

A Procesos de decisión de Markov

Los procesos de decisión de Markov y, en especial, los parcialmente observables, están siendo objeto de un amplio estudio por parte de la comunidad científica que se dedica a planificación y control en dominios estocásticos, dentro del área de Inteligencia Artificial.

En este apéndice se hace una breve introducción a los procesos de decisión de Markov (MDP). En la primera parte se describe el concepto y soluciones básicas de los Procesos de Decisión de Markov Completamente Observable (COMDP). A partir de éstos, en la segunda parte, se definen los procesos de Decisión de Markov Parcialmente Observable (POMDP). Una descripción más detallada se puede encontrar en diversas referencias [Hauskrecht 1996] [Cassandra 1998].

A.1 COMDP

Un sistema de decisión de Markov completamente observable (COMDP) representa el comportamiento dinámico de un proceso sometido a diferentes acciones. La situación de dicho proceso se caracteriza mediante una serie de estados en los que se puede encontrar. Las transiciones entre los mismos no son deterministas y viene dada en forma de probabilidades. Sobre el sistema se puede efectuar una serie de acciones que permiten cambiar su estado de acuerdo con las probabilidades de transición. Se trata de un proceso discreto con un número finito o infinito de pasos. A cada paso es posible la ejecución de una acción diferente, determinada de acuerdo con un sistema de decisión. Además, con objeto de caracterizar el comportamiento deseado del proceso, se define un sistema de costes o bonificación.

En definitiva, un COMDP se define formalmente como:

$$\Xi \equiv (S, A, R, T)$$

donde:

- ✓ S: conjunto de estados del proceso a modelar.
- ✓ A: conjunto de acciones posibles a realizar.
- ✓ R: recompensa o ganancia.

$$R : S \times A \rightarrow \Re$$

- ✓ T: probabilidad de transición entre estados.

$$T : S \times A \times S \rightarrow \Pi(s)$$

$$t(s, a, s') = P(s^{n+1} = s' / s^n = s, A^n = a)$$

Definido el modelo del proceso, el objetivo de este planteamiento consiste en encontrar una acción o secuencia de acciones a ejecutar a lo largo del tiempo que optimize los beneficios obtenidos de acuerdo con una *función valor*. Esta función combina la bonificación a lo largo del tiempo, calculando un valor que representa la ganancia acumulada.

La propiedad de Markov implica que el proceso no tiene memoria. Es decir, que la decisión acerca de la acción que se ejecutará en el próximo intervalo n depende únicamente del estado del sistema al finalizar el intervalo anterior que es cuando se toma dicha decisión.

En lo que sigue, se va a considerar los COMDP con un número finito de estados y actuaciones.

A.1.1 Clasificación

Existen distintas formas de clasificar los MDP de acuerdo a la naturaleza del proceso. La más común se realiza en función del horizonte con el cual se desea planificar la solución, obteniéndose una sencilla división en dos grupos:

- ✓ *Horizonte finito.* En este caso el interés se centra en maximizar la bonificación para un número finito de pasos. La ganancia esperada en T pasos se puede calcular a partir de la ecuación siguiente:

$$E \left[\sum_{n=0}^{T-1} (r)^n r^n \right] \quad 0 \leq r \leq 1 \quad [Eq. A.1]$$

donde r es un factor de descuento que permite penalizar el hecho que una acción se ejecute demasiado tarde ($r < 1$) y r^n es la bonificación en el paso n .

- ✓ *Horizonte infinito.* El objetivo, en este caso, es maximizar la bonificación a largo plazo que se puede calcular de acuerdo con la ecuación A.2.

$$E \left[\sum_{n=0}^{\infty} (r)^n r^n \right] \quad 0 \leq r < 1 \quad [Eq. A.2]$$

A.1.2 Políticas de decisión. Planes de actuación

La decisión sobre la siguiente acción a ejecutar se toma en función del estado actual, usando la propiedad de Markov.

Se define una *regla de decisión* m como una relación que a cada estado le asigna una acción:

$$m: S \rightarrow A$$

$$m(s) = a$$

A partir de este concepto, se define una *política*¹ o *plan de actuación* a T pasos (p^T) como una secuencia de reglas de decisión.

$$p = (m^1, m^2, m^3, \dots, m^T)$$

¹ También es conocido como *estrategia* o *plan de control*.

El problema a resolver por cualquier tipo de proceso de decisión de Markov es encontrar la política óptima, ya sea para un determinado estado inicial o para un caso general.

Entre otras divisiones de las políticas, la más difundida se refiere a su variación temporal:

- ✓ *Estacionarias*. Asignan a cada estado una acción independientemente del tiempo ($\mu^0 = \mu^1 = \mu^2 = \dots = \mu^T$). Existe un teorema que garantiza para cualquier COMDP con horizonte infinito la existencia de una política de Markov estacionaria que es óptima [Hauskrecht 1996].
- ✓ *No estacionarias*. La acción a ejecutar en un estado depende del tiempo. La política de control óptimo para los modelos de horizonte finito es de este tipo [Hauskrecht 1996] y está formada por una secuencia de reglas de decisión.

A.1.3 Funciones Valor

Se usa este término para referirse a las funciones que cuantifican la ganancia (bonificación acumulada) obtenida al actuar de acuerdo con una determinada política p' . La forma de calcular su valor depende del horizonte del COMDP, realizándose normalmente de forma recurrente para los de horizonte finito como puede apreciarse en la ecuación A.3.

Horizonte finito

Dada una política p^T estacionaria de horizonte finito de T pasos, se define $V_n(p^T, s)$ como el valor de bonificación esperada a partir del paso n ejecutando la política p^T cuando el sistema se encuentra en el estado s . Dicho valor se puede calcular de acuerdo con la ecuación:

$$V_n(p^T, s) = r(s, m^n(s)) + r \sum_{s' \in S} t(s, m^n(s), s') V_{n+1}(p^T, s') \quad [Eq. A.3]$$

donde $m^n(s)$ es la acción determinada por la política p^T para el paso n . El primer término de la ecuación ($r(s, m^n(s))$) se corresponde con la recompensa obtenida en el paso n , mientras que el segundo término se refiere a la suma de las recompensas $V_{n+1}(p, s')$ que se obtienen a partir del paso $n+1$ para los distintos estados posibles (s') ponderadas por la probabilidad $t(s, m^n(s), s')$ de que el sistema se encuentre en ese

estado (s') en el siguiente paso. El cálculo se ha de realizar de forma recurrente comenzando con el último paso en el cual sólo interviene el primer término de la ecuación A.3 ($r(s, m(s))$).

Horizonte infinito

En este caso la política óptima es estacionaria y el cálculo de la función valor se restringirá a este tipo de políticas. El sistema resultante es una serie de ecuaciones con tantas ecuaciones e incógnitas como el número de estados $|S|$.

$$V(p', s) = r(s, m(s)) + r \sum_{s' \in S} t(s, m(s), s') V(p', s') \quad [Eq. A.4]$$

Una propiedad importante que explotan algunos de los algoritmos de cálculo de la política óptima viene dada por la siguiente ecuación:

$$\lim_{T \rightarrow \infty} \|V_1(p^T, s) - V(p, s)\| = 0 \quad [Eq. A.5]$$

donde p^T es la política óptima para el horizonte finito y p es la política del horizonte infinito, $V(p^T)$ es la función valor de la política óptima para el COMDP de horizonte T pasos y $V(p)$ es la correspondiente al horizonte infinito.

Aplicando el principio de optimalidad [Cassandra 1998] se puede observar que los primeros T pasos de cualquier política óptima con T' pasos, donde $T' \geq T$, coinciden, por lo tanto se cumple la siguiente igualdad:

$$V_n(p^T, s) = V_n(p^{T'}, s) \quad \forall T', T / T' \geq T \quad [Eq. A.6]$$

Una importante consecuencia de esta igualdad es que para calcular una política de T pasos a partir de otra de $T-I$ basta con obtener una regla de decisión (la del primer paso).

A.1.4 Cálculo de la política óptima

Cabe recordar que el objetivo del COMDP es obtener un plan o política óptima, entendiendo por tal aquella que maximice la función valor. Es decir, la política que haga máximo el valor de la ecuación A.3 o A.4.

Existen básicamente dos algoritmos que obtienen la política óptima por iteración, a partir de los cuales se han desarrollado distintas versiones con objeto de acelerar los cálculos. Dichos algoritmos son *Iteración de la función valor* e *Iteración de la política*.

A.1.4.1 Iteración de la función valor

Este algoritmo se basa en el cálculo recurrente de las acciones que maximizan la función valor. Hay que distinguir dos casos según el horizonte del COMDP.

Horizonte finito

Al igual que antes, se supone T el número de pasos y r el factor de descuento.

En cada paso de la iteración se obtienen las acciones que hagan máximo el valor de la ecuación A.3. Se calcula de forma simultánea la política óptima y la función valor para dicha política comenzando por el último paso (T). En general, para realizar los cálculos para el paso n se conocen ya las reglas de decisión y funciones valor para los pasos $n+1 ..T$ y se obtiene la regla de decisión y función valor de acuerdo con la ecuación A.7:

$$V_n(p^T, s) = \max_{a \in A} \left[r(s, a) + r \sum_{s' \in S} t(s, a, s') V_{n+1}(p^T, s') \right] \quad [Eq. A.7]$$

donde $V_n(p^T)$ es la función valor para la política óptima p^T desde el paso n al T . La política óptima se puede calcular a partir de la ecuación anterior comenzando por el último paso T . Este método es conocido como iteración de la función valor y su funcionamiento se presenta en el algoritmo A.1.

Algoritmo A.1 Cálculo de la política óptima mediante iteración de la función valor

Iteración de la función Valor (X, r, T)

Para cada $s \in S$
 $V_{T+1}(s) = 0$

Fin para cada s

Para cada $m \in \{T, T-1, T-2, \dots, 1\}$
Para cada $s \in S$

$$V_m(s) = \max_{a \in A} \left[r(s, a) + r \sum_{s' \in S} t(s, a, s') V_{m+1}(p^T, s') \right]$$

Añadir a la regla de decisión m^m la acción a para s

Fin para cada s

Añadir a la política p^T la regla de decisión m^m

Fin para cada m

Resultado { función valor primer paso $V_1(\cdot)$ y política p^T }

Fin Iteración de la función Valor.

Horizonte Infinito

Se puede hacer de forma similar al caso anterior incrementando T pues cuando T tiende a infinito, la política obtenida tenderá a la óptima (ecuación A.5). Cabe recordar que calcular una política para T pasos a partir de la de $T-1$ consiste simplemente en obtener una regla de decisión más, propiedad utilizada por algunos algoritmos para obtener la política óptima.

Por otra parte, se sabe que dicha política es estacionaria, con lo cual la ecuación A.7 pasará a tener la siguiente forma:

$$V^p(s) = \max_{a \in A} \left[r(s, a) + r \sum_{s' \in S} t(s, a, s') V^p(s') \right] \quad [Eq. A.8]$$

El sistema de ecuaciones A.8 define la función valor para cada uno de los estados y , a partir de ésta, se pueden obtener la política p compuesta por una única regla de decisión m definida de la siguiente forma:

$$m(s) = \arg \max_{a \in A} \left[r(s, a) + \gamma \sum_{s' \in S} t(s, a, s') V^p(s') \right]$$

A.1.4.2 Iteración de la política

Este método se aplica sólo para el caso de horizonte infinito, puesto que se supone que la política óptima es estacionaria. Se parte de una política inicial aleatoria que se va mejorando en cada paso mientras sea posible, como se muestra en el algoritmo A.2. La evaluación de la política (*evalúa_política*) se realiza a través de la función valor y el procedimiento de mejora (cálculo de m' mejor que la actual m) se resume en el algoritmo A.3.

Algoritmo A.2 Cálculo de la política óptima mediante iteración de la política

```

Iteración de la política (X, r )
    m' = cualquier regla de decisión.
    Para cada paso
        m = m'
        V = evalúa_política(X, r, m)
        m' = mejora_política(X, r, V, m)
    hasta que m = m'
Fin Iteración de la política
    
```

Algoritmo A.3 Mejora de la política

```

Mejora_política( $\mathcal{X}, r, V, m$ )
  para cada  $s \in S$ 
    para cada  $a \in A$ 
       $V(s) = \max_{a \in A} V^a(s)$ 
       $V^a(s) = r(s, a) + r \sum_{s' \in S} t(s, a, s') V(s')$ 
      si  $V(s) > V^{m(s)}(s)$ 
         $m'(s) = \operatorname{argmax}_{a \in A} V^a(s)$ 
      sino
         $m'(s) = m(s)$ 
    fin para cada  $a$ 
  fin para cada  $s$ 
  resultado  $m'$ 
fin Mejora_política

```

A.2 POMDP

El comportamiento de los procesos de decisión de Markov parcialmente observable (POMDP) es básicamente igual a los COMDP descritos anteriormente. Sin embargo, se elimina la restricción de que el estado es conocido en todo momento y en su lugar se añade una serie de observaciones. La eliminación de esta hipótesis complica el sistema de decisión, puesto que ahora no se conoce con exactitud el estado del proceso sino a través de las observaciones que no son deterministas.

A.2.1 Modelo

Al añadir incertidumbre en la determinación del estado del sistema, se hace necesaria la adición de nuevos parámetros para definir las observaciones \mathbf{Z} y probabilidades asociadas \mathbf{O} . De esta forma, el POMDP puede definirse formalmente como:

$$\Xi \equiv (S, A, Z, R, T, O)$$

donde:

- ✓ S: conjunto de estados.
- ✓ A: conjunto de acciones.
- ✓ Z: conjunto de observaciones.
- ✓ R: recompensa o ganancia.

$$R : S \times A \rightarrow \Re$$

- ✓ T: transiciones.

$$T : S \times A \times S \rightarrow \Pi(s)$$

$$t(s, a, s') = P(s^{n+1} = s' / s^n = s, A^n = a)$$

- ✓ O: probabilidades de observación.

$$O : S \times A \times Z \rightarrow \Pi(z)$$

$$o(a, s, z) = P(o^n = z / s^n = s, A^n = a)$$

Al igual que en el caso COMDP, el objetivo es encontrar la política óptima pero, puesto que ahora no se conoce el estado del sistema, las decisiones se han de hacer en función de la historia completa acerca del proceso.

Este problema se ha intentado resolver de diversas formas, como por ejemplo tratar de encontrar una política que asocie observaciones con acciones, obteniendo resultados insatisfactorios al realizar las decisiones sin tener en cuenta toda la información relevante. La mayoría de las soluciones exactas replantean este problema en la forma de un COMDP, tal como se muestra en el siguiente apartado.

A.2.2 Transformación de POMDP en COMDP

Para poder plantear el problema en forma de un COMDP se define un nuevo parámetro que condensa toda la información acerca de la historia del proceso. Ese nuevo parámetro es la *información de estado* definida como una distribución de probabilidades B entre los distintos estados (s). Es decir:

$$B : S \rightarrow \Pi(s)$$

$$b(s) = P(s)$$

donde $P(s)$ es la probabilidad de que el sistema se encuentre en el estado s . Esta distribución de probabilidades es un estadístico suficiente.

La información de estado después de ejecutar la acción a y observar z se puede calcular, usando reglas básicas de teoría de la probabilidad, a través de la siguiente ecuación:

$$b_z^a(s') = \frac{o(a, s', z) \sum_s t(s, a, s') b(s)}{\sum_{s, s'} o(a, s'', z) t(s, a, s'') b(s)} \quad [Eq. A.9]$$

Nuevo modelo

Utilizando este nuevo estadístico, el POMDP puede ser planteado como un COMDP donde:

- ✓ El nuevo espacio de estados es continuo y esta formado por los valores que puede obtener la información de estado b .
- ✓ Las acciones son las mismas que el POMDP
- ✓ La bonificación ahora viene dada por la ecuación:

$$w(b, a) = \sum_{s \in S} b(s) r(s, a) \quad [Eq. A.10]$$

- ✓ Las probabilidades de transición vienen dadas por la ecuación A.13 que se deduce a continuación.

Suponiendo un estado inicial dado por \mathbf{b} y ejecutando la acción \mathbf{a} se obtendrá el estado final \mathbf{b}' con probabilidad:

$$P(b, a, b') = \sum_z P(z / b, a) I(b', b_z^a) \quad [Eq.A.11]$$

donde

$$I(b', b_z^a) = \begin{cases} 1 & \text{si } \dots b' = b_z^a \\ 0 & \text{En cualquier otro caso.} \end{cases} \quad [Eq.A.12]$$

Aplicando principios estadísticos básicos se obtiene:

$$\begin{aligned} P(b, a, b') &= \sum_z \left[\sum_s \sum_{s''} p(z, s, s'' / b, a) \right] I(b', b_z^a) \\ &= \sum_z \left[\sum_s \sum_{s''} p(z / s, s'', b, a) p(s'' / s, b, a) p(s / b, a) \right] I(b', b_z^a) \end{aligned}$$

Teniendo en cuenta además la independencia de variables :

$$P(b, a, b') = \sum_z \left[\sum_s \sum_{s''} p(z / s'', a) p(s'' / s, a) p(s / b) \right] I(b', b_z^a)$$

Por último, identificando términos se obtiene:

$$P(b, a, b') = \sum_z \left[\sum_{s, s''} o(a, s'', z) t(s, a, s'') b(s) \right] I(b', b_z^a) \quad [Eq.A.13]$$

El nuevo sistema cumple la propiedad de Markov, dado que la información de estado es un estadístico suficiente y se pueden aplicar entonces los métodos descritos en el capítulo anterior con la complejidad añadida de que el espacio de estados es continuo.

