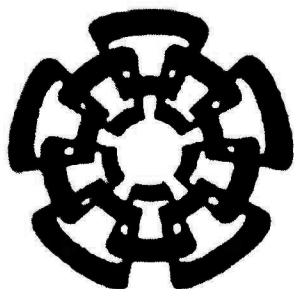




xx(101582,1)





# CINVESTAV

Centro de Investigación y de Estudios Avanzados del IPN  
Unidad Guadalajara

---

## ARQUITECTURA PARA AYUDA EN LA TOMA DE DECISIONES EN PRESENCIA DE INCERTIDUMBRE

TESIS QUE PRESENTA  
MARIO ENRIQUE AGUEDA HERRERA



PARA OBTENER EL GRADO DE  
MAESTRO EN CIENCIAS

EN LA ESPECIALIDAD DE  
INGENIERÍA ELÉCTRICA

Guadalajara, Jal. Noviembre del 2001

CINVESTAV I.P.N.  
SECCION DE INFORMACION  
Y DOCUMENTACION

CLASIFICACION	
ASUBJETO	tesis 2002
FECHA	6-agosto-02
PROCESO	Serv Bibli

# **ARQUITECTURA PARA AYUDA EN LA TOMA DE DECISIONES EN PRESENCIA DE INCERTIDUMBRE**

Tesis de Maestría en Ciencias  
Ingeniería Eléctrica

Por:

**Mario Enrique Agueda Herrera**

Ingeniero en Sistemas Computacionales  
Instituto Tecnológico de Tuxtla Gutiérrez

Becario de CONACYT, expediente No. 143776

Directores de Tesis:

**Dr. Pablo Héctor Ibargüengoytia González**

**Dr. Luis Ernesto López Mellado**

CINVESTAV del IPN Unidad Guadalajara, Noviembre del 2001

## **Dedicatoria**

Dedico este trabajo a ti, hijo. Quiero decirte que tengo la bendición de que hayas nacido en el mismo mes en que este trabajo ha llegado a su fin. Por ti, es todo esto. Te amo.

## Agradecimientos:

A ti *Αββα ο πατηρ*, Dios de Israel, te agradezco la bendición que me has dado. Pude sentir tu ayuda en la realización de esta tesis. Tú dijiste:

“ Clama a mí y yo te responderé,  
y te enseñaré cosas grandes y ocultas  
que tú no conoces” (Jer. 33.3)

Pues bien:

“Busqué a Jehová, y Él me oyó,  
y me libró de todos mis temores.  
Los que miraron a Él fueron alumbrados,  
y sus rostros no fueron avergonzados.  
Este pobre clamó, y le oyó Jehová  
y lo libró de todas sus angustias.”  
(Sal. 34.4-6)

A mis padres porque con su dirección fui encaminando mis esfuerzos para lograr lo que hoy tengo.

A mi esposa, por haber aceptado la distancia y haber resistido. Sé que este trabajo también a ti te ha costado esfuerzo.

A mis hermanos, porque siempre han creído en mí.

A mis compañeros, amigos y hermanos en Cristo, por compartir momentos agradables conmigo. Porque ustedes han hecho más placentera la estancia en Guadalajara y en Cuernavaca mientras realizaba el trabajo.

A mis asesores, por toda la orientación que me dieron. La experiencia de ustedes que me han transmitido me ha sido de gran importancia.

Al Centro de Investigación y de Estudios Avanzados del IPN, unidad Guadalajara, por la enseñanza que me impartieron.

Al Instituto de Investigaciones Eléctrica, por permitirme hacer la estancia con ustedes.

A todos los que colaboraron con el desarrollo de la presente Tesis, gracias.

A Consejo Nacional de Ciencia y Tecnología (Conacyt), por el apoyo financiero proporcionado en estos dos años. (No. Becario: 143776)



# INDICE

<b>RESUMEN</b> .....	<b>1</b>
<b>CAPÍTULO 1 PANORAMA GENERAL</b> .....	<b>3</b>
1.1 INTRODUCCIÓN .....	4
1.2 INTELIGENCIA ARTIFICIAL EN EL SECTOR ELÉCTRICO .....	5
1.3 ANTECEDENTES .....	6
1.4 PLANTEAMIENTO DEL PROBLEMA .....	10
1.5 OBJETIVOS DE LA TESIS .....	11
1.6 ESTRUCTURA DE LA TESIS .....	13
<b>CAPÍTULO 2 PROCESOS DE DECISIÓN CON INCERTIDUMBRE</b> .....	<b>15</b>
2.1 INTRODUCCIÓN .....	16
2.2 DECISIÓN CON INCERTIDUMBRE .....	18
2.3 LOS PROCESOS DE DECISIÓN DE MARKOV (MDP).....	19
2.3.1 <i>Definición de los MPD</i> .....	19
2.3.2 <i>Cálculo de la política óptima</i> .....	21
2.4 ALGORITMOS PARA LOS MDPS .....	22
2.4.1 <i>Cálculo de la máxima suma esperada de recompensas</i> .....	22
2.4.2 <i>Cálculo de la política óptima</i> .....	24
2.5 MDPS PARCIALMENTE OBSERVABLES (POMDPs).....	25
2.5.1 <i>Definición de los POMDPs</i> .....	25
2.5.2 <i>Cálculo de la distribución en estados</i> .....	26
2.5.3 <i>Cálculo de la acción óptima</i> .....	27
2.6 ALGORITMOS PARA ESTIMAR EL ESTADO EN LOS POMDP .....	28
<b>CAPÍTULO 3 MODELADO DEL DOMINIO DE APLICACIÓN</b> .....	<b>33</b>
3.1 INTRODUCCIÓN .....	34
3.2 EL DOMO DE UNA CTE .....	36
3.3 COMPONENTES DE LOS POMDP EN EL DOMINIO.....	39
3.3.1 <i>Estados</i> .....	41
3.3.2 <i>Acciones</i> .....	42
3.3.3 <i>Función de transición</i> .....	43
3.3.4 <i>Función de recompensa</i> .....	44
3.3.5 <i>Observaciones</i> .....	44
3.3.6 <i>Función de observación</i> .....	45

## CAPÍTULO 4 ARQUITECTURA PARA GENERAR ACCIONES EN DOMINIOS CON

<b>INCERTIDUMBRE .....</b>	<b>47</b>
4.1 INTRODUCCIÓN .....	48
4.2 DESCRIPCIÓN GENERAL DE AUDA .....	48
4.3 EL AMBIENTE .....	51
4.4 MÓDULO DE OPERADORES (ACTOR).....	52
4.5 MÓDULO DE BASE DE CONOCIMIENTO.....	53
4.5.1 <i>Matrices de recompensas (MR)</i> .....	53
4.5.2 <i>Matriz de Transición (MT)</i> .....	54
4.5.3 <i>Matriz de Observación (MO)</i> .....	55
4.6 MÓDULO DE PROCESOS DE DECISIÓN (PD).....	56
4.6.1 <i>Metas</i> .....	58
4.6.2 <i>Módulo de los MDP</i> .....	59
4.6.3 <i>Percepción</i> .....	59
4.6.4 <i>Observación Parcial (OP)</i> .....	60
4.6.5 <i>Acción</i> .....	60
4.6.6 <i>Distribución de estados (DE)</i> .....	60
4.6.7 <i>Módulo de los POMDP</i> .....	61
4.7 MÓDULO DE APRENDIZAJE (MA).....	61
4.7.1 <i>Modificación de Recompensas</i> .....	62
4.7.2 <i>Modificación de Transición</i> .....	63
4.7.3 <i>Modificación de Observación</i> .....	64

## CAPÍTULO 5 EXPERIMENTOS Y RESULTADOS..... 65

5.1 INTRODUCCIÓN .....	66
5.2 PROTOTIPO.....	66
5.3 CONSTRUCCIÓN DE MATRICES .....	69
5.3.1 <i>Los datos históricos</i> .....	70
5.3.2 <i>Matriz de transición</i> .....	70
5.3.3 <i>Matriz de recompensas</i> .....	75
5.3.4 <i>Matriz de observación</i> .....	76
5.4 OBTENCIÓN DE POLÍTICAS.....	78
5.4.1 <i>Consideraciones</i> .....	78
5.4.2 <i>Desarrollo del experimento</i> .....	78
5.4.3 <i>Resultados</i> .....	78
5.5 LOS POMDP.....	81
5.5.1 <i>Consideraciones</i> .....	82

5.5.2	<i>Desarrollo del experimento</i> .....	82
5.5.3	<i>Resultados</i> .....	83
5.6	APRENDIZAJE A TRAVÉS DEL OPERADOR.....	83
5.6.1	<i>Consideraciones</i> .....	83
5.6.2	<i>Desarrollo del experimento</i> .....	84
5.6.3	<i>Resultados</i> .....	84
<b>CAPÍTULO 6 CONCLUSIONES Y TRABAJOS FUTUROS</b> .....		<b>87</b>
6.1	CONCLUSIÓN GENERAL .....	88
6.2	DISCUSIONES .....	90
6.3	TRABAJOS FUTUROS .....	91
<b>REFERENCIAS</b> .....		<b>93</b>
<b>APÉNDICE</b> .....		<b>97</b>

# Resumen

La planificación es de gran importancia en la inteligencia artificial. Existen en la actualidad dominios donde se requieren de planificadores inteligentes de tal forma de que se puedan conocer las acciones óptimas para alcanzar una meta. Se ha atacado este problema con técnicas que resultan ser deterministas. Sin embargo, existe un dominio en especial que resulta interesante para ser estudiado, éste es aquel donde está presente la incertidumbre. La incertidumbre puede estar presente en los efectos de las acciones o bien en las creencias en un determinado momento, lo cual representa un problema para las técnicas deterministas.

