

Arquitecturas Basadas en el Conocimiento

Introducción a la Robótica Inteligente



Álvaro Gutiérrez
21 de febrero de 2025

a.gutierrez@upm.es
www.robolabo.etsit.upm.es

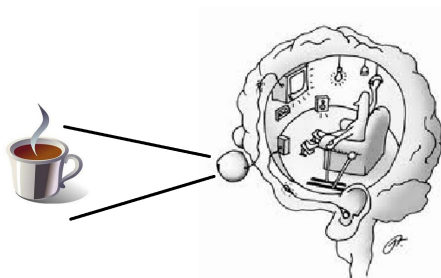
- ❶ Introducción
- ❷ Mapas
- ❸ Experimentos

1 Introducción

2 Mapas

3 Experimentos

- Existe controversia sobre el uso de representaciones y conocimientos en la inteligencia, y por consiguiente en la robótica



¿Qué es el Conocimiento?



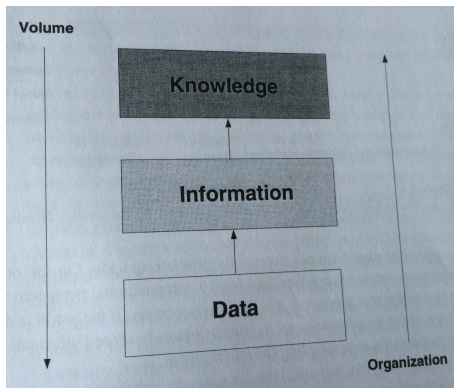
- Una definición complicada

- ▶ Una definición complicada
- ▶ **RAE:** Acción y efecto de conocer → Averiguar por el ejercicio de las facultades intelectuales la naturaleza, cualidades y relaciones de las cosas.
- ▶ **Wikipedia:** Hechos o información adquiridos por una persona a través de la experiencia o la educación, la comprensión teórica o práctica de un asunto referente a la realidad.

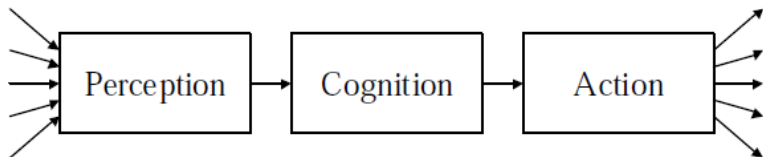
¿Qué es el Conocimiento?

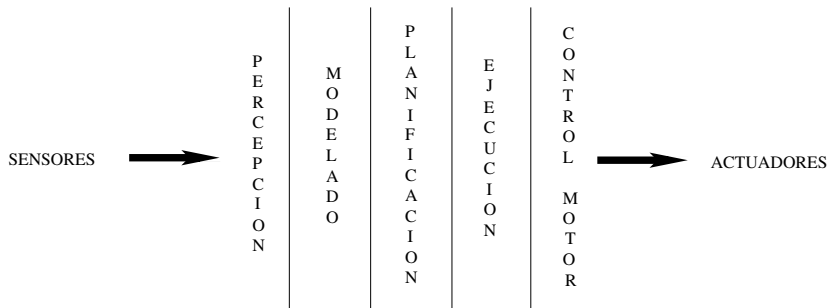


- **Tanimoto, 1990:** Información en contexto, organizada de tal manera que puede ser rápidamente aplicada a solución de problemas, percepción y aprendizaje.



- ▶ **Steels, 1995:** La representación del conocimiento debe incluir “relaciones físicas con correlaciones con aspectos del entorno y tener capacidad predictiva para el sistema”
 - ▶ **Correlación con el entorno:** Las representaciones tienen que estar relacionadas con el entorno.
 - ▶ Duración: Corto o largo plazo
 - ▶ Espacio: Mapas métricos, relacionales,...
 - ▶ **Capacidad predictiva:** Debe ser la capacidad central del conocimiento
 - Si no hay que predecir → arquitectura reactiva
 - Si hay que predecir → cuidado con los modelos!!