A.2.3 Función valor

Al igual que se ha hecho con los COMDP, hay que diferenciar dos casos de acuerdo a la naturaleza del proceso.

Horizonte finito

A partir de la ecuación A.3, identificando términos con los del nuevo COMDP se obtiene la ecuación A.14.

$$V_n(\mathbf{p}^T, b) = w(b, \mathbf{m}(b)) + r \sum_{b' \in B'(b, \mathbf{m}(b))} P(b, \mathbf{m}(b), b') V_{n+1}(\mathbf{p}^T, b') \quad [Eq. A.14]$$

donde $B'(b, \mathbf{m}(b))$ es el conjunto de los estados resultantes posibles (valores posibles de la información de estado) al aplicar la acción $\mathbf{m}(b)$ al proceso cuando éste se encuentra en estado b . Se ha restringido la suma a este conjunto, dado que ahora el espacio de estados B es continuo (infinitos valores posibles). La política óptima será aquella que haga máxima dicha función:

$$V_n(\mathbf{p}^T, b) = \max_{a \in A} \left\{ w(b, a) + r \sum_{b' \in B'(b, a)} P(b, a, b') V_{n+1}(\mathbf{p}^T, b') \right\} \quad [Eq. A.15]$$

De la ecuación A.13 se deduce que el sumatorio sobre el conjunto de valores de información finales se puede convertir en un sumatorio sobre las observaciones obteniéndose:

$$V_n(\mathbf{p}^T, b) = \max_{a \in A} \left\{ w(b, a) + r \sum_{z \in Z} \left[\sum_{s, s'} o(a, s'', z) t(s, a, s'') b(s) \right] V_{n+1}(\mathbf{p}^T, b_z^a) \right\} \quad [Eq. A.16]$$

Esto se puede expresar de forma más compacta:

$$V_n(p^T, b) = \max_{a \in A} V_n^a(p^T, b) \quad [Eq. A.17]$$

$$V_n^a(p^T, b) = \sum_{z \in Z} V_n^{a,z}(p^T, b) \quad [Eq. A.18]$$

donde:

$$V_n^{a,z}(p^T, b) = \frac{1}{|Z|} w(b, a) + r \left[\sum_{s, s''} o(a, s'', z) t(s, a, s'') b(s) \right] V_{n+1}(p^T, b_z^a) \quad [Eq. A.19]$$

Se puede demostrar [Cassandra 1998] que la función valor óptima para el caso de horizonte finito es PWLC (lineal por tramos y convexa) en un espacio $|S|-1$ dimensional. Esto significa que se puede usar un vector \mathbf{g} para representar un único segmento de la función valor y \mathbf{G} para representar el conjunto de vectores que componen la función valor. Entonces:

$$V(b) = \max_{g \in \Gamma} \sum_{s \in S} b(s) \cdot g(s) \quad [Eq. A.20]$$

Horizonte infinito

La determinación de la función valor mediante un proceso similar al realizado con la ecuación A.4 no es de gran ayuda en el caso del POMDP. El problema reside en que el espacio de estados es ahora continuo. En su lugar, se suele partir de las ecuaciones obtenidas para el caso de horizonte finito, dado que la función valor de la política óptima para el horizonte infinito puede ser aproximada tanto como se desee por la de horizonte finito considerando éste suficientemente largo [Sawaki 1978].

$V_n(\pi^T)$ es lineal por tramos y convexa. Además, se sabe que:

$$\lim_{T \rightarrow \infty} \|V(p^T, s) - V(p, s)\| = 0 \quad [Eq. A.21]$$

Pero esto no implica que $V(\cdot)$ sea lineal por tramos y convexa y de hecho, existen ejemplos que demuestran lo contrario [Sondik 1971]. No obstante, se puede aproximar por una función lineal por tramos.

A.2.4 Cálculo de la política óptima

En principio, dado que se ha convertido el POMDP en un COMDP, se podrían aplicar los algoritmos de iteración de la política e iteración del valor presentados anteriormente. A continuación se analiza la dificultad de aplicar dichos algoritmos y soluciones presentadas por distintos autores.

A.2.4.1 Iteración de la política

Para una política de horizonte infinito arbitraria se desconoce incluso si su función valor es representable de forma finita [Cassandra 1998]. Esto cuestiona la generalización del algoritmo de iteración de la política al caso POMDP. En la práctica, no existen algoritmos exactos de iteración de la política y se suele recurrir a la solución mediante métodos aproximados.

A.2.4.2 Iteración de la función valor

La generalización de este tipo de métodos para procesos POMDP consiste en aplicar el algoritmo A.1 para un espacio de estados continuo. En dicho algoritmo se puede observar que existen dos bucles que iteran sobre el tiempo y el espacio de estados. Esta última iteración se complica para el POMDP dado que, en la nueva situación, dicho espacio es continuo. De otra forma, el problema radica en el cálculo² de $V_m(\cdot)$ a partir de $V_{m-1}(\cdot)$. Es en este paso donde se han concentrado los esfuerzos en obtener soluciones exactas de este tipo de procesos. En lo que sigue, se analizará el caso de horizonte infinito.

La mayor parte de los algoritmos de iteración de la función valor utilizan la representación mostrada en la ecuación A.20. Suponen que la función valor se puede

² El índice de iteración m no es lo mismo que el índice temporal n . La relación entre ambos viene dada por: $m = T - n$. Utilizando el índice temporal se calcula $V_n(\cdot)$ a partir de $V_{n+1}(\cdot)$. No obstante, se ha utilizado esta notación utilizando la propiedad mostrada en la ecuación A.6.

representar mediante un conjunto de vectores \mathbf{G} que se actualiza a cada paso. De esta forma, la obtención de $V_m()$ a partir de $V_{m-1}()$ es lo mismo que obtener G_m a partir de G_{m-1} .

En líneas generales, el funcionamiento de estos algoritmos puede describirse de la siguiente forma:

Algoritmo A.4. Estructura de los algoritmos basados en iteración de la función valor

Iteración de la función Valor
 Calculo de los valores iniciales G_1
Para cada paso m
 Calcular G_m en función de G_{m-1}
Hasta que: $\sup_b |V_m(b) - V_{m-1}| < \epsilon$
Fin Iteración de la función Valor.

donde ϵ es el error mínimo permitido que determinará el número de iteraciones necesario para alcanzar la precisión requerida.

En cada paso de la iteración se calculan los vectores γ cuyo conjunto Γ_m define la función valor V_m .

Para un estado inicial b por cada acción se pueden obtener $|Z|$ posibles observaciones con lo cual, una política de T pasos p^T (ecuación A.19) tiene un número total de situaciones a contemplar igual a $(|Z|^T - 1) / (|Z| - 1)$. Como en cada situación se puede seleccionar cualquier acción, el número total de posibles políticas es:

$$|A|^{\frac{|Z|^T - 1}{|Z| - 1}}$$

El calculo exhaustivo de todas las políticas es inviable, incluso para procesos de horizonte finito con valores de T relativamente grandes. Sin embargo, no es necesario calcular todas las políticas posibles. Como el factor de ramificación es muy elevado, en cada paso se trata de encontrar el conjunto de vectores mínimo denominado *conjunto parco* (*parsimonious set*) que define completamente a Γ_m . El algoritmo se convierte entonces en el A.5.

Algoritmo A.5. Algoritmos basados en iteración de la función valor con métodos de poda

Iteración de la función Valor

Calculo de los valores iniciales G_1

Para cada paso m

 Calcular G'_m en función de G_{m-1}

 Calcular G_m aplicando métodos de poda sobre G'_m

Hasta que: $\sup_b |V_m(b) - V_{m-1}| < \epsilon$

Fin Iteración de la función Valor.

Entre las soluciones presentadas se encuentran el algoritmo de Monahan [Monahan 1982] [Zhang 1996], el algoritmo de testigos (witness algorithm) [Littman 1994] y algoritmo de poda incremental [Cassandra 1997] sobre los cuales se han desarrollado distintas modificaciones [Hauskrecht 1996].

En particular, aprovechando las propiedades de las funciones convexas lineales a trozos, algunos algoritmos como el de poda incremental y el de testigos obtienen Γ_n^a para cada una de las acciones. Detalles acerca de las propiedades, implementación y mejoras de estos algoritmos pueden verse en [Hauskrecht 1996].

Además de las soluciones exactas existen soluciones aproximadas basadas en técnicas de aprendizaje [Barto 1990] y otras específicas para determinado tipo de aplicaciones [Simmons 1995].

B Ejemplo sencillo de supervisión usando POMDP

B.1 Definición de la situación

En este ejemplo se presenta una versión simplificada de la solución estocástica al problema de supervisión y detección de excepciones en robots móviles. El objetivo de este ejemplo es obtener una idea de la complejidad que conllevan los distintos tipos de soluciones planteadas y su forma de funcionamiento.

Se supone que las situaciones excepcionales provienen solamente del entorno. Éste puede clasificarse como **NN** (No Navegable) o **NV** (Navegable) y las acciones que el robot puede ejecutar son: seguir el camino planeado **SC**, verificar el camino **CC**, o elegir un camino alternativo **CA**. Además, como consecuencia de **CC** pueden observarse que existe un bloqueo **BD** (Bloqueo Detectado) ó no observar nada **NO** (No hay Observación) que equivale en este caso a que no existe bloqueo **BN** (Bloqueo No observado). Para cualquiera de las demás acciones se supone que no hay observaciones (**NO** = **BN**). Los parámetros del sistema son por tanto:

- ✓ Estados: **NN**, **NV**.
- ✓ Acciones: **SC**, **CC**, **CA**.
- ✓ Observaciones: **BD**, **BN**.

Se establece un sistema de bonificación/penalización tal como el que se muestra en la tabla B.1:

Tabla B.1. bonificación/penalización de las acciones

Estados	Acciones		
	SC	CC	CA
NV	10	-10	-100
NN	-100	-10	10

En cuanto a las probabilidades de transición, se definen de la siguiente manera:

- ✓ La acción **CC** no cambia el estado del sistema porque es una simple acción de observación.
- ✓ Un nuevo paso del camino tiene probabilidad 0,1 de ser **NN** con lo cual la probabilidad de pasar de **NV** a **NN** ejecutando **SC** ó **CA** es 0,1, mientras que la probabilidad de pasar de **NN** a **NV** es cero para el caso de **SC** y 0,9 para **CA**.

Estas probabilidades se encuentran resumidas en la tabla B.2.

Tabla B.2. Transiciones entre estados

	Acciones					
	SC		CC		CA	
INICIAL \ FINAL	NV	NN	NV	NN	NV	NN
NV	0,9	0,1	1,0	0,0	0,9	0,1
NN	0,0	1,0	0,0	1,0	0,9	0,1

El problema consiste en determinar cuál es la mejor acción a ejecutar en cada momento, teniendo en cuenta las anteriores probabilidades y sistema de bonificación. Dicha tarea se puede abordar desde diferentes perspectivas:

Plan local (en línea). La primera solución consiste en calcular a cada paso una plan para la situación del sistema en ese momento de forma que se obtuviera la mayor bonificación estimada teniendo en cuenta las consecuencias futuras de tal decisión.

Plan global (fuera de línea). En este caso la solución consiste en calcular una política global para todas las situaciones posibles y, a cada paso, seleccionar la acción para el estado actual determinada por dicha política.

B.2 Plan local

Para determinar cuál es la mejor acción hay que tener en cuenta el efecto de dicha acción en el futuro, calculando la bonificación media esperada para todas las posibles opciones. Supóngase, en principio, que no hay incertidumbre en las

observaciones y que la distribución de probabilidades sobre estados es inicialmente 0,9 para el estado NV y 0,1 para el NN. Haciendo previsión a un solo paso en el futuro, se obtiene el árbol de la figura B.1.

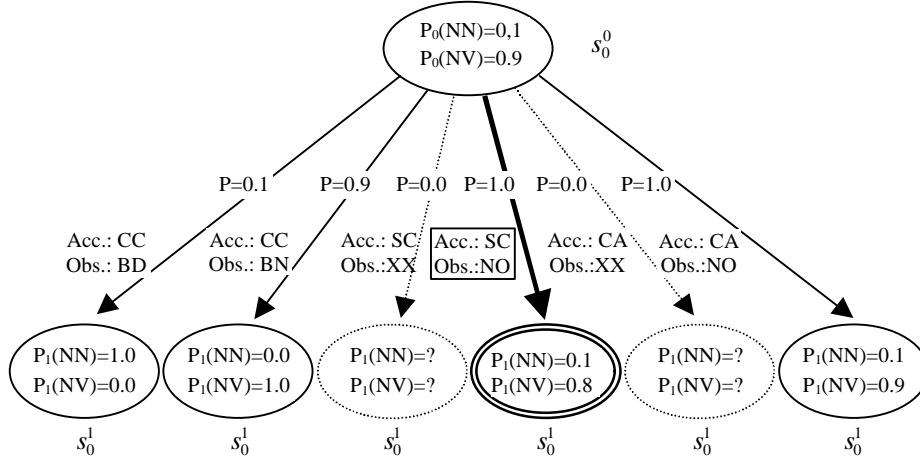


Figura B.1. Previsión de un paso en el futuro

Las distintas posibilidades (cada rama del árbol) vienen dadas por las posibles combinaciones de acción/observación. Cada una de estas combinaciones tiene una probabilidad asociada que es la probabilidad $p(\text{observación/acción})$. Las probabilidades de los estados a posteriori se han calculado de acuerdo con la ecuación 7.5 que, en este ejemplo, se puede simplificar bastante, al no haber incertidumbre en las observaciones. El estado después de ejecutarse CC viene dado por la observación obtenida mientras que, para las demás acciones se puede calcular a través de la ecuación B.1.

$$b^n(s') = p^n(s'/b^{n-1}, a) = \sum_s p(s'/a, s) b^{n-1}(s) \quad [Eq. B.1]$$

Las recompensas R para cualquier acción a se pueden calcular a partir de la tabla B.1 teniendo en cuenta la acción y la información de estado b , ya que el sistema

de bonificación es independiente de las observaciones y probabilidades de estado final (usando la Eq. B.2).

$$R_a^b = \sum_s b(s)R(a, s) \quad [Eq. B.2]$$

Obteniéndose una bonificación de -1 para **SC**, -10 para **CC** y -89 para **CA**, con lo cual la acción con mayor bonificación a un paso vista es **SC**. Por tanto, la acción a ejecutar si sólo se tiene en cuenta la bonificación inmediata sería **SC**, conclusión lógica porque la probabilidad de que el camino sea navegable es bastante alta.

Repitiendo el mismo proceso, se podría estimar la bonificación a n pasos en el futuro, creando un árbol de n pasos con la bonificación en las hojas. La acción a seleccionar en ese caso sería aquella con *bonificación esperada* mayor. La bonificación esperada para una acción a es la bonificación del mejor plan que comience con la acción a .

En la figura B.2 se muestra un ejemplo de la planificación con un horizonte de tres pasos. La probabilidad asociada a un arco que une los nodos correspondientes a dos informaciones de estado b^{n-1} y b^n es la probabilidad que, ejecutando la acción a correspondiente a ese arco, la información de estado final sea la correspondiente al extremo del arco (b^n). O lo que es lo mismo, la probabilidad de obtener la observación o correspondiente a ese arco cuando se ejecuta la acción a en el estado determinado por b^{n-1} (Eq. B.3).

$$\begin{aligned} p(o / b^{n-1}, a) &= \sum_s p(o, s / b^{n-1}, a) = \sum_s p(o / s, a, b^{n-1}) p(s / a, b^{n-1}) \\ &= \sum_s p(o / s, a) p(s / b^{n-1}) = \sum_s p(o / s, a) b^{n-1}(s) \end{aligned} \quad [Eq. B.3]$$

Los valores que aparecen en los nodos, dentro de las elipses, son las probabilidades de los estados **NN** y **NV**. En las hojas del árbol se muestra la recompensa **R** y, para aquellos nodos cuya acción en el último paso ha sido **CC**, se muestra también la probabilidad (**P**) que se haya observado **BN**.

