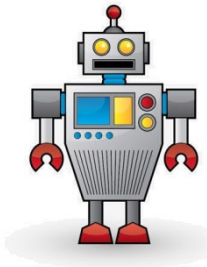
ECCI
Escuela de Ciencias de la Computación e Informática

Localización

CI-2657 Robótica

Prof. Kryscia Ramírez Benavides



Problemas de Navegación de los Robots

- ¿Dónde estoy?
  - Localización.
- ¿Dónde he estado?
  - Mapa de decisiones.
- ¿A dónde voy?
  - Planificación de misiones.
- ¿Cuál es la mejor manera de llegar?
  - Planificación de trayectoria.

El Problema de Localización

- La detección de la posición de un robot con respecto a su medio ambiente, utilizando la información sobre el entorno recogida por el robot.
- El robot utiliza normalmente sus sensores para recoger información sobre el medio ambiente.

Lecturas de los Sensores

- Tipos de lecturas de los sensores utilizados en la localización:
  - Visión.
  - Lecturas del sonar de distancia.
  - Lecturas del escáner láser de distancia.
  - Las señales GPS.
  - Lecturas de brújula electrónica.
  - Potencia de la señal WLAN.
- Posar en dos dimensiones al robot:
  - Coordenadas x,y.
  - Partida.

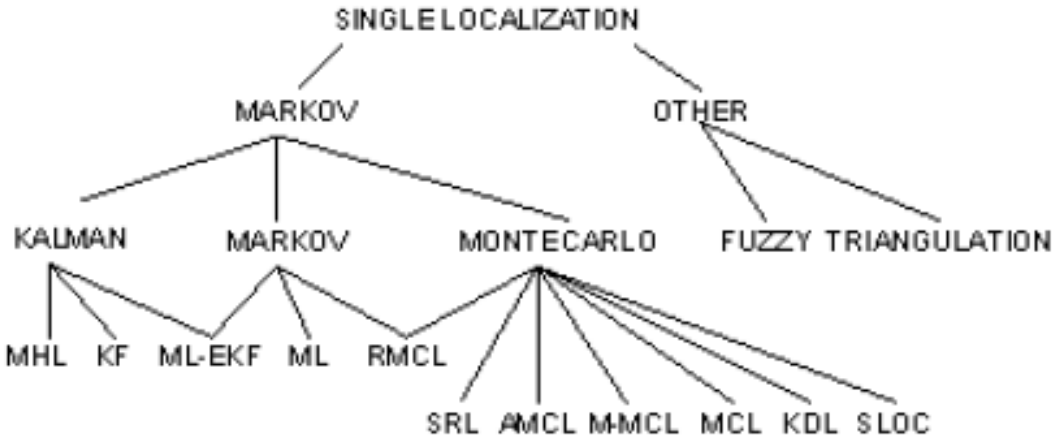
Tipos de Problemas de Localización

- **Posición (pose) seguimiento:** Hacer un seguimiento de la pose del robot utilizando odometría, suponiendo que la posición inicial del robot es conocida.
- **Localización global:** Encontrar su ubicación en el medio ambiente sin el uso de la información a priori.
- **Secuestro:** El robot fue tomado de su ubicación actual y fue llevado a otro lugar sin que el robot fuera consciente. El robot debe darse cuenta de esto, y relocalizarse a sí mismo en su nueva posición. Este es posiblemente el problema más difícil, entre otros.

Localización Actica vs. Pasiva

- **Localización activa:** El robot busca activamente en su entorno para encontrar puntos de referencia y así localizarse mejor, y disminuir su error de localización.
- **Localización pasiva:** El robot está ocupado con su tarea, y ejecuta la localización como una tarea de fondo, con las observaciones que recibe durante su implementación de la tarea principal.