1 Introducción

2 Mapas

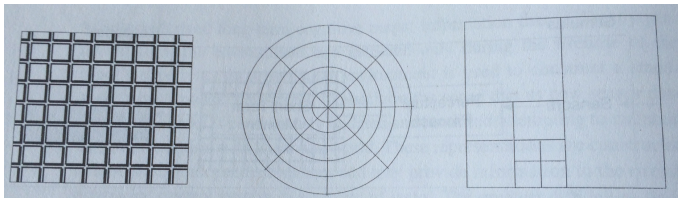
3 Experimentos

- ▶ El ejemplo por excelencia (35 años)

- ▶ El ejemplo por excelencia (35 años)
- ▶ **SLAM**: Simultaneous Localization and Mapping
- ▶ Investiga los problemas que plantea la construcción de modelos matemáticos, geométricos o lógicos de entornos físicos, empleando como herramienta un robot móvil
 - ▶ Ruido en los sensores
 - ▶ Errores en el movimiento
 - ▶ Aproximaciones de modelos
 - ▶ Dificultad de representación
- ▶ Formulación bayesiana, filtros de Kalman, algoritmos de optimización de caminos, mapas de ocupación, ...

- ▶ **Métrico:** El robot representa un mapa métrico, donde se almacenan distancias y longitudes de todos los elementos del mapa (muro de 30 cm, curvatura de 30° , ..)
- ▶ **Odométrico** El robot representa los desplazamientos de las ruedas realizados durante su movimiento (avanzar las 2 ruedas 1.5 vueltas, rueda izq. parada y derecha 0.3 vueltas, ...)
- ▶ **Basado en Marcas:** El robot representa la secuencia de movimientos basado en marcas del entorno (giro a la derecha en el primer cruce, recto hasta la tercera rotonda,...)
- ▶ **Topológico:** El robot representa una acción ligada a una marca (en el cruce verde a la derecha, en el rojo recto, en el azul a la izquierda,...)

- ▶ **Rejillas**
- ▶ **Quadtree**
- ▶ **Árboles**
- ▶ **Grafos**



- ▶ **Tiempo real**

- ▶ **Búsqueda aleatoria:** El robot avanza, y si encuentra un obstáculo intenta rodearlo con movimientos aleatorios
- ▶ **Trazado de obstáculos:** El robot avanza, y si encuentra un obstáculo almacena su posición e intenta rodearlo

- ▶ **Planificación**

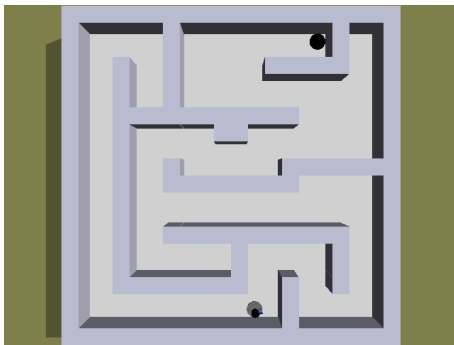
- ▶ **Búsqueda en anchura** (Breadth First Search): Se explora el grafo por niveles, si en un nivel se ha llegado a la solución se acaba la búsqueda.
- ▶ **Búsqueda en profundidad** (Depth First Search): Se explora el grafo por ramas hasta que se recorren todas las ramas
- ▶ **Búsqueda del más prometedor** (Best First Search): Es una búsqueda en anchura pero explorando el vecino más prometedor primero.
- ▶ **Dijkstra** (Caminos mínimos): Calcula el coste asociado de cada origen a cada destino
- ▶ **A*:** Modificación de Dijkstra pero con el uso de heurísticos para ver qué vecino es más prometedor.

