

Agentes Inteligentes

Javier García

Departamento de Electrónica y Computación
Universidad de Santiago de Compostela

November 10, 2021

Part VI

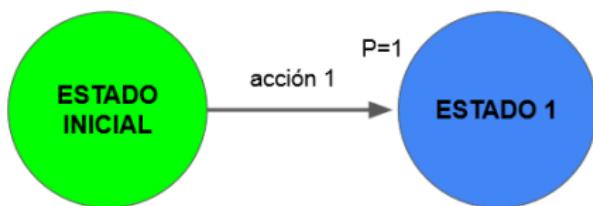
Incertidumbre en robótica

- 1 Incertidumbre en robótica
- 2 Toma de decisiones con incertidumbre
 - Caracterización de problemas
 - MDPs
 - MDPs Parcialmente Observables
- 3 Intercalado de planificación y ejecución

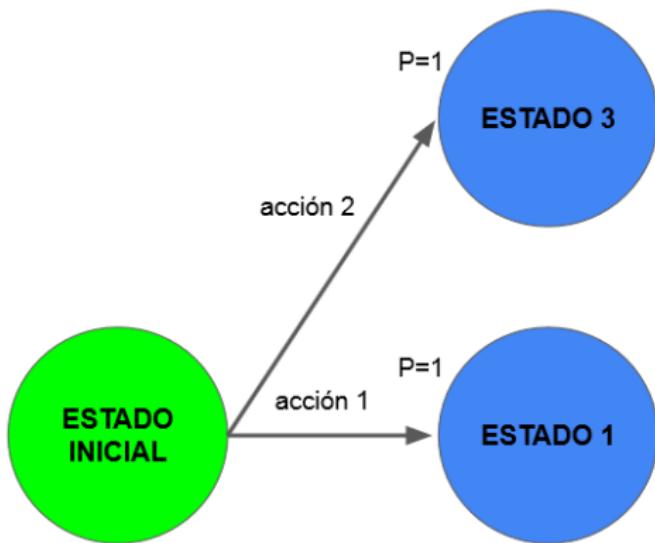
Hasta ahora...



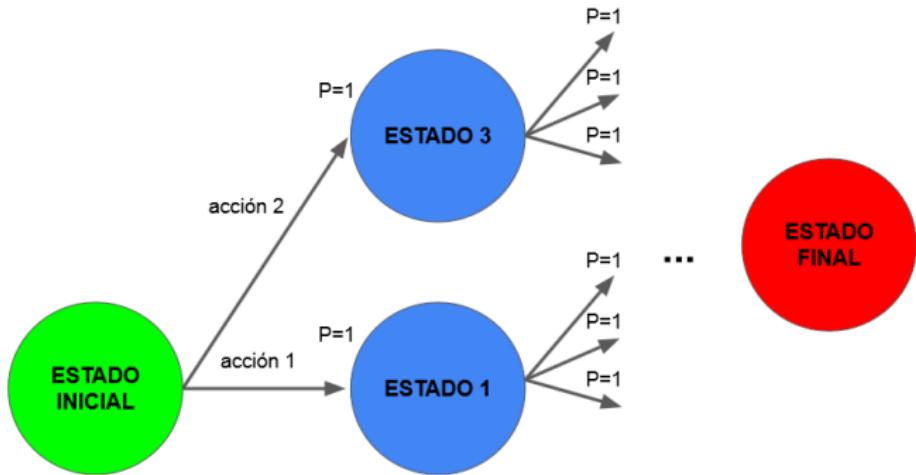
Hasta ahora...



Hasta ahora...



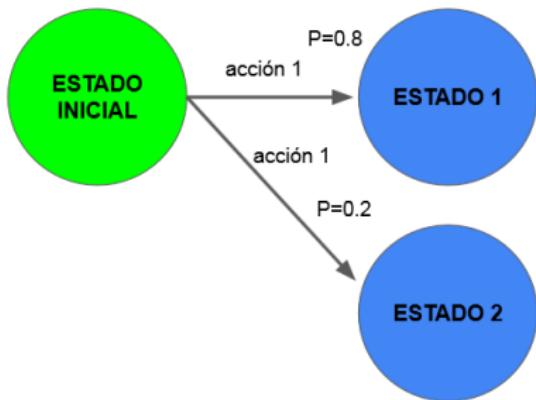
Hasta ahora...