Clasificación de Algoritmos de Localización



[Köse 2006]

Odometría

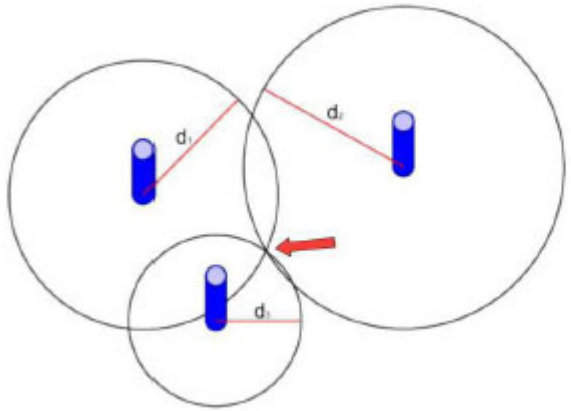
- Es utilizada por los robots móviles para estimar su posición relativa de acuerdo a una posición de partida.
- Utiliza datos de la rotación de las ruedas o las piernas, para estimar el cambio en la posición con el tiempo.
- A menudo son muy sensibles a error.
- Se requieren en la mayoría de los casos por odometría la recolección de datos rápida y precisa, la calibración del equipo, y el procesamiento; para que sea utilizada con eficacia.

Triangulación

- El más simple método de localización
- Usa las propiedades geométricas de triángulos sobre la base de las distancias y/o mediciones de ángulo entre el robot y los puntos de referencia observados.

Triangulación utilizando mediciones de distancia y ángulo

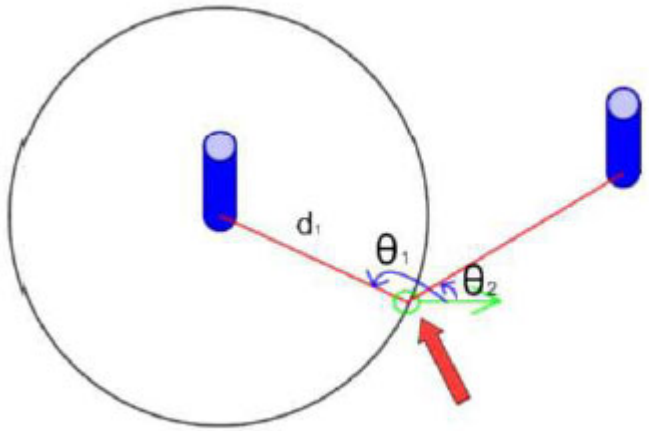
- La posición del robot se calcula utilizando la distancia medida entre el robot y las posiciones de referencia múltiples.
- Para la estimación de la posición en dos dimensiones son necesarias las medidas de distancia a partir de tres puntos no colineales:



[Çelik 2005]

Triangulación utilizando mediciones de distancia y ángulo (cont.)

- También se puede utilizar dos medidas de ángulo y una medida de distancia:



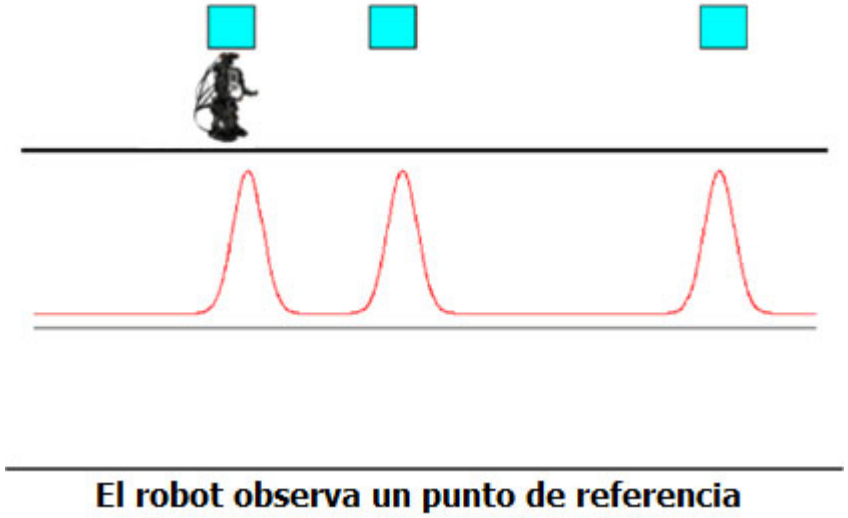
[Çelik 2005]

