



Arquitecturas Híbridas

Introducción a la Robótica Inteligente



Álvaro Gutiérrez
19 de marzo de 2025

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

- 1 Introducción
- 2 Arquitecturas Híbridas
- 3 Ejemplo

1 Introducción

2 Arquitecturas Híbridas

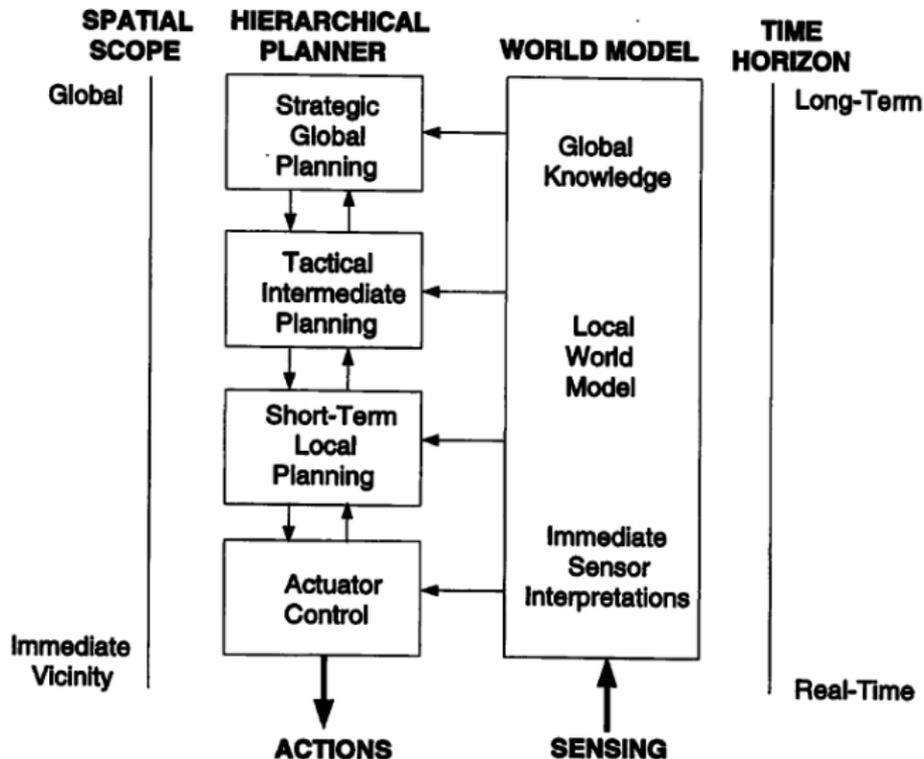
3 Ejemplo

- ▶ Arq. Basadas en el Conocimiento
 - ▶ Entornos estáticos
 - ▶ Modelos precisos
 - ▶ Sensores perfectos
 - ▶ Necesidad de representación simbólica
- ▶ Arq. Basadas en el Comportamiento
 - ▶ Entorno cambiante y dinámico
 - ▶ Ruido en los sensores
 - ▶ Dificultad de localización
 - ▶ No necesidad de representación simbólica
- ▶ Arq. Híbridas
 - ▶ **Planifiquemos pero sin perder autonomía, robustez y flexibilidad**