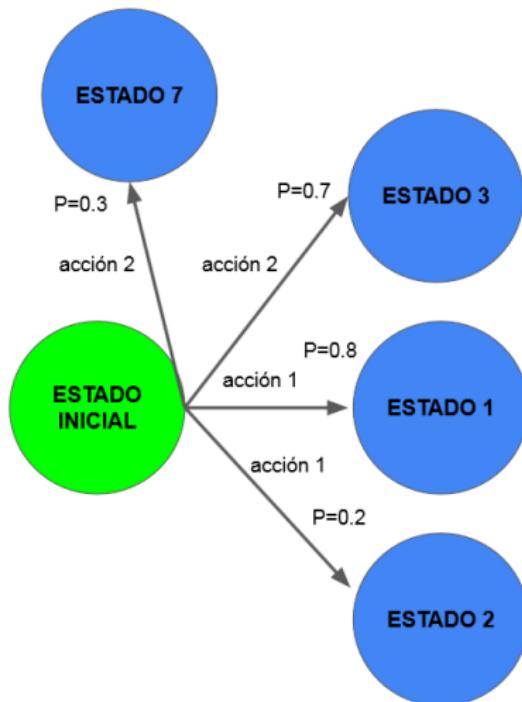
- Espacios de estados y acciones discretos
- **Entornos deterministas**
- **Totalmente observables**

Incertidumbre en robótica

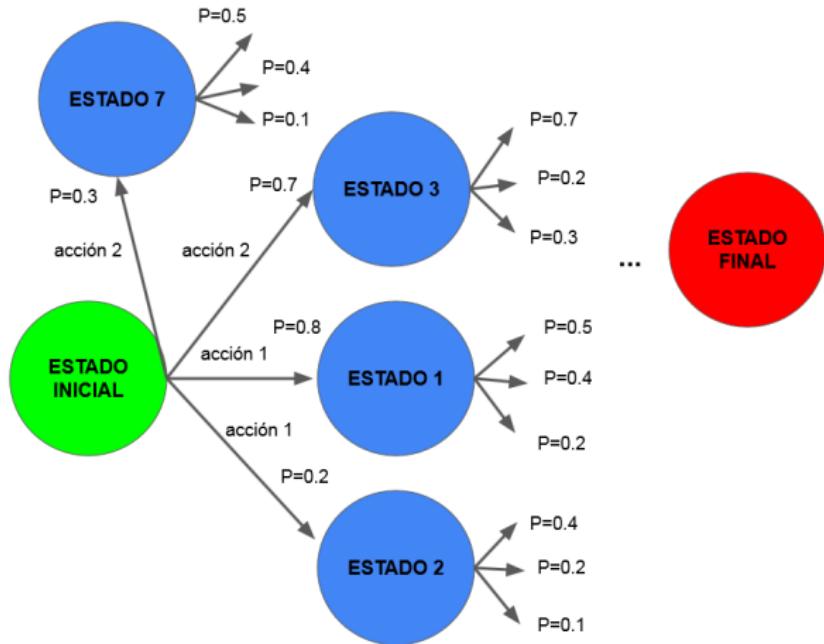
Más cercano al mundo real...



Más cercano al mundo real...



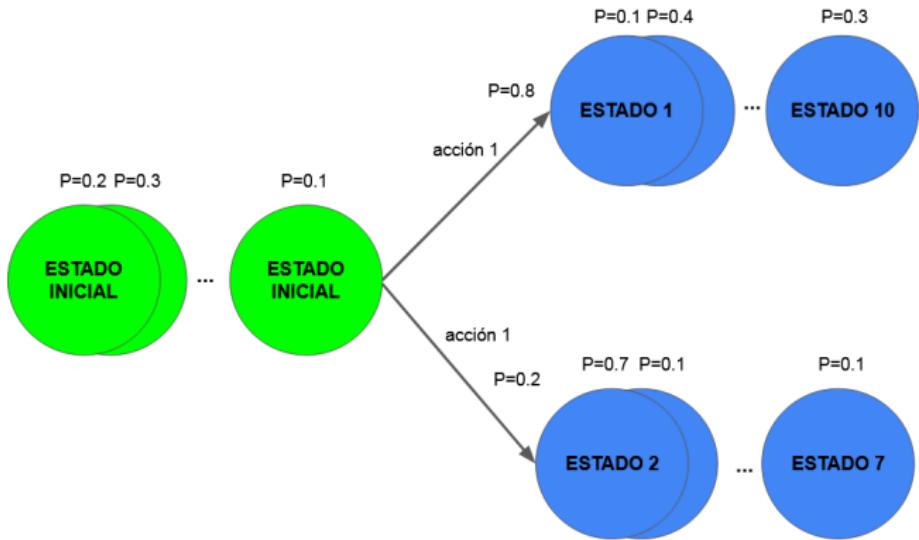
Más cercano al mundo real...



...el mundo en la mayoría de los casos es estocástico

Incertidumbre en robótica

Aún más cercano al mundo real...