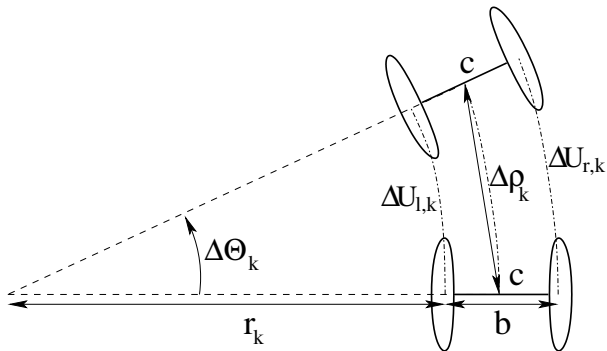
1 Introducción

2 Mapas

3 Experimentos

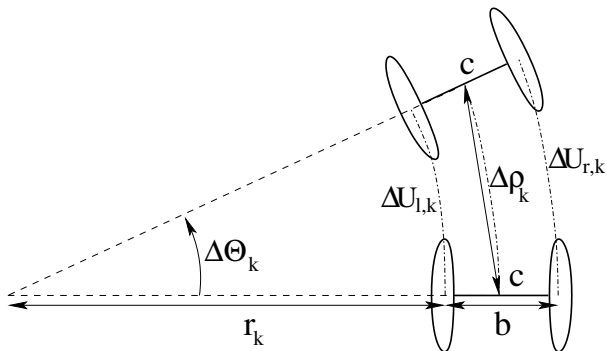
- ▶ 5 experimentos
- ▶ Mapa en rejilla rectangular
- ▶ Proporcionamos el mapa al robot
- ▶ Objetivo: Avanzar hasta llegar a la meta





$$\Delta\theta_k = (\Delta U_{r,k} - \Delta U_{l,k})/b$$

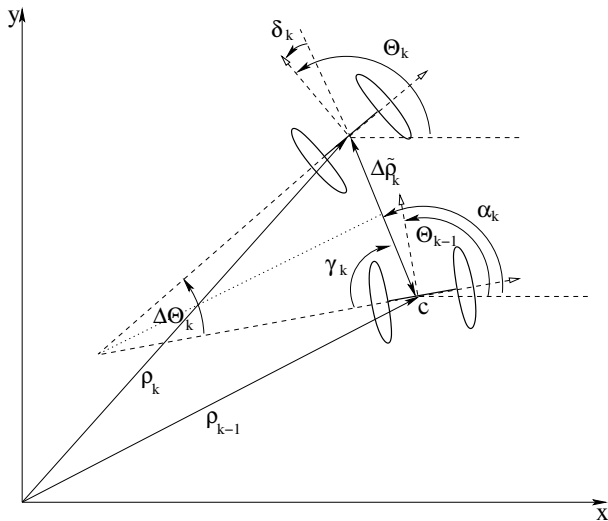
$$\Delta\rho_k = (\Delta U_{r,k} + \Delta U_{l,k})/2$$



$$\Delta\theta_k = (\Delta U_{r,k} - \Delta U_{l,k})/b$$

$$\Delta\rho_k = (\Delta U_{r,k} + \Delta U_{l,k})/2$$

$$\Delta\tilde{\rho}_k = (r_k + b/2) \cdot 2 \cdot \sin(\Delta\theta_k/2)$$



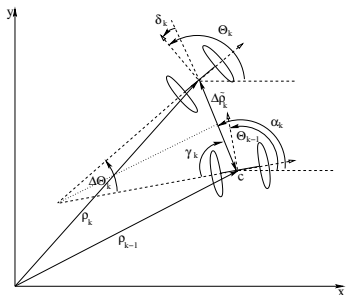
$$\mathbf{x}_{k-1} = \begin{bmatrix} x_{k-1} & y_{k-1} & \theta_{k-1} \end{bmatrix}^T \quad (1)$$

$$\mathbf{x}_k = \begin{bmatrix} x_{k-1} + \Delta \tilde{\rho}_k \cos(\alpha_k) \\ y_{k-1} + \Delta \tilde{\rho}_k \sin(\alpha_k) \\ \alpha_k + \delta_k \end{bmatrix} \quad (2)$$