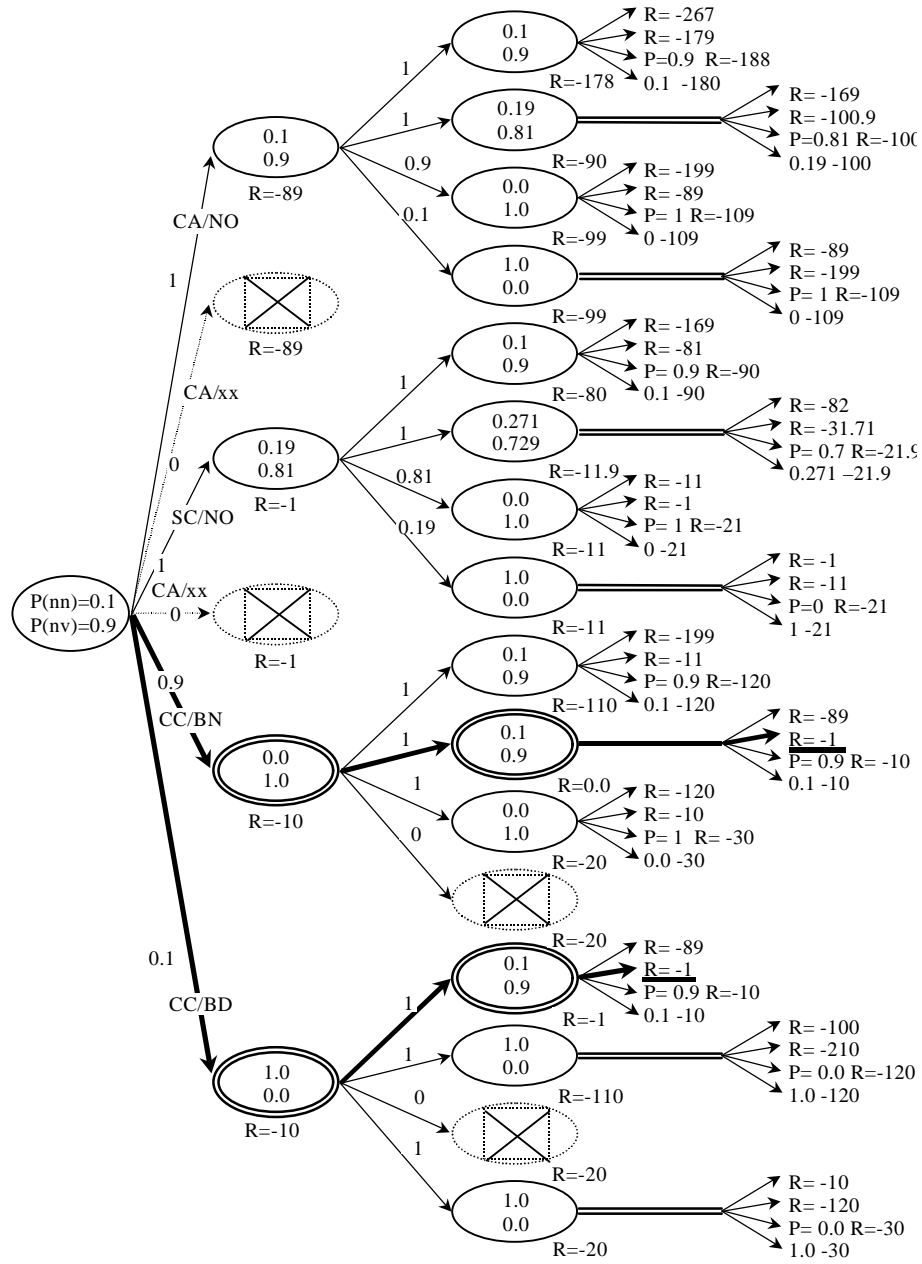


Figura B.2. Previsión de tres pasos en el futuro

Para decidir cuál es la mejor acción, es necesario tener en cuenta tanto el peso de los arcos, como la bonificación de las hojas del árbol. La *bonificación esperada* ($E(i)$) de un nodo i se puede calcular de forma recurrente a partir de la de sus descendientes:

$$E^i = \max_{a \in A} \left[\sum_o p(o / b, a) E_{a,o}(i) \right] \quad [Eq. B.4]$$

donde $E_{a,o}(i)$ es la bonificación esperada de la rama del nodo i asociada a la acción a y observación o . La bonificación esperada de las hojas E es igual a la bonificación R . A partir de ahí se calcula la bonificación esperada del resto de los nodos hasta llegar al raíz. La bonificación asociada a los nodos de la figura es la bonificación R . La acción a seleccionar se determinará de acuerdo a la bonificación esperada para las acciones de los nodos que cuelgan del raíz según la ecuación B.5:

$$a_{opt} = \arg \max_a \left[\sum_o p(o / b, a) E_{a,o}(raiz) \right] \quad [Eq. B.5]$$

De acuerdo con la figura B.2, el plan a dos pasos vista sería, en primer lugar, verificar el camino **CC** y el segundo depende de la observación del primero: ejecutar seguir el camino **SC** en caso que no se detectase ningún objeto bloqueando el camino (observación **BN**) o elegir un camino alternativo **CA** si se hubiese detectado algún objeto (observación **BD**). Por tanto, la acción a ejecutar en primera instancia es **CC**.

Siguiendo el mismo desarrollo, a tres pasos vista el plan sería el mismo que en el caso anterior para los dos primeros pasos. Mientras que, en el tercero, se ejecutaría la acción de seguir camino independientemente de lo que se haga en el segundo.

Una característica a tener en cuenta es el factor de ramificación. En un principio, el factor de ramificación es el producto de acciones por observaciones obteniendo un total de 6. Sin embargo, al no poder percibir todas las observaciones en todos los estados, se reduce a 4. En el modelo real presentado en el siguiente apéndice, el número de acciones es 7 y observaciones 64, con lo cual el factor de ramificación es 448. Por tanto, para un horizonte de n pasos, los nodos necesarios en el paso n serían 448^n , siendo inviable encontrar una solución dado que estos cálculos se realizan cada vez que es necesario tomar una nueva decisión (se trata de planificación en línea). La

situación empeora aun más si se tiene en cuenta que las observaciones no son deterministas como se describe en el siguiente apartado.

B.2.1 Incertidumbre en las observaciones.

Si ahora se considera un modelo más realista con observaciones no deterministas, el árbol obtenido se muestra en la figura B.3. También se ha añadido un factor de descuento γ (0,9) a la bonificación de las acciones para disminuir el efecto de las acciones que se realizan en un futuro muy lejano. El modelo así descrito es un POMDP de horizonte finito (n) con un factor de descuento γ .

Hay que resaltar que tanto este caso como el anterior se consideran parcialmente observables dado que el estado del sistema no es siempre conocido.

En la tabla B.3 se representan las probabilidades de observación para los distintos estados del robot. Estas probabilidades se refieren a la acción **CC** dado que en los demás casos no hay observaciones, esto es, se observará **NO** con probabilidad 1

Tabla B.3. Probabilidades de observación

Estados	Observaciones (sólo acción CC)	
	BN	BD
NV	0,9	0,1
NN	0,1	0,9

Para calcular las probabilidades a posteriori en este caso hay que tener en cuenta las probabilidades de observación y es necesario aplicar la ecuación 7.5.

Considerando un horizonte de tres pasos, en el primero la acción a ejecutar es **CC**. Si la observación indica que no se detectó ningún objeto bloqueando el camino (**BN**) se ejecutará la acción de seguir camino (**SC**) y en el tercer paso de nuevo (**SC**). Si, por el contrario, se observa algún obstáculo obstruyendo el camino (**BD**) al ejecutar la primera acción (**CC**), a continuación se selecciona la acción de verificar el camino. De nuevo pueden darse dos situaciones, si la observación es **BN** la siguiente acción a ejecutar será **SC**. Si la observación es **BD** se seleccionará un camino alternativo (**CA**).

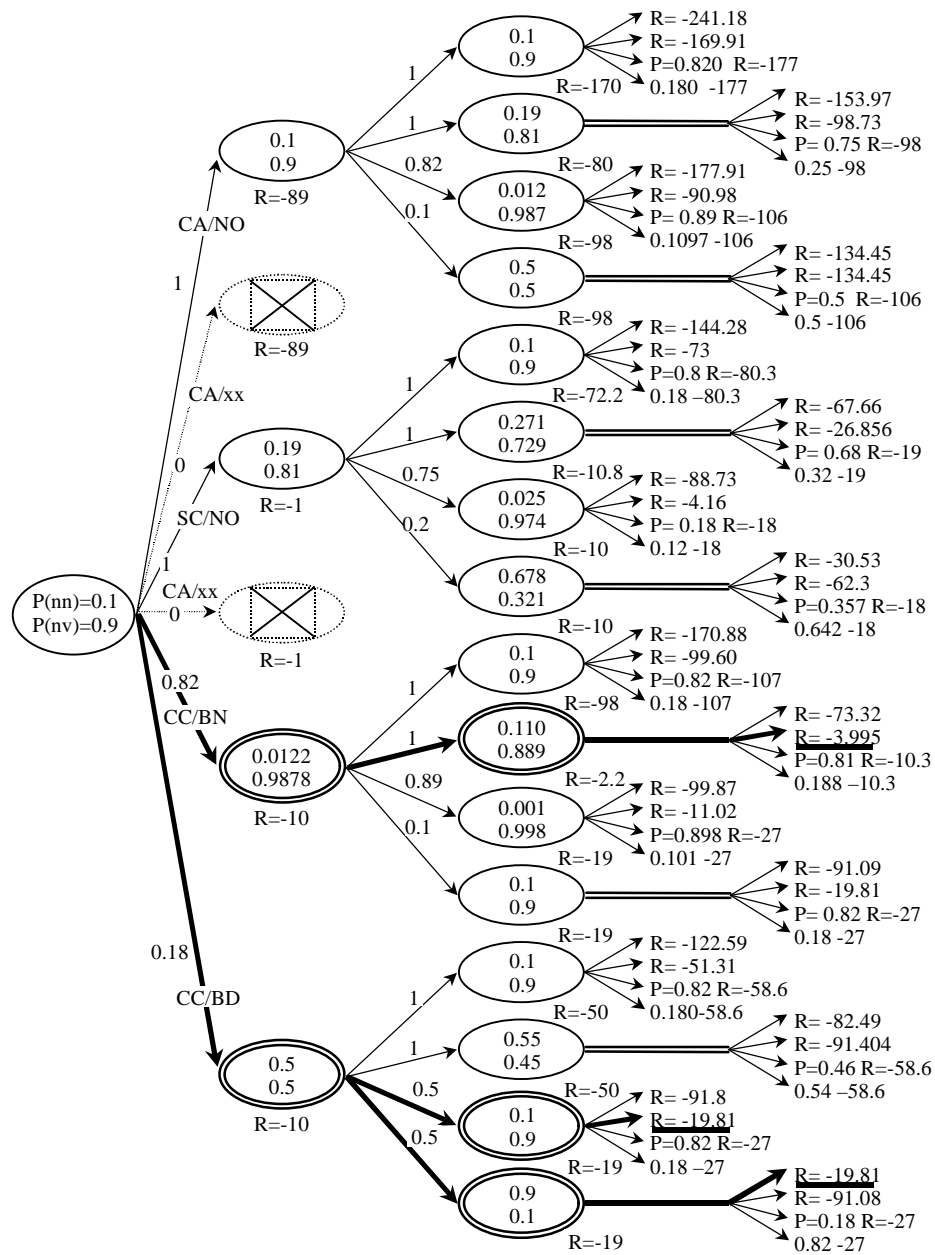


Figura B.3. Previsión de tres pasos en el futuro con incertidumbre en las observaciones

En este caso, el plan óptimo (figura B.3) difiere del anterior (figura B.2) principalmente porque el estado del sistema se hace completamente indeterminado si en el primer paso se observa un obstáculo (**BD**) al verificar el camino (**CC**). Otra causa de esta diferencia también hay que buscarla en el factor de descuento. El plan, en el segundo paso, es ejecutar de nuevo la acción de verificar el camino cuando la observación del primer paso sea obstáculo detectado (**BD**). Esto es debido a que, en el caso de observación determinista, el estado del sistema se conocía con certeza, fuese cual fuese la observación, pero ahora, al obtener una observación que viene de alguna forma a *contradecir* lo que el sistema cree, existe la incertidumbre de si falló la observación o es que la información de estado no es acorde con la realidad con lo cual decide hacer una segunda verificación.

Considerando un número suficiente de pasos, dado el factor de descuento, se llegaría a una solución óptima. En éste ultimo caso existe una mayor dispersión de estados al considerar el sistema parcialmente observable. El problema es que el factor de ramificación es exponencial con el número de acciones y observaciones como se ha mencionado antes, resultando muy costosa en tiempo, incluso para un modelo tan sencillo con sólo dos estados y dos observaciones. Además, hay que tener en cuenta que estos cálculos han de hacerse a cada paso, con lo cual para el modelo real se hace intratable un planificador de este tipo.

B.3 Plan global

Si el problema en el caso anterior era el cálculo del plan a cada paso y esto se hacía inabordable en tiempo real, en el segundo caso, se trata de encontrar una política fuera de línea (off line) de forma que, en tiempo de ejecución del sistema, el decisor sólo tenga que consultar esta política para tomar decisiones. El problema ahora se reduce a calcular una política que contemple todas las posibles situaciones.

Así, siguiendo con el mismo ejemplo usado en la política local, la política global depende de la información de estado que, en este caso, viene dada por la probabilidad que el camino sea no navegable al existir dos estados posibles. En un caso general, la información de estado se define sobre un espacio n -dimensional siendo $n+1$ el número de estados posibles (la suma de las probabilidades sobre todos los estados tiene que ser 1).

La política a un paso vista puede entonces representarse mediante intervalos como se indica en la tabla B.4.

Tabla B.4. Política a seguir en el primer paso para obtener máxima ganancia en dicho paso

Estado	S_1	S_2	S_3
Acción	Seguir Camino	Verificar Camino	Camino Alternativo
Intervalo $[P(NN)]$	$[0,0 \ 0,1818]$	$[0,1818 \ 0,8182]$	$[0,8182 \ 1,0]$

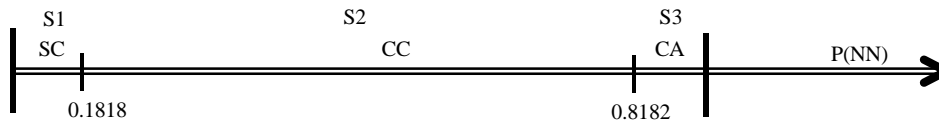


Figura B.4. Política a seguir en el primer paso para obtener máxima ganancia en dicho paso

En la figura B.4 puede observarse que si la probabilidad de camino no navegable es menor que 0,18 la acción a ejecutar es seguir camino. Si está entre 0,18 y 0,81 verificar el camino y si es superior a 0,81 elegir un camino alternativo. Para el caso de planificación a dos pasos, la política óptima que determina la acción a ejecutar en el primer paso viene dada por la tabla y figura B.5.

Tabla B.5. Política a seguir en el primer paso para obtener máxima ganancia en los dos siguientes pasos

Estado	S_1	S_2	S_3	S_4	S_5
Acción	Seguir Camino	Verificar Camino	Verificar Camino	Verificar Camino	Camino Alternativo
Intervalo $[P(NN)]$	$[0,0 \ 0,0604]$	$[0,017 \ 0,333]$	$[0,333 \ 0,666]$	$[0,666 \ 0,867]$	$[0,867 \ 1,0]$

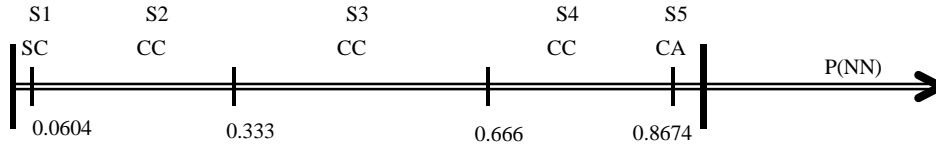


Figura B.5. Política a seguir en el primer paso para obtener máxima ganancia en los dos siguientes pasos

La política óptima no estacionaria para los dos pasos junto con los rangos de información de estado se puede entonces representar como en la figura B.6

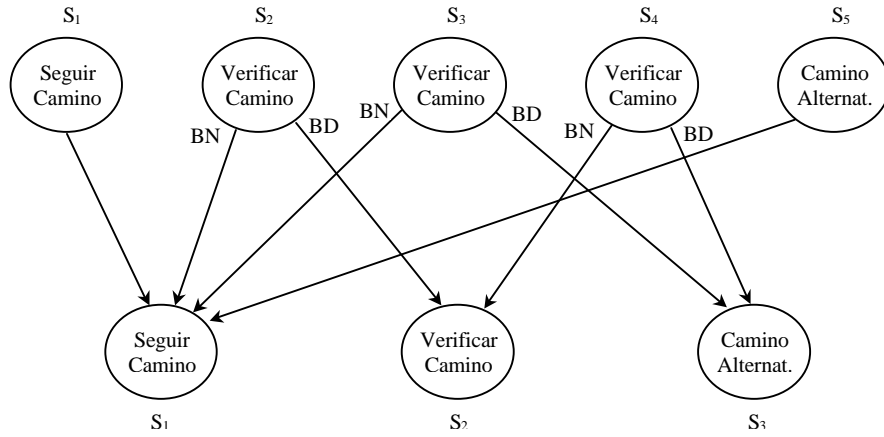


Figura B.6 . Política a seguir en los dos siguientes pasos para obtener máxima ganancia en los mismos

La función que determina la política a partir de las probabilidades de estado no es lineal pero sí lo es por trozos y, además, se puede demostrar que es convexa [Hauskrecht 1996], propiedades que son explotadas por la mayoría de los algoritmos de resolución de POMDP [Littman 1994].