...el mundo en la mayoría de los casos es estocástico

...y el robot puede tener incertidumbre del estado en el que se encuentra, i.e., el entorno es parcialmente observable

Incertidumbre en robótica

- En robótica dos fuentes principales de incertidumbre:
 - **Modelo de acción:** En muchos problemas de planificación no se puede suponer una dinámica determinista
 - **Observabilidad:** En muchos problemas, manejar una descripción completa y perfecta del estado es imposible
- Si la incertidumbre es pequeña, habitualmente se ignora y se delega en la **replanificación**
- En otras ocasiones no se puede ignorar...

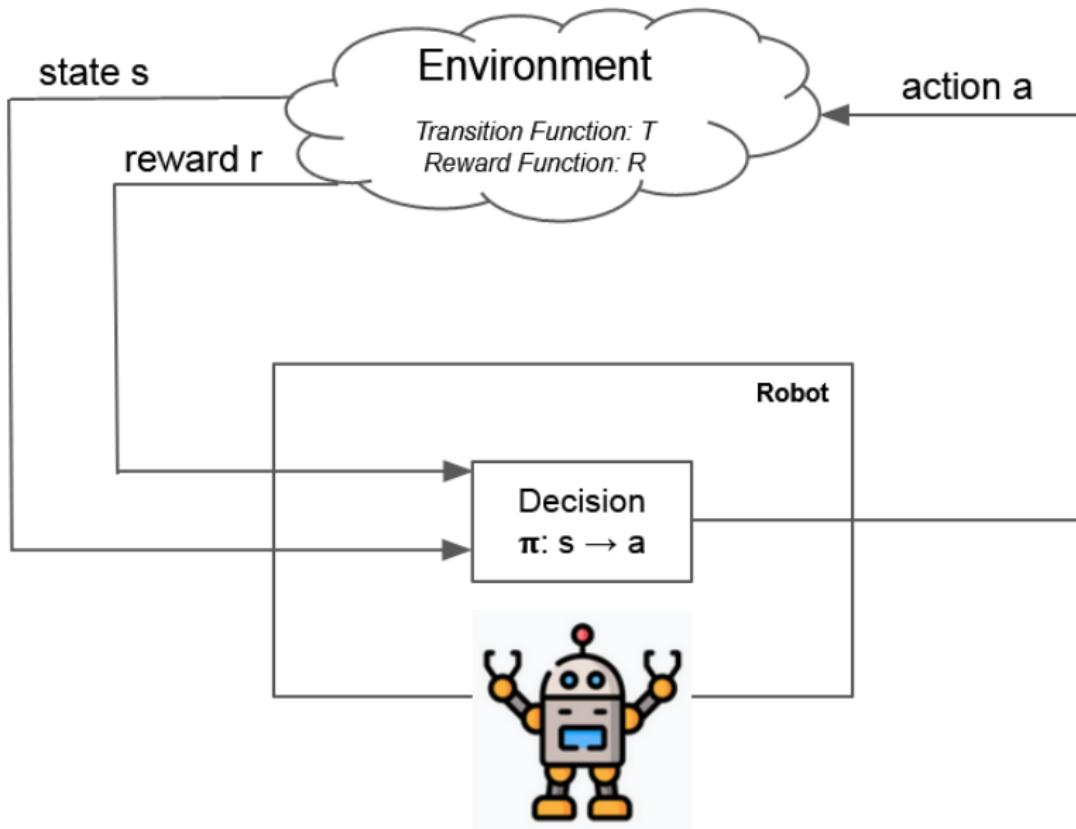
- **Planificación probabilística:**

- Asumimos probabilidades en el modelo de transiciones, i.e., **aborda la incertidumbre en el modelo de la acción pero no en la observabilidad**
- Los problemas se modelan como Procesos de Decisión de Markov (**MDPs**, por sus siglas en inglés)
- Programación dinámica y aprendizaje por refuerzo

- **Planificación probabilística contingente:**

- Además de asumir probabilidades en las transiciones, observaciones parciales, i.e., **aborda la incertidumbre en el modelo de acción y la observabilidad**
- Los problemas se modelan como Procesos de Decisión de Markov Parcialmente Observables (**POMDPs**, por sus siglas en inglés)