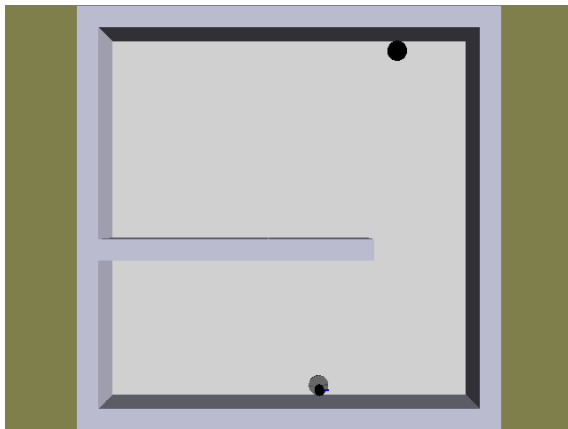
$$\delta_k = \Delta \theta_k / 2 \quad (3)$$

$$\alpha_k = \theta_{k-1} + \Delta \theta_k / 2 \quad (4)$$

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \begin{bmatrix} \Delta \tilde{\rho}_k \cos(\theta_{k-1} + \Delta \theta_k / 2) \\ \Delta \tilde{\rho}_k \sin(\theta_{k-1} + \Delta \theta_k / 2) \\ \Delta \theta_k \end{bmatrix} \quad (5)$$

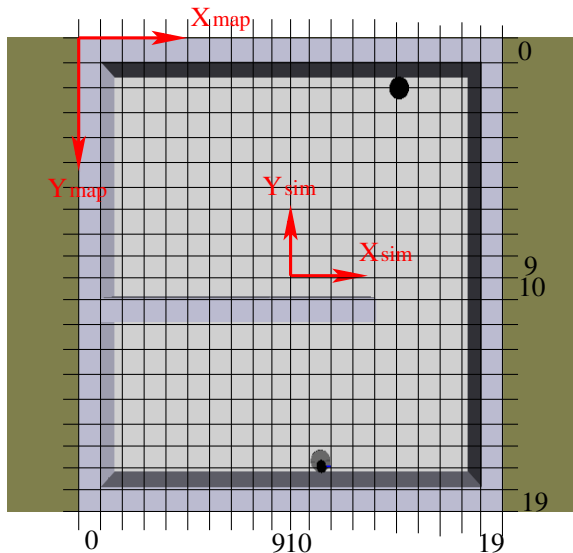


Experimento 1



- ▶ 20 x 20 celdas
- ▶ $3.0 \times 3.0 \text{ m}^2$
- ▶ 0.15m/celda

Experimento 1



- ▶ 20 x 20 celdas
- ▶ $3.0 \times 3.0 \text{ m}^2$
- ▶ 0.15m/celda

- ▶ Start:
Celda (11,18)
- ▶ End:
Celda (15,1)
- ▶ Obstáculo:
(0-13,11)



- ▶ 20 x 20 celdas
- ▶ 3.0x3.0 m²
- ▶ 0.15m/celda
- ▶ S0: Giro +90°
- ▶ S1: Avance 6*0.15 m
- ▶ S2: Giro -90°
- ▶ S3: Avance 4*0.15 m
- ▶ S4: Giro +90°
- ▶ S5: Avance 11*0.15 m
- ▶ S6: Stop

Limpiemos el código!!