1 Introducción

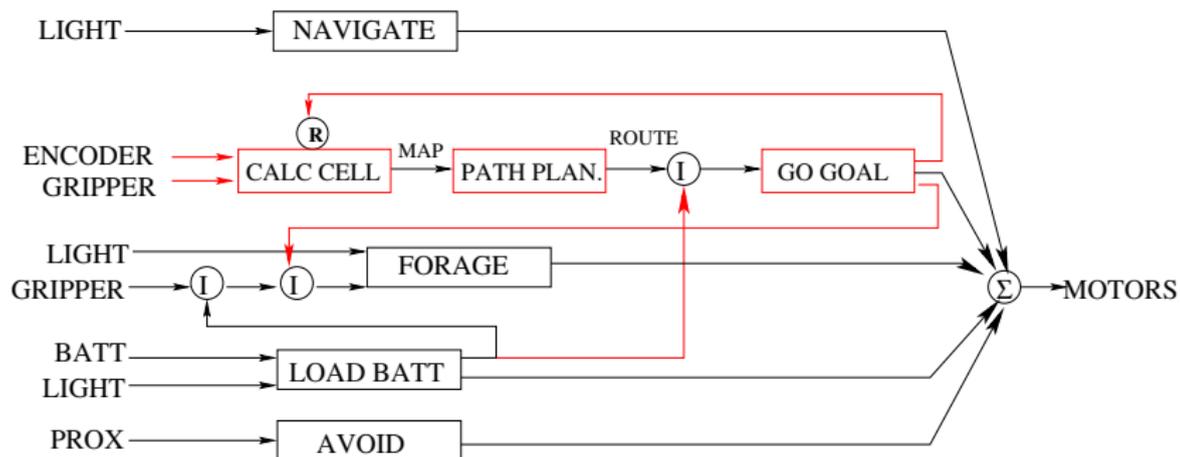
2 Arquitecturas Híbridas

3 Ejemplo

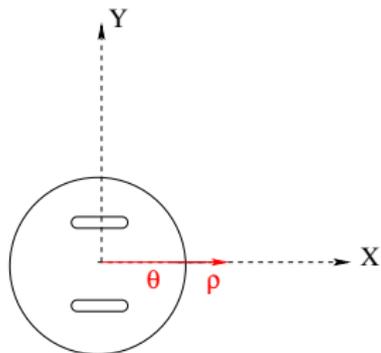


-
- 1 Introducción
 - 2 Arquitecturas Híbridas
 - 3 Ejemplo**

Ejemplo IRSIM

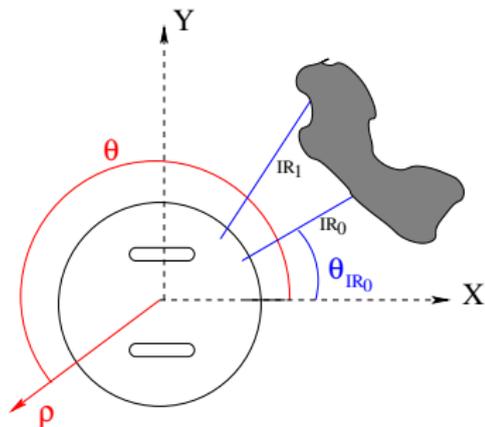


- ▶ $\theta = 0$
- ▶ $\rho = 0.1$

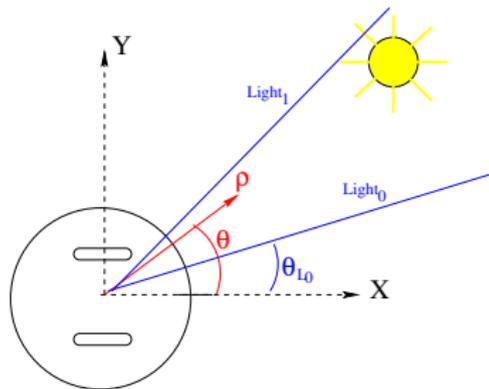


$$\blacktriangleright \theta = \pi - \operatorname{arctg} \left(\frac{\sum_{i=0}^7 IR_i \cdot \sin(\theta_{IR_i})}{\sum_{i=0}^7 IR_i \cdot \cos(\theta_{IR_i})} \right)$$