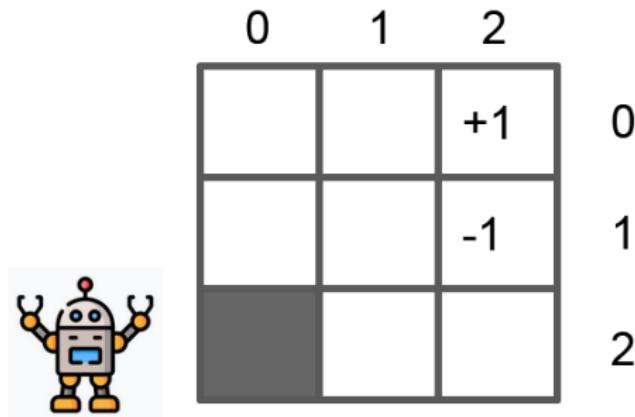
Procesos de Decisión de Markov



Procesos de Decisión de Markov

- Formalmente un **MDP** se describe como una tupla $\mathcal{M} = \langle S, A, T, R \rangle$ donde:
 - S es el conjunto de estados donde se puede encontrar el agente
 - A el conjunto de acciones que puede ejecutar en cada estado
 - $T(s, a, s')$ es la probabilidad de transitar del estado $s \in S$ al estado $s' \in S$ cuando se ejecuta la acción $a \in A$
 - $R(s, a)$ es el refuerzo que percibe el agente cuando ejecuta la acción $a \in A$ en el estado $s \in S$
- El objetivo es encontrar una política de comportamiento $\pi(s)$ que determine qué acción ejecutar en cada estado de forma que se maximice el refuerzo obtenido a lo largo del tiempo
- **Métodos:** programación dinámica (si se conoce el modelo), aprendizaje por refuerzo (si se desconoce el modelo)
- **Con búsqueda/planificación hemos resuelto:**
 - MDPs deterministas asumiendo que conocíamos el modelo del mundo
 - Los refuerzos se pueden transformar en costes

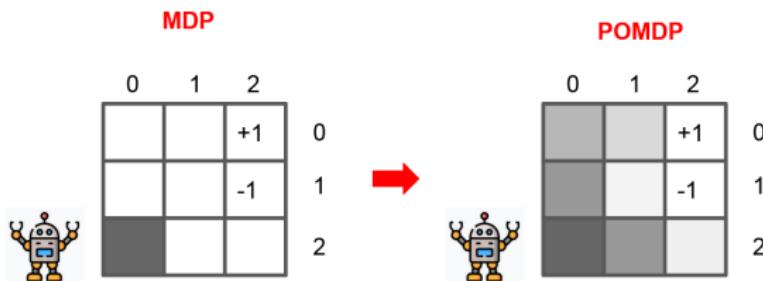
Procesos de Decisión de Markov



- S : Celdas donde se puede encontrar el robot, $s_{00}, s_{01}, \dots, s_{22}$
- A : Norte, Sur, Este, Oeste
- T : $T(s,a,s') = P(s' | s, a) = 1$, i.e., determinista aunque no tendría por qué ser así
- R : +1 en estado s_{02} , -1 en s_{12} , 0 en cualquier otro caso
- **En cada instante de tiempo el robot está en un estado $s \in S$ con probabilidad 1, en la imagen la casilla sombreada s_{20}**

MDPs Parcialmente Observables

- En los MDPs Parcialmente Observables se desconoce cuál es la posición exacta del robot en el entorno...
- ...pero sí se conoce la distribución de probabilidades de pertenecer a cada uno de estos estados
- A esta distribución de probabilidades se le conoce como *belief state* y habitualmente se representa con una b