Las acciones a ejecutar serán las mismas mientras la información de estado del sistema se encuentre dentro de la misma región. Dada la política, la acción a ejecutar en un determinado paso viene dada por la región en la que se encuentre la información de estado en ese momento. Por ejemplo, para una información de estado igual a la considerada en el apartado anterior ($P(NN) = 0,1$), la política a un paso vista establece que la acción a ejecutar en él es seguir camino (SC), pues la información de estado se encuentra en el intervalo S_1 .

Tabla B.6. Intervalos $[P(NN)]$ para los estados de los distintos pasos de la figura B.7

Paso		S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9
1	Desde	0,0	0,087	0,333	0,456	0,500	0,666	0,839	0,867	0,880
	Hasta	0,087	0,333	0,456	0,500	0,666	0,839	0,867	0,880	1,0
2	Desde	0,0	0,085	0,182	0,366	0,421	0,818	0,9		
	Hasta	0,085	0,182	0,366	0,421	0,818	0,9	1,0		
3	Desde	0,0	0,060	0,333	0,667	0,867				
	Hasta	0,060	0,333	0,667	0,867	1,0				
4	Desde	0,0	0,182	0,500						
	Hasta	0,182	0,500	1,0						

Para cuatro pasos vista, la política se muestra en la figura B.7, mientras que los intervalos de cada estado se presentan en la tabla B.6.

Añadiendo un factor de descuento para disminuir el efecto de futuras bonificaciones o penalizaciones, la estructura de la política óptima no estacionaria varía ligeramente. El resultado de la política a cinco pasos vista para este caso se muestra en la figura B.8.

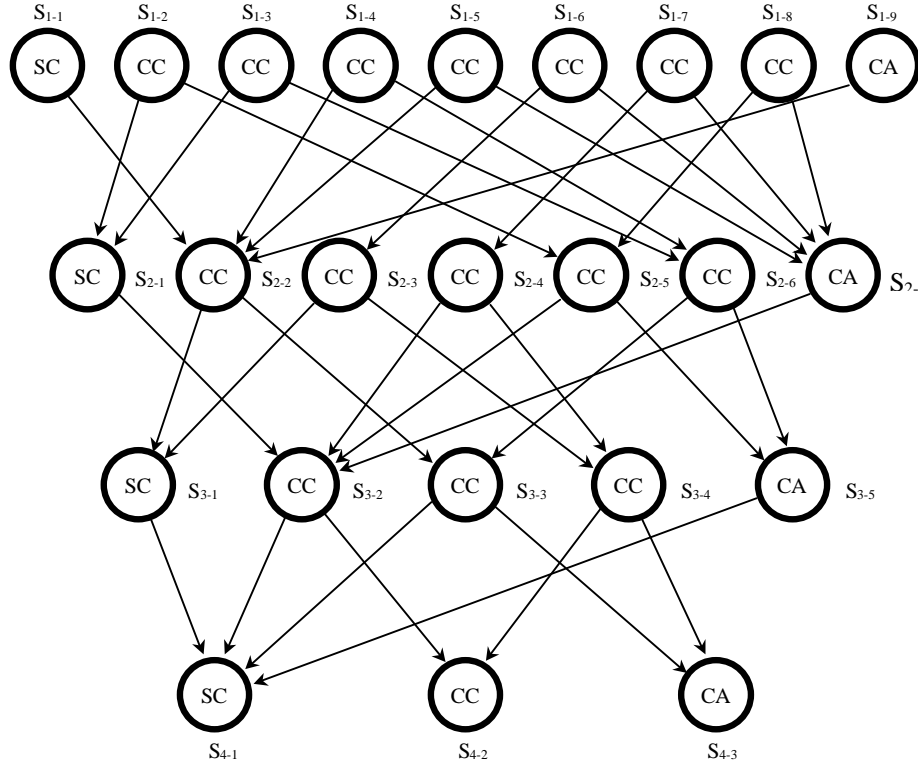


Figura B.7. Política para cuatro pasos en el futuro con bonificación máxima en los mismos

Considerando la información de estado inicial igual al apartado anterior ($P(\text{NN}) = 0,1$) y restringiéndose solamente a los tres últimos pasos, en el primero, dicha información de estado se encuentra en el $S_{3,2}$ y el plan a seguir es ejecutar **CC** en el primer paso. Si la observación indica que no se detectó ningún objeto bloqueando el camino (**BN**) se pasará a $S_{4,1}$ y se ejecutará la acción de seguir camino (**SC**), pasando a $S_{5,1}$ y ejecutando de nuevo seguir camino (**SC**). Si, por el contrario, se observa algún obstáculo obstruyendo el camino (**BD**) al ejecutar la primera acción (**CC**), se pasará a $S_{4,3}$ y se ejecutará la acción de verificar el camino. De nuevo pueden darse dos

situaciones, si la observación es **BN**, la siguiente información de estado estará en el intervalo S_{5-1} para ejecutar **SC**. Si la observación es **BD**, se pasa a S_{5-3} cuya acción asociada es seleccionar un camino alternativo (**CA**). Se puede comprobar que este plan coincide con el calculado en el apartado anterior.

Tabla B.7. Intervalos $[P(NN)]$ para los estados de los distintos pasos de la figura B.8

		S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}
Paso1	Desde	0,0	0,100	0,182	0,397	0,407	0,435	0,473	0,492	0,818	0,886
	Hasta	0,100	0,182	0,397	0,407	0,435	0,473	0,492	0,818	0,886	1,0
Paso2	Desde	0,0	0,090	0,097	0,333	0,475	0,489	0,666	0,855	0,861	0,873
	Hasta	0,090	0,097	0,333	0,475	0,489	0,666	0,855	0,861	0,873	1,0
Paso3	Desde	0,0	0,0914	0,182	0,396	0,407	0,818	0,896			
	Hasta	0,0914	0,182	0,396	0,408	0,818	0,896	1,0			
Paso4	Desde	0,0	0,068	0,333	0,666	0,861					
	Hasta	0,068	0,333	0,666	0,861	1,0					
Paso5	Desde	0,0	0,182	0,500							
	Hasta	0,182	0,500	1,0							

A medida que se considera un horizonte mayor, la bonificación de los pasos finales tiene menos efecto en la bonificación total. En muchos problemas POMDP con factor de descuento inferior a 1, la política óptima para n pasos futuros y para $n+1$ son bastante parecidos cuando n es muy grande (figura B.9). En dicha situación se puede calcular la política óptima usando la iteración por valor [Sondik 1971].

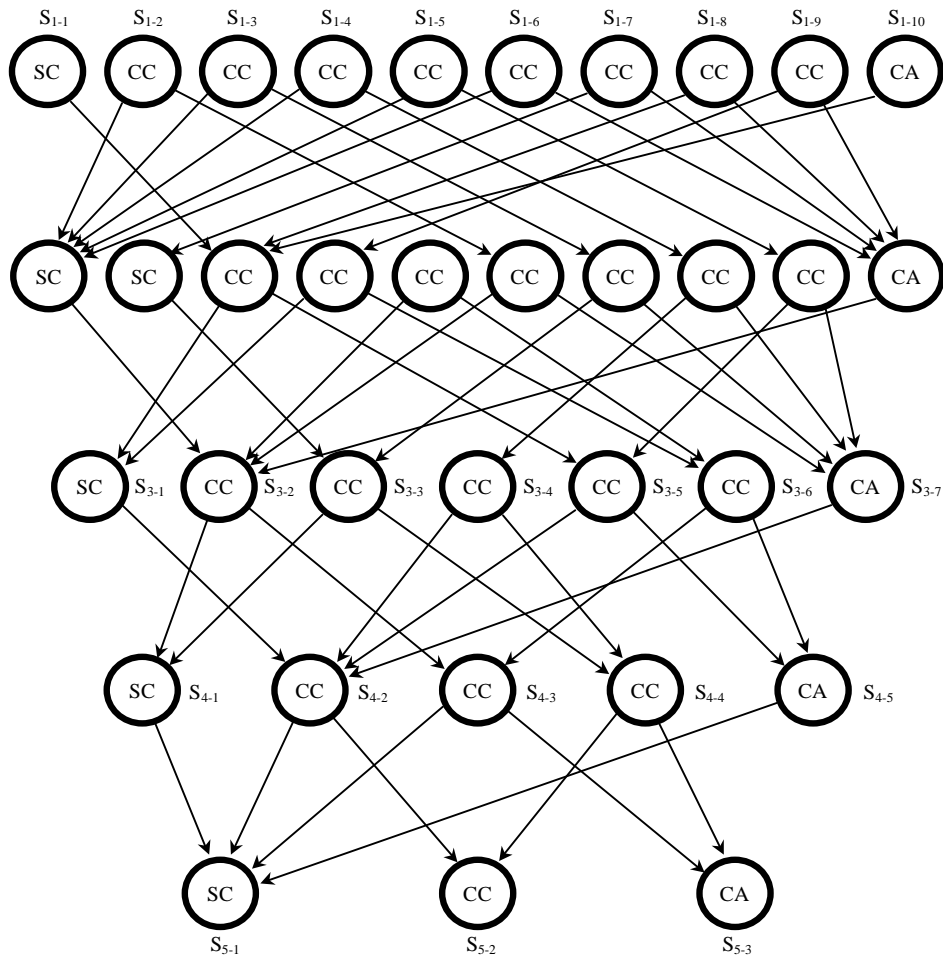


Figura B.8. Política para cinco pasos en el futuro con bonificación máxima en los mismos

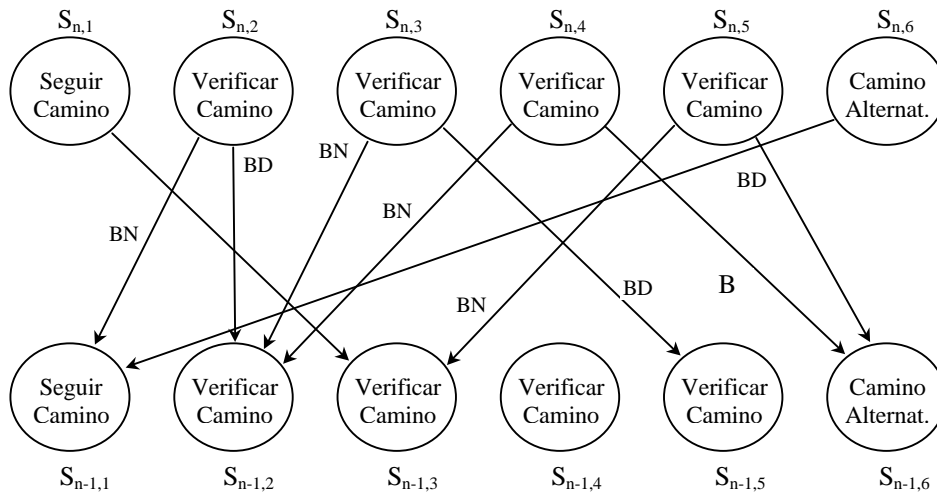


Figura B.9. Política para el paso n en el futuro

En la figura B.9 se puede observar la solución para este caso, mientras que en la tabla B.8 se presenta el rango de valores de información asociado a cada *estado*. En la figura B.10 se presenta el grafo correspondiente a dicho plan. Para ejecutar este plan, la información de estado inicial determina el nodo inicial y, a continuación, se ejecuta la acción asociada. A partir de la acción y observación, la política (fig B.10) determina cuál es el próximo estado (región de valores de información de estados) y se pasa a ejecutar la acción asociada. Este proceso se puede continuar de forma indefinida con la certeza que se está ejecutando la política óptima.

Tabla B.8

Estado		S_1	S_2	S_3	S_4	S_5	S_6
Intervalo P(NN)	Inicio	0,0	0,00636	0,103027	0,450820	0,5082	0,8807
	Fin	0,00636	0,103027	0,450820	0,5082	0,8807	1

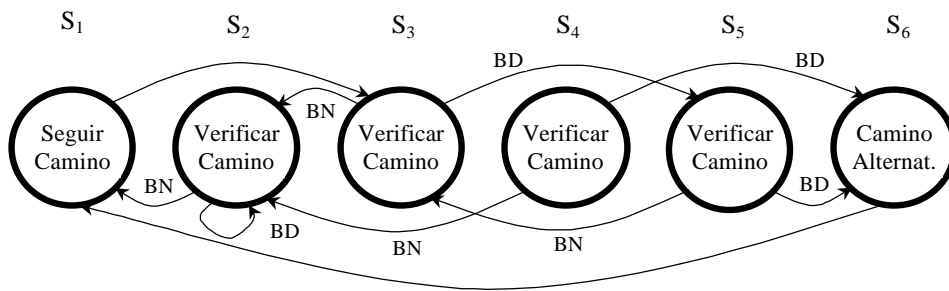


Figura B.10 política óptima de actuación

C Definición del modelo

En este apéndice se presenta el archivo de definición de los parámetros del modelo empleado en esta tesis. La sintaxis está basada en la usada por algunos autores para la descripción del modelo POMDP pero aquí se ha extendido con objeto de aumentar las posibilidades de expresión y hacer menos tediosa la tarea de especificación de parámetros para sistemas complejos.

C.1 Sintaxis del archivo de definición del modelo.

La descripción completa del POMDP para sistemas medianamente complejos implicaría la definición de un gran número de probabilidades y bonificaciones que puede ser simplificada para determinados casos. La sintaxis empleada permite la descripción en detalle del sistema pero incluye además la posibilidad de utilizar expresiones generales o bien en función de componentes simplificando considerablemente la tarea de determinación de parámetros.

La división en componentes se puede usar para definir estados y observaciones. Así, es posible definir el estado del robot como la combinación de cero, uno o varias situaciones de excepción con lo cual se pueden usar éstas como componentes. La utilidad, como se verá más adelante, se fundamenta en la posibilidad de especificar datos aislados de componentes cuando éstos sean independientes del resto reduciendo el número de parámetros a definir.

Lo mismo ocurre con las observaciones. La obtenida en un período puede verse como la combinación de las obtenidas por cada monitor más los resultados de las acciones. Si dos de ellas son independientes se puede especificar sus probabilidades por separado y el programa calculará las conjuntas suponiendo independencia entre ellas.

C.1.1 Conceptos generales

Comentarios

Los comentarios comienzan con el carácter '#'. El texto que aparece desde ese carácter al final de la línea es considerado como comentario y no se tiene en cuenta.

Parámetros generales

Lo primero que se definen son tres parámetros:

components: <yes/no>

values: <reward/cost>

discount: <valor real>

Si se desea definir los estados y observaciones como combinación de componentes en lugar de indicarlos uno a uno es necesario poner en la línea de componentes (*yes*). En cuanto a la bonificación, se puede indicar un sistema de bonificación (*reward*) o bien penalización (*cost*). El factor de descuento es un valor real entre cero y uno que indica la atenuación de la ganancia con el tiempo (ver capítulo 7).

Estados, acciones y observaciones

A continuación se ha de especificar la lista de estados, acciones y observaciones de la siguiente forma:

states: <estado 1/componente 1> <estado 2/componente 2> [... <estado n/componente n>]

actions: <acción 1> <acción 2> [... <estado m>]

observations: <observación 1/componente 1> <observación 2/componente 1> [... <observación t/componente t>]

Los estados, al igual que las observaciones, se pueden enumerar todos o bien enunciar los componentes si anteriormente se especificó esta opción en los parámetros globales. En caso de usar componentes, los estados serán todas las combinaciones posibles de éstos.

Por compatibilidad con la sintaxis de Cassandra, cualquiera de estos tres campos se puede expresar mediante su cardinal. En caso de enumerar los elementos o componentes, éstos se han de separar por espacios o tabuladores. Por ejemplo:

actions: 3

actions: fp np rl

En el resto del archivo se puede hacer referencia a la acción *0*, *1* y *2* o también se puede usar *fp*, *np*, y *rl*.

C.1.2 Probabilidades de transición

Una vez definidos los estados, acciones y observaciones se definen las probabilidades de transición. Hay que distinguir dos posibilidades según se usen componentes o no.

En el primer caso (componentes), se expresará de la siguiente forma:

TC: <acción a>: <expresión de componentes 1> : <expresión de componentes 2> <probabilidad p>

Indicando que realizando la acción *a* cuando el estado del robot es el definido por la *expresión de componentes 1* pasará al estado indicado por la *expresión de componentes 2* con probabilidad *p*. La expresión de componentes puede ser de la forma:

<+/-><componente 1> & <+/-><componente 2> [& ... & <+/-><componente n>]

Cuando a un componente se le antepone el símbolo ‘-’ quiere decir que en ese estado el robot no presenta esa componente. Si no aparece ningún signo, por defecto es como si hubiera un ‘+’. Por ejemplo, para indicar que mientras el robot está realizando la acción de seguir camino (SC) puede ocurrir un problema en el módulo reactivo (RT) con probabilidad 0,05 se indicaría:

TC: SC : -RT : RT 0.05

Para el caso de que no se especifique el estado por componentes, se sigue usando la notación definida por Cassandra donde las transiciones se pueden especificar de tres formas:

T: <acción a> : <estado inicial i> : <estado final f> <probabilidad p>

Indicando que la probabilidad de pasar del estado *i* al estado *f* mientras se realiza la acción *a* es *p*.