Localización Probabilística

- La ubicación del robot se representa como una densidad de probabilidad sobre las posibles localizaciones de robot o posturas.
- Estado Creencia:
  - Representa la creencia del robot en su posición actual.
  - En él se resume toda la información que el robot tiene sobre su posición actual.
  - Se actualiza para los movimientos del robot y para la información de los sensores.

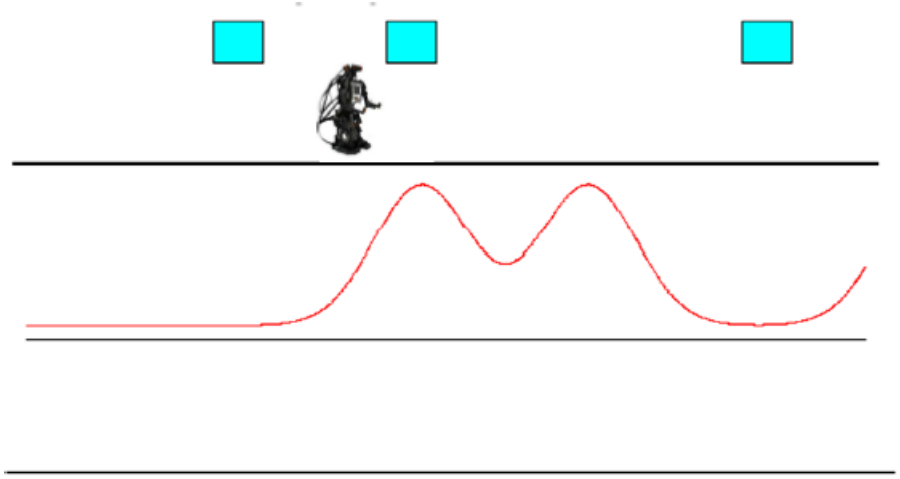
Localización Probabilística (cont.)

Inicialmente, el robot no sabe dónde está.

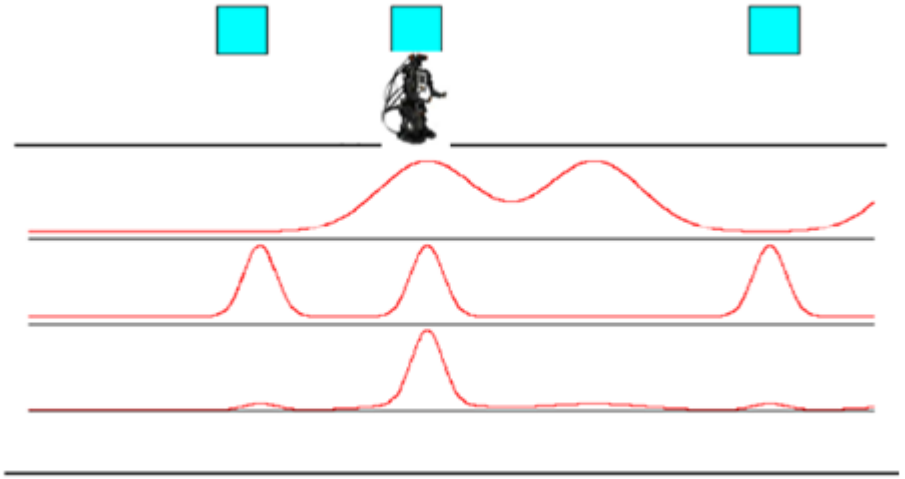


El robot observa un punto de referencia

Localización Probabilística (cont.)