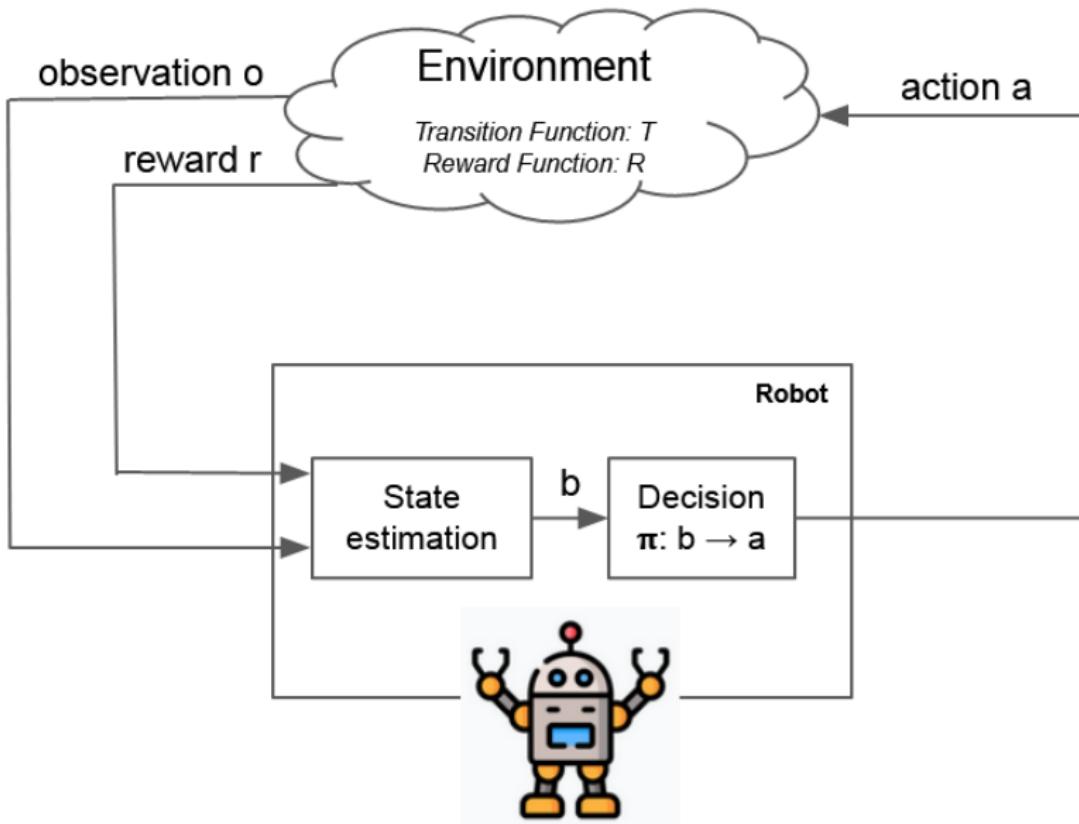
El robot ya no está en s_{20} con probabilidad 1, sino que puede estar en s_{20} con probabilidad 0.7, en s_{00} con probabilidad 0.1...
Con estas probabilidades se compone el *belief state*: $b = \langle 0.1, 0.05, \dots, 0.7 \rangle$, donde $b(s)$ es la probabilidad en particular del estado s en b

- El estado de creencias b se actualiza en cada paso utilizando dos modelos probabilísticos:
 - **El modelo de acción**, i.e., la función de transición $T(s, a, s') = P(s'|s, a)$
 - **El modelo sensorial**, i.e., la probabilidad $P(o|s)$ de percibir la observación o en el estado s
- Si el agente tiene la creencia $b(s)$, ejecuta la acción a y percibe la observación o , la nueva creencia $b'(s')$ es:

$$b'(s') = \alpha P(o|s') \sum_s P(s'|s, a) b(s) \quad (1)$$

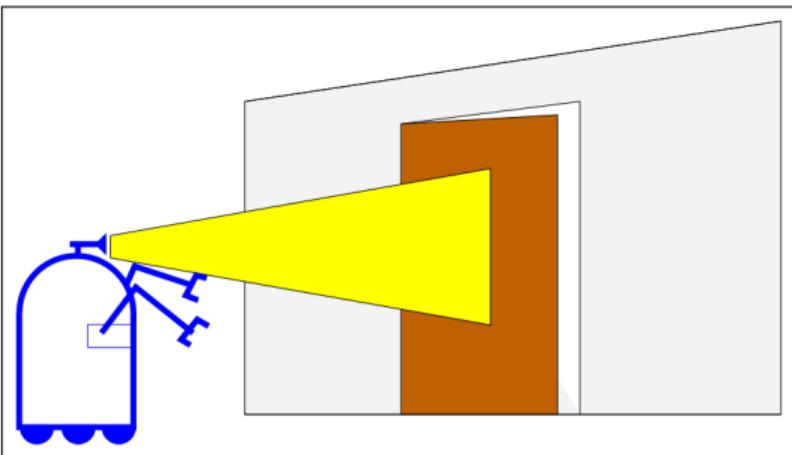
- Con todos estos ingredientes ya podemos definir formalmente un POMDP...

MDPs Parcialmente Observables



- Formalmente los **POMDPs** se pueden describir como una tupla $\mathcal{M} = \langle B, A, T, R, O \rangle$
 - B es el conjunto de *belief states*. Cada $b \in B$ es una distribución de probabilidad sobre el conjunto de estados. La probabilidad de un estado en particular se denota $b(s)$
 - A el conjunto de acciones que puede ejecutar en cada estado
 - $T(s, a, s')$ es la probabilidad de transitar del estado $s \in S$ al estado $s' \in S$ cuando se ejecuta la acción $a \in A$
 - $R(s, a)$ es el refuerzo que percibe el agente cuando ejecuta la acción $a \in A$ en el estado $s \in S$
 - O es el conjunto de observaciones (información sensorial). Por cada $o \in O$, y $s \in S$ hay una probabilidad $P(o|s)$ de observar o en el estado s

MDPs Parcialmente Observables - Ejemplo



- Un robot estima el estado de una puerta a través de su cámara
- La puerta solo puede estar en dos posibles estados: *abierta*, *cerrada*
- El robot puede ejecutar dos posibles acciones: empujar la puerta, o no hacer nada