También se puede indicar una fila para todos los estados finales posibles:

**T: <acción a> : <estado inicial i>
<probabilidad p₁> <probabilidad p₂> [... <probabilidad p_n>]**

Donde *p₁* es la probabilidad de pasar de *i* al estado *0* ejecutando la acción *a*, *p₂* es la probabilidad de pasar de *i* al estado *1* ejecutando la acción *a* y así sucesivamente.

En la tercera forma, se especifica una matriz de probabilidades para una acción:

T: <acción a>
<probabilidad p_{11} > <probabilidad p_{12} > [... <probabilidad p_{1n} >]
<probabilidad p_{21} > <probabilidad p_{22} > [... <probabilidad p_{2n} >]
...
<probabilidad p_{n1} > <probabilidad p_{n2} > [... <probabilidad p_{nn} >]

Donde p_{ij} es la probabilidad de que el sistema pase del estado i al j mientras ejecuta la acción a . Para esta matriz, se pueden especificar dos casos particulares:

identity
uniform

El primero sustituye a la matriz identidad indicando que el estado del sistema con esa acción no cambia y el segundo la matriz uniforme lo que implica que, sea cual sea el estado, el estado final puede ser cualquiera con igual probabilidad.

C.1.3 Probabilidades de observación

Al igual que en el apartado anterior, hay que distinguir dos situaciones dependiendo de si se hace uso de las componentes. La primera, usando componentes, es útil cuando las observaciones son combinaciones de varias y existe independencia entre ellas. La sintaxis es:

OC: <acción a>: <componentes de estado e_1 > : < componente de observación o_1 > <probabilidad p >

Donde las componentes de estado se pueden expresar como en el caso de las transiciones pudiéndose además incluir el ‘*’ para indicar todos los estados posibles. Esta posibilidad existe también para indicar las acciones y observaciones.

Es importante hacer notar que, en la definición a través de los componentes, los casos particulares se deben definir después de los generales dado que éstos se van sobrescribiendo. Por ejemplo, para indicar que por defecto la acción FP (seguir camino) no tiene observaciones de tipo EPS (error de posición) a menos que el robot se encuentre en un entorno no navegable (NNP) en cuyo caso se disparará dicho monitor con probabilidad 0,95, se utilizará:

OC: FP : * : EPS 0.0
OC: FP: NNP : EPS 0.95

El estado al que se refieren las observaciones es siempre el estado final.

Para el segundo caso de indicar las observaciones de forma individual, la sintaxis no varía con respecto a la utilizada en los POMDP:

O : <acción a_1 > : <estado e_1 > : <observación o_1 > <probabilidad>

Se puede utilizar el carácter * en cualquier posición.

La segunda forma consiste en indicar todas las probabilidades para una acción y un estado:

O : <acción a_1 > : <estado e_1 >
<probabilidad p_1 > <probabilidad p_2 > [... <probabilidad p_i >]

Donde p_1 es la probabilidad de obtener la observación o_1 , p_2 de obtener la observación o_2 y así sucesivamente.

Por último, se puede utilizar la forma matricial:

O : <acción a_1 >
<probabilidad p_{11} > <probabilidad p_{12} > [... <probabilidad p_{1i} >]
<probabilidad p_{21} > <probabilidad p_{22} > [... <probabilidad p_{2i} >]
...
<probabilidad p_{n1} > <probabilidad p_{n2} > [... <probabilidad p_{ni} >]

Siendo p_{ij} la probabilidad de observar o_j al ejecutar a_1 para un estado final s_i . Se puede usar 'uniform' para indicar la matriz uniforme.

C.1.4 Bonificación

También aquí se ha extendido la sintaxis para el caso de la especificación de estados y observaciones a través de sus componentes:

RC: <acción a >: <componentes de estado e_1 > : < componente de estado e_2 > <valor v >

Donde las componentes de estado se pueden expresar como en los casos anteriores incluyendo el asterisco. Esta posibilidad del asterisco existe también para las acciones. El valor v será considerado como una bonificación si el parámetro *values* del comienzo del archivo es *reward*. Si, por el contrario, dicho parámetro es *cost* se interpretará como penalización o coste.

De nuevo, los casos particulares se deben definir después de los generales. Por ejemplo, para indicar que por defecto todas las acciones tienen valor 0 se puede expresar:

$$RC: * : * : * : 0$$

A continuación se puede definir un coste de 5 para la acción *fp*:

$$RC: fp : * : * : -5$$

Se ha puesto el signo menos porque se supone que al campo *values* se ha establecido como *reward*.

Más tarde se puede indicar que la acción *fp* cuando no existe ningún problema debe recibir una bonificación independientemente del estado final:

$$RC: fp : -NNP \& -RD \& -PP \& -HP \& -WD : * : 30$$

Por otra parte, también se pueden expresar los valores siguiendo la nomenclatura de los modelos POMDP:

$$R : \langle \text{acción } a_1 \rangle : \langle \text{estado inicial } i_1 \rangle : \langle \text{estado final } f_1 \rangle : \langle \text{observación } o_1 \rangle \langle \text{valor} \rangle$$

Se puede utilizar el carácter * en cualquier posición (estados y acciones).

La forma vectorial define los valores para las distintas acciones:

$$R : \langle \text{acción } a_1 \rangle : \langle \text{estado inicial } i_1 \rangle : \langle \text{estado final } f_1 \rangle \\ \langle \text{valor } v_1 \rangle \langle \text{valor } v_2 \rangle [\dots \langle \text{valor } v_t \rangle]$$

Donde v_1 es el valor de ejecutar la acción a_1 cuando el estado inicial es i_1 , el final será f_1 y la observación la o_1 .

En la forma matricial cada fila se refiere a un estado final y cada columna a una observación distinta:

$$R : \langle \text{acción } a_1 \rangle : \langle \text{estado inicial } i_1 \rangle \\ \langle \text{valor } v_{11} \rangle \langle \text{valor } v_{12} \rangle [\dots \langle \text{valor } v_{1t} \rangle] \\ \langle \text{valor } v_{21} \rangle \langle \text{valor } v_{22} \rangle [\dots \langle \text{valor } v_{2t} \rangle] \\ \dots \\ \langle \text{valor } v_{n1} \rangle \langle \text{valor } v_{n2} \rangle [\dots \langle \text{valor } v_{nt} \rangle]$$

Siendo v_{ij} el valor (penalización/coste) de ejecutar a_1 cuando el estado inicial es i_1 , el estado final i y se obtiene la observación j .

C.2 Listado del archivo

En lo que sigue, se lista el contenido del archivo de descripción del modelo usado en las pruebas presentadas en el capítulo séptimo:

```
# Error detection and recovery in a mobile robot
# Version to try to keep the robot doing a nominal action FP.
#
# NOTE: THE PROBABILITIES THAT A PROBLEM WILL HAPPEN ARE INCREASED
#       IN ORDER TO TEST THE SYSTEM.
# The format of this file should be full compatible with
# Tony Cassandra's format for POMDP. We added new features to the
# language so that now it's possible to describe the states as a
# combination of independent components adding:
# components: yes
# In that case we describe the transition states as combination of the
# states of the different components considered independent.
# We also express the observation probabilities depending on the
# components.
# Without this option or setting the option not (components: not)
# the format still be the same as described in Cassandra's POMDP page.
#
# ***** basic description *****
# Number of states: 128
# Number of actions: 7
# Number of observations: 64
#
# Error States: Combination of all the possible components:
#   NNP: Path Non Navigable -- ENVIRONMENT
#   NNA: Non Navigable Alternative. -- ENVIRONMENT
#   RT: Reactive problems.(basically spinning) -- SOFTWARE (REACTIVE)
#   WD: Wrong Direction. -- SOFTWARE (NAVEGATION)
#   PP: Perception Problems. -- SOFTWARE (PERCEPTION)
#   HP: Hardware and other problems. -- HARDWARE, etc
#   LST: Lost.
#
#
# Actions: numbered from 0 to 6
#   0: FP - Follow Path -- Follow the planned path.
#   1: GO - Go trough an Opening.
#   2: NP - New Path -- Follow an alternative path.
#   3: MA - Move Away -- Move away from the actual position.
#   4: CA - Call Assistance -- When a hardware problem is detected
#   5: GU - Give Up -- Give up this task. Goal unreachable.
#   6: RL - Relocalize -- Try to relocalize the robot.
#
#
# Observations: Combination of all the possible components:
#   0: EPS - Error position (robot not moving)
```

```
#      1: EPM - Error position (robot moving)
#      2: LD - Loop Detected
#      3: SD - Spinning Detected
#      4: BO - Blockage overtaken (when a go action succeed)
#      5: ALT - Alternative path (every time a path is requested)
#
#
discount: 0.80
values: reward
components: yes
states: nnp nna rt wd pp hp lst
actions: fp go np ma ca gu rl
observations: eps epm ld sd bo alt
#
# %%%%%%%%%%%
#
#
# TRANSITIONS
#
# %%%%%%%%%%%
# Format:
# TC: <action> : <states from components> : <states from components> prob
# tipical example:
# TC: <action> : -component : component => prob that component will show
#                                     up doing action.
# TC: <action> : component : component => prob that component will persists
#                                     doing action.
#
# Transitions for the state NNP (Non Navigable)
#
# A blocked path happens with 1% while navigating and a fp action
# never solves the problem.
TC: fp : -nnp : nnp 0.01
TC: fp : nnp : nnp 1.0
#
# A blocked path happens with 1% while navigating and a go action
# never solves the problem.
TC: go : -nnp : nnp 0.01
TC: go : nnp : nnp 1.0
#
# A blocked path executing np depends mostly if it was an alternative.
TC: np : -nna : nnp 0.05 #even that there is an alternative path, the
                        #planner could give me an non navigable one
TC: np : nna : nnp 0.89 # If there isn't an alternative, could try again....
TC: np : nna & nnp : nnp 1.00
TC: np : nna & pp : nnp 1.00
#
# A blocked path happens with 1% while ma action and a ma action
# never solves the problem.
TC: ma : -nnp : nnp 0.01
TC: ma : nnp : nnp 1.0
#
# Call Assistance has nothing to do with the environment
TC: ca : -nnp : nnp 0.0
```

TC: ca : nnp : nnp 1.0

give up means start a new task. nnp will keep if the new path is the same
TC: gu : -nnp : nnp 0.0
TC: gu : nnp : nnp 0.50

Relocalize has nothing to do with the environment.
TC: rl : -nnp : nnp 0.0
TC: rl : nnp : nnp 1.0

Transitions for NNA (Non navigable alternative path)

TC: fp : -nna : nna 0.0
TC: fp : nna : nna 0.90 # The alternatives increase going ahead.
TC: fp : nna & nnp : nna 1.0 # Not if it can't go ahead.

TC: go : -nna : nna 0.0
TC: go : nna : nna 1.0

TC: np : -nnp : nna 0.0
TC: np : -nnp & -nna : nna 0.5 # it depends on the alternative/s
TC: np : -nnp & nna : nna 1.0 # To avoid execute np twice in a row.
TC: np : nnp & -nna : nna 0.5 # it depends on the alternative/s.
TC: np : nnp & nna : nna 1.0

TC: ma : -nna : nna 0.0
TC: ma : nna : nna 1.0

TC: ca : -nna : nna 0.0
TC: ca : nna : nna 1.0

#I assume that gu "RESETS" everything
TC: gu : -nna : nna 0.1
TC: gu : nna : nna 0.1

TC: rl : -nna : nna 0.0
TC: rl : nna : nna 1.0

Transitions for the state PP (Perception Problems)
PP is only interesting with other problems.
A PP problem happens rarely while nav.
fp never solves the problem
TC: fp : -pp : pp 0.05
TC: fp : pp : pp 0.95

A PP problem happens rarely while executing go.
So far we are only interested in the effect with nnp.
TC: go : -pp : pp 0.05
TC: go : pp & nnp : pp 0.99
TC: go : pp & -nnp : pp 0.1

TC: np : -pp : pp 0.1
TC: np : pp : pp 0.2

```
#Looking from another point could solve the problem
TC: ma : -pp : pp 0.05
TC: ma : pp : pp 0.9

TC: ca : -pp : pp 0.0
TC: ca : pp : pp 0.80

# gu depends if the new path is going to be the same.
TC: gu : -pp : pp 0.05
TC: gu : pp : pp 0.2

TC: rl : -pp : pp 0.0
TC: rl : pp : pp 1.0

# Transitions for the state HP (Hardware Problems)

# A HP problem happens rarely while nav. fp never solves the problem
TC: fp : -hp : hp 0.005
TC: fp : hp : hp 1.0

# A HP problem happens rarely while go. fp never solves the problem
TC: go : -hp : hp 0.005
TC: go : hp : hp 1.0

# A HP problem happens rarely while np. fp never solves the problem
TC: np : -hp : hp 0.005
TC: np : hp : hp 1.0

# A HP problem happens rarely while ma. ma never solves the problem
TC: ma : -hp : hp 0.005
TC: ma : hp : hp 1.0

# ca never provoques a hp problem and Assistance always! solve the hp problem
TC: ca : -hp : hp 0.0
TC: ca : hp : hp 0.0

# give up means start a new task. Does not modify the hardware
# problems. Assumption of no hardware prob if the robot is still.
TC: gu : -hp : hp 0.0
TC: gu : hp : hp 1.0

# rl never solves a hardware problem but does not cause it either.
TC: rl : -hp : hp 0.0
TC: rl : hp : hp 1.0

# Transitions for RT (reactive problems)
TC: fp : -rt : rt 0.01
TC: fp : rt : rt 0.95

#These problems happen more often when going through narrow spaces.
#TC: go : -rt : rt 0.02 El
TC: go : -rt : rt 0.15
```



```

TC: go : rt : rt 0.95

#Restart and probably new direction
TC: np : -rt : rt 0.02
TC: np : rt : rt 0.45

# ma usually solves the problem because moves away from the problematic
# area
TC: ma : -rt : rt 0.01
TC: ma : rt : rt 0.3

#
TC: ca : -rt : rt 0.001
TC: ca : rt : rt 0.25

# Restart and probably new direction
TC: gu : -rt : rt 0.02
TC: gu : rt : rt 0.65

#
TC: rl : -rt : rt 0.0
TC: rl : rt : rt 1.0

# Transitions for WD (perception)

# Sometimes these problems fix themselves for example
# looking from another point of view.
TC: fp : -wd : wd 0.01
TC: fp : wd : wd 0.85

#reset
# TC: go : -wd : wd 0.01 El
TC: go : -wd : wd 0.10
TC: go : wd : wd 0.65

#reset and replan
TC: np : -wd : wd 0.01
TC: np : wd : wd 0.20

#reset
TC: ma : -wd : wd 0.01
TC: ma : wd : wd 0.65

TC: ca : -wd : wd 0.00
TC: ca : wd : wd 0.45

#reset and replan
TC: gu : -wd : wd 0.01
TC: gu : wd : wd 0.20

TC: rl : -wd : wd 0.00
TC: rl : wd : wd 1.00

```

```
# Transitions for LST (perception)

# Navigating can relocalize the robot. Because of the POMDP model
TC: fp : -lst : lst 0.01
TC: fp : lst : lst 0.85

# It is not too helpfull.
TC: go : -lst : lst 0.01
TC: go : lst : lst 0.97

# Same as fp
TC: np : -lst : lst 0.01
TC: np : lst : lst 0.85

# Moving could help a little bit.
TC: ma : -lst : lst 0.01
TC: ma : lst : lst 0.95

#The assistance is not for solve it but he can (probably).
TC: ca : -lst : lst 0.00
TC: ca : lst : lst 0.20

#Even Give Up is not able to solve it
TC: gu : -lst : lst 0.01
TC: gu : lst : lst 0.99

#Relocalize should make it
TC: rl : -lst : lst 0.1 # wrong location.
TC: rl : lst : lst 0.35 #

#####
#
# OBSERVATIONS
#
# #####

# Observations expressed in function of the observation components and
# the state components.
# O: <action> : nnp & hp & rt & wd : <observation component> %f
#

# For the action fp(follow path) only the monitors are observed.
# the monitors are eps, epm, ld, sd

OC: fp : * : * 0.0 # By default no observation.
OC: fp : +nnp : eps 0.50 # if it's non navegable...
OC: fp : -nnp & hp : eps 0.80 # Depends on the hardwr.
OC: fp : -nnp & -hp & +rt : eps 0.90 #
OC: fp : -nnp & -hp & -rt & wd : eps 0.05 #
OC: fp : -nnp & -hp & -rt & -wd & pp : eps 0.1 #
OC: fp : -nnp & -hp & -rt & -wd & -pp & lst : eps 0.05 #
OC: fp : -nnp & -hp & -rt & -wd & -pp & -lst : eps 0.01 #Everything ok
```

```

OC: fp : +nnp : epm 0.98 # if it's non navegable...
OC: fp : -nnp & hp : epm 0.40 # Depending on the hp failure...sens..mot
OC: fp : -nnp & -hp & +rt : epm 0.05 # I probably get eps
OC: fp : -nnp & -hp & -rt & wd : epm 0.90 #
OC: fp : -nnp & -hp & -rt & -wd & pp : epm 0.1 #
OC: fp : -nnp & -hp & -rt & -wd & -pp & lst : epm 0.05 #
OC: fp : -nnp & -hp & -rt & -wd & -pp & -lst : epm 0.01 # Everything is ok

OC: fp : nnp : ld 0.05 # if it's non navegable...
OC: fp : -nnp & hp : ld 0.05 # Depending on the hp failure...sens..mot
OC: fp : -nnp & -hp & +rt : ld 0.05 #
OC: fp : wd : ld 0.95 #
OC: fp : -nnp & -hp & -rt & -wd & lst : ld 0.05 #
OC: fp : -nnp & -hp & -rt & -wd & -lst : ld 0.01 # When everything is ok

OC: fp : +rt : sd 0.90 # rt is the main reason to spin
OC: fp : -rt : sd 0.01 # All the other problems....

# For the action np I don't get the monitors fired (because of the delay)

OC: np : * : * 0.0

OC: np : -nna : alt 0.90 # Could be that the alternative is non nav.
OC: np : nna : alt 0.0

# For the action go (go trough opening) only bo can be observed.
# Important: bo is the result of check path basically.!!
#
OC: go : * : * 0.0
OC: go : nnp & pp : bo 0.05 # if it's non navegable and has perception prob.
OC: go : nnp & -pp : bo 0.0 # if it's non navegable and no percep. prob.
OC: go : -nnp & pp : bo 0.95 # if it's navegable but has perception problems
OC: go : -nnp & -pp : bo 0.99 # if it's navegable...
# For the action move away no observation is done.

OC: ma : * : * 0.0

# For the action Call Assistance no observation is done.

OC: ca : * : * 0.0

# For the action Give Up no observation is done.

OC: gu : * : * 0.0

# For the action Relocalize no observation is done.

OC: rl : * : * 0.0

# %%%%%%%%%%%%%%
#
#
# REWARDS