El robot se mueve.



El robot observa otro punto de referencia

Localización de Markov

- No hace ninguna hipótesis de distribución y permite utilizar cualquier tipo de distribución.
- Se utilizan mediciones de odometría y mediciones perceptivas para calcular la posición actual del robot.
- El robot mantiene una creencia sobre su posición, que se denota como $Bel^{(t)}(L)$ en el tiempo t.
- El método ML utiliza un filtro de histograma para la creencia posterior.

Localización de Markov (cont.)

- Observaciones generales:
  - Pesado.
  - Muy robusto.
  - No supone una distribución, así es más flexible, más costoso.
  - Alta complejidad computacional.
  - Recuperación rápida de secuestro.

Localización de Monte Carlo

- Localización de Monte Carlo (MCL) es una versión de ML, que se basa en una muestra basada en la representación y el muestreo / importancia del algoritmo de re-muestreo para la propagación de creencias.
- Las creencias se representan por un conjunto de K muestras que se pesaron (partículas) que son de tipo $((x,y,\theta), w)$, donde $w \geq 0$ es un factor de ponderación numérica no negativo, tal que la suma de todos los w es uno.
- Los factores de ponderación se denominan pesos de importancia.
- Actualizaciones odométricas y sensoriales son similares a ML.
  - Se llevan a cabo dentro de la predicción y pasos de corrección.

Localización de Monte Carlo (cont.)

- Observaciones generales:
  - Se produce un error en caso de secuestro.
  - La precisión y robustez dependen del número de muestras.
  - Alto número de muestras -> menos sesgo.
  - Menos complejidad computacional que ML.
  - Menor consumo de memoria que ML -> depende del número de muestras.
  - No tan rápido como ML en la convergencia.

Algoritmo de MCL

procedure *MCL(S, a, o)*

1: **for *i = 0 to n do***

2: draw by replacement random sample *l from S according to* $w_1..w_n$ *(resampling)*

3: draw sample *l* $p(l|a, l)$*(sampling)*

4: calculate $w=p(l|o)$ *(importance sampling)*

5: add *(l, w) to S*

6: **end for**

7: normalize the importance factors *w in S*

8: return *S*

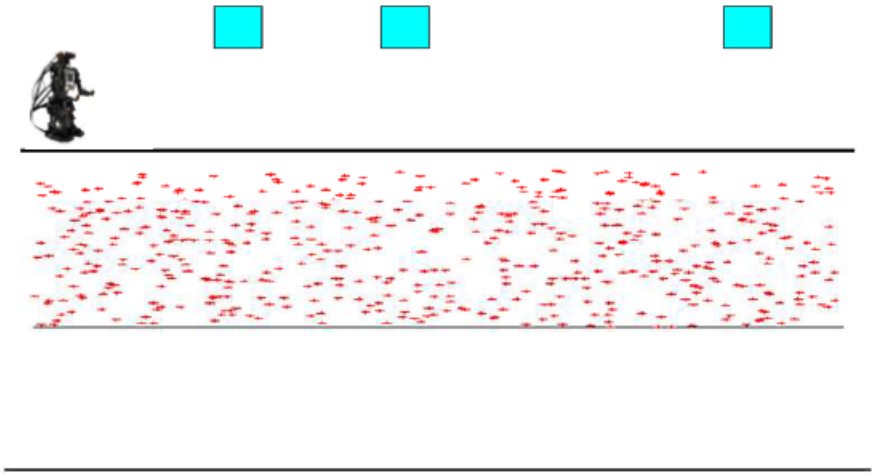
Algoritmo de MCL (cont.)

- MCL representa la distribución de probabilidad de ubicación por un conjunto de muestras.
- Las muestras se encuentra de tal manera que, el tamaño de la población de muestras es proporcional a la probabilidad de que el robot está en esa ubicación y el entorno.
- Después de varios pasos, las muestras convergen en algún lugar del medio ambiente.
  - La posición del robot se obtiene tomando la media de las posiciones de las muestras, y la desviación estándar da la incertidumbre del robot.