Los Procesos de Decisión de Markov (MDP: Markov Decision Processes) y los MDP Parcialmente observables (POMDP: Partially Observable MDP), son un medio a través del cual se puede resolver el problema planteado. En esta tesis se propone una arquitectura para la ayuda en la toma de decisiones en presencia de incertidumbre utilizando los MDP y los POMDP. La arquitectura AUDA (Acting on Uncertainty Domain Architecture) está basada en el paradigma de *creencias, deseos e intenciones*.

La solución que se presenta se ha estudiado en el ámbito de la operación de una Central Termoeléctrica. Por lo tanto, se ha desarrollado una metodología para modelar el sistema físico definiendo cada elemento necesario para resolver el problema con los procesos de decisión de la arquitectura.





# Capítulo 1      Panorama General

---

La planificación es una área de la Inteligencia Artificial de gran importancia en las ciencias de la computación. Este problema está inmerso en muchos campos de aplicación. En particular, en los procesos relacionados con la generación de electricidad, es importante contar con herramientas que ayuden a los operadores a llevar a las centrales a un nivel óptimo de operación.

En dominios con incertidumbre se complica el desarrollo de planificadores. Una solución al problema es desarrollar una arquitectura para ayuda en la toma de decisiones en presencia de incertidumbre. En este capítulo se dará una descripción sobre el origen del presente trabajo. Se comentarán los trabajos realizados que son antecedentes a éste.

---

## 1.1 Introducción

Con la globalización, automatización y competencia en las industrias, se ha hecho evidente la necesidad de optimizar los recursos para el mejor desempeño de las funciones dentro de ellas. Se han convertido en una gran herramienta los sistemas de computación que permiten crear estrategias que pueden ayudar en la toma de decisiones. Una de las áreas de las ciencias de la computación que se ha explorado en este sentido es la Inteligencia Artificial (IA).

El primer sistema experto comercial, R1, inició sus actividades en Digital Equipment Corporation. En años recientes en el área de la comprensión del lenguaje, han prevalecido los enfoques basados en los Modelos de Markov ocultos (HMM, hidden Markov models). Los tempranos trabajos de Tate (1977) continuados por Chapman (1987) han dado por fruto una mejor representación de programas de planificación ya existentes, convirtiéndolos en un sencillo marco de trabajo. Similares revoluciones se han dado en robótica, visión por computadora, autoaprendizaje de máquina (incluidas las redes neuronales) y representación del conocimiento, el reconocimiento de voz (HMM) y la incertidumbre (Bayes). [Russell y Norving 94].

Las raíces de la planificación en la IA se encuentran parcialmente en la solución de problemas mediante la búsqueda en espacios de estados y otras técnicas similares, como son la reducción de problemas y el análisis de medios y fines; en parte, esto se puede observar en la demostración de teoremas y el cálculo de situaciones, especialmente presentes en sistemas para demostración de teoremas, por ejemplo el QA3. La planificación también fue motivada por la robótica. STRIPS, el primer sistema de planificación importante, se diseñó como el elemento de planificación del software para el proyecto del robot Shakey en SRI International [Russell y Norving 94].

Por otro lado, el desarrollo de planificadores para dominios con incertidumbre, obliga tener algoritmos poco eficientes. Por lo tanto, se debe contar con herramientas y técnicas de IA que sustituyan a los planificadores, de tal manera que se obtengan respuestas aceptables a los operadores.

## 1.2 Inteligencia Artificial en el sector eléctrico

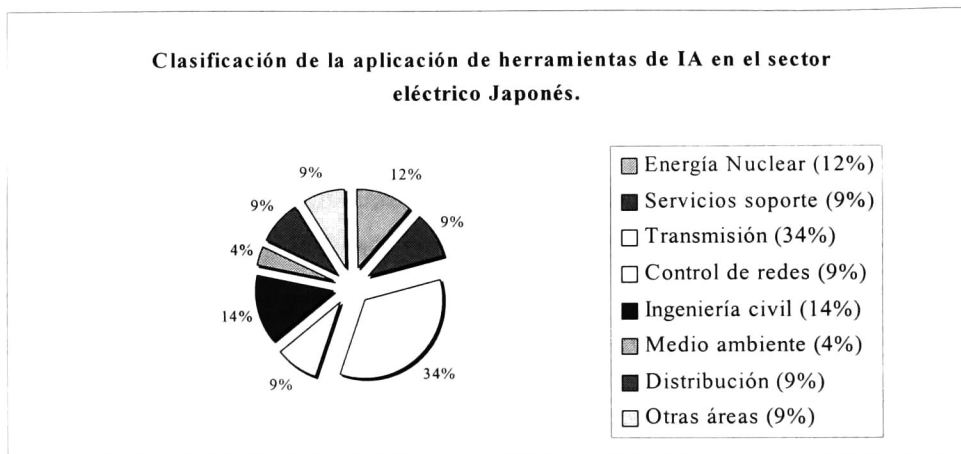
En la actualidad las industrias del sector energético se apoyan en sistemas expertos, los cuales son diseñados para actuar como especialista en un campo particular. Típicamente están constituidos por una base de conocimientos expansible, dada por expertos. Esta base de conocimientos consiste en eventos del campo y reglas aplicables al registro y control de los mismos. Han probado su utilidad para el análisis químico, la exploración geotérmica y la configuración de sistemas computacionales.

Por otro lado, el desarrollo de planes de negocios o de complejos proyectos industriales requiere de la recopilación y evaluación de grandes cantidades de información. Los programas *inteligentes de planificación* están diseñados para apoyar la toma de decisiones y se han convertido en herramientas gerenciales útiles.

A partir de la década de los ochenta, los organismos relacionados con la generación, transmisión y distribución de la energía eléctrica de todo el mundo han doblado esfuerzos para aprovechar el rápido crecimiento de la IA. Tanto en los Estados Unidos como en Europa y Japón se han obtenido resultados halagadores que ya vinculan ampliamente a la IA con los sectores eléctricos de todo el mundo. A principios de esa década, iniciaron las primeras aplicaciones de herramientas de IA en Japón para resolver problemas relacionados con el diagnóstico y apoyo a la operación eficiente de planta. Fue a partir de 1990 que se multiplicó la investigación en torno de la aplicación de tecnologías de IA a líneas de transmisión, subestaciones eléctricas y a la operación y control de sistemas de distribución. La Gráfica 1.1 muestra un informe sobre las áreas de aplicación de las herramientas de IA en el sector eléctrico japonés. En él se reveló que casi 35% de éstas se enfocaron hacia el control de redes y 14%, a ingeniería civil de construcción de plantas [Boletín IIE 98].

Uno de los proyectos que demuestra la enorme potencialidad de la IA en este campo es el iniciado por Kansai Electric Power Company, cuyos especialistas construyeron un avanzado analizador de sistemas a escala física de generadores, líneas de transmisión, transformadores y otros elementos. En un boletín publicado en el IIE [Boletín IIE 98], se menciona que la utilización de sistemas expertos y módulos de redes neuronales para el mejoramiento del desempeño de plantas de generación eléctrica han sido probadas con buenos resultados en los Estados Unidos. En 1996, la empresa San Diego Gas & Electric

probó con éxito el Intelligent Power Plan Advisor System (IPPAS) en la unidad 4 de su planta termoeléctrica Encinas. El IPPAS es un sistema de asesoría que opera en tiempo real para interpretar la información recabada por sistemas de control digital y de monitoreo de la planta. Los resultados obtenidos a partir de las primeras pruebas demostraron su capacidad para desempeñarse como un útil asistente para los operadores de la planta. Desarrollos similares se han aplicado con excelentes resultados en Canadá, Inglaterra, Francia e Italia.