```

```
#
# %%%%%%%%%%%

# R: <action> : <start_state> : <end_state> : <observation> %f
# RC: <action> : <start_state from components> : <end_state from components> %f

# By default, 0 reward
RC: *: *: *: 0

# these are regular actions cost
RC: fp : *: *: -5
RC: go : *: *: -10
RC: np : *: *: -
RC: ma : *: *: -5
RC: rl : *: *: -5

# Try to reach the state where everything is ok
RC: *: *: -nnp & -hp & -rt & -wd & -lst : 10
# Force fp action when ok.
RC: fp : -nnp & -hp & -rt & -wd & -pp & -lst: *: 30

# these are the actions cost. Very costly actions
RC: ca : *: *: -40      #I don't want to bother Assistance
RC: gu : *: *: -30      #I don't want to give up easily

# END OF FILE
```

D Solución al COMDP asociado al modelo.

En este apéndice se presenta la solución al proceso COMDP asociado al modelo del apéndice anterior. Dicha solución es utilizada por algunos algoritmos de decisión como el MLS y AV. Para cada estado se presenta la acción a ejecutar de acuerdo con dicha solución y la bonificación media esperada para cada acción. Esta bonificación es la obtenida de ejecutar dicha acción en el siguiente paso suponiendo que a partir de ahí se va a aplicar la política óptima y desaparecerá la incertidumbre acerca del estado tal y como establecen los procesos COMDP. Esta bonificación sirve al método Q-MDP y sus derivados para la toma de decisiones.

Los estados se representan por los componentes que forman parte de éste suponiendo que los demás problemas no existen en dicho estado.

En la primera columna aparecen los estados, en la segunda la acción con mayor bonificación y en el resto las bonificaciones para cada una de las acciones.

<u>Estados</u>	<u>A</u>	<u>fp</u>	<u>go</u>	<u>Np</u>	<u>ma</u>	<u>ca</u>	<u>gu</u>	<u>rl</u>
<i>nhh</i>	<i>fp</i>	136.720	106.720	107.432	116.057	59.350	76.498	115.365
<i>nnp</i>	<i>np</i>	79.937	76.397	107.432	79.937	35.927	64.047	78.003
<i>nna</i>	<i>fp</i>	131.987	100.514	34.450	110.798	55.564	76.498	111.685
<i>nnp_nna</i>	<i>gu</i>	45.577	43.785	25.379	45.577	1.227	64.047	44.259
<i>rt_</i>	<i>ma</i>	79.103	77.267	83.343	104.656	52.963	60.765	76.063
<i>nnp_rt</i>	<i>np</i>	61.997	61.353	83.343	74.403	31.128	50.407	59.267
<i>nna_rt</i>	<i>ma</i>	74.575	71.792	27.584	99.468	49.086	60.765	72.020
<i>nnp_nna_rt</i>	<i>gu</i>	35.418	35.196	20.741	42.443	-1.490	50.407	33.650
<i>wd</i>	<i>np</i>	77.017	83.293	94.685	86.312	44.224	70.153	68.088
<i>nnp_wd</i>	<i>np</i>	71.468	71.091	94.685	73.484	31.339	59.982	68.088
<i>nna_wd</i>	<i>ma</i>	63.250	71.090	32.325	73.406	34.484	70.153	51.785
<i>nnp_nna_wd</i>	<i>gu</i>	42.880	42.081	24.773	43.523	-0.236	59.982	41.097
<i>rt_wd</i>	<i>ma</i>	61.963	66.044	74.518	80.059	39.400	56.099	56.894
<i>nnp_rt_wd</i>	<i>np</i>	55.982	57.414	74.518	68.527	26.892	47.354	52.403
<i>nna_rt_wd</i>	<i>ma</i>	53.710	57.860	26.002	68.610	30.492	56.099	48.066
<i>nnp_nna_rt_wd</i>	<i>gu</i>	33.352	33.843	20.200	40.536	-2.863	47.354	31.275
<i>pp</i>	<i>np</i>	93.098	85.939	103.985	94.373	34.482	72.332	89.905

Estados	A	fp	go	Np	ma	ca	gu	rl
nnp_pp	np	77.474	73.926	103.985	77.610	33.307	61.766	75.321
nna_pp	gu	71.397	62.583	25.198	71.309	10.245	72.332	65.286
nnp_nna_pp	gu	43.948	42.166	25.198	44.038	-0.507	61.766	42.485
rt_pp	ma	65.115	63.462	80.514	85.587	30.681	57.699	61.199
nnp_rt_pp	np	59.948	59.242	80.514	72.197	28.625	48.698	57.067
nna_rt_pp	ma	50.275	46.633	20.580	64.747	8.485	57.699	44.956
nnp_nna_rt_pp	gu	34.173	33.911	20.580	41.016	-3.115	48.698	32.321
wd_pp	np	71.729	73.530	91.824	77.345	29.566	66.709	65.863
Nnp_wd_pp	np	69.356	68.851	91.824	71.411	28.920	58.068	65.863
nna_wd_pp	gu	53.276	52.677	24.598	56.452	7.266	66.709	46.493
nnp_nna_wd_pp	gu	41.471	40.604	24.598	42.142	-1.844	58.068	39.609
rt_wd_pp	np	55.168	57.158	72.115	71.379	25.732	53.466	50.651
nnp_rt_wd_pp	np	54.194	55.482	72.115	66.554	24.576	45.867	50.534
nna_rt_wd_pp	gu	41.531	41.364	20.044	52.284	5.149	53.466	36.152
nnp_nna_rt_wd_pp	gu	32.244	32.653	20.044	39.250	-4.374	45.867	30.119
hp	ca	41.101	39.374	16.660	41.101	59.350	15.850	42.078
nnp_hp	ca	23.515	22.653	16.660	23.515	35.927	5.970	23.271
nna_hp	ca	37.507	34.998	-16.472	37.108	55.564	15.850	39.066
nnp_nna_hp	gu	-0.360	-0.770	-20.452	-0.360	1.227	5.970	-0.557
rt_hp	ca	36.417	35.472	14.599	39.656	52.963	12.701	36.963
nnp_rt_hp	ca	19.915	19.610	14.599	22.404	31.128	3.242	19.453
nna_rt_hp	ca	32.813	31.138	-17.534	35.659	49.086	12.701	33.878
nnp_nna_rt_hp	gu	-2.407	-2.507	-21.386	-0.992	-1.490	3.242	-2.737
wd_hp	ca	31.356	33.084	14.239	33.676	44.224	13.549	29.967
nnp_wd_hp	ca	20.442	20.663	14.239	21.174	31.339	4.496	19.621
nna_wd_hp	ca	24.488	26.381	-16.989	26.912	34.484	13.549	22.236
nnp_nna_wd_hp	gu	-1.346	-1.410	-20.674	-1.111	-0.236	4.496	-1.736
rt_wd_hp	ca	27.602	29.701	12.292	32.450	39.400	10.551	26.113
nnp_rt_wd_hp	ca	17.064	17.743	12.292	20.115	26.892	1.858	16.083
nna_rt_wd_hp	ca	21.221	23.346	-18.030	25.812	30.492	10.551	19.041
nnp_nna_rt_wd_hp	gu	-3.336	-3.115	-21.602	-1.729	-2.863	1.858	-3.842
pp_hp	ca	23.457	21.882	14.093	24.437	34.482	12.670	22.294
nnp_pp_hp	ca	21.632	20.712	14.093	21.736	33.307	4.229	21.186
nna_pp_hp	gu	8.366	4.845	-20.591	8.379	10.245	12.670	4.839
nnp_nna_pp_hp	gu	-1.612	-2.057	-20.591	-1.542	-0.507	4.229	-1.943
rt_pp_hp	ca	20.495	19.425	12.131	23.494	30.681	9.723	19.235
nnp_rt_pp_hp	ca	18.111	17.738	12.131	20.649	28.625	1.605	17.462
nna_rt_pp_hp	gu	5.960	2.947	-21.522	7.616	8.485	9.723	2.478
nnp_nna_rt_pp_hp	gu	-3.588	-3.732	-21.522	-2.153	-3.115	1.605	-4.039
wd_pp_hp	ca	19.814	19.673	11.883	21.403	29.566	10.622	18.306
nnp_wd_pp_hp	ca	18.681	18.804	11.883	19.483	28.920	2.883	17.696
nna_wd_pp_hp	gu	6.198	3.873	-20.811	6.371	7.266	10.622	3.198
nnp_nna_wd_pp_hp	gu	-2.520	-2.645	-20.811	-2.237	-1.844	2.883	-3.020
rt_wd_pp_hp	ca	16.884	17.212	10.023	20.479	25.732	7.801	15.238
nnp_rt_wd_pp_hp	ca	15.374	15.950	10.023	18.446	24.576	0.338	14.241
nna_rt_wd_pp_hp	gu	3.934	2.029	-21.736	5.661	5.149	7.801	0.939
nnp_nna_rt_wd_pp_hp	gu	-4.446	-4.291	-21.736	-2.836	-4.374	0.338	-5.052

Estados	A	fp	go	Np	ma	ca	gu	rl
<i>lst</i>	<i>rl</i>	83.130	71.867	60.706	79.211	54.329	52.340	105.339
<i>nnp_lst</i>	<i>rl</i>	55.421	49.324	60.706	52.502	30.042	38.101	70.646
<i>nna_lst</i>	<i>rl</i>	79.519	67.511	14.593	75.346	50.753	52.340	101.921
<i>nnp_nna_lst</i>	<i>rl</i>	29.078	25.408	8.926	27.114	-2.730	38.101	39.312
<i>rt_lst</i>	<i>ma</i>	56.510	51.904	47.864	71.377	47.866	35.599	69.407
<i>nnp_rt_lst</i>	<i>rl</i>	41.675	38.299	47.864	48.416	25.510	25.533	53.248
<i>nna_rt_lst</i>	<i>ma</i>	52.982	47.894	9.980	67.537	44.201	35.599	65.634
<i>nnp_nna_rt_lst</i>	<i>rl</i>	21.294	19.114	5.373	24.800	-5.297	25.533	29.460
<i>wd_lst</i>	<i>rl</i>	53.211	54.666	53.257	57.242	39.069	45.788	61.439
<i>nnp_wd_lst</i>	<i>rl</i>	48.930	45.435	53.257	47.736	25.709	34.176	61.439
<i>nna_wd_lst</i>	<i>ma</i>	44.194	47.059	13.254	49.167	30.093	45.788	46.938
<i>nnp_nna_wd_lst</i>	<i>rl</i>	27.011	24.158	8.462	25.596	-4.112	34.176	36.376
<i>rt_wd_lst</i>	<i>ma</i>	43.109	43.479	42.402	53.141	34.317	31.477	51.510
<i>nnp_rt_wd_lst</i>	<i>rl</i>	37.065	35.412	42.402	44.076	21.509	22.949	46.874
<i>nna_rt_wd_lst</i>	<i>ma</i>	37.293	37.955	8.933	45.836	26.086	31.477	43.511
<i>nnp_nna_rt_wd_lst</i>	<i>rl</i>	19.710	18.122	4.958	23.391	-6.593	22.949	27.256
<i>pp_lst</i>	<i>rl</i>	65.525	56.578	58.062	63.185	30.842	49.310	81.697
<i>nnp_pp_lst</i>	<i>rl</i>	53.532	47.512	58.062	50.783	27.568	36.442	68.156
<i>nna_pp_lst</i>	<i>rl</i>	48.894	39.535	8.788	46.151	7.951	49.310	58.837
<i>nnp_nna_pp_lst</i>	<i>rl</i>	27.829	24.221	8.788	25.977	-4.367	36.442	37.664
<i>rt_pp_lst</i>	<i>ma</i>	45.553	41.477	45.678	57.208	26.807	33.336	55.523
<i>nnp_rt_pp_lst</i>	<i>rl</i>	40.104	36.751	45.678	46.785	23.146	24.274	51.205
<i>nna_rt_pp_lst</i>	<i>ma</i>	33.938	28.864	5.249	41.729	5.828	33.336	40.353
<i>nnp_nna_rt_pp_lst</i>	<i>rl</i>	20.339	18.171	5.249	23.745	-6.831	24.274	28.227
<i>wd_pp_lst</i>	<i>rl</i>	49.145	47.428	51.056	50.605	25.225	43.281	59.373
<i>nnp_wd_pp_lst</i>	<i>rl</i>	47.311	43.792	51.056	46.204	23.424	32.783	59.373
<i>nna_wd_pp_lst</i>	<i>gu</i>	36.051	32.983	8.328	36.193	4.315	43.281	41.807
<i>nnp_nna_wd_pp_lst</i>	<i>rl</i>	25.931	23.075	8.328	24.576	-5.630	32.783	34.994
<i>rt_wd_pp_lst</i>	<i>ma</i>	37.470	36.506	40.536	46.596	21.386	29.521	45.547
<i>nnp_rt_wd_pp_lst</i>	<i>rl</i>	35.694	33.995	40.536	42.617	19.322	21.847	45.139
<i>nna_rt_wd_pp_lst</i>	<i>ma</i>	27.087	24.900	4.838	33.203	2.053	29.521	32.099
<i>nnp_nna_rt_wd_pp_lst</i>	<i>rl</i>	18.860	17.248	4.838	22.440	-8.020	21.847	26.182
<i>hp_lst</i>	<i>ca</i>	37.766	35.544	13.432	37.368	54.329	11.980	41.074
<i>nnp_hp_lst</i>	<i>ca</i>	19.568	18.189	13.432	19.099	30.042	1.801	22.094
<i>nna_hp_lst</i>	<i>ca</i>	34.303	31.373	-19.306	33.538	50.753	11.980	38.104
<i>nnp_nna_hp_lst</i>	<i>gu</i>	-3.159	-3.963	-23.247	-3.492	-2.730	1.801	-1.391
<i>rt_hp_lst</i>	<i>ca</i>	33.032	31.602	11.351	35.906	47.866	8.792	35.943
<i>nnp_rt_hp_lst</i>	<i>ca</i>	16.137	15.308	11.351	18.046	25.510	-0.887	18.329
<i>nna_rt_hp_lst</i>	<i>ca</i>	29.563	27.472	-20.361	32.074	44.201	8.792	32.901
<i>nnp_nna_rt_hp_lst</i>	<i>gu</i>	-5.180	-5.675	-24.170	-4.114	-5.297	-0.887	-3.563
<i>wd_hp_lst</i>	<i>ca</i>	27.932	29.197	11.024	29.868	39.069	9.664	28.936
<i>nnp_wd_hp_lst</i>	<i>ca</i>	16.640	16.305	11.024	16.880	25.709	0.337	18.495
<i>nna_wd_hp_lst</i>	<i>ca</i>	21.475	22.925	-19.817	23.532	30.093	9.664	21.358
<i>nnp_nna_wd_hp_lst</i>	<i>gu</i>	-4.139	-4.598	-23.467	-4.238	-4.112	0.337	-2.568
<i>rt_wd_hp_lst</i>	<i>ca</i>	24.208	25.824	9.062	28.645	34.317	6.640	25.097