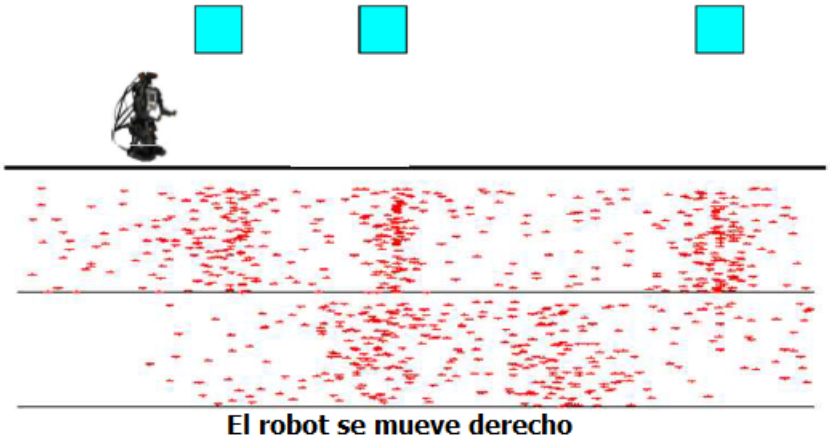
MCL y Secuestro

- El algoritmo original sufre en caso de secuestro.
  - Varias extensiones MCL se han propuesto para superar este problema.
- Estas extensiones añaden nuevas muestras para el conjunto de muestra de diferentes partes del campo, y difieren en el número de muestras nuevas y cuando añadir estas muestras.
- Algunos de estos métodos son:
  - Restablecimiento del sensor de localización (SRL)
  - Mezcla MCL (Mix-MCL)
  - Adaptativo MCL (A-MCL)
  - Distancia Kullback-Leibler (KLD)-Muestreo

Algoritmo de MCL (cont.)