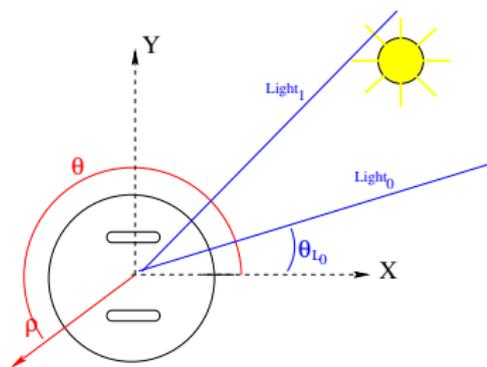
$$\blacktriangleright \rho = \begin{cases} 0 & \text{if } IR_{max} < \epsilon \\ 1 & \text{if } IR_{max} \geq \epsilon \end{cases}$$



$$\begin{aligned} \blacktriangleright \theta &= \arctg \left(\frac{\sum_{i=0}^7 L_i \cdot \sin(\theta_{L_i})}{\sum_{i=0}^7 L_i \cdot \cos(\theta_{L_i})} \right) \\ \blacktriangleright \rho &= \begin{cases} 0 & \text{if } Batt > \epsilon \\ Light_{max} & \text{if } Batt \leq \epsilon \end{cases} \end{aligned}$$



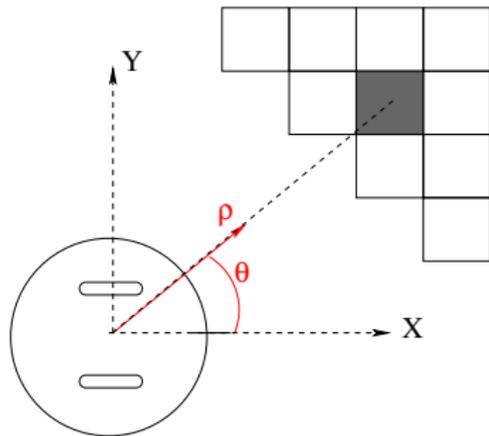
- ▶ $\theta = \pi - \operatorname{arctg} \left(\frac{\sum_{i=0}^7 L_i \cdot \sin(\theta_{L_i})}{\sum_{i=0}^7 L_i \cdot \cos(\theta_{L_i})} \right)$
- ▶ $\rho = \begin{cases} 0 & \text{if } \mathit{GroundMem} = 0 \\ 1 - \mathit{Light}_{max} & \text{if } \mathit{GroundMem} = 1 \end{cases}$
- ▶ Nota: Tener en cuenta los inhibidores



- ▶ **Inicialización:** Crea un mapa del tamaño de la arena con todas las celdas marcadas como obstáculos
- ▶ **Ejecución:**
 - ▶ Marca las celdas por las que va pasando como libre de obstáculos
 - ▶ Si encuentra un objeto (baldosa gris, *groundMemory* = 1): Lo marca como **Prey**
 - ▶ Si deja un objeto (baldosa negra, *groundMemory* = 0): Lo marca como **Nest**
 - ▶ Si ha encontrado *Prey* y *Nest* se lo comunica a *PathPlanning*
 - ▶ Mantiene el estado de búsqueda (*m_nForageStatus*): Búsqueda de *Prey* o *Nest*
 - ▶ Inicializa el estado de la Planificación (*m_nPathPlanningDone*, *m_nState*)
- ▶ **Reset:** Elimina las posiciones de *Prey* y *Nest*

- ▶ **Inicialización:** Crea una copia del mapa de `Calc Cell`
- ▶ **Ejecución:** Planifica una ruta en base a un algoritmo A^* si:
 - ▶ Se ha localizado un *Prey*
 - ▶ Se ha localizado un *Nest*
 - ▶ No hay ninguna planificación ya realizada
(`m_nPathPlanningDone = 0`)

- ▶ $\theta = \arctg \left(\frac{Y_{Grid} - Y_{Robot}}{X_{Grid} - X_{Robot}} \right)$
- ▶ $\rho = 1$
- ▶ Nota: Tener en cuenta la conversión de coordenadas



GRACIAS!!

GRACIAS!!