Estados	A	fp	go	Np	ma	ca	gu	rl
<i>nnp_rt_wd_hp_lst</i>	<i>ca</i>	13.419	13.541	9.062	15.877	21.509	-2.255	15.006
<i>nna_rt_wd_hp_lst</i>	<i>ca</i>	18.198	19.869	-20.850	22.424	26.086	6.640	18.160
<i>nnp_nna_rt_wd_hp_lst</i>	<i>gu</i>	-6.100	-6.277	-24.384	-4.847	-6.593	-2.255	-4.665
<i>pp_hp_lst</i>	<i>ca</i>	20.945	18.983	10.944	21.575	30.842	8.959	21.566
<i>nnp_pp_hp_lst</i>	<i>ca</i>	17.773	16.352	10.944	17.413	27.568	0.148	20.038
<i>nna_pp_hp_lst</i>	<i>gu</i>	5.835	1.976	-23.380	5.503	7.951	8.959	4.097
<i>nnp_nna_pp_hp_lst</i>	<i>gu</i>	-4.357	-5.188	-23.380	-4.618	-4.367	0.148	-2.760
<i>rt_pp_hp_lst</i>	<i>ca</i>	17.844	16.401	8.961	20.586	26.807	5.962	18.460
<i>nnp_rt_pp_hp_lst</i>	<i>ca</i>	14.417	13.537	8.961	16.383	23.146	-2.441	16.366
<i>nna_rt_pp_hp_lst</i>	<i>gu</i>	3.386	0.049	-24.299	4.729	5.828	5.962	1.725
<i>nnp_nna_rt_pp_hp_lst</i>	<i>gu</i>	-6.311	-6.841	-24.299	-5.221	-6.831	-2.441	-4.849
<i>wd_pp_hp_lst</i>	<i>ca</i>	16.928	16.499	8.740	18.236	25.225	6.883	17.438
<i>nnp_wd_pp_hp_lst</i>	<i>ca</i>	14.960	14.546	8.740	15.277	23.424	-1.195	16.597
<i>nna_wd_pp_hp_lst</i>	<i>gu</i>	3.627	0.996	-23.598	3.503	4.315	6.883	2.450
<i>nnp_nna_wd_pp_hp_lst</i>	<i>gu</i>	-5.263	-5.774	-23.598	-5.311	-5.630	-1.195	-3.835
<i>rt_wd_pp_hp_lst</i>	<i>ca</i>	13.978	13.988	6.864	17.296	21.386	4.026	14.369
<i>nnp_rt_wd_pp_hp_lst</i>	<i>ca</i>	11.808	11.843	6.864	14.295	19.322	-3.699	13.190
<i>nna_rt_wd_pp_hp_lst</i>	<i>gu</i>	1.340	-0.874	-24.512	2.785	2.053	4.026	0.184
<i>nnp_nna_rt_wd_pp_hp_lst</i>	<i>gu</i>	-7.164	-7.397	-24.512	-5.901	-8.020	-3.699	-5.860

Referencias Bibliográficas

- [Agre 1992] G. Agre (1992). *"An approach for Solving Technical Diagnostic Problems using causal and probabilistic knowledge"*, PhD thesis, Technical University, Sofia 1992.
- [Agre 1997] G. Agre (1997). *"Diagnostic Bayesian Networks"*, Computers and artificial intelligence, Vol. 16, N° 1, pp: 47-67.
- [Altrock 1995] C. V. Altrock (1995). *"Fuzzy logic and NeuroFuzzy Applications Explained"*, Prentice Hall PTR, 1995.
- [Angle 1990] C. Angle and R. Brooks (1990). *"Small Planetary Rovers"*, Proceedings of the IEEE International Workshop on Intelligent Robots and Systems, Ibaraki, Japan, 1990, pp: 383-388.
- [Arkin 1998] R. C. Arkin (1998). *"The 1997 AAAI mobile robot competition and exhibition"*, AI magazine, Vol.19 N° 3, pp: 13-17, 1998.
- [Ayoubi 1996] M. Ayoubi (1996). *"Fuzzy systems design based on a hybrid neural structure and application to the fault diagnosis of technical processes"*, Control Engineering practice, Vol. 4, N° 1, (1996), pp: 35-41.
- [Ayoubi 1997] M. Ayoubi and R. Isermann (1997). *"Neuro-fuzzy systems for diagnosis"*, International Journal for Fuzzy Sets and Systems. Vol. 89 (1997), pp: 289-307.
- [Balch 1998] T. Balch and C. Arkin (1998). *"Behavior-Based Formation Control for Multirobot Teams"*, IEEE transactions on robotics and automation, vol. 14, N° 6 (1998) pp: 926-939.
- [Bares 1990] J. Bares and W. Whittaker (1990). *"Walking Robot with a Circulating Gait"*, Proceedings of IEEE International Workshop on Intelligent Robots and Systems, Ibaki, Japan, july 1990, pp: 809-815.
- [Barraquand 1992] J. Barraquand, B. Langlois and J. Latombe (1993). *"Numerical Potencial Field Techniques for Robot Path Planning"*, IEEE Transactions on systems, man, and cybernetics, Vol. 22, N° 2, March/april 1992.

- [Barto 1990] A. G. Barto, R. S. Sutton and C. J. Watkins (1990). “*Learning and sequential decision making*”, In M. Gabriel and J. W. Moore editors, *Learning and Computational Neuroscience: Foundations of Adaptive Networks*, pp: 539-602 MIT Press, Cambridge, Massachusetts, 1990.
- [Beetz 1998] M. Beetz, W. Burgard, D. Fox and A. B. Cremers (1998). “*Integrating active localization into high-level robot control systems*”, *Robotics and Autonomous Systems*, Vol. 23, 1998, pp: 205-220.
- [Behringer 1998] R. Behringer and N. Müller (1998). “*Autonomous Road Vehicle Guidance from Autobahnen to Narrow Curves*”, *IEEE transactions on robotics and automation*, vol. 14, N° 5 (1998), pp: 810-815.
- [Bellingham 1989] J. Bellingham, T. Consi, R. Beaton and W. Hall (1989). “*Keeping layered control simple*”, *Proceedings of the sixth international Symposium on Autonomous Underwater Vehicle Technology*, 3-8.
- [Bernardino 1999] A. Bernardino and J. Santos-victor (1999). “*Visual behaviours for binocular tracking*”, *Robotics and Autonomous Systems*, Vol. 27, April (1999), pp: 225-245.
- [Bishop 1995] C. Bishop (1995). “*Neuronal Networks for pattern recognition*”, Clarendon Press, Oxford.
- [Boehmke 1995] S. Boehmke and J. Bares (1995). “*Electronic and telemetry systems of dante II*”, *proceedings of the 41th International Instrumentation Symposium*, Instrument Society of America, pp: 223-232.
- [Borenstein 1991] J. Borenstein and Y. Koren (1991). “*The vector Field Histogram-Fast Obstacle Avoidance for Mobile Robots*”, *IEEE Transactions on Robotics and automation*, Vol. 7, N° 3, June 1991.
- [Brooks 1986] R. A. Brooks (1986). “*A Robust Layered Control System for a Mobile Robot*”, *IEEE Journal of Robotics and Automation*, Vol.2, N° 1, pp: 14-23, March 1986.
- [Burgard 1998] W. Burgard, A. B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schalz, W. Steiner and S. Thrun (1998). “*Experiences with an interactive Museum Tour-Guide Robot*”, June 1998. Carnegie Mellon University report: CMU-CS-98-139.
- [Cassandra 1994] A. R. Cassandra, L. P. Kaelbling and M. L. Littman (1994). “*Acting optimally in partially observable stochastic domains*” In *Proceedings of the twelfth National Conference on Artificial Intelligence*, Seattle, Washington, 1994, pp: 1023-1028.

- [Cassandra 1997] A. R. Cassandra, M. Littman and N. Zhang (1997). “*Incremental pruning: a simple, fast, exact method for Partially Observable Markov Decision Processes*”, Proceedings of the 13 Annual Conference on Uncertainty in Artificial Intelligence (UAI-97), 1997.
- [Cassandra 1998] A. R. Cassandra (1996). “*Exact and approximate algorithms for partially observable Markov decision Process*”, PhD thesis, Department of Computer Science, Brown University, Providence, Rhode Island, May 1998.
- [Chatila 1996] R. Chatila, S. Lacroix, S. Betge-Brezetz, M. Dery and T. Simeon (1996). “*Autonomous mobile robot navigation for planet exploration –The EDEN project*”, Proceedings of IEEE International Conference Robot. Automat. 1996.
- [Chauvin 1995] Y. Chauvin and D. Rumelhart (1995). “*BACKPROPAGATION: Theory, architectures, and applications (edited collections)*”, Hillsdale, NJ: Lawrence Erlbaum Assoc. 1995.
- [Chen 1998] J. Chen and R.J. Patton (1998). “*Robust Model-Based Fault Diagnosis for Dynamic Systems*”, Kluwer Academic Publishers, 1998.
- [Clark 1978] R. N. Clark (1978). “*A simplified instrument detection scheme*”, IEEE Trans. Aerospace Electron. Systems, Vol. 14, pp: 12-23.
- [Congdom 1993] C. Congdom (1993). “*Carmel Versus Flakey: A Comparison of Two Winners*”, AI Magazine, Vol. 14, N° 1, pp: 49-57.
- [Cooper 1990] G. F. Cooper (1990). “*The computational complexity of Probabilistic Inference on Bayesian Belief Network*” Artificial Intelligence, Vol. 42, N° 2-3, pp: 353-405.
- [Dasarthy 1990] B. Dasarthy (1990). “*Nearest neighbour pattern classification techniques*”, IEEE Computer Society Press, Los Alamitos, CA.
- [Diéguez 1995] A. R. Diéguez, C. Raimundez, R. Sanz, J. L. Fernández and E. Delgado (1995). “*An Intelligent Supervisory Model for Path Planning and Guidance of Mobile Robots in Non-Structured Environments*”, In proc. International Congress on Intelligent Autonomous Vehicles IFAC Spoo (Finlandia) 1995.
- [Diéguez 1998] A. R. Diéguez, R. Sanz and J. López (1998). “*Improving global motion planning for mobile robots by experimental measurement of traveling time*”, 3rd IFAC Conference on Intelligent Autonomous Vehicles IAV. 1998 Madrid, Spain.

- [Domonte 1997] E. P. Domonte (1997). “*Sistema de posicionamiento basado en visión y planos CAD para navegación de robots móviles de interiores*”, PhD thesis, Dpto. Ing. Sistemas y automática, E.T.S.I.I.M. Universidad de Vigo. 1997.
- [Driankov 1993] D. Driankov, H. Hellendoorn and M. Reinfrnak (1993). “*An introduction to fuzzy control*”, Springer-Verlag, Berlin Heidelberg 1993.
- [Escalada 1994] G. Escalada-Imaz and A. M. Martínez-Enríquez (1994). “*Motores de inferencia de complejidad óptima de encadenamiento hacia delante para diversas clases de sistemas de reglas.*” Informática y automática. Vol.27-3, 1994.
- [Fedor 1993] C. Fedor (1993). “TCX. “*An interprocess Communication System for building robotic architectures. Programmer’s guide to version 10.xx.*”, Carnegie Mellon University, Pittsburgh, PA 15213, 12, 1993.
- [Fernández 1997] J. L. Fernández and R. G. Simmons (1997). “*Calibrating the lasers in Xavier*”, Robot learning laboratory, department of Computer Science, Carnegie Mellon University. Interman note.
- [Fernandez 1998] J.L. Fernandez and R. G. Simmons (1998). “*Robust Execution Monitoring for Navigation Plans*” Proc. 1998 IEEE/RSJ International Conference Victoria, B.C., Canada, 1993, pp: 551-557.
- [Ferrell 1993] C. Ferrel (1993). “*Robust Agent Control of an Autonomous Robot with Many Sensors and Actuators*”, MIT Artificial Intelligence Lab. Technical Report 1443, 1993.
- [Ferrell 1994] C. Ferrel (1994). “*Failure Recognition and Fault Tolerance of an Autonomous Robot*”, Adaptive Behavior, 2:4, pp: 375-398, 1994.
- [Firby 1989] R. J. Firby (1998). “*Adaptive execution in complex dynamic worlds*” Technical report YALEU/CSD/RR 672, Yale University, 1989.
- [Fleury 1994] S. Fleury, M. Herrb and R. Chatila (1994). “*Design of a modular architecture for autonomous robot*”, IEEE International Conference on Robot. Automat., San Diego, CA, 1994.
- [Forgy 1982] C. L. Forgy (1982). “*Rete: A fast Algorithm for the Many Pattern/Many Object Pattern Match Problem*”, Artificial Intelligence, Vol. 19 (1982) pp: 17-37.

- [Fox 1998] D. Fox, W. Burgard and S. Thrun (1998). “*Active Markov localization for mobile robots*”, Robotics and Autonomous Systems, Vol. 25, Nov. (1998), pp: 195-207.
- [Friedman 1997] N. Friedman, D. Geiger and M. Goldszmidt (1997). “*Bayesian networks classifiers*”, Machine Learning, Vol. 29, pp: 131-163, 1997.
- [Gat 1991] E. Gat (1991). “*Integrating Reaction and Planning in a Heterogeneous Asynchronous Architecture form Mobile Robot Navigation.*” ACM SIGART bulletin. Vol. 2, N° 4, pp: 70-74.
- [Gertler 1991] J. Gertler (1991). “*Analitical redundancy methods in fault detection and isolation*”, IFAC Safeprocess Symposium, Baden-Baden. Vol. 1, pp: 239-255. Pergamon Press.
- [Ghallab 1985] M. Ghallab (1985). “*Task execution monitoring by compiled production rules in an advanced multi-sensor robot*”, Robotics Research, the second International Symposium, pp: 393-401, Cambridge, MA, 1985.
- [Gómez 1996] J. M. Gómez de Gabriel, J. L. Martínez, A. Ollero, A. Mandow and V.F. Muñoz (1996). “*Autonomous and teleoperated control of the aurora mobile robot*”, IFAC 1996. San Francisco. Junio 1996.
- [Goodwin 1996] R. Goodwin (1996). “*Meta level control for Decision-Theoretic Planners*”, PhD thesis, School of Computer Science, Carnegie Mellon University (CMU-CS-96-186), Pittsburgh PA, Octubre 1996.
- [Gray 1990] J. Gray (1990). “*A Census of Tandem System Availability Between 1985 and 1990*”, IEEE Transactions on Reliability 39, 409-417. 1990.
- [Haigh 1998] K. Z. Haigh and M. M. Veloso (1998). “*Planning, Execution and Learning in a Robotic Agent*”, Proceedings 4th international conference on Artificial Intelligence Planing Systems, Pittsburgh PA, USA, Junio 1998, pp: 120-127.
- [Hamscher 1991] W. Hamscher (1991). “*Principles of diagnosis: current trends and a report on the first international workshop*” AI magazine. Vol. 12, 1991. N° 4, pp: 15-23.
- [Hanazawa 1989] T. Hanazawa, G. Hinton, K. Shikano, K. Lang and A. Waibel (1989). “*Pnoneme recognition using time-dalay neural networks.*”, IEEE Transactions on Acoustics, Speech and Signal Processing 1989.

- [Hauskrecht 1996] M. Hauskrecht (1996). "*Planning and control in stochastic domains with imperfect information*", PhD thesis, EECS, MIT, 1996.
- [Hayes 1989] B. Hayes Roth, R. Washington, R. Hewett, M. Hewett and A. Seiver (1989). "*Intelligent Monitoring and control*", in Proc. International Joint Conference on AI, Detroit, MI, August 1989, pp: 243-249.
- [Howe 1999] A. E. Howe (1999). "*Evaluating Robot Planning Systems: Some Issues and Successes*", IJCAI'99 workshop on Robot Planning, July 31 1999, Stocholm.
- [Hunt 1992] K. J. Hunt, D. Sbarbaro, R. Zbikowski and P. J. Gawthrop (1992). "*Neuronal networks for Control Systems – A survey*", Automatica, Vol. 28, N° 6, pp: 1083-1112.
- [Isermann 1992] R. Isermann (1992). "*Estimation of phisical parameters for dynamic processes with aplication to an industrial robot*", Int. J. Control, Vol. 55, pp: 1287-1289.
- [Isermann 1996] R. Isermann and M. Ayoubi (1996). "*Fault detection and diagnosis with neuro-fuzzy systems*", EUFIT September 1996, Aachen, Germany.
- [Isermann 1997] R. Isermann and P. Ballé (1997). "*Trends in the aplication of model-based fault detection and diagnosis of technical processes*", Control Eng. Practice 5 (5), pp: 709-719.
- [Isermann 1997b] R. Isermann (1997). "*Supervision, fault-detection and fault-diagnosis methods -An introduction*" Control Eng. Practice, Vol. 5, N° 5, pp: 639-652.
- [Jetto 1999] L. Jetto, S. Longhi and G. Venturini (1998). "*Development and Experimental Validation of an Adaptive Extended Kalman Filter for the Localization of Mobile Robots*", IEEE transactions on robotics and automation, Vol. 15, N° 2, April (1999) pp: 219-229.
- [Kabuka 1990] M. Kabuka, S. Harjadi and A. Younis (1990). "*A Fault Tolerant Architecture for an Automatic Vision-Guided Vehicle.*", IEEE Transactions on systems, man, and cybernetics 20, 381-393. 1990.
- [Kaelbling 1996] A R. Cassandra, L. P. Kaelbling and J. A. Kurien (1996). "*Acting under uncertainty: Discrete bayesian models for mobile-robot navigation*", In international conference: "Intelligent Robots and Systems", IEE/RSJ 1996.