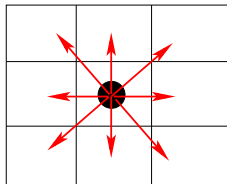
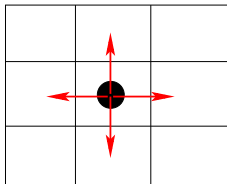
- ▶ CalcPositionAndOrientation(Ul, Ur)
- ▶ Movimiento:
 - ▶ TurnLeft (speed)
 - ▶ TurnRight (speed)
 - ▶ GoForwards (speed)
 - ▶ GoBackards (speed)
 - ▶ Stop
- ▶ GoGoal (x, y);

Planifiquemos la ruta

- ▶ PathPlanning()
 - ▶ Representamos los destinos
 - ▶ `m_nPathPlanningStops`: Número de destinos intermedios
 - ▶ `m_vPositionsPlanning = new dVector2[m_nPathPlanningStops]`: Vectores de coordenadas

Resolvamos el mapa

- ▶ Importar el mapa
- ▶ Proporcionar coordenadas
- ▶ Definir movimientos

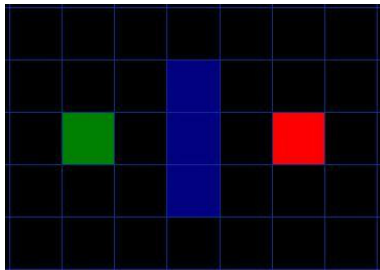


A*

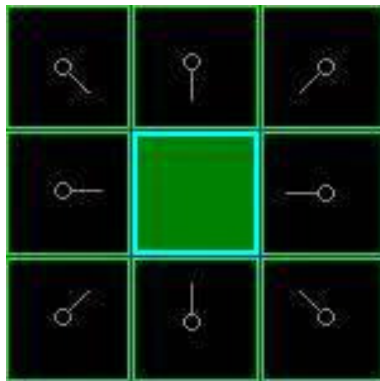
(<http://homepages.abdn.ac.uk/f.guerin/pages/teaching/CS1013/practicals/aStarTutorial.htm>)

- ▶ Nodo Inicio: El comienzo del grafo
- ▶ Nodo final: El destino
- ▶ Nodo padre: El nodo desde el que se ha llegado
- ▶ Lista abierta: Nodos por explorar
- ▶ Lista cerrada: Nodos explorados
- ▶ Distancia al origen
- ▶ Distancia al destino (estimación)