**Gráfica 1.1 Aplicaciones de IA en el sector eléctrico Japonés**

La gama de aplicaciones de las herramientas de IA en el sector eléctrico es enorme: entrenamiento de operadores a partir de sistemas expertos; lectores maestros de diagramas de planta; utilización de redes neuronales para el reconocimiento y control de patrones de consumo eléctrico, y simuladores en tiempo real para el análisis del comportamiento de sistemas de potencia, entre muchos otros.

### **1.3 Antecedentes**

El presente trabajo es consecuencia de proyectos realizados en el sector energético en México, principalmente en la industria eléctrica. En particular, el trabajo de investigación aquí presentado ha surgido a partir de algunas aplicaciones desarrolladas



dentro de la Gerencia de Control e Instrumentación (GCI) del Instituto de Investigaciones Eléctricas<sup>1</sup> (IIE) sobre planificadores para la Comisión Federal de Electricidad (CFE).

Los primeros pasos en el desarrollo de herramientas de IA en el IIE comenzaron en 1989 con el desarrollo de EXEB, sistema experto en barnices aislantes, encargado al Instituto por el *Grupo Condumex* para la planta de barnices de su planta *Magneto*. La base de conocimientos del sistema estuvo compuesta por cerca de 500 reglas. Manejaba conocimiento sobre diagnóstico y operación, integraba gráficos de equipo, sus partes y su distribución, así como programas externos de cálculo. Al analizar en retrospectiva el proyecto, el grupo de trabajo encontró que básicamente el mayor problema al que se enfrentaron era que el conocimiento cambiaba mucho y rápidamente.

A partir de esos trabajos, se comenzó a incursionar en otras áreas. Las redes neuronales tuvieron su aparición con el desarrollo de la red neuronal SOLNET, sucesora en el diseño de secuencias de manufactura de soleras de cobre del sistema experto SOL, también desarrollado y puesto en operación en 1990 por el IIE. Con las redes neuronales holográficas se buscó resolver el problema que la continua actualización de la base de datos del sistema experto representaba para manufacturar nuevos diseños o rutas.

A partir de entonces se desarrollaron proyectos para el diagnóstico y reconocimiento de fallas de un cuadro de alarmas en el cuarto de control de alarmas de una central nucleoelectrónica. En la Unidad de Turbomaquinaria del IIE, un proyecto para la CFE comprobó la eficacia de la utilización de herramientas de IA en el campo de diagnóstico de fallas. El sistema experto para el diagnóstico de fallas de turbinas de vapor se basó principalmente en variables mecánicas, posteriormente, se le agregó un módulo adicional que hacía uso de los parámetros de operación para conocer tanto el estado dinámico, como la condición de eficiencia y potencia que tenía el equipo en el momento del diagnóstico. El sistema experto denominado Turmaya se implantó en Guaymas II, en la región Noreste de la CFE, en 1997. [IIE 98].

---

<sup>1</sup> El Instituto de Investigaciones Eléctricas se encuentra ubicado en Avenida Reforma #113, col. Palmira. Temixco, Morelos, México.



En el campo de los sistemas inteligentes para el apoyo de procesos de centrales termoeléctricas, se desarrolló el proyecto Sistema Inteligente de Ayuda para el Arranque de Unidades Termoeléctricas (SIAAT) en 1998. El SIAAT es un sistema informático ejecutable en PC, que recibe información en línea del estado de la unidad termoeléctrica y la procesa mediante técnicas de inteligencia artificial utilizando redes bayesianas, las cuales toman en cuenta las sugerencias de operadores expertos. De esta manera, el SIAAT determina una secuencia de acciones que llevan a la unidad termoeléctrica del estado de paro frío al estado de unidad sincronizada con el Sistema Eléctrico Nacional. Esta versión fue probada en acoplamiento con el simulador Termos II del Centro de Adiestramiento a Operadores Ixtapantongo (CAOI) y revisada por instructores de ese Centro. Las pruebas de validación en sitio consistieron en normalizar el sistema eléctrico siguiendo las instrucciones generadas por el SIAAT. El SIAAT fue diseñado con base en módulos de funciones de programación estructurada y se utilizó un planificador comercial para la generación de los planes. [Ibargüengoytia et al. 99]

Entre los resultados obtenidos por parte del grupo de desarrollo del SIAAT, se puede hacer notar que la fase de encendido en frío de la planta, el plan consistía en procedimientos en la mayoría secuenciales y conocidos. Concluyeron que en el arranque en caliente de una Central Termo Eléctrica (CTE) existen condiciones termodinámicas que dificultan la creación de un plan óptimo. Con base a estos resultados, en el año 2000 se inició en la GCI el desarrollo del proyecto llamado “*Desarrollo de técnicas de inteligencia artificial para procesos dinámicos*”

Este nuevo proyecto debería tener como objetivo, crear una técnica que pudiera obtener un plan dada la presencia de una falla en la central o bien en un arranque en caliente en una Central de Ciclo Combinado (CCC) . El sistema no ejercería ninguna acción directa sobre la central si no que presentaría al operador mensajes de apoyo, mencionándole las acciones que debe tomar sobre la central para lograr el objetivo deseado. Esto se ilustra en la Figura 1.1

El Asistente al Operador esta basado en una arquitectura Multiagentes independientes y acoplados, todos trabajando concurrentemente para alcanzar una meta en

común. La arquitectura es presentada en la Figura 1.2, para mayor información referirse a [Ibargüengoytia y Reyes 01] y [Valdez 01].