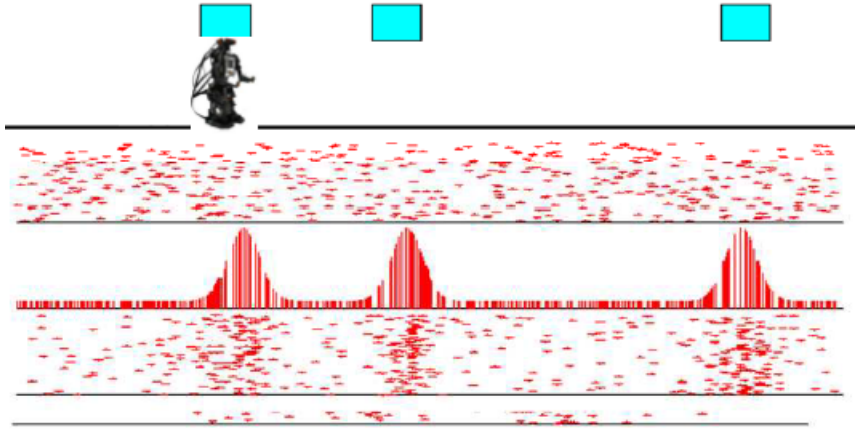


El robot está inicialmente perdido

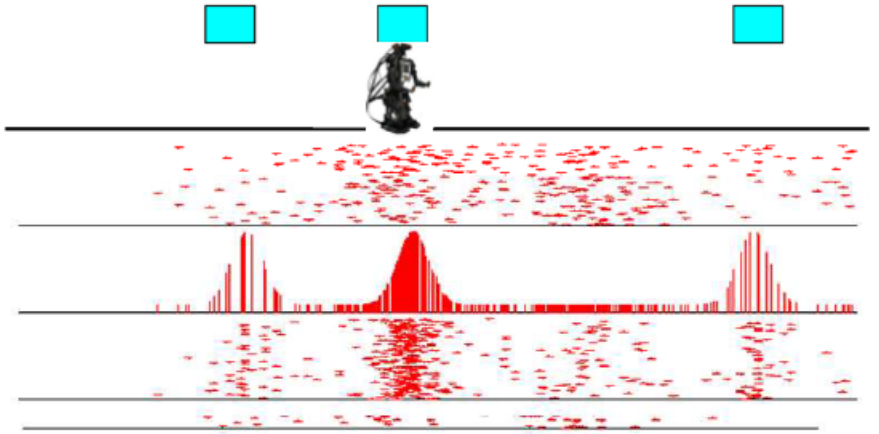


El robot se mueve derecho

Algoritmo de MCL (cont.)



El robot observa un punto de referencia



El robot se sigue moviendo derecho, observa otro punto de referencia

Algoritmo de Filtro de Kalman

- Implementa el cálculo de creencia en los estados continuos.
- Es similar a la ML.
- Este filtro integra incertidumbre en los cálculos, haciendo la suposición de distribuciones gaussianas para representar todas las densidades, incluidas las posiciones odométricas y mediciones sensoriales.
- Las estimaciones de posición son actualizadas por odometría y sintiendo alternativamente con la propiedad de que las distribuciones gaussianas se pueden combinar usando multiplicación.

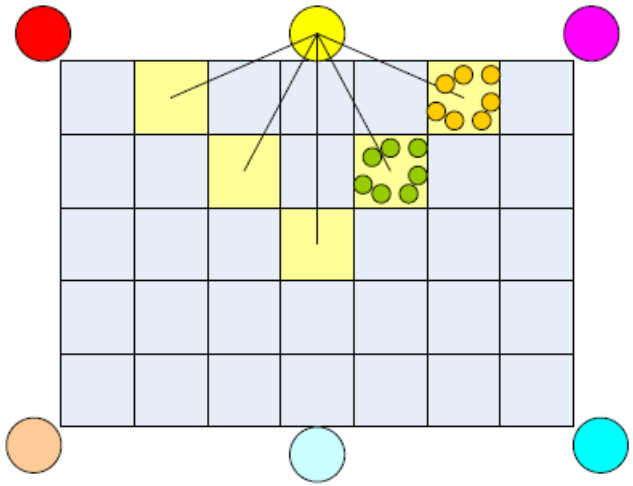
Algoritmo R-MCL (MCL Reverso)

- ML es un método basado en una red robusta.
  - Debido a los grandes tamaños de celda de cuadrícula que no son muy precisos.
- MCL no es tan robusto y converge más lentamente que ML.
  - Demasiadas muestras se utilizan para obtener resultados precisos caros.
- ¿Por qué no combinar estos dos métodos?

Algoritmo R-MCL (cont.)

Algorithm 1 The R-MCL Algorithm

```
1: if bool_ML==TRUE then
2:    ML_update
3:    if ML_number_of_grid_cells_in_max_grid_array < Th_ML then
4:       MCL_init(ML_samples)
5:       bool_ML=FALSE
6:    end if
7: else
8:    MCL_update
9:    MCL_init(ML_samples)
10:   if MCL_lost()==TRUE then
11:      ML_reset()
12:      bool_ML=TRUE
13:   end if
14: end if
```



[Köse 2006]

Referencias Bibliográficas

- Fu, K.S.; González, R.C. y Lee, C.S.G. Robotics: Control, Sensing, Vision, and Intelligence. McGraw-Hill. 1987.


¿Preguntas?

¡Gracias!

Dra. Kryscia Daviana Ramírez Benavides
Profesora e Investigadora
Universidad de Costa Rica
Escuela de Ciencias de la Computación e Informática

Sitio Web: http://www.kramirez.net/
E-Mail: kryscia.ramirez@ucr.ac.cr
kryscia.ramirez@ecci.ucr.ac.cr

Redes Sociales: