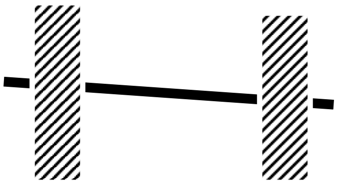




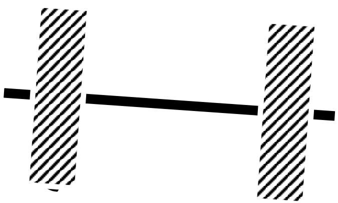
Causas de error en el movimiento



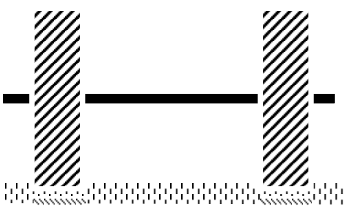
Caso ideal



Ruedas con
distinto diámetro



Suelo con sobresaltos



Terrenos rugosos

Modelos de movimiento probabilístico

- Para implementar un filtro Bayesiano, necesitamos el modelo de transición $p(x \mid x', u)$.
- El término $p(x \mid x', u)$ especifica una probabilidad a posteriori, que la acción u traslada al robot desde x' a x.
- En esta sección hablaremos de como $p(x \mid x', u)$ puede modelarse basándonos en las ecuaciones de movimiento.

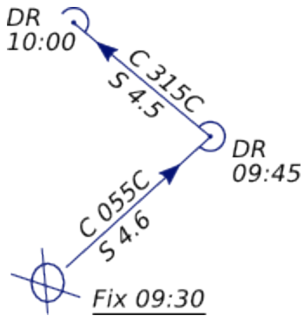
Tipos de modelos de movimiento

- En la práctica, a menudo se encuentran dos tipos de modelos de movimiento:
  - Basados en odometría
  - Basados en velocidad (dead reckoning)
- Los modelos basados en la odometría se utilizan cuando los sistemas están equipados con encoders.
- Los modelos basados en la velocidad tienen que ser aplicados cuando no se dan encoders.
  - Calculan la nueva pose basándose en las velocidades y el tiempo transcurrido.

10

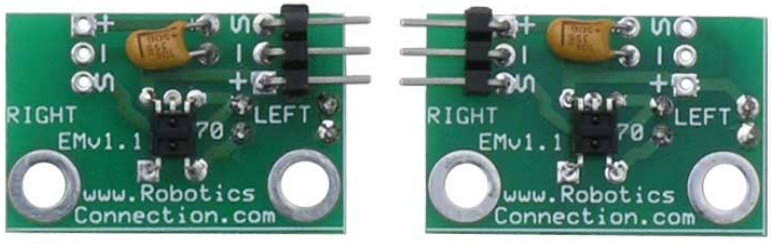
Dead Reckoning

- Derivado de “deduced reckoning.”
- Procedimiento matemático para determinar la ubicación actual de un vehículo.
- En navegación, el “dead reckoning” es el proceso de calcular la posición actual de un robot utilizando una posición previamente determinada, o fija, y avanzar esa posición basándose en las velocidades conocidas o estimadas sobre el tiempo y el curso transcurridos.
- Utilizado tradicionalmente para conocer la posición de los barcos



Ejemplo de encoders para ruedas

Estos módulos requieren +5V y GND para su alimentación, y proporcionan una salida de 0 a 5V. Proporcionan una salida de +5V cuando "ven" blanco, y una salida de 0V cuando "ven" negro.



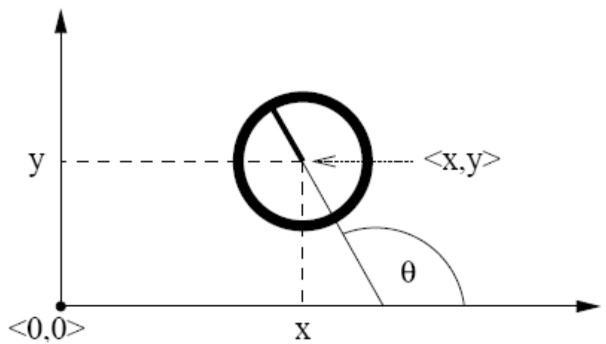
Estos discos se fabrican en plástico de color laminado de alta calidad para ofrecer una transición de negro a blanco muy nítida. Esto permite a un sensor de codificador de ruedas ver fácilmente las transiciones.

Fuente: http://www.active-robots.com/

Sistema de coordenadas

- En el caso general, la configuración de un robot puede describirse mediante seis parámetros:
  - Coordenadas cartesianas tridimensionales más tres ángulos de Euler: pitch, roll y tilt.
- A lo largo de esta sección asumiremos que el robot se mueve en un plano 2D (3 parámetros.

- El espacio de configuracíon es tridimensional (x,y,θ).



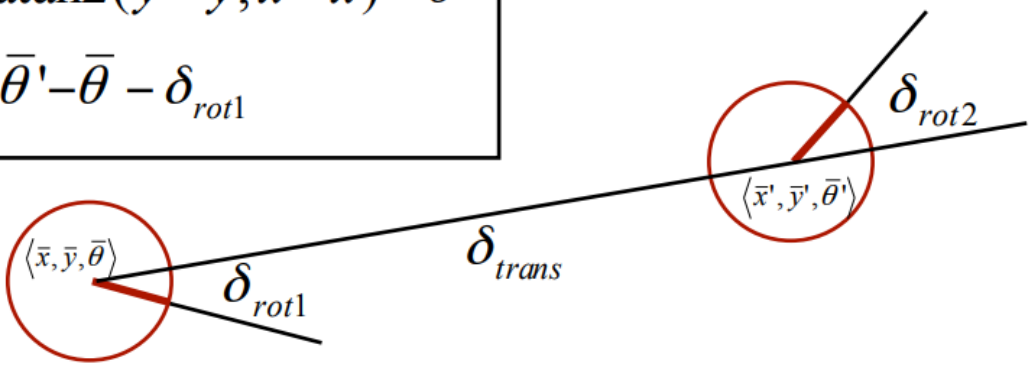
Modelo de odometría

- La odometría nos dice de $\langle \bar{x}, \bar{y}, \bar{\theta} \rangle$ a $\langle \bar{x}', \bar{y}', \bar{\theta}' \rangle$
- Descomponemos odometría $u = \langle \delta_{rot1}, \delta_{rot2}, \delta_{trans} \rangle$

$$\delta_{trans} = \sqrt{(\bar{x}' - \bar{x})^2 + (\bar{y}' - \bar{y})^2}$$

$$\delta_{rot1} = \text{atan2}(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\theta}$$

$$\delta_{rot2} = \bar{\theta}' - \bar{\theta} - \delta_{rot1}$$



Modelo de error para la odometría

- El movimiento medido viene dado por el movimiento verdadero contaminado por el ruido.