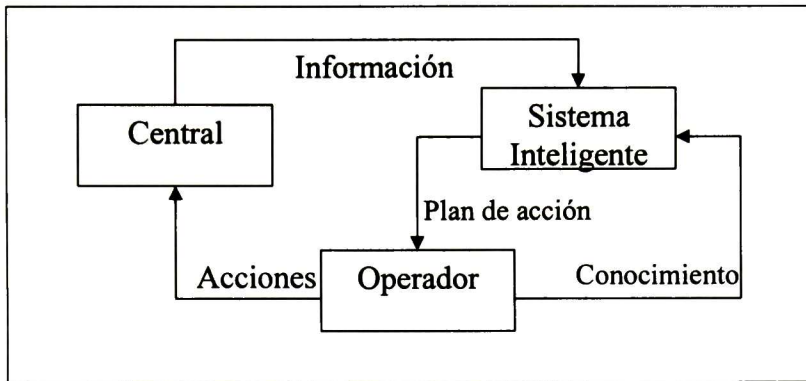


Figura 1.1 Asistente al operador

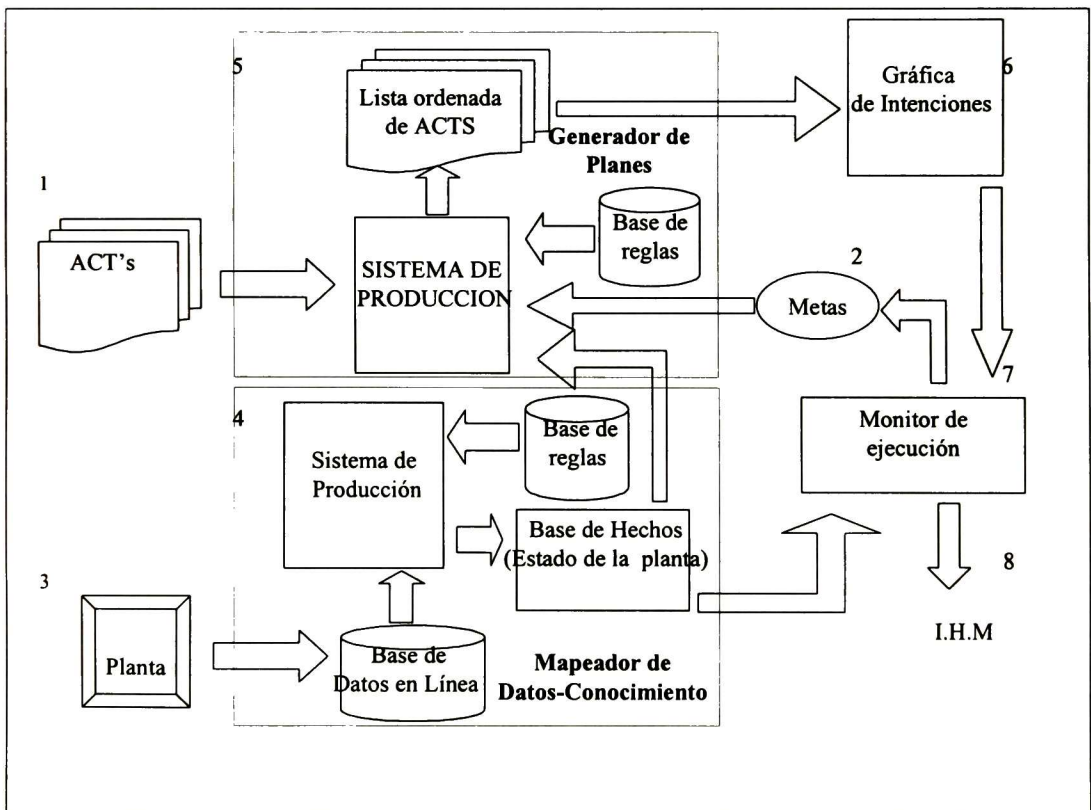


Figura 1.2 Arquitectura del asistente al operador

En la biblioteca de ACT's (1), se encuentra toda la información recolectada de la experiencia de los expertos en la operación de las CCC. Toda la información se escribe en un lenguaje formal llamado ACT. Los ACTS, junto con la(s) meta(s) (2) se dan como entrada al *agente generador de planes* (5). El *agente generador de planes* encuentra una solución que satisfaga la meta planteada por el operador de la central. El estado actual de la central se obtiene a través del *agente "mapeador" de datos a conocimiento* (4); el cual recibe los datos "crudos" de la central y con base a las reglas que obtiene de una base de datos puede saber en que estado se encuentran los diferentes sistemas que conforman la CCC. De la lista de ACTs se extrae un conjunto de acciones puntuales para generar un *gráfico de intenciones* (6) que es recibida por el *agente monitor* (7), el cual verifica el cumplimiento de las acciones [Valdez 01].

## 1.4 Planteamiento del problema

Durante los proyectos realizados en IIE para desarrollar planificadores, se ha observado que los planificadores tradicionales son deterministas. Se consideran reglas causales que son invariantes; las relaciones causales siempre ocurren conforme a lo esperado. Sin embargo, en la realidad esto no es así. Se ha observado que existen procesos donde pueden ocurrir comportamientos inciertos. Durante el arranque en caliente de una central eléctrica, por ejemplo, las condiciones termodinámicas en las que se encuentran los equipos hacen inciertas algunas de las tareas para llevar la planta a un nivel de operación normal. Los operadores de las centrales, realizan las tareas sin importarles la optimización, con el único propósito de normalizar la operación. Asimismo, existen equipos, como el domo, en los que siempre está presente la incertidumbre. Otro factor que provoca que exista incertidumbre en los sistemas de las centrales es el desgaste de los equipos.

Se debe pues considerar la incertidumbre presente en los sistemas, para obtener una acción que permita llegar de forma óptima a una meta dada. En particular, no se debe ignorar la incertidumbre en los procesos relacionados con la operación de una central eléctrica. Para este problema, se han hecho algunas aproximaciones con Redes Bayesianas y con la Lógica Difusa [Arroyo et al. 98] para diagnóstico de fallas. No obstante, no se cuenta con alguna arquitectura formal de un agente que solucione el problema de planificación en dominios donde esté presente la incertidumbre, tanto en los efectos de las



acciones como en las observaciones que se están realizando. Dicha arquitectura, en este tipo de dominios, debe ser capaz también de aprender del comportamiento (incierto) del sistema y del conocimiento del experto.

El grupo de trabajo del IIE dedicado al desarrollo de planificadores necesita contar con alguna arquitectura de un agente para obtener acciones y que pueda aprender, apropiado para el tipo de dominio descrito. Puntualmente, se debe contar con alguna herramienta de Inteligencia Artificial desarrollada a través de una arquitectura, e implementada en un prototipo para hacer pruebas. Con dichas pruebas se debe demostrar que es factible una solución eficiente que ataque el problema de planificación en dominios estocásticos. Dicho agente, debe contar con interfaces para cooperar con otros agentes de planificadores, para que pueda ser integrado al proyecto global del planificador.

## 1.5 Objetivos de la tesis

Para la solución del problema planteado anteriormente, la presente tesis tiene el objetivo principal de **desarrollar una técnica para resolver el problema de planificación dinámica en presencia de incertidumbre, mediante un agente de ayuda en la toma de decisiones, para la aplicación en una Termo-Eléctrica.** En la actualidad, hay suficiente marco teórico para desarrollar planificadores en dominios inciertos; aunque no existen aplicaciones reales conocidas. Las aplicaciones (la mayoría) que se han realizado son del tipo académico, en especial en la navegación de “robots”. Es por eso, que este objetivo se refiere en el contexto de desarrollar la técnica para ser utilizada en una aplicación real dentro de una CTE.

### **Objetivos particulares.**

Para el cumplimiento del objetivo general de la tesis, se deben cumplir con algunos objetivos específicos. Estos objetivos se enumeran a continuación:

#### **1) Establecer el dominio de aplicación**

Se debe establecer claramente cuál es la parte de la CTE que será estudiada para el desarrollo de la tesis. Es necesario definir correctamente un dominio donde se presente incertidumbre en las acciones realizadas y en las observaciones. En caso contrario, se

estaría hablando de un dominio determinista que se puede atacar con planificadores tradicionales [Agueda e Ibarzüengoytia 01 a].

## **2) Estudiar la herramienta de Inteligencia Artificial a utilizar**

De acuerdo al problema expresado, se observa la necesidad de contar con alguna herramienta de la IA que pueda utilizarse como máquina de razonamiento o generador de acciones, en dominios con incertidumbre. Los *Procesos de Decisión de Markov* (MDP) y los *Parcialmente Observables MDP* (POMDP), aceptados en la comunidad de Inteligencia Artificial, son de gran interés en la investigación sobre dominios con incertidumbre, por lo que una aproximación clara a una aplicación real será de gran aportación a la comunidad interesada en los MDP y POMDP. Se mostrará cómo éstos pueden ser útiles para solucionar el problema planteado en la tesis [Agueda e Ibarzüengoytia 01 a].

## **3) Especificar los requerimientos necesarios para utilizar la herramienta**

Se deben especificar los elementos que necesita la herramienta de IA a utilizar. Del dominio de aplicación se deben obtener todos los elementos necesarios para aplicar los MDP y los POMDP.

## **4) Desarrollar una arquitectura**

Otra aportación importante del presente trabajo es la arquitectura que se propone debido a que permite obtener acciones óptimas en dominios con incertidumbre. Se mostrará que esta arquitectura (AUDA: Acting on Uncertain Domain Architecture), es una arquitectura de un agente capaz calcular políticas de acción óptimas y determinar el estado actual del sistema, aun en presencia de incertidumbre. De esta forma AUDA puede auxiliar a otros agentes sugiriendo la acción óptima y/o el estado del sistema más probable. AUDA es de las primeras arquitecturas que son creadas para solucionar el problema de planificación en dominios con incertidumbre. Se mostrará esta arquitectura a la comunidad interesada para ser estudiada y discutida [Agueda e Ibarzüengoytia 01 b].

## **5) Desarrollar un prototipo de la arquitectura**

Se desarrollará un prototipo en donde puedan integrarse los módulos de la arquitectura propuesta. Se debe observar en el prototipo las características funcionales de un agente con AUDA. Este prototipo es de gran importancia para el grupo de trabajo del



IIE. Debe ser una aproximación muy cercana a la aplicación real en el dominio definido en el presente trabajo de investigación.

#### **6) Realizar pruebas con el prototipo**

Una vez desarrollado el prototipo, el grupo del IIE podrá hacer pruebas para verificar que el problema de planificación en dominios con incertidumbre haya sido correctamente atacado con la técnica propuesta en esta tesis. Se harán pruebas utilizando datos históricos reales y se presentará la arquitectura en condiciones de ser acoplada a una central.

#### **7) Concluir a partir del trabajo**

Dadas la herramienta de IA, el dominio, la arquitectura, el prototipo y las pruebas; se podrá hacer una evaluación objetiva del avance y aportaciones a partir de la presente investigación.

Se debe observar que dentro del objetivo general y los particulares, se establecen también los alcances del trabajo. Únicamente lo planteado en los objetivos, será desarrollado en la tesis.