A*



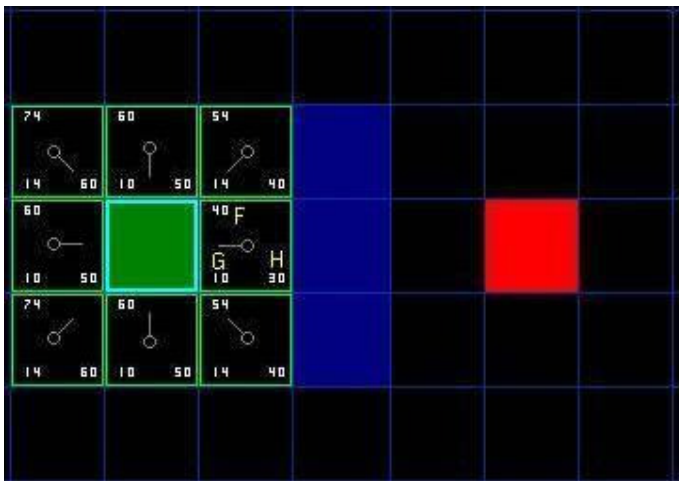
A*



Experimento 4



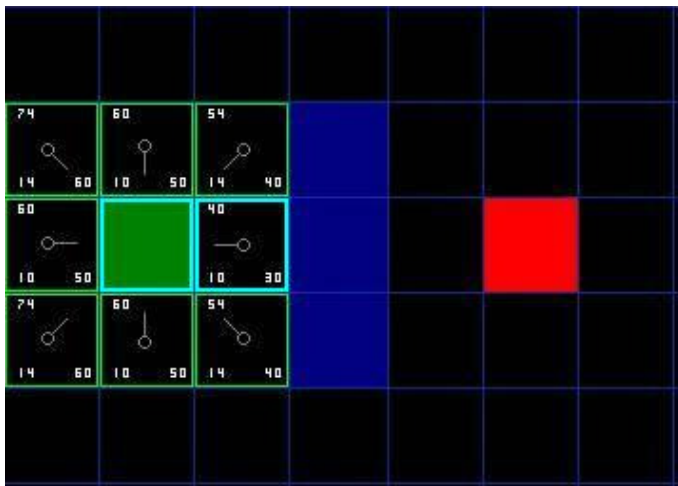
A*



Esperimento 4



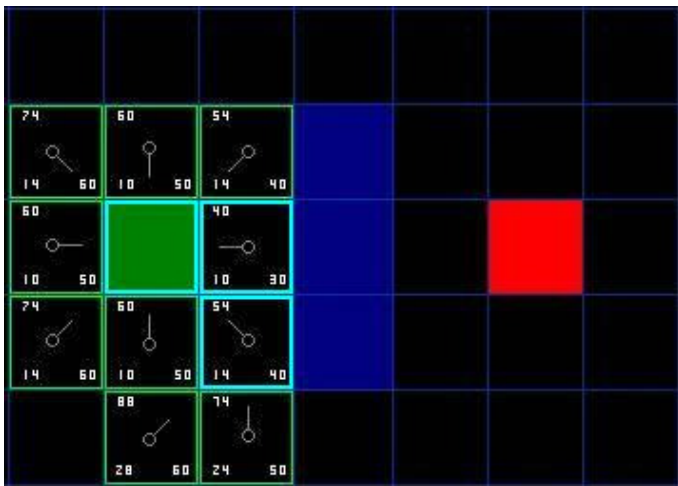
A*



Experimento 4



A*

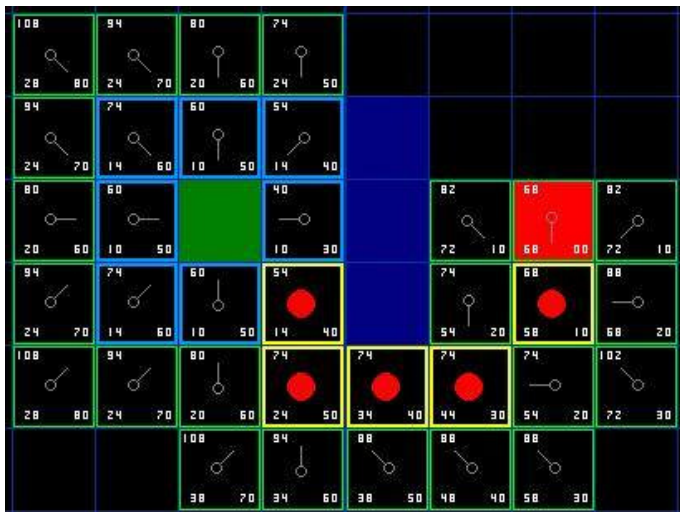


108 28 80	94 24 70	80 20 60	74 24 50				
94 24 70	74 14 60	60 10 50	54 14 40				
80 20 60	60 10 50		40 10 30		82 72 10	68 68 00	82 72 10
94 24 70	74 14 60	60 10 50	54 14 40		74 54 20	68 58 10	88 68 20
108 28 80	94 24 70	80 20 60	74 24 50	74 34 40	74 44 30	74 54 20	102 72 30
		108 38 70	94 34 60	88 38 50	88 48 40	88 58 30	

Experimento 4



A*



- ▶ Una vez obtenido el camino:
 - ▶ Calculamos los destinos intermedios
 - ▶ Los introducimos en el planificador

Lo vemos en el código!!

- ▶ Problemas con la odometría
- ▶ Problemas con los modelos (muros)

Introducamos sensores!!

- ▶ Movimientos en arco: `TurnAngle`
- ▶ $\theta_{goal} = (\theta_{goal} + \theta_{repellent})/2$
 - ▶ θ_{goal} : Dirección planificada de movimiento
 - ▶ $\theta_{repellent}$: Dirección de esquivación de obstáculos

GRACIAS!!

GRACIAS!!