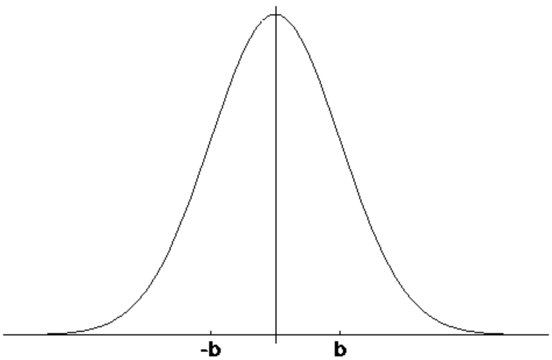
$$\hat{\delta}_{rot1} = \delta_{rot1} + \varepsilon_{\alpha_1 |\delta_{rot1}| + \alpha_2 |\delta_{trans}|}$$

$$\hat{\delta}_{trans} = \delta_{trans} + \varepsilon_{\alpha_3 |\delta_{trans}| + \alpha_4 |\delta_{rot1} + \delta_{rot2}|}$$

$$\hat{\delta}_{rot2} = \delta_{rot2} + \varepsilon_{\alpha_1 |\delta_{rot2}| + \alpha_2 |\delta_{trans}|}$$

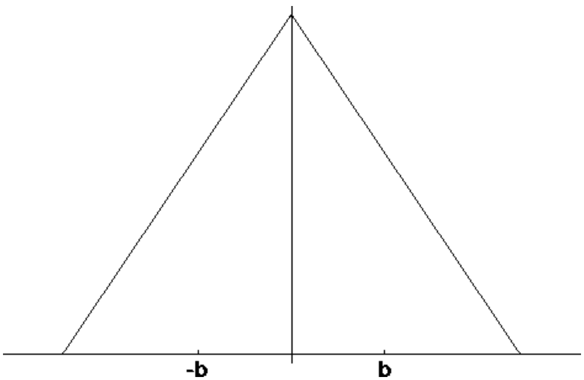
Distribuciones típicas para modelos probabilísticos de movimiento

Distribución normal



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

Distribución triangular



$$\varepsilon_{\sigma^2}(x) = \begin{cases} 0 \text{ if } |x| > \sqrt{6\sigma^2} \\ \dfrac{\sqrt{6\sigma^2} - |x|}{6\sigma^2} \end{cases}$$

Calculando la probabilidad

- Para una distribución normal

Query point

1. Algoritmo prob_normal_distribution(a,b):
2. devuelve $$\frac{1}{\sqrt{2\pi\, b^2}} \exp\left\{-\frac{1}{2}\frac{a^2}{b^2}\right\}$$

Std deviation

- Para una distribución triangular

1. Algorithmo prob_triangular_distribution(a,b):
2. devuelve $$\max\left\{0, \frac{1}{\sqrt{6}\, b} - \frac{|a|}{6\, b^2}\right\}$$

18

Calculando la probabilidad a posteriori

Algoritmo motion_model_odometry(x,x' ,u)

1. $\delta_{trans} = \sqrt{(\bar{x}' - \bar{x})^2 + (\bar{y}' - \bar{y})^2}$
2. $\delta_{rot1} = \text{atan2}(\bar{y}' - \bar{y}, \bar{x}' - \bar{x}) - \bar{\theta}$
3. $\delta_{rot2} = \bar{\theta}' - \bar{\theta} - \delta_{rot1}$

valores odometría (u)

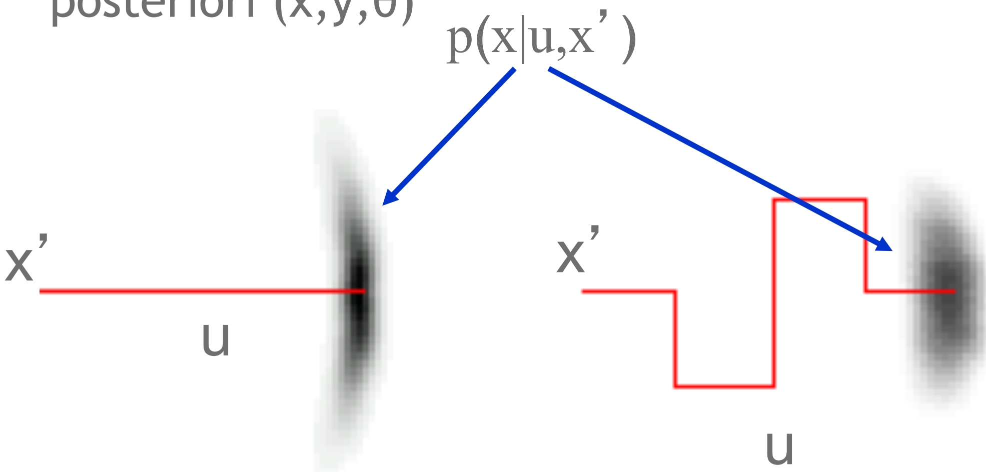
4. $\hat{\delta}_{trans} = \sqrt{(x' - x)^2 + (y' - y)^2}$
5. $\hat{\delta}_{rot1} = \text{atan2}(y' - y, x' - x) - \bar{\theta}$
6. $\hat{\delta}_{rot2} = \theta' - \theta - \hat{\delta}_{rot1}$

valores de interés (x,x')

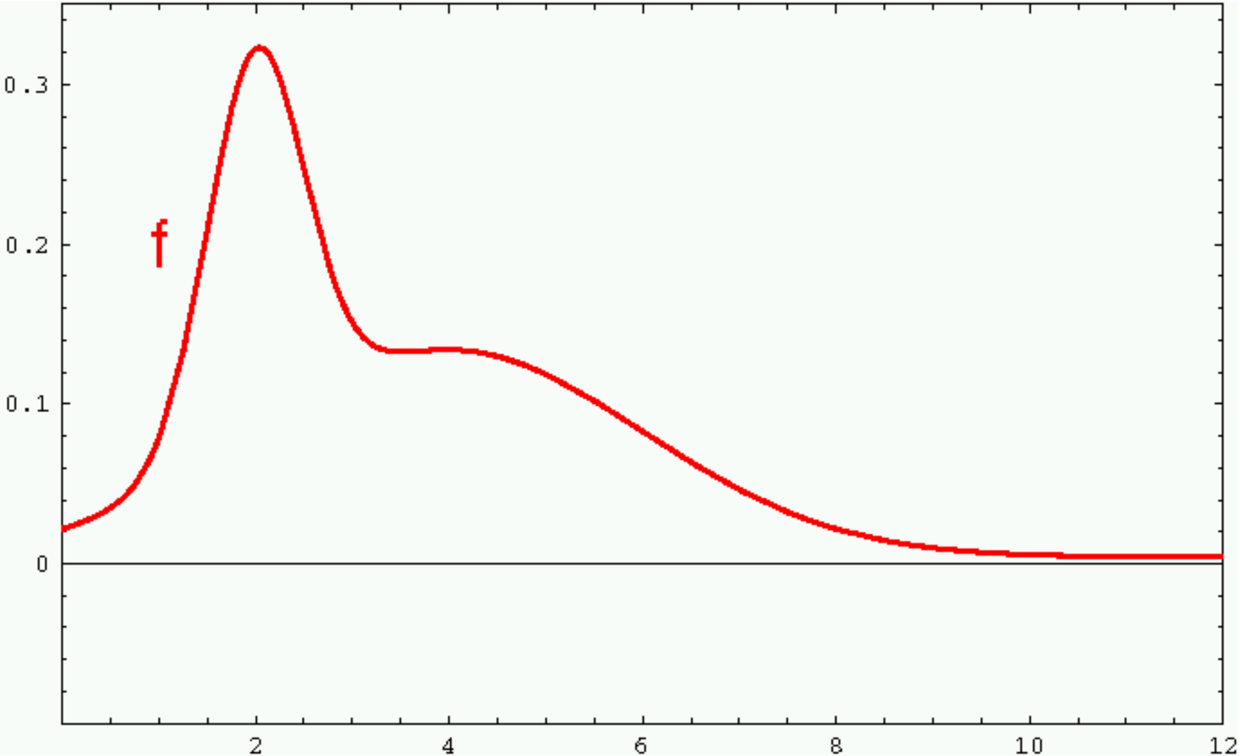
7. $p_1 = \text{prob}(\delta_{\text{rot1}} - \hat{\delta}_{\text{rot1}}, \alpha_1 |\hat{\delta}_{\text{rot1}}| + \alpha_2 \hat{\delta}_{\text{trans}})$
8. $p_2 = \text{prob}(\delta_{\text{trans}} - \hat{\delta}_{\text{trans}}, \alpha_3 \hat{\delta}_{\text{trans}} + \alpha_4 (|\hat{\delta}_{\text{rot1}}| + |\hat{\delta}_{\text{rot2}}|))$
9. $p_3 = \text{prob}(\delta_{\text{rot2}} - \hat{\delta}_{\text{rot2}}, \alpha_1 |\hat{\delta}_{\text{rot2}}| + \alpha_2 \hat{\delta}_{\text{trans}})$
10. devuelve $p_1 \cdot p_2 \cdot p_3$

Aplicación

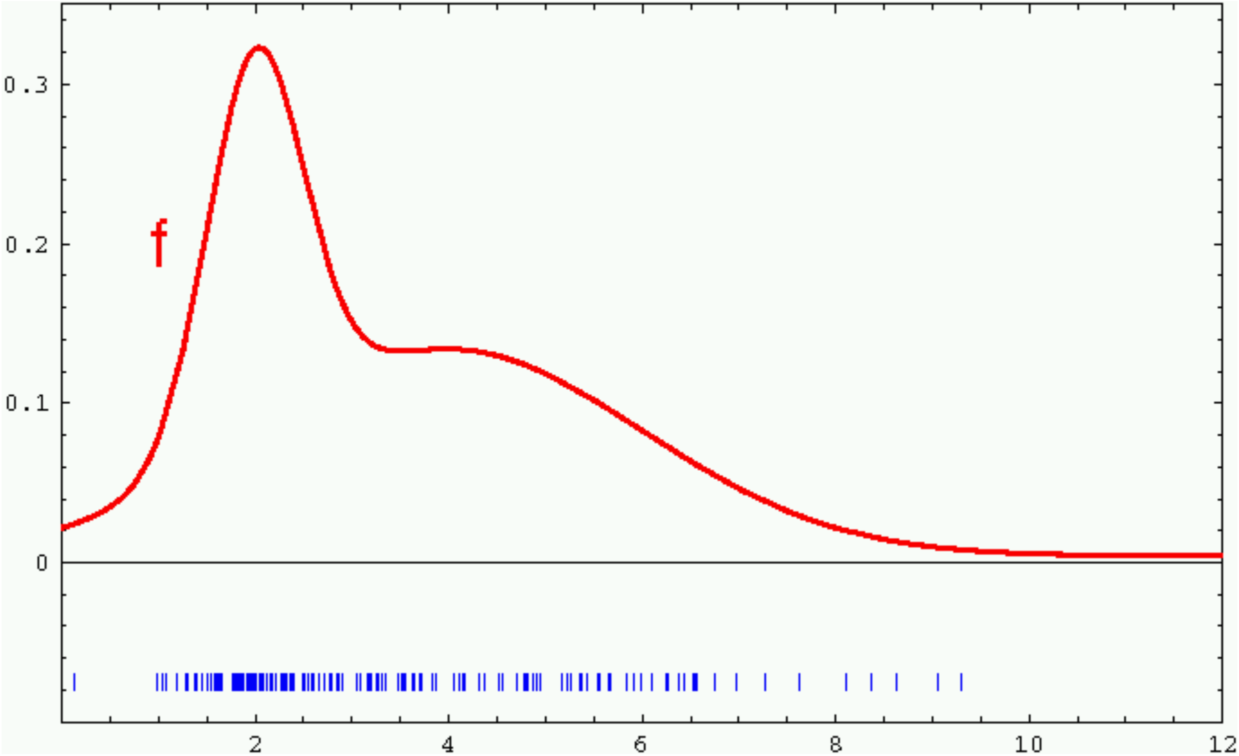
- Aplicación repetida del modelo de sensor para movimientos cortos
- Distribuciones típicas en forma de banana obtenidas para la proyección 2D de la probabilidad 3D a posteriori (x,y,θ)



Representación de la densidad basada en muestras



Representación de la densidad basada en muestras



Cómo tomar muestras de distribuciones normales o triangulares?

- Muestreando una distribución normal

1. Algoritmo sample_normal_distribution
2. devuelve $\frac{1}{2}\sum_{i=1}^{12} rand(-b,b)$

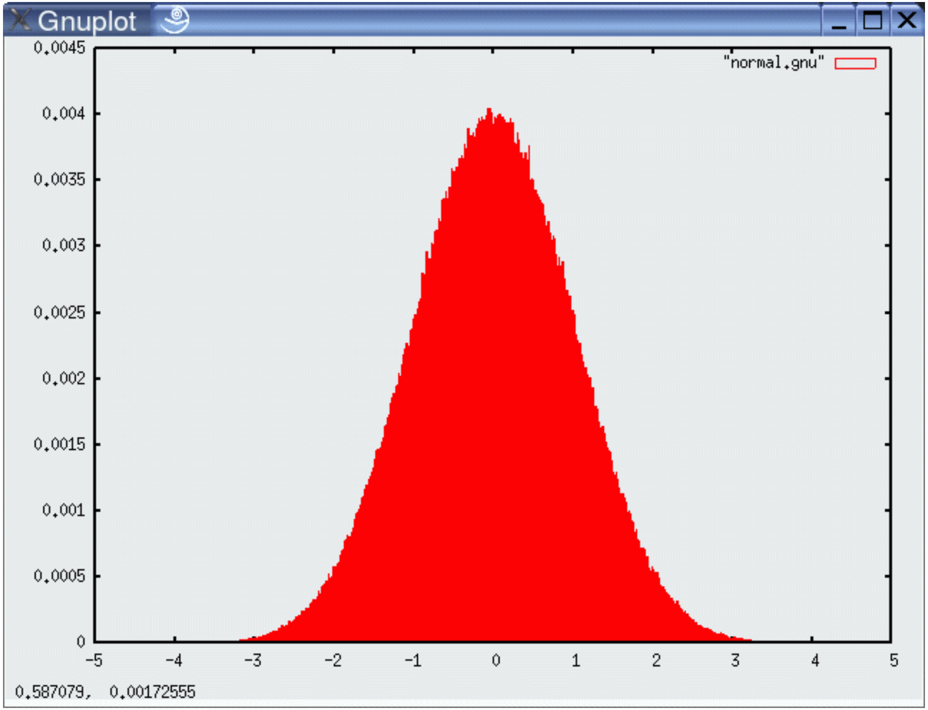
Distribución gaussiana

- Muestreando una distribución triangular

1. Algoritmo sample_triangular_distribution
2. devuelve $\frac{\sqrt{6}}{2}\left[\mathbf{rand}(-b,b)+\mathbf{rand}(-b,b)\right]$

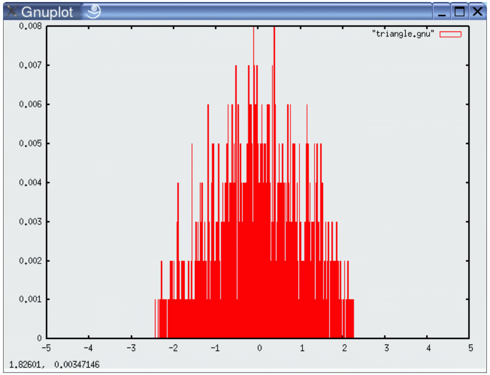
23

Muestras normalmente distribuidas

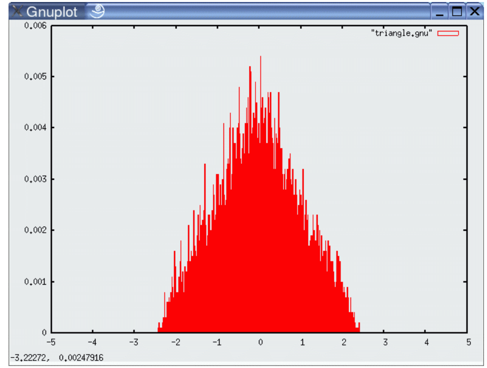


10^6 muestras

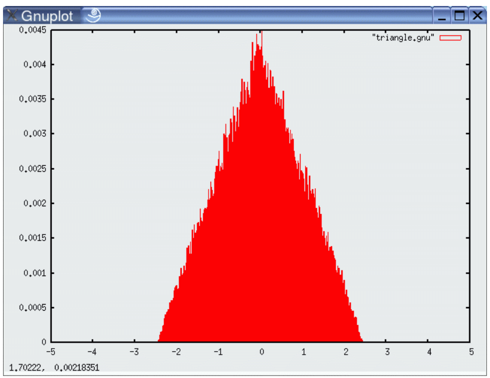
Distribución triangular (aproximación Gausiana)



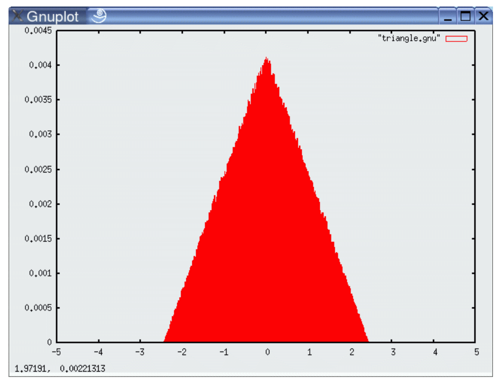
10^3 muestras



10^4 muestras



10^5 muestras



10^6 muestras

25

Ejemplo de Modelo de Movimiento de Odometría

1. Algoritmo sample_motion_model

$$u = \langle \delta_{rot1}, \delta_{rot2}, \delta_{trans} \rangle, x = \langle x, y, \theta \rangle$$

$$\hat{\delta}_{rot1} = \delta_{rot1} + \text{sample}(\alpha_1 \, |\delta_{rot1}| + \alpha_2 \, \delta_{trans})$$

$$\hat{\delta}_{trans} = \delta_{trans} + \text{sample}(\alpha_3 \, \delta_{trans} + \alpha_4 \, (|\delta_{rot1}| + |\delta_{rot2}|))$$

$$\hat{\delta}_{rot2} = \delta_{rot2} + \text{sample}(\alpha_1 \, |\delta_{rot2}| + \alpha_2 \, \delta_{trans})$$

sample_normal_distribution

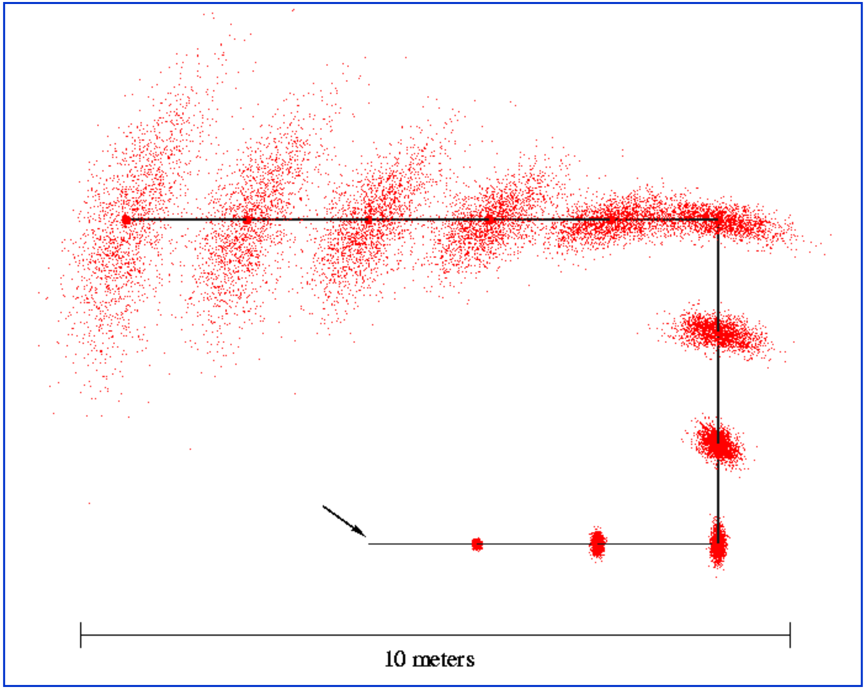
$$x' = x + \hat{\delta}_{trans} \cos(\theta + \hat{\delta}_{rot1})$$

$$y' = y + \hat{\delta}_{trans} \sin(\theta + \hat{\delta}_{rot1})$$

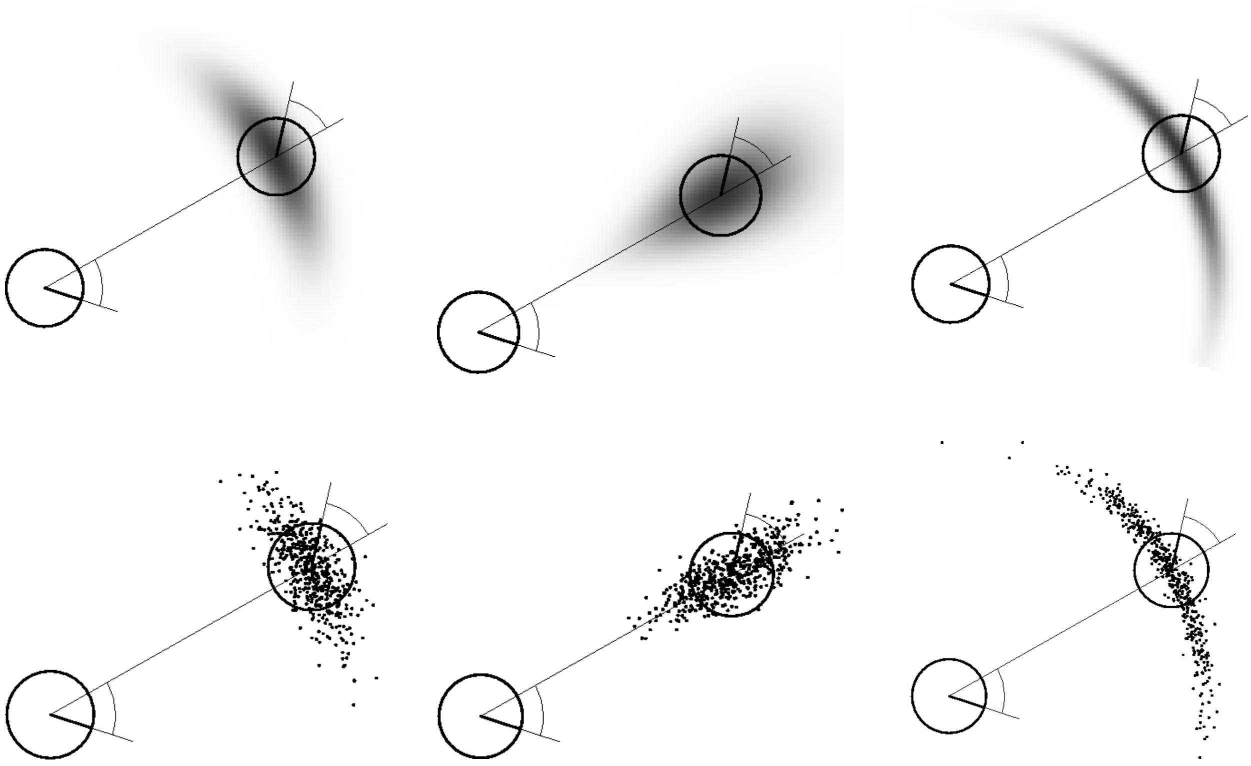
$$\theta' = \theta + \hat{\delta}_{rot1} + \hat{\delta}_{rot2}$$

8. devuelve $\langle x', y', \theta' \rangle$

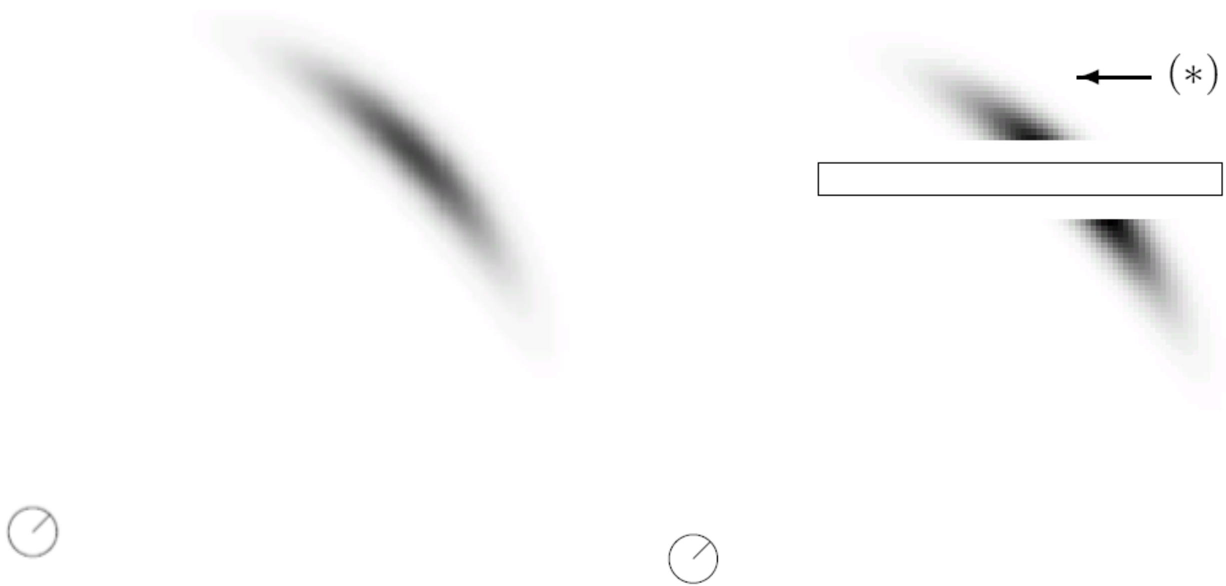
Muestreando el modelo de movimiento



Ejemplos (basados en odometría)



Modelo de movimiento consistente con mapa



$$p(x \mid u, x') \neq p(x \mid u, x', m)$$

Aproximación: $$p(x \mid u, x', m) = \eta \; p(x \mid m) \; p(x \mid u, x')$$

Índice

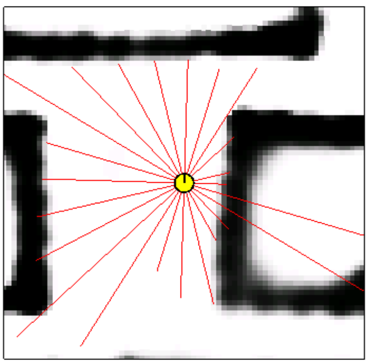
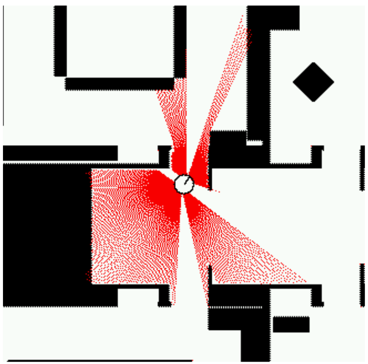
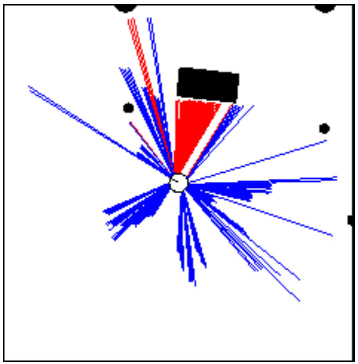
Incertidumbre en robótica móvil
Modelo de error movimiento
Modelo de error de los sensores

Sensores para robots móviles

- Sensores por contacto: Bumpers
- Sensores internos
  - Acelerómetros
  - Giroscopios
  - Brújula
- Sensores de proximidad
  - Sonar (time of flight)
  - Radar
  - Laser range-finders
  - Infrarrojos (intensidad)
- Sensores visuales: Cameras
- Sensores basados en satélites: GPS

Sensores de proximidad

- La tarea central es determinar *P(z|x)*, es decir, la probabilidad de una medición *z* dado que el robot está en la posición *x*
- Pregunta: de dónde vienen las probabilidades?
- Aproximación: tratar de explicar las medidas

Modelo sensor basado en haz de luz

- La lectura z consiste de K medidas.

$$z = \{z_1, z_2, \ldots, z_K\}$$

- Las mediciones individuales son independientes para la posición del robot x

$$P(z \mid x, m) = \prod_{k=1}^{K} P(z_k \mid x, m)$$

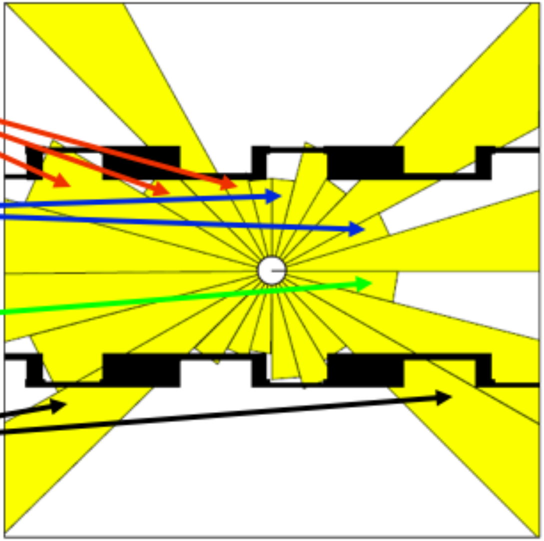
Errores típicos de las mediciones de rango

Haces de luz reflejados por obstáculos

Haces de luz reflejados por personas / interferencias

Lecturas aleatorias

Lecturas rango máximo

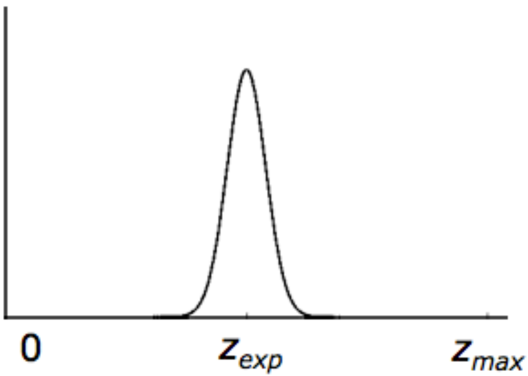


Lecturas de proximidad

- Las lecturas pueden ser causadas por …
  - obstáculos
  - interferencias
  - obstáculos dinámicos (personas, muebles, …).
  - Obstáculos no detectables (cristales, zonas que no reflejan laser/sonar…).
- El ruido es causado por la incertidumbre…
  - en la medición de la distancia al obstáculo conocido.
  - en posición de obstáculos adicionales (inesperados).
  - si se ha pasado por alto un obstáculo.

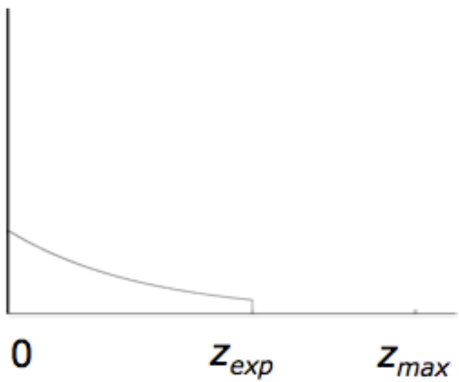
Modelo de proximidad basado en haz de luz

Ruido en la lectura



$$P_{hit}(z \mid x,m) = \eta \frac{1}{\sqrt{2\pi b}} e^{-\frac{1}{2}\frac{(z-z_{\text{exp}})^2}{b}}$$

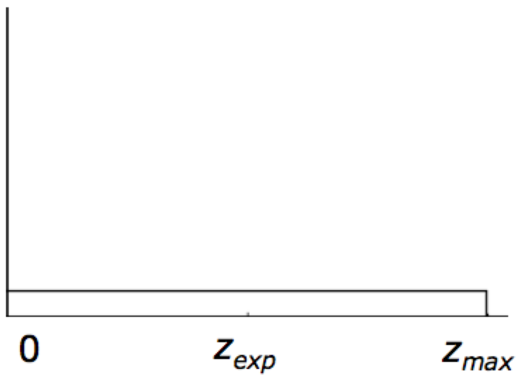
Obstáculos inesperados



$$P_{\text{unexp}}(z \mid x,m) = \begin{cases} \eta\ \lambda\ e^{-\lambda z} & z < z_{\text{exp}} \\ 0 & otherwise \end{cases}$$

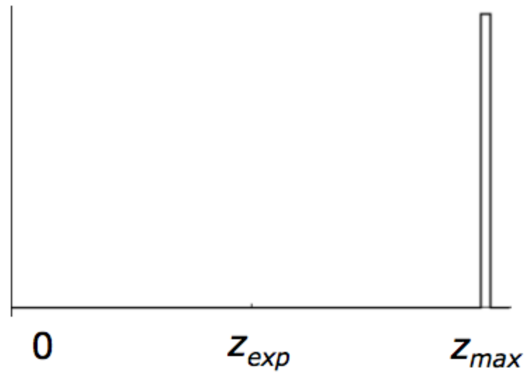
Modelo de proximidad basado en haz de luz

Lectura aleatoria



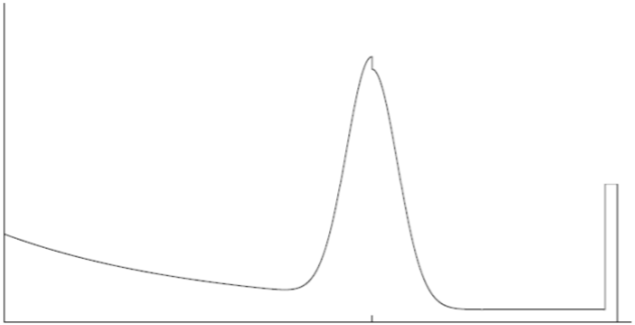
$$P_{rand}(z \mid x, m) = \eta \frac{1}{z_{\max}}$$

Lectura máxima



$$P_{max}(z|x,m) = \begin{cases} 1 \; si \; z = z_{max} \\ 0 \; en \; otro \; caso \end{cases}$$

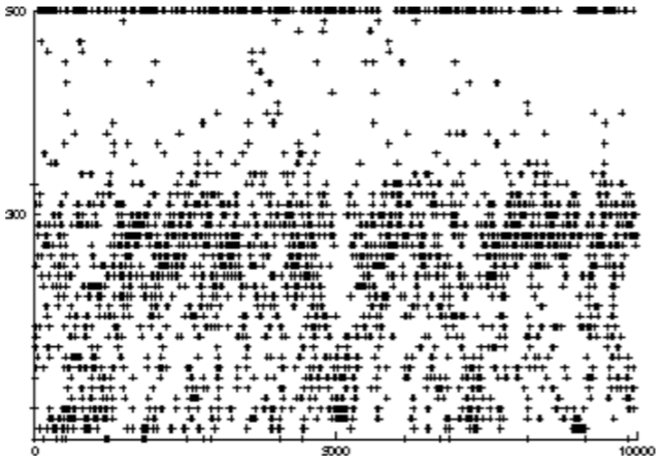
Modelo lecturas sensor (mixture)



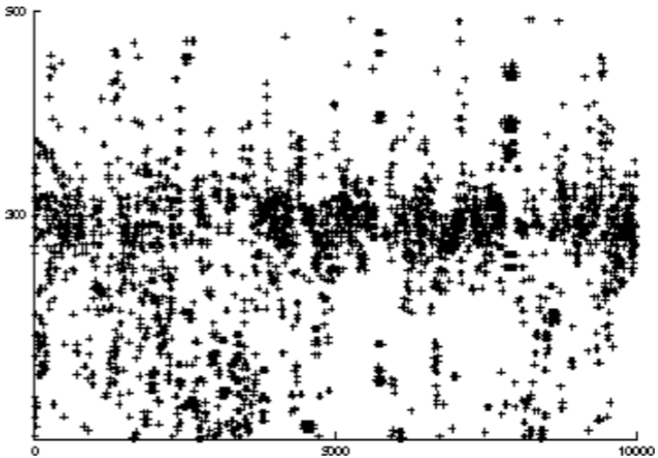
$$P(z \mid x, m) = \begin{pmatrix} \alpha_{\text{hit}} \\ \alpha_{\text{unexp}} \\ \alpha_{\text{max}} \\ \alpha_{\text{rand}} \end{pmatrix}^T \cdot \begin{pmatrix} P_{\text{hit}}(z \mid x, m) \\ P_{\text{unexp}}(z \mid x, m) \\ P_{\text{max}}(z \mid x, m) \\ P_{\text{rand}}(z \mid x, m) \end{pmatrix}$$

¿Cómo podemos determinar los parámetros del modelo?

Raw data (sensor)



Sónar



Láser

Lecturas obtenidas para un obstáculo situado a 300 cm.

eje X: número de lectura, eje Y: distancia de la lectura

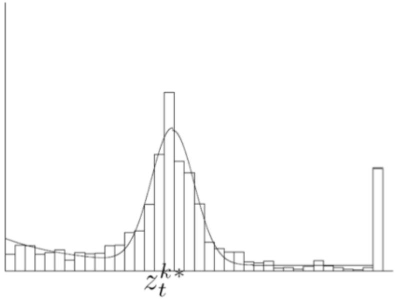
Método

- Maximizar la probabilidad de registro de los datos

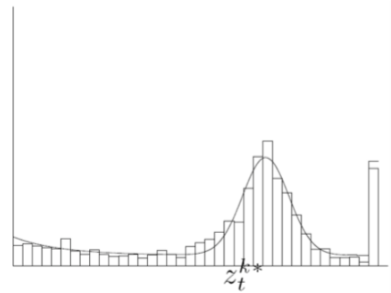
- Espacio de búsqueda de los parámetros n-1
  - Hill climbing
  - Gradient descent
  - Algoritmos genéticos
  - …

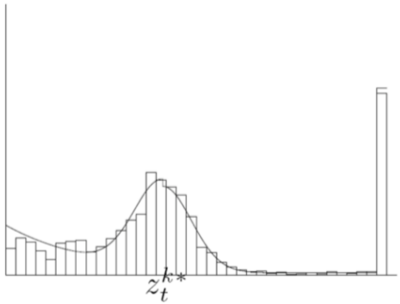
- Calcular el parámetro n-ésimo para satisfacer la restricción de normalización.

Resultados método optimización

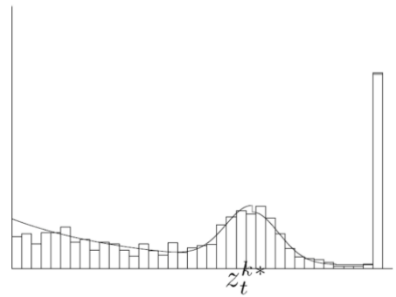


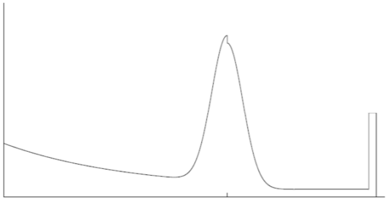
Láser



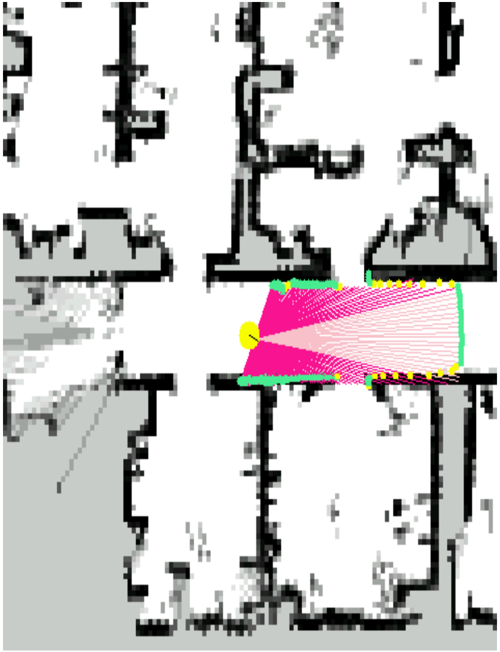


Sónar

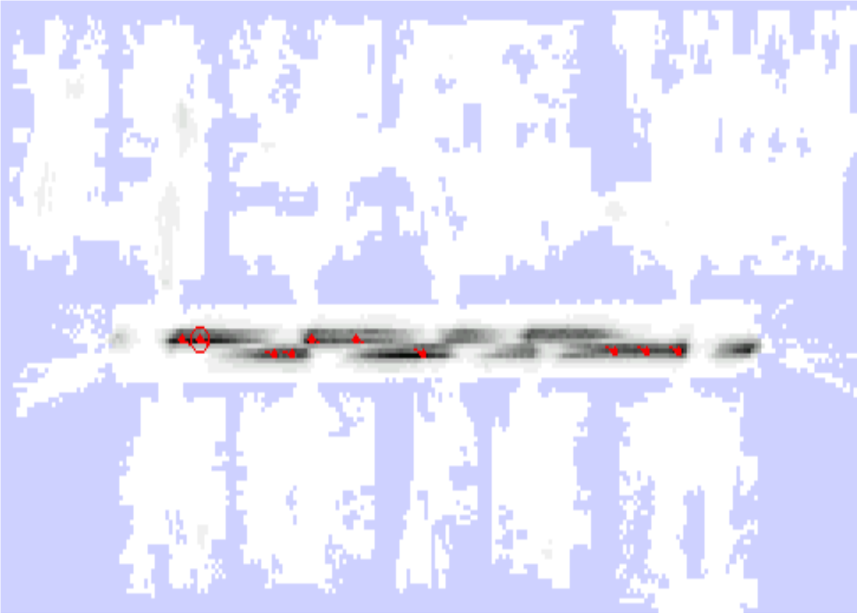




$$P(z \mid x, m) = \begin{pmatrix} \alpha_{\text{hit}} \\ \alpha_{\text{unexp}} \\ \alpha_{\text{max}} \\ \alpha_{\text{rand}} \end{pmatrix}^T \cdot \begin{pmatrix} P_{\text{hit}}(z \mid x, m) \\ P_{\text{unexp}}(z \mid x, m) \\ P_{\text{max}}(z \mid x, m) \\ P_{\text{rand}}(z \mid x, m) \end{pmatrix}$$



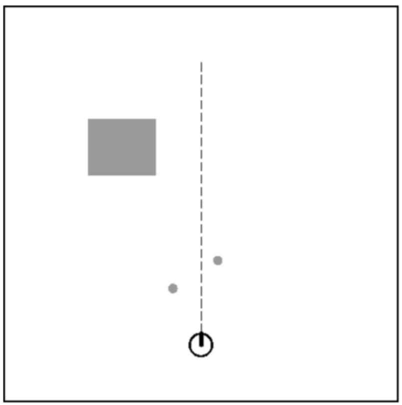
z



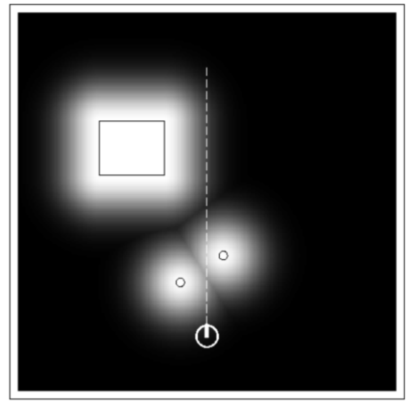
P(z|x,m)

z_{exp}

Modelo basado en lecturas del mapa
Scan-based model

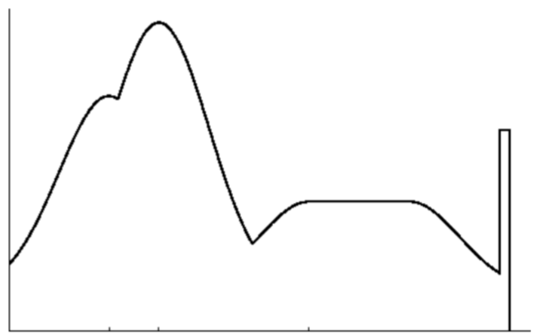


Map *m*



Likelihood field

$P(z|x,m)$



Landmarks

- Activos (radio, GPS)
- Pasivos (visuales)
- Triangulación

- El sensor nos proporciona
  - distancia
  - orientación
  - Distancia y orientación

Marcadores de distancia y orientación

Probabilidad de la distancia de un marcador

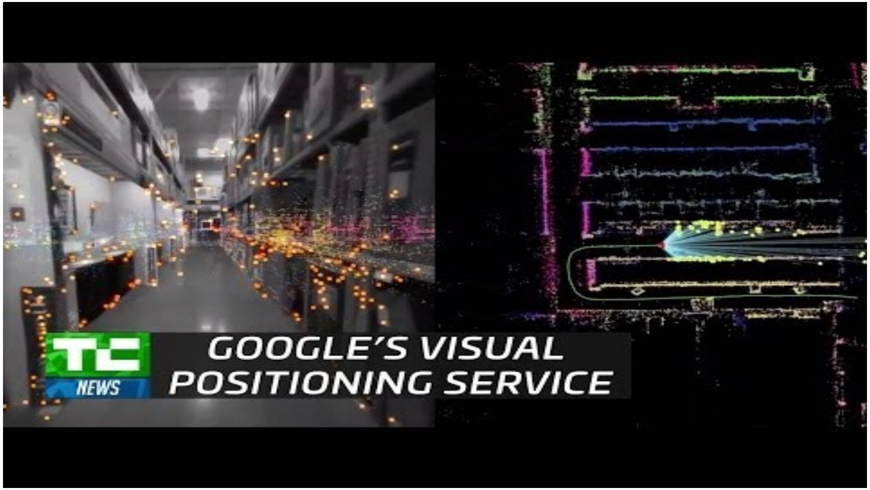
1. Algorithm **landmark_detection_model**(z,x,m):

   $z = \langle i, d, \alpha \rangle, x = \langle x, y, \theta \rangle$

2. $\hat{d} = \sqrt{(m_x(i) - x)^2 + (m_y(i) - y)^2}$
3. $\hat{a} = \text{atan2}(m_y(i) - y, m_x(i) - x) - \theta$
4. $p_{\text{det}} = \text{prob}(\hat{d} - d, \varepsilon_d) \times \text{prob}(\hat{\alpha} - \alpha, \varepsilon_\alpha)$
5. Return p_{det}

Problema: asociación de landmarks

- Asumir que los landmarks no pueden ser identificados de manera única
- Problema de asociación de datos: las características observadas deben coincidir con los puntos de referencia registrados anteriormente.
- Métodos basados en la probabilidad de las lecturas de las características dados los landmarks anteriores



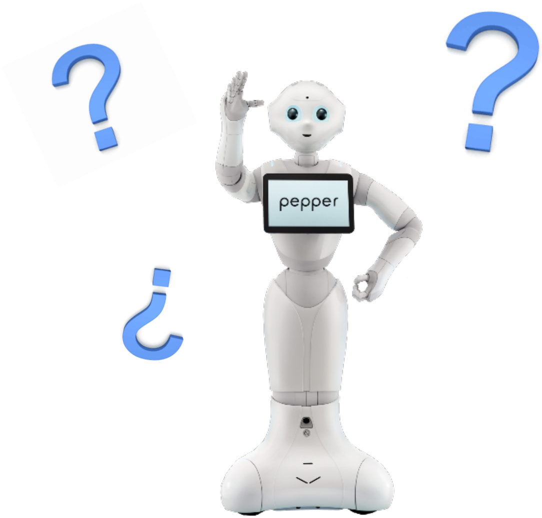
Bibliografía

Teoría

- Chapter 5. Introduction to Autonomous Mobile robots. Roland Siegwart and Illah R. Nourbakhsh. A Bradford Book.
- Chapter 5 & 6. Probabilistic Robotics. MIT Press. Thrun, Burgard, Fox.

Robots Móviles

Modelo de error del movimiento y de los sensores



Sergio Orts Escolano
Otto Colomina Pardo

sorts @ ua.es