## **1.6 Estructura de la tesis**

Para los Procesos de Decisión de Markov, y los Parcialmente observables, se han conservado las siglas del inglés, MDP y POMDP respectivamente.

Los índices utilizados a lo largo del documento, se han establecido que inicien en 0. De esta forma, se numeran los conjuntos de 0 en adelante. Esto se observa en conjuntos, vectores y matrices.

La presente tesis está organizada de la siguiente forma:

En el capítulo uno se presenta un panorama general del problema de planificación y su aplicación en el sector eléctrico. Primero se mencionan algunas aproximaciones que se han hecho en todo el mundo y en particular en México para Comisión Federal de Electricidad (CFE) a través del Instituto de Investigaciones Eléctricas. Se menciona la importancia que tiene la planificación aún en presencia de incertidumbre, debido a que es una tendencia actual.

En el capítulo dos se presentan los MDP y los POMDP como la forma apropiada para atacar el problema. Estos se definen formalmente y se describen cada uno de los elementos que lo componen. Se muestran los algoritmos fundamentales que son utilizados para resolver los procesos de decisión.

En el capítulo tres se describe el dominio de aplicación, el cual será el caso de estudio para mostrar una solución al problema de planificación con incertidumbre. Se hace una relación entre los elementos del dominio con los elementos de los procesos de decisión, logrando mostrar una metodología para el modelado del sistema físico.

En el capítulo cuatro se presenta la arquitectura AUDA. Ésta provee una solución al problema de planificación en dominios con incertidumbre. Se describe cada uno de sus módulos y se muestra cómo se utilizan las propiedades de los MDP y los POMDP para ser integrados en esta arquitectura.

En el capítulo cinco se muestran algunos experimentos realizados en el IIE con un prototipo de un agente que utiliza la arquitectura AUDA. Se presentan los resultados de los experimentos, los cuales son necesarios para demostrar que el problema de planificación en dominios estocásticos puede encontrar una solución a través de un generador de acciones óptimas, que pueden ser de ayuda para un operador o algún otro agente de planificación.

Para finalizar, en el capítulo seis se escriben las conclusiones del trabajo y se mencionan los trabajos futuros que se pueden desarrollar a partir de los presentes resultados.

## Capítulo 2 Procesos de Decisión con incertidumbre

---

En muchas aplicaciones es importante contar con un mecanismo para decidir las acciones que se deben realizar para alcanzar una meta, de tal manera que se obtenga al final la mayor ganancia. Se debe obtener una política óptima de acciones. Es necesario contar con dicha política, aún en presencia de incertidumbre sobre los efectos de las acciones y sobre el estado actual del sistema.

En este capítulo se presentarán los *Procesos de Decisión de Markov (MDP)* y los *MDP Parcialmente Observables (POMDP)*, como la manera de solucionar las necesidades anteriores. Se dan a conocer los algoritmos principales necesarios para solucionar los MDP y los POMDP.

---

## 2.1 Introducción

En diversas áreas en las que se deben tomar decisiones, existe cierta incertidumbre sobre los efectos de las decisiones que se toman. Muchos problemas de ingeniería se modelan de forma determinista, de tal forma que siempre se saben los efectos de las acciones realizadas. Se puede considerar que cuando se presiona un interruptor para mover un “robot” sobre una superficie de navegación, éste cambiara de lugar; lo cual, en la aplicación real no siempre es cierto. Existen factores que pueden provocar que no suceda el efecto esperado: el “robot” puede atorarse por alguna irregularidad del terreno, el interruptor puede fallar, la energía del “robot” puede estar agotada, el sistema mecánico puede fallar, etc. Es cierto pues, que en muchas aplicaciones existe incertidumbre referente a las acciones realizadas, por lo que se tiene un dominio con incertidumbre, sobre el cual se tienen que tomar decisiones con las que se pueda obtener el máximo de ganancias esperadas, para alcanzar una meta determinada.

En este capítulo se presentan los *Procesos de Decisión de Markov (MDP: Markov Decision Processes)*, para modelar el problema del cálculo de la política óptima en un dominio con incertidumbre. Estos procesos tienen la propiedad de *Markov*, es decir que la probabilidad de alcanzar algún estado dada una acción, depende solamente del estado actual y no de la historia.

En ocasiones, también se presenta incertidumbre para determinar el estado de un sistema. Por ejemplo, en el domo de una *Central de Ciclo Combinado (CCC)*, los sensores pueden estar indicando un nivel determinado de líquido; siendo la medición incorrecta debido a las condiciones termodinámicas presentes ( p.e. puede existir “burbujeo” dentro del domo) [Delgadillo 96]. Se presentan los *Procesos de Decisión de Markov Parcialmente Observables (POMDP: Partially Observable MDP)* para determinar el estado del sistema y calcular las acciones óptimas en cada estado, en dominios donde no se tiene la certidumbre del estado actual [Cassandra et al. 94]. Debido a que no existe una observación exacta o completa del ambiente, se dice que se tiene un dominio estocástico parcialmente observable. El agente puede realizar observaciones, no necesariamente confiables, a partir de las cuales se calcula una distribución de probabilidad sobre los estados en los que puede estar el sistema.



Los *MDP* y lo *POMDP* son utilizados típicamente en navegación de “robots”; aunque, se puede transportar para modelar problemas de controladores, observadores y planificadores. Además se pueden utilizar en otras áreas como la medicina, economía, mercadotecnia, diseño, deportes, juegos, etc. Cualquiera que sea el dominio, para modelar estos procesos se deben definir un conjunto de estados finitos del sistema, un conjunto de acciones que se pueden realizar sobre los estados y un conjunto de observaciones en cada estado; además de funciones de transición, de observación y recompensas [Agueda e Ibarгүйengoytia 01 a]. Estos elementos se definirán más adelante en el capítulo.

Los *POMDP* se utilizan como un modelo estocástico de un agente que estará interactuando con el mundo [Cassandra et al. 94]. Dada una acción desde un estado conocido del mundo, se puede determinar probabilísticamente el siguiente estado del agente y se puede sugerir la acción óptima. La estructura de un sistema que se puede representar con *POMDP* es la mostrada en la Figura 2.1 [Cassandra 98].

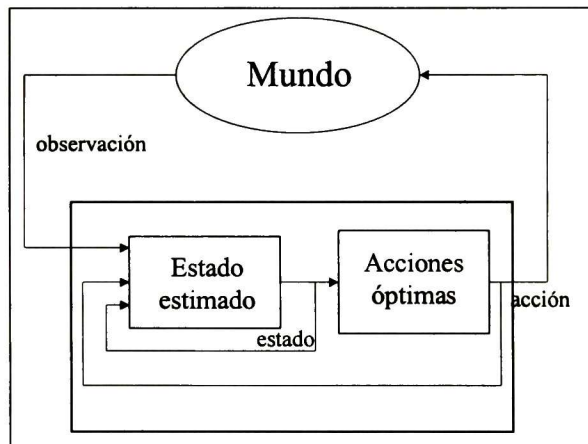


Figura 2.1 Estructura con POMDP

En esta figura se observa que el estado más probable depende de las observaciones y de las acciones realizadas; teniendo como entrada también una distribución probabilista de estados donde puede estar el sistema. Una vez estimado el estado más probable, se busca en el conjunto de acciones óptimas, la correspondiente a dicho estado. Esta acción es dada la mundo para ser ejecutada, repitiendo así el ciclo.

Existen algoritmos para resolver los *POMDP*, los cuales serán los utilizados en la construcción del prototipo de la arquitectura. En dominios donde es necesario respetar

restricciones de tiempo, los algoritmos se pueden modificar para cumplir con las condiciones de tiempo [Dean et al. 95].

## 2.2 Decisión con incertidumbre

En la aplicación de sistemas inteligentes, muchos dominios reales presentan incertidumbre, por lo que es necesario contar con algún agente que proporcione un conjunto de acciones adecuadas para alcanzar una meta predeterminada en dicho dominio. Es necesario que el agente, para calcular la mejor acción, tome en cuenta la incertidumbre presente, la utilidad esperada y el cumplimiento futuro de una meta dada.

La **teoría de la probabilidad** permite abstraerse del conocimiento histórico sobre el sistema que se va a modelar. No importa tener toda una red causal dada por la historia global del sistema, sino que permite tener únicamente las estadísticas obtenidas de la historia. Se cuenta con las relaciones de probabilidad entre las causas y los efectos dentro del sistema.