- [Kaelbling 1998] L. P. Kaelbling, M. L. Littman and A. R. Cassandra (1998). "*Planning and acting in Partially Obaservable Stochastic Domains*", Artificial Intelligence (101) 1-2 (1998), pp: 99-134.
- [Kiszka 1990] J. Kiszka and M. Gupta (1990). "*Fuzzy logic Neural Network*", BUSEFAL 4, pp: 104-109.
- [Kleer 1987] Johan de Kleer and B. C. Williams (1987). "*Diagnosing Multiple Faults*", Artificial Intelligence 32 (1987) 97-130.
- [Koenig 1996] S. Koenig and R. G. Simmons (1996). "*Easy and Hard Testbeds for Real-Time Search Algorithms*", Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI), pp: 279-285; 1996.
- [Koenig 1997] S. Koenig and Y. Smirnov (1997). "*Sensor-Based Planning with the Freespace Assumption*", Proceedings of the IEEE International Conference on Robotics and Automation (ICRA). 1997.
- [Korf 1990] R. E. Korf (1990). "*Real time heuristic search*", Artificial Intelligence, Vol. 42, N° 2-3, pp: 189-211.
- [Kortenkamp 1993] D. Kortenkamp, M. Huber, C. Cohen, U. Raschke, C. Bidlack, C. B. Congdom, F. Koss and T. Weymouth (1993). "*Integrated Mobile-Robot Design Winning the AAAI'92 Robot competition*", August 1993 IEEE Expert, pp: 61-73.
- [Krotov 1992] E. Krotov and R. G. Simmons (1992). "*Performance of a Six-Legged Planetary Rover: Power, Positioning, and Autonomous Walking*", Proceedings del 'IEEE International Conference on Robotics and Automation', Nice, France, pp: 169-174.
- [Kurihara 1998] S. Kurihara, S. Aoyagi, R. Onai, T. Sugawara (1998). "*Adaptive selection of reactive/deliberate planning for a dynamic environment*", Robotics and Autonomous Systems, Vol. 24, Sept. (1998), pp: 183-195.
- [Leonhardt 1997] S. Leonhardt and M. Ayoubi (1997). "*Methods of fault diagnosis*", Control Eng. Practice. Vol. 5, N° 5, pp: 683-692.
- [Lewis 1991] L. M. Lewis (1991). "*A Time-Oriented Architecture for Integrating Reflexive and Deliberative Behavior*" ACM SIGART Bulletin, Vol. 2, N° 4 August 1991.
- [Lin 1994] Y. Lin and G. Cunningham (1994). "*A new fuzzy approach for setting the initial weights in a neural network*", 3rd IEEE Conference on Fuzzy Systems , Orlando 1994, pp: 40-45.

- [Lin 1997] C. Lin and L. Wang (1997). "Intelligent collision avoidance by fuzzy logic control", *Robotics and Autonomous Systems*, Vol. 20, Nº 1, April 1997.
- [Liscano 1995] R. Liscano, A. Manz, E. R. Stuck, R. E. Fayek and J. Tigli (1995). "Using a blackboard to integrate multiple activities and achieve strategic reasoning for mobile robot navigation", *IEEE Expert*, April 1995, pp: 24-36.
- [Littman 1994] M. L. Littman (1994). "The witness algorithm for solving partially observable Markov decision processes", Technical Report CS-94-40, Brown University, Providence, Rhode Island, 1994.
- [Littman 1995] M. Littman, A. R. Cassandra and L. Kaelbling (1995). "Learning policies for partially observable environments: Scaling up", *Proceedings of the 12 International conference on Machine Learning*, San Francisco, CA, 1995 pp: 362-370.
- [Masliah 1998] M. R. Masliah and R. W. Albrecht (1998). "The Mobile Robot Surrogate Method for Developing Autonomy", *IEEE Transactions on Robotics and Automation*, Vol. 14, Nº 2, April 1998, pp: 314-320.
- [Matos 1999] L. M. Camarinha-Matos and W. Vieira (1999). "Intelligent mobile agents in elderly care", *Robotics and Autonomous Systems*, Vol. 27, June (1999), pp: 59-75.
- [McDermott 1999] D. McDermott (1999). "The 1998 AI planning systems competition", To be published in *AI Magazine*.
- [Milliken 1986] K. R. Milliken (1986). "YES/MVS and the automation of operations for Large Computer Complexes", *IBM System Journal*, Vol. 25 (nº2) pp: 159-180.
- [Mitchell 1997] T. M. Mitchell (1997). "Machine learning", McGraw-Hill (ISBN 0-07-042807-7).
- [Monahan 1982] G. Monahan (1982). "A survey of partially observable Markov Decision processes: theory, models, and algorithms", *Management Science*, Vol.28, Nº 1, pp: 1-16.
- [Moravec 1988] H. Moravec (1988). "Sensor fusion in certainty grids for mobile robots" *AI Magazine*. Vol. 9. Nº 2, pp: 61-74.
- [Murphy 1992] R. Murphy and R. Arkin (1992). "SFX: An architecture for action-oriented sensor fusion", In 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp: 225-250.

- [Murphy 1996] R. R. Murphy and D. Hershberger (1996). “*Classifying and Recovering from Sensing Failures in Autonomous Mobile Robots*”, In proceedings of AAAI/IAAI, pp: 922-929, August, 1996.
- [Murphy 1997] R. Murphy (1997). “*AAAI Mobile Robot Competitions 1992-1996 Robot Competitions Corner*”, Robotics and Autonomous Systems, Vol. 20, (1997), pp: 6-9.
- [Murphy 1999] R. Murphy, K. Hughes, A. Marzilli and E. Noll (1999). “*Integration explicit path planning with reactive control of mobile robots using Trulla*”, Robotics and Autonomous Systems, Vol. 27, June 1999, pp: 225-245.
- [N.Y. Ko 1998] N.Y. Ko and R. G. Simmons (1998). “*The Lane-Curvature Method for Local Obstacle Avoidance*” Proc. 1998 IEEE/RSJ International Conference Victoria, B.C., Canada, 1993.
- [Nayak 1996] P. P. Nayak and Brian C. Williams (1996). “*Fast Context Switching in Real-time Propositional Reasoning*”, Proceedings of AAAI-97.
- [Noreils 1990] F. R. Noreils (1990). “*Integrating Error Recovery in a Mobile Robot Control System*”, IEEE International Conference on Robotics And Automation. Cincinnati , OH, May 1990 pp: 396-401.
- [Noreils 1995] F. R. Noreils and R. Chatila (1995). “*Plan Execution Monitoring and Control Architecture for Mobile Robots*”, IEEE Transactions on Robotics and Automation, Vol. 11, N° 2, pp: 255-266. 1995.
- [O’Sullivan 1996] J. O’Sullivan and K. Z. Haigh (1996). “*Xavier manual Versión 0.3*”, Robot learning laboratory, department of Computer Science, Carnegie Mellon University. Internal note.
- [Ogasawara 1991] G. H. Ogasawara (1991). “*A Distributed, Decision-Theoretic Control System for a Mobile Robot*”, ACM SIGART Bulletin, Vol. 2, N° 4, pp: 140-145, August 1991.
- [Parker 1998] L. E. Parker (1998). “*ALLIANCE: An Architecture for Fault Tolerant Multirobot Cooperation*”, IEEE Transactions on Robotics and Automation, Vol. 14. N° 2, April 1998, pp: 220-240.
- [Parr 1995] R. Parr and S. J. Russell (1995). “*Approximating Optimal Policies for Partially Observable Stochastic Domains*”, In proceedings IJCAI 1995, pp: 1088-1095.
- [Patton 1991] R. J. Patton and J. Chen (1991). “*A review of parity space approaches to fault diagnosis*”, IFAC Safeprocess Symposium, Baden-Baden. Vol. 1, pp: 239-255. Pergamon Press.

- [Payton 1992] D. Payton, D. Keirsey, D. Kimble, J. Krozel and K. Rosenblat (1992). "*Do whatever works: A robust approach to fault-tolerant autonomous control*", Journal of Applied Intelligence, Vol. 2, pp: 225-250.
- [Perret 1996] J. Perret and R. Alami (1996). "*Planning with non-deterministic events for enhanced robot autonomy*" Robotics and Autonomous Systems Vol. 18 (1996) pp: 311-317.
- [Piaggio 1998] M. Piaggio and R. Zaccaría (1998). "*Using roadmaps to classify regions of space for autonomous robot navigation*", Robotics and Autonomous Systems, Vol. 25, Nov. (1998), pp: 209-217.
- [Rabiner 1986] L. R. Rabiner and B. H. Juang (1986). "*An introduction to hidden Markov Models*", IEEE ASSP Magazine, pages 4-16, January 1986.
- [Raimundez 1998] J. C. Raimundez and E. Delgado Romero (1998). "*Pose acquisition through laser measures in structured environments*", IFAC-ICV, pp: 195-199, 1998.
- [Reignier 1994] P. Reignier (1994). "*Fuzzy logic techniques for mobile robot obstacle avoidance*", Robotics and Autonomous systems, Vol. 12 (1994), pp: 143-153.
- [Rumelhart 1994] D. Rumelhart, B. Widrow and M. Lehr (1994). "*The basic ideas in neuronal networks*", Communications of the ACM, Vol. 37, N°3, pp: 87-92.
- [Russell 1996] S. Russell and P. Norving (1996). "*Inteligencia Artificial: un enfoque moderno*", Ed. Prentice Hall Hispanoamericana, (ISBN 968-880-682), pp: 161-313.
- [Saffiotti 1993] A. Saffiotti and E.H. Ruspini (1993). "*Blending Reactivity and Goal-Directedness in a Fuzzy Controller*", Proc. IEEE International Conference Fuzzy Systems, IEEE Computer Society Press. Los Alamitos, Calif., 1993.
- [Sankaran 1977] S. Sankaran (1977). "*Error recovery in robot systems*", PhD thesis, Dept. of Comp. Sci., University of Carnegie Mellon. 1977.
- [Sawaki 1978] K. Sawaki and A. Ichikawa (1978). "*Optimal control for partially observable Markov decision processes over an infinit horizon*", Journal of the Operations Reseach Society of Japan, Vol. 21, N° 1, pp: 1-14, March 1978.

- [Simmons 1991] R. Simmons and E. Krotkov (1991). “*An integrated walking system for the Ambler planetary rover*” Proc. IEEE International Conference on Robotics and Automation, pages 2086-2091, Sacramento, CA, April 1991.
- [Simmons 1991b] R. G. Simmons (1991). “*Coordinating Planning, Perception, and Action for Mobile Robots*”, ACM SIGART Bulletin, Vol. 2, N° 4, August. 1991, pp: 156-159.
- [Simmons 1994] R. G. Simmons (1994). “*Structured Control for Autonomous Robots*”, IEEE Transactions on Robotics and Automation. Vol. 10, N° 1, February 1994.
- [Simmons 1995] R. G. Simmons and S. Koenig (1995). “*Probabilistic Navigation in Partially Observable Environments*”, IJCAI Intl. Conference, Montreal Canada, July 1995.
- [Simmons 1996] R. G. Simmons (1996). “*The Curvature-Velocity Method for Local Obstacle Avoidance*”, Proc. International Conference on Robotics and Automation, Minneapolis MN, April 1996.
- [Simmons 1997] R. G. Simmons and G. Whelan (1997). “*Visualization Tools for Validating Software of Autonomous Spacecraft*”, i-Sairas, Tokyo Japan, July 1997.
- [Simmons 1997b] R. G. Simmons, R. Goodwin, K. Haigh, S. Koenig and J. O'Sullivan (1997). “*A Layered Architecture for Office Delivery Robots*”, First International Conference on Autonomous Agents, Marina del Rey, CA, February 1997, pp: 245-252.
- [Simmons 1997c] R. G. Simmons, R. Goodwin, C. Fedor and J. Basista (1997). “*Task Control Architecture, Programmer's guide to version 8.0*” (including release notes through TCA version 8.5)” Carnegie Mellon University, School of Computer Science, may 1997.
- [Simmons 1999] R. G. Simmons, J. L. Fernandez, R. Goodwin, S. Koenig and J. O'Sullivan (1999). “*Xavier: An Autonomous Mobile Robot on the Web*”, To appear in IEEE Robotics and Automation Magazine.
- [Sondik 1971] E. Sondik (1971). “*The Optimal Control of Partial Observable Markov Processes*”, PhD thesis, Standford University, 1971.
- [Sprave 1994] J. Sprave (1994). “*Linear Neighborhood Evolution Strategies*” Proceedings of 3rd anual conference on Evolutionary Programming (EP'94), pp: 42-51, San Diego CA, Febrero 24-26, 1994. World Scientific, Singapore.

- [Stergios 1998] S. I. Roumeliotis, G. S. Sukhatme and G. A. Bekey (1998). “*Fault Detection and identification in a Mobile Robot using Multiple-Model Estimation*”, In proceedings of the 1998 IEEE International Conference on Robotics and Automation Leuven, Belgium, may 16-21, 1998, pp: 2223-2228.
- [Stergios 1998b] S. I. Roumeliotis, G. S. Sukhatme and G. A. Bekey (1998). “*Sensor Fault Detection and identification in a Mobile Robot using Multiple-Model Estimation*”, In proceedings of the 1998 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems, Victoria, B.C., Canada, October, 1998 pp: 1383-1388.
- [Stone 1998] P. Stone and M. Veloso (1998). “*Towards collaborative and adversarial Learning: a case study in robotic soccer*”, International Journal of Human-Computer Studies (IJHCS), Vol. 48, number1, 1998.
- [Stuck 1992] E. R. Stuck (1992). “*Detecting and Diagnosing Mistakes in Inexact Vision-Based Navigation*”, PhD thesis, Dept. of Computer Science, University of Minnesota, Minneapolis, MN, November 1992.
- [Stuck 1995] E. R. Stuck (1995). “*Detecting and Diagnosing Navigational Mistakes*”, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Pittsburgh, PA, August 5-9, 1995.
- [Thrun 1998] S. Thrun (1998). “*Learning metric-topological for indoor mobile robot navigation*”, Artificial Intelligence, Vol. 99, N°1, pp: 21-71.
- [Tsui 1993] C. C. Tsui (1993). “*A general failure detection, isolation and accommodation system with model uncertainty and measurement noise*”, 12th IFAC World Congress, Sydney, Australia. Vol. 6, pp: 231-238.
- [Ulieru 1996] M. Ulieru (1996). “*Fuzzy logic in diagnostic decision: Possibilistic Networks*”, Dissertation, Technical University Darmstadt.
- [Vázquez 1994] F. Vázquez and E. García (1994). “*A local guidance method for low-cost mobile robots using fuzzy logic*”, IFAC Artificial Intelligence in Real Time Control. Valencia, Spain, 1994.
- [Veloso 1995] M. M. Veloso, J. Carbonell, M. A. Perez, D. Borrajo, E. Fink and J. Blythe (1995). “*Integrating planning and learning: The PRODIGY architecture*” Journal of Experimental and Theoretical Artificial Intelligence, Vol. 7, N°1, January 1995.

- [Vemuri 1998] A. T. Vemuri, M. M. Polycarpou and S. A. Diakourtis (1998). "*Neuronal Network Based Fault Detection in Robotic Manipulators*", IEEE transactions on Robotics and Automations, Vol. 14 N° 2. April 1998 pp: 342-348.
- [Warren 1976] D.H.D. Warren (1976). "*Generating conditional plans and programs*", AISB-76 Summer Conference, Edinburgh.
- [Washington 1996] R. Washington (1999). "*Incremental Markov-Model Planning*", 8th IEEE International Conference on Tools with Artificial (ICTAI'96). November 16-19, 1996.
- [Washington 1998] R. Washington (1998). "*Markov Tracking for Agent Coordination*", Proceedings of Agents'98: Second International Conference on Autonomous Agents, May 10-13, 1998.
- [Wettergreen 1999] D. Wettergreen, D. Bapna, M. Maimone and G. Thomas (1999). "*Developing Nomad for robotic exploration of the Atacama Desert*", Robotics and Autonomous Systems, Vol. 26, February (1999), pp: 127-148.
- [Williams 1997] B. C. Williams and P. P. Nayak (1997). "*A Reactive Planner for a Model-based Executive*", In Proceedings of IJCAI-97.
- [Willsky, 1976] A. S. Willsky (1976). "*A survey of design methods for failure detection systems*", Automatica, Vol. 12, pp: 601-611.
- [Xu 1997] H. Xu and H. Van Brussel (1997). "*A behaviour-based blackboard architecture for reactive and efficient task execution of an autonomous robot*". Robotics and Autonomous Systems, Vol. 22, N° 2, Nov. 1997, pp: 115-132.
- [Xu 1998] W L. Xu, S. K. Tso and Y. H. Funy (1998). "*Fuzzy reactive control of a mobile robot incorporating a real/virtual target switching strategy*", Robotics and Autonomous Systems, Vol. 23, N° 3, April 1998, pp: 171-186.
- [Zhang 1996] N. L. Zhang and W. Liu (1996). "*Planing in stochastic domains: Problem characteristics and approximation*", Technical Report HKUST-CS96-31, Department of Computer Science, Hong Kong University of science and Technology, 1996.