MDPs Parcialmente Observables - Ejemplo

Formalización del problema:

- $S = \{abierta, cerrada\}$
- $A = \{empujar, no_op\}$
- $T(s, a, s') = P(s'|s, a) :$
 - Para la acción *empujar*:
 - $P(s' = abierta|s = abierta, a = empujar) = 1$
 - $P(s' = cerrada|s = abierta, a = empujar) = 0$
 - $P(s' = cerrada|s = cerrada, a = empujar) = 0.2$
 - $P(s' = abierta|s = cerrada, a = empujar) = 0.8$
 - Para la acción *no_op*:
 - $P(s' = abierta|s = abierta, a = no_op) = 1$
 - $P(s' = cerrada|s = abierta, a = no_op) = 0$
 - $P(s' = abierta|s = cerrada, a = no_op) = 0$
 - $P(s' = cerrada|s = cerrada, a = no_op) = 1$

MDPs Parcialmente Observables - Ejemplo

Formalización del problema:

- Modelo sensorial: la cámara tiene ruido que sigue la siguiente distribución
 - $P(o = \text{sensado_abierta} | s = \text{abierta}) = 0.6$
 - $P(o = \text{sensado_cerrado} | s = \text{abierta}) = 0.4$
 - $P(o = \text{sensado_abierta} | s = \text{cerrada}) = 0.2$
 - $P(o = \text{sensado_cerrada} | s = \text{cerrada}) = 0.8$
- El robot inicialmente no conoce el estado de la puerta por lo tanto, el vector de creencias b inicial $b = \langle 0.5, 0.5 \rangle$, i.e., $b(\text{abierta}) = 0.5$ y $b(\text{cerrada}) = 0.5$

MDPs Parcialmente Observables - Ejemplo

- En el siguiente instante de tiempo, el robot no hace nada y detecta la puerta abierta:

$$b'(s') = \alpha P(o|s') \sum_s P(s'|s, a) b(s) \quad (2)$$

$$\begin{aligned} b'(s' = \text{abierta}) &= \alpha P(o = \text{abierta}|s' = \text{abierta}) \times \\ &\quad (P(\text{abierta}|s = \text{abierta}, a = \text{no_op}) b(s = \text{abierta}) \\ &\quad + P(\text{abierta}|s = \text{cerrada}, a = \text{no_op}) b(s = \text{cerrada})) \\ &= \alpha 0.3 \end{aligned}$$

$$\begin{aligned} b'(s' = \text{cerrada}) &= \alpha P(o = \text{abierta}|s' = \text{cerrada}) \times \\ &\quad (P(\text{cerrada}|s = \text{abierta}, a = \text{no_op}) b(s = \text{abierta}) \\ &\quad + P(\text{cerrada}|s = \text{cerrada}, a = \text{no_op}) b(s = \text{cerrada})) \\ &= \alpha 0.1 \end{aligned}$$

- $b'(s' = \text{abierta}) + b'(s' = \text{cerrada}) = 1$, luego
 $\alpha = 1/(0.3 + 0.1) = 2.5$
- $b' = \langle 0.75, 0.25 \rangle$, i.e., $b'(s' = \text{abierta}) = 0.75$ y
 $b'(s' = \text{cerrada}) = 0.25$

- Si en el siguiente paso el robot empuja la puerta y vuelve a detectar a través de su cámara que está abierta:
 $b''(s'' = \text{abierta}) = 0.983$
 $b''(s'' = \text{cerrada}) = 0.017$
- En este punto podríamos considerar simplemente que la puerta está abierta y que los dos sensados anteriores han sido correctos, pero esta suposición no es válida en entornos críticos
 - Imagina volar un avión en piloto automático con una posibilidad de 0.983 de no estrellarse
 - **¡Es necesario considerar todas las probabilidades en el proceso de toma de decisiones!**
- Este proceso también se conoce como **Localización de Markov** (Thrun, 2002)