En ciertos puntos de la búsqueda de la solución se pueden tener más de una acción para escoger y todas haciendo factible la solución. Pero, existe una acción  $a_1$  tal que no existe alguna otra acción  $a_2$  donde la utilidad esperada de  $a_2$  sea mayor que la de  $a_1$ . La **teoría de la utilidad** sostiene que cada estado implica cierto grado de utilidad para el agente, y que el agente prefiere aquellos estados que le representan una mayor utilidad [Russel y Norvig 95]. Por ejemplo, en el recorrido de un grafo puede existir más de un camino para llegar a un vértice determinado, pero se puede considerar uno de ellos con mayor utilidad dependiendo de cuál representa la distancia más corta.

### **Definición 2.1: Utilidad esperada (UE)**

Sea  $H=[s_1, a_1, s_2, a_2, \dots, a_{n-1}, s_n]$  un camino que lleva a la meta, donde  $s_i$  y  $a_j$ , representan estados y acciones del sistema, respectivamente, para  $i=1, 2, \dots, n-1$  y  $j=1, 2, \dots, n$ . Se puede obtener la utilidad esperada (UE) de  $H$  como se muestra en la siguiente **función aditiva**:

$$UE(H) = R(s_1, a_1) + UE([s_2, a_2, s_3, \dots, a_{n-1}, s_n]) \quad (2.1)$$



En donde  $R(s_i, a_i)$  se le conoce como la *función de recompensa*, y su valor representa la ganancia de realizar la acción  $a_i$  en el estado  $s_i$ .

### **Definición 2.2 Políticas**

Una **política**  $\pi$  es una secuencia de acciones  $\pi = [a_1, a_2, \dots, a_{n-1}]$  donde  $a_i$  es la acción que se debe realizar en el estado  $s_i$ , para  $i = 1, 2, \dots, n$ .

Una **política estacionaria** es una situación en la que, por cada estado se especifica la acción que se debe tomar, la cual sólo depende del estado y no del tiempo. Una **política no estacionaria** es una situación en la que la acción que se debe tomar en algún estado depende también del tiempo en que se realiza la decisión [Kealbling et al. 98].

La política  $\pi$  que lleva a alguna meta  $g$  a través del camino  $H$ , es una **política óptima** si y sólo si no existe algún otro camino  $H'$  que lleve a la meta  $g$  tal que  $UE(H') > UE(H)$ .

## **2.3 Los Procesos de Decisión de Markov (MDP)**

El propósito de modelar un problema con los *MDP* es el de tener un agente que proporcione la política óptima, para llegar a la meta establecida, dados algunos criterios de recompensa y en presencia de incertidumbre. Con este modelo se obtendrán las mejores decisiones por cada estado con base a una función de recompensa y una de transiciones construida sobre bases probabilistas; con estas funciones se deben obtener las utilidades máximas esperadas en cada estado.

### **2.3.1 Definición de los MPD**

De manera formal, un modelo MDP consiste de la tupla  $(S, A, T, R)$ , donde:

- $S$  es un conjunto de estados finitos del mundo.
- $A$  es un conjunto finito de acciones que puede realizar el agente. En particular,  $A(s)$  es el conjunto de acciones que pueden ejecutarse en el estado  $s$ .
- $T: S \times S \times A \rightarrow P = \{x \in \mathcal{R} / x \leq 1 \text{ y } x \geq 0\}$  es una función de transición de estados. Por cada acción y estado del mundo, se tiene una distribución de probabilidad sobre los

estados del mundo a los que se puede llegar después de realizar la acción. Se define  $T(s' / s, a)$  como la probabilidad de llegar al estado  $s'$  desde  $s$  dada la acción  $a$ .

- $R: S \times A \rightarrow \mathcal{R}$  es una función de recompensa (reward). Por cada acción y estado del mundo, se asigna un número real. Se define  $R(s, a)$  como la recompensa de ejecutar la acción  $a$  en el estado  $s$ .

En este tipo de modelo, tanto el siguiente estado y la recompensa esperada dependen únicamente del estado actual del sistema y de la acción realizada y no de la historia. En otras palabras, los *MDP* cumplen con la propiedad de Markov [Cassandras y Lafortene]. Posteriormente se mostrará que las políticas pueden cambiar debido a una arquitectura en la que el agente irá modificando su conocimiento históricamente. No se estará violando la propiedad de Markov, ya que una vez calculada la política, esa ya no cambia con el tiempo hasta que se requiera calcular una nueva política dado el nuevo conocimiento.

### **Definición 2.3: *MDP de horizonte infinito***

Un *MDP* de horizonte infinito es aquel que modela dominios donde se requiere una evaluación de secuencias infinitas de recompensas en todos los estados de  $S$  [Puterman 94]. Utiliza algoritmos en el que no está acotado el número de iteraciones para los cálculos. La acción óptima se obtendrá de acuerdo al valor máximo de la suma infinita de recompensas

$$\sum_{t=0}^{k \rightarrow \infty} r_t \quad (2.2)$$

donde  $r_t$  es la recompensa en el tiempo  $t$ . Se debe notar que  $k$  tiende a infinito, de otra forma se estaría acotando el número de iteraciones.

### **Definición 2.4: *MDP desocontados (discounted MDP)***

Los *MDP* desocontados contienen un *factor de descuento (discount factor)*  $0 < \gamma < 1$  que se introduce para realizar los cálculos. Este factor garantiza obtener los resultados en un número finito de iteraciones [Kealbling et al. 98]. Por lo tanto se modifica la ecuación 2.2. como se muestra a continuación:

$$\sum_{t=0}^{k \rightarrow \infty} \gamma^t r_t \quad (2.3)$$

### 2.3.2 Cálculo de la política óptima

Se ha mencionado que a cada acción que se realiza en el modelo, se le atribuye una recompensa. En este sentido, los MDP obtendrán una política de acciones que logren maximizar la suma de recompensas al realizar dichas acciones. Para el presente trabajo, se considera un horizonte infinito al calcular la política, y se utilizan los MDP descontados.

#### Definición 2.5. Suma esperada de recompensas (SER)

Sea  $V_{\pi,t}(s)$  la *suma esperada de recompensas (SER)* ganadas hasta la iteración  $t$  (horizonte finito) al llevar al cabo la política **no estacionaria**  $\pi$ , entonces se define

$$V_{\pi,0}(s') = 0 \quad (2.4)$$

para toda  $s' \in S$ . Se define  $V_{\pi,t}(s)$  recursivamente como la ganancia acumulada cuando se inicia en el estado  $s$ :

$$V_{\pi,t}(s) = \left[ R(s, \pi_t(s)) + \gamma \sum_{s' \in S} T(s'|s, \pi_t(s)) V_{\pi,t-1}(s') \right] \quad (2.5)$$

donde  $\gamma$  ( $0 \leq \gamma < 1$ ) es el factor de descuento y  $\pi_t(s)$  es la acción que se debe realizar en el estado  $s$  en el tiempo  $t$  bajo la política  $\pi$ . Se debe observar que para la primera iteración la recompensa esperada queda dada por

$$V_{\pi,1}(s) = R(s, \pi_1(s)); \quad (2.6)$$

es decir, la recompensa de realizar la acción correspondiente a la política en el estado  $s$  en el que se está evaluando.

En el caso de **política estacionarias**, se puede calcular la utilidad (suma de recompensas) máxima esperada en cada estado como sigue:

$$V_t(s) = \max_{a \in A(s)} \left[ R(s, a) + \gamma \sum_{s' \in S} T(s'|s, a) V_{t-1}(s') \right]; \quad (2.7)$$

donde  $\gamma ( 0 \leq \gamma < 1 )$ . Se puede observar que se hace una evaluación por cada acción que se puede realizar en  $s$  y se toma sólo el valor máximo relacionada a una de las acciones.

Dada una función  $V: S \rightarrow \mathcal{R}$  de la SER, en un modelo de **horizonte infinito descontado**, la política óptima  $\pi_v$  se calcula como sigue:

$$\pi_v (s) = \underset{a \in A(s)}{\operatorname{arg\,max}} \left[ R(s, a) + \gamma \sum_{s' \in S} T(s' | s, a) V (s') \right] \quad (2.8)$$

donde  $\gamma ( 0 \leq \gamma < 1 )$  es el mismo factor de descuento que se utilizó para calcular la función  $V$ .

Hasta este momento, se han dado las definiciones de lo que se va a utilizar en el presente trabajo. Se calcularán políticas estacionarias utilizando los MDP descontados de horizonte infinito. A su vez, se ha mostrado la forma de resolver estos procesos de decisión. A continuación se mostrarán los algoritmos para resolver las Ecuaciones 2.7 y 2.8.

## 2.4 Algoritmos para los MDPs

Los algoritmos que se necesitan para el cálculo de la política óptima, dado un modelo con los MDP, son realizados a través de un método de *programación dinámica* llamada *iteración de valor* [Bellman 57]. Debido a que se están trabajando para MDP descontadas, se tiene una respuesta en un número finito de iteraciones [Kealbling et al. 98].

### 2.4.1 Cálculo de la máxima suma esperada de recompensas

El Algoritmo 2.1, descrito a continuación calcula para cada estado, el máximo valor de la SER. Para ejecutar el algoritmo, se cuenta con los conjuntos de estados  $S$  y de acciones  $A$ ; y la función  $R$  de recompensas. También se tienen los vectores  $V_t$  y  $V_{t-1}$  con el número de elementos igual a la cardinalidad del conjunto  $S ( \#(S) )$ , que contendrán el valor de la recompensa máxima esperada en la iteración  $t$  y  $t-1$  respectivamente. Se define una constante  $\gamma ( 0 \leq \gamma < 1 )$  como el factor de descuento.



```

1)  $V_t[s] \leftarrow 0$  para toda  $s$ 
2)  $V_{t-1} \leftarrow V_t$ ;
3) para  $s \leftarrow 0$  hasta  $\#(S)-1$  hacer
4)     para  $a \leftarrow 0$  hasta  $\#(A)-1$  hacer
5)         sum  $\leftarrow 0$ ;
6)         para  $s' \leftarrow 0$  hasta  $\#(S)-1$  hacer
7)             sum  $\leftarrow$  sum +  $T(s'|s, a) * V_{t-1}[s']$ ;
8)             val  $\leftarrow R(s,a) + \gamma * sum$ ;
9)             si (val > max) entonces max  $\leftarrow$  val
10)      $V_t[s] \leftarrow$  max;
11) si cambio(  $V_{t-1}, V_t$ ) entonces ir a 2

```

Algoritmo 2.1 Máxima suma esperada de recompensas

En la línea 1 se inicia el vector  $V$  con el valor de 0 para cada elemento. Dicho vector corresponde a la *SER* (ver Ecuación 2.4). En otras palabras, al inicio no se tiene ninguna estimación sobre las recompensas futuras. En las líneas subsiguientes, el algoritmo iterará para calcular la suma esperada de recompensas de acuerdo a la ecuación 2.7.

Para cada estado se hará el cálculo de la *SER*, lo cual se puede observar en la condición de la línea 3. De acuerdo a la ecuación 2.7, se debe tomar el valor máximo que se genere de la ecuación después de haber hecho los cálculos para cada acción; por lo tanto en la línea 4 se garantiza que se evalúe para toda acción del modelo. Ahora bien, para todo estado  $s'$  a los que se puede llegar (Línea 6) a partir del estado actual  $s$  dada la acción  $a$  se debe verificar y sumar las ganancias que se obtendría de todos los estados  $s'$

$$\sum_{s' \in S} T(s'|s, a) V_{t-1}(s'); \quad (2.9)$$

esto se realiza en la línea 7.

En la línea 8 se multiplica el *factor de descuento*  $\gamma$  a la sumatoria

$$\gamma \sum_{s' \in S} T(s'|s, a) V_{t-1}(s'); \quad (2.10)$$

y se le suma la recompensa obtenida al llevar al cabo la acción  $a$  en el estado actual. En la línea 9 se asegura que “max” contenga el valor máximo de la recompensa esperada, y en la línea 10 se asigna al vector  $V_t$  en el elemento correspondiente al estado actual  $s$ .

El algoritmo termina cuando ya no se presentan cambios entre  $V_t$  y  $V_{t-1}$ . Correctamente dicho, cuando la diferencia entre dos funciones sucesivas de valores es menor a un valor de error  $\varepsilon$  dado, conocido como el *error de Bellman* (línea 11).

Por lo tanto la función *cambio* queda definida como

*cambio*:  $V \times V \rightarrow \{Verdad, Falso\}$

$$V_1, V_2 \mapsto \text{cambio}(V_1, V_2) = \begin{cases} \text{Falso, si } \forall s \in S, \|V_1(s) - V_2(s)\| \leq \varepsilon \\ \text{Verdad,} & \text{en otro caso} \end{cases}$$

En trabajos realizados en el grupo de Michael L. Littman [Littman et al. 95], se ha demostrado que la complejidad de este algoritmo está en un orden polinomial, por lo que se está hablando de un algoritmo teóricamente eficiente.

### 2.4.2 Cálculo de la política óptima.

El Algoritmo 2.2 descrito a continuación obtiene las reglas de decisión de la política óptima; es decir, por cada estado, la acción que producirá la mejor utilidad. El resultado se verá en el vector  $\pi$ , cuya semántica es de la forma  $\pi(s) = a$ ; que significa que la acción  $a$  es la que se debe ejecutar en el estado  $s$  para obtener la política óptima. Para llevar al cabo este algoritmo se considera como hipótesis que ya se ha calculado el vector  $V$  de la *SER*; se debe contar con los conjuntos de estados  $S$  y de acciones  $A$ ; y la función  $R$  de recompensas. La política óptima es calculada de acuerdo a la ecuación 2.8. El algoritmo es descrito a continuación:

```

1) para s ← 0 hasta #(S)-1 hacer
2)   para a ← 0 hasta #(A)-1 hacer
3)     sum ← 0;
4)     para s' ← 0 hasta #(S)-1 hacer
5)       sum ← sum + T(s'|s, a) * V[s'];
6)     val ← R[s][a] + γ * sum
7)     si (val > max) entonces max ← val, action ← a
8) π[s] ← action;

```

Algoritmo 2.2 Política óptima

Para cada estado se calculará la acción óptima que se debe realizar, por lo que en condición de la línea 1 se incluyen todos los estados. De acuerdo a la ecuación, para toda



acción  $a$ , se debe tomar aquella que genere el valor máximo; por lo tanto en la línea 2 se han puesto la condición de que se haga el cálculo para toda acción del modelo.

Para todo estado  $s'$  a los que se puede llegar (Línea 4) a partir del estado actual  $s$  dada la acción  $a$  se deben sumar las ganancias que se obtendría de todos los estados  $s'$   $R(s, a) + \gamma \sum_{s' \in S} T(s'|s, a)V(s')$  (Línea 5). Por último, se suma la recompensa obtenida al llevar al cabo la acción  $a$  en el estado actual y se multiplica por el *factor de descuento* (Línea 6). En la línea 7 se asegura mantener en “max” el valor máximo de la recompensa esperada y en “action” la acción que generó dicho valor.

En la línea 8 se añade a la política  $\pi$  la acción que generó el máximo valor.

## 2.5 MDPs Parcialmente Observables (POMDPs)

Con los MDPs se pueden obtener las políticas óptimas para un determinado problema. La política se pudiera utilizar si, a través de un modelo, se conociera con certeza el estado en el que se encuentra el sistema en determinado momento. En algunos modelos existe incertidumbre para conocer el estado actual de éstos. Es conveniente entonces utilizar alguna técnica o herramienta para determinar el estado actual. Los POMDP proveen una forma de estimar de forma probabilista dicho estado y llevar a cabo un plan óptimo de acciones. Los POMDP son una extensión de los MDP, que proveen una solución apropiada al problema de planificación en dominios parcialmente observables [Cassandra et al. 94].

### 2.5.1 Definición de los POMDPs

Los POMDPs se pueden describir como la tupla  $(S, A, T, R, \Omega, O)$ , donde:

- $S, A, T$  y  $R$  se definen como en los *MDPs*.
- $\Omega$  es un conjunto finito de observaciones que el agente puede realizar del mundo.
- $O: \Omega \times S \rightarrow P$  es una función de observación. Por cada estado, se puede obtener la probabilidad de realizar una observación determinada. Se define a  $O(o/s)$  como la probabilidad de realizar la observación  $o \in \Omega$  cuando se está en el estado  $s \in S$ .

## 2.5.2 Cálculo de la distribución en estados

Con los POMDPs se obtiene una distribución de probabilidad  $E$  sobre los estados; donde  $E(s)$  es la probabilidad de encontrarse en el estado  $s$ . Esta distribución indicará por cada estado la probabilidad de encontrarse en éste. Esta distribución es modificada después de realizar una acción o bien cuando se tiene alguna observación. Se supone que cuando se realiza una acción, hay cambio de estado; así también, cuando se lleva a cabo una observación se tiene mayor seguridad para determinar el estado actual.

### 2.5.2.1 Distribución después de realizar una acción

Sea  $E$  la distribución sobre los estados, donde  $E(s)$  es la probabilidad de encontrarse en el estado  $s$ . Dada una acción  $a$ , se calcula probabilidad resultante  $E^a(s)$  considerando todas las posibilidades de llegar a  $s$  a partir de los demás estados (Figura 2.1) Se hace una suma de la probabilidad de llegar al estado  $s$  a partir del estado  $s_i$  multiplicado por la probabilidad  $E'(s_i)$  de encontrarse actualmente en el estado  $s_i$ , con  $i=0,1,2,\dots,\#(S)-1$ . Dicha distribución esta dada por:

$$E^a(s) = \frac{1}{\beta} \sum_{s' \in S | a \in A} T(s | s', a) E'(s') \quad (2.11)$$

[Kealbling et al. 98], donde  $E'$  es la distribución que existe actualmente y  $\beta$  es una constante para normalizar la distribución.

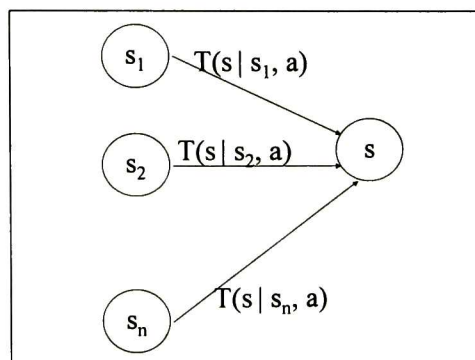


Figura 2.2 Probabilidad de alcanzar el estado  $s$ .

### 2.5.2.2 Distribución después de realizar una observación

Para obtener la probabilidad de  $E^o(s)$  después de que se ha realizado una observación  $o$ , se considera la probabilidad condicionada de realizar dicha observación en el estado  $s$ , multiplicada por la probabilidad  $E'(s)$  de encontrarse actualmente en dicho estado. Por lo tanto la distribución queda dada por:

$$E^o(s) = \frac{1}{\beta} O(o | s) E'(s) \quad (2.12)$$

[Kealbling et al. 98], donde  $E'$  y  $\beta$  son la distribución actual y una constante de normalización, respectivamente.

### 2.5.3 Cálculo de la acción óptima

Con los POMDP se puede calcular la política óptima. Se consideran las posibles acciones a ejecutar y las observaciones que se pueden realizar. Dada una creencia, (basada en  $E$ ) en un momento, se debe escoger la acción con la que se obtenga mayor ganancia. Esto se hace actualizando un valor nuevo de SER  $V'$  dado una SER  $V$ , a través de programación dinámica [Cassandra et al. 97].

La función  $V'$  obtiene un valor a partir de la información de estados  $E$ , y se define como:

$$V'(E) = \max_{a \in A} \left( r^a \cdot E + \gamma \sum_{o \in \Omega} \Pr(o | E, a) V(E_o^a) \right) \quad (2.13)$$

donde  $r^a$  es un vector de recompensas ( $r^a(s)$  = la recompensa de realizar la acción  $a$  en el estado  $s$ ),  $\Pr(o|E,a)$  es la probabilidad de hacer la observación  $o$  dada la distribución  $E$  y la acción  $a$ ; y  $E_o^a$  es la distribución de estados después de que ha ocurrido  $o$  y  $a$ .

$V'(E)$  se debe calcular, bajo el estado actual de creencias  $E$ , cada vez que se ha realizado una acción  $a$  y observado  $o$ . Con esto se obtiene la acción óptima en el estado actual de creencias.

De la ecuación 2.13, se define que [Cassandra 94, 97]:

$$r^a \cdot E = \sum_{s \in S} R(s, a) E(s) \quad (2.14)$$

$$E_o^a(s) = \frac{1}{\beta} \left( \Pr(o | s, a) \sum_{s' \in \mathcal{S}} T(s | s', a) E(s') \right) \quad (2.15)$$

donde  $\beta$  es una factor de normalización.

Se observa en la ecuación 2.15 una composición de las ecuaciones 2.11 y 2.12:

$$E_o^a(s) = \frac{1}{\beta'} \left( \Pr(o | s, a) E^a(s) \right) \quad (2.16)$$

Sea  $E^a$  la distribución antes de realizar la observación en 2.12. Suponiendo que se ha hecho la acción  $a$  se tiene que  $\Pr(o | s, a) = O(o | s)$ . Entonces:

$$E_o^a(s) = \frac{1}{\beta''} \left( O(o | s) E^a(s) \right) \quad (2.17)$$

por lo tanto,

$$E_o^a(s) = \frac{1}{\beta''} E^o(E^a(s)) \quad (2.18)$$

La distribución  $E_o^a$  utilizada para el cálculo de políticas con los POMDP es una composición de la distribución después de una observación y/o después de una acción. Se calcula la distribución a partir de una distribución anterior, y con base a una observación o acción dada. En otras palabras, si se lleva a cabo una acción y posteriormente se realiza una observación modificando las distribuciones de acuerdo a las ecuaciones 2.11 y 2.12 respectivamente; entonces la distribución resultante se puede calcular también con la Ecuación 2.15. Con esta idea, se ha creado la arquitectura mostrada en el capítulo 3.

## 2.6 Algoritmos para estimar el estado en los POMDP

La planificación sobre dominios estocásticos con incertidumbre es un problema de los llamados “difíciles” [Brassard 97]. En algunos casos caen dentro del conjunto de los P-difícil (PSPACE-hard) [Madani 99]. Se dice entonces que es posible tener algoritmos rápidos. Algunos algoritmos conocidos son los que resuelven el problema a través de *iteración de valor* [Sawaki y Ichikawa 78][Cassandra et al.94], *iteración de política* [White y Scherer 89], *representaciones estructurales* [Boutilier y Poole 96]; y *aproximaciones*



[Zhang y Liu 96]. Hay algunos que son modificaciones a los de *programación dinámica* [Cassandra et al 97]. En la literatura, los algoritmos se han clasificado como exactos y aproximaciones [Lusena et al. 01][Cassandra 97]. Los que convergen en tiempo polinomial no dan garantía de encontrar las políticas óptimas [Lusena et al. 01], por lo que son algoritmos de aproximación o de Monte Carlo [Brassard y Bratley 97].

Los algoritmos exactos para resolver los POMDP están en orden exponencial (NP). Sin embargo, se pueden utilizar heurísticas para reducir la complejidad [Lusena et al 01][Cassandra 98]. Uno de los algoritmos exactos de gran eficiencia es el de *Poda Incremental (Incremental Pruning)* el cual está en orden del tamaño del conjunto de estados, acciones y observaciones dadas como entrada [Cassandra 98]. Algunas otras **heurísticas** pueden ser:

- 1) reducción de estados parecidos,
- 2) reducción de número de acciones y
- 3) manipulación del número de iteraciones (hacerlos de horizonte finito o cambiar el valor de *error de Bellman*).

Para esta tesis se ha dividido el problema de obtener la distribución, calculándola por separado dada una acción y dada una observación. Es decir, se resuelven las ecuaciones 2.11 y 2.12 por separado. A continuación se presentan dos algoritmos que permiten obtener una distribución probabilista sobre los estados a través de las ecuaciones mencionadas. El Algoritmo 2.3 es utilizado para calcular la distribución después de que se ha ejecutado una acción; el Algoritmo 2.4, cuando se ha realizado una observación. Estos algoritmos tienen un tiempo de respuesta polinomial, por lo que son algoritmos eficientes. Ambos se encuentran en orden cuadrado del número de estados del modelo.

```

1) para s ← 0 hasta #(S)-1 hacer
2)     sum ← 0
3)     para s' ← 0 hasta #(S)-1 hacer
4)         sum ← sum + T(s | s', a) * E'[s']
5)     E[s] ← sum
6) E ← E / β
    
```

Algoritmo 2.3 Distribución de estados dada una acción

Para el Algoritmo 2.3 se considera que se conoce la distribución de estados  $E'$  existente antes de que ocurriera la acción. Se observa que en el algoritmo se está haciendo el cálculo de la ecuación 2.11, la cual obtiene una distribución de probabilidad sobre los estados del modelo. Para todo estado  $s \in S$ , se le calcula la probabilidad de que el sistema se encuentre en dicho estado una vez que ha ocurrido una acción  $a$ . En la línea 1 se observa la condición de “para todo estado”

Para calcular