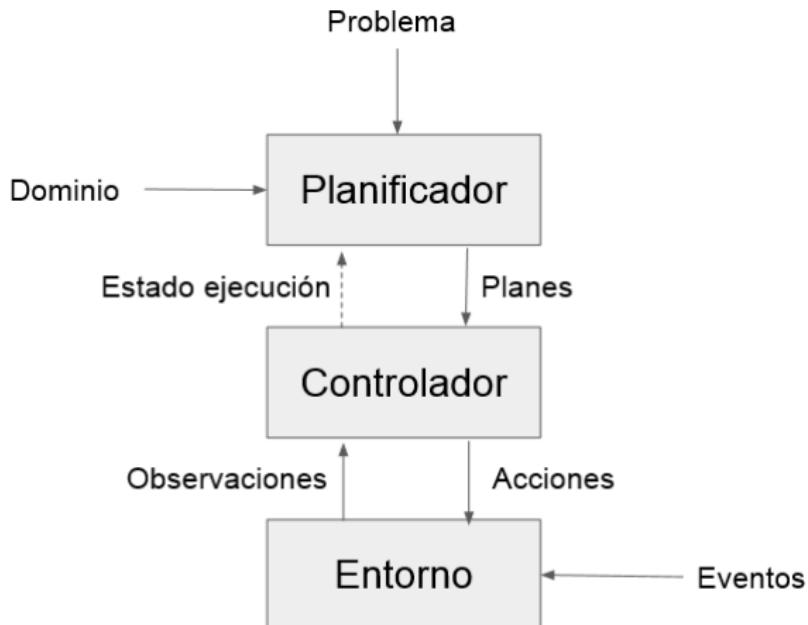
MDPs Parcialmente Observables - Ejemplo

- El objetivo es aprender una política $\pi(b) = a$ que permita determinar la acción a a ejecutar dada la creencia b de forma que se maximice alguna medida de refuerzo a largo plazo
- Se puede transformar el POMDP a un MDP:
 - Un estado por cada *belief state*
 - Definición de una función de transición $P(b'|b, a)$
 - ...
- **¡Impracticable en la mayoría de los casos!**

Intercalado de planificación y ejecución

- Los algoritmos que resuelven MDP y POMDP son ideales para robótica, sin embargo:
 - Algunos requieren conocer *a priori* el modelo de acción y sensorial
 - Otros, que no requieren conocer *a priori* estos modelos, como los algoritmos de aprendizaje por refuerzo, requieren probar acciones para decidir cuál es la mejor en cada estado, lo que es inviable en robótica
 - Ademas, calcular políticas óptimas para los POMDP resulta impracticable en la mayoría de los casos
- Por todo ello, utilizar algoritmos con incertidumbre no siempre merece la pena en el mundo real
- **Habitualmente se prefiere intercalar planificación con el seguimiento de la ejecución para adaptar los planes cuando aparecen conflictos tras la ejecución de una acción**

Intercalado de planificación y ejecución



Intercalado de planificación y ejecución

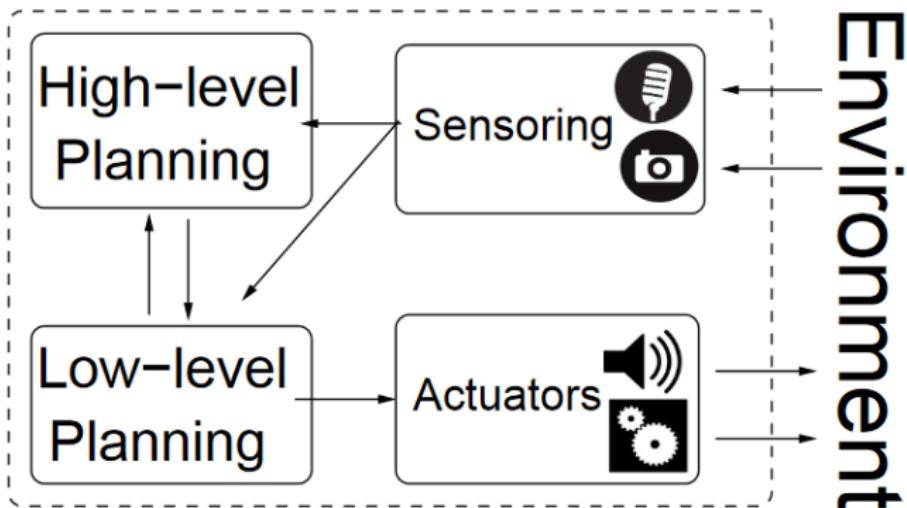
- Un subplan se considera ejecutable cuando todas las precondiciones de las acciones restantes se pueden cumplir
- Subplan: 1. unstack(C,A), 2. putdown(C), 3. pickup(B), 4. stack(B,C), 5. pickup(A), 6. stack(A,B)

on(C,A) clear(C) hand empty	unstack(C,A)				
	holding(C)		putdown(C)		
on(B,table)			hand empty	pickup(B)	
			clear(C)	holding(B)	stack(B,C)
on(A,table)	clear(A)			hand empty	pickup(A)
				clear(B)	holding(A)
		on(C,table)		on(B,C)	stack(A,B)
					on(A,B) clear(A)

Intercalado de planificación y ejecución

- Cuando se detecta que el subplan no se puede ejecutar
 - **Replanificación:** Se crea un nuevo plan desde 0 tomando como estado inicial el actual del robot, con las mismas o con otras metas
 - **Reparación:** Se adapta el nuevo plan a la nueva situación
- La reparación habitualmente es menos costosa computacionalmente que generar un nuevo plan desde 0...
- ...pero requiere (*re*)*planificadores* específicos (Koenig et al., 2002)
- Pero aún queda problemas por resolver en el modelo conceptual anterior:
 - De alguna forma hay que *traducir* las acciones de alto nivel del planificador a comandos interpretables por el robot
 - De alguna forma hay que *traducir* la información sensorial del robot a predicados de alto nivel interpretables por el planificador

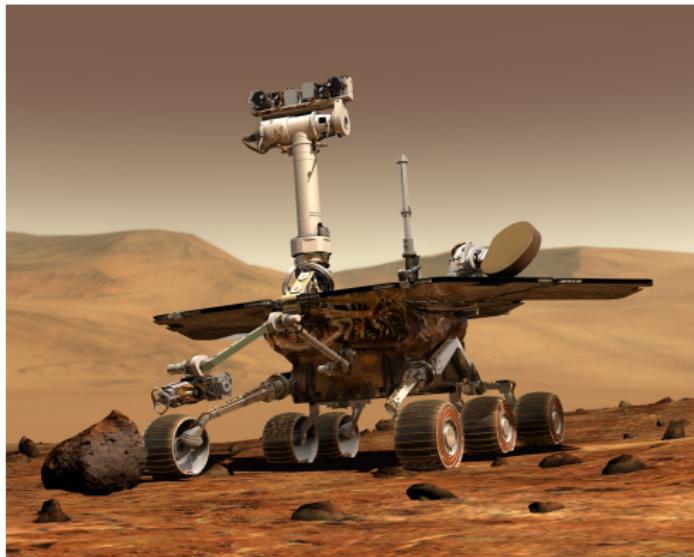
Intercalado de planificación y ejecución



- El controlador anterior se divide en dos nuevos módulos de sensorización y actuación
- El planificador anterior se considera un planificador de alto nivel (**task planning**) y aparece un nuevo planificador de bajo nivel (**motion/path planning**)

Intercalado de planificación y ejecución

- Rovers
 - **Acciones:** *navigate, sample-rock, communicate-rock-data,...*
 - **Metas:** *communicated_rock_data*



Intercalado de planificación y ejecución

1. Sensado

rover $x = 0.65$
 $y = -0.3$
 $z = 0.11$

laser [0.01, 0.111, 1.0, ..., 1.0]

cam



(at rover1 wp1)
(at_rock_sample wp2)
...
(visible wp1 wp2)

Información sensorial en bruto

Estado simbólico

Los datos sensoriales en bruto son demasiado detallados/ruidosos y los planificadores simbólicos no pueden razonar con ellos

2. Planificación de alto nivel

(at rover1 wp1)
(at_rock_sample wp2)

...
(visible wp1 wp2)

Estado simbólico

(communicated_rock_sample)

Meta



1. (navigate rover1 wp1 wp2)
2. (sample-rock rover1 wp2)
3. (communicate-rock-data rover1)

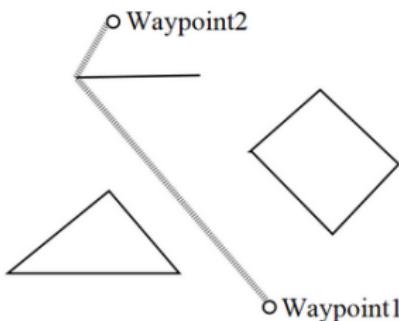
El planificador se encarga de construir un plan compuesto de acciones de alto nivel

3. Planificación de bajo nivel

1. (navigate rover1 wp1 wp2)



1. turn 90°
2. move
3. turn 23°
4. ...



El planificador de bajo nivel es el encargado de traducir las acciones de alto nivel a comandos interpretables por el robot y ejecutarlos con ayuda de los actuadores. Utiliza algoritmos como Dijkstra, A* o RRT

Intercalado de planificación y ejecución

- Una vez ejecutada la acción *navigate*, se volvería a *sensar* el entorno
- Si el subplan restante se puede ejecutar, se continua con la siguiente acción, sino se invoca nuevamente al planificador de alto nivel
- El proceso se repite hasta que se han ejecutado todas las acciones del plan

El paradigma de intercalar planificación y ejecución es ampliamente utilizado, sin embargo, no es una panacea. Dado que la planificación clásica no asume fallos al ejecutar las acciones, puede producir planes demasiado frágiles que deben repararse continuamente o incluso peor aún, planes que no se pueden arreglar

Intercalado de planificación y ejecución

Arquitecturas basadas en este paradigma: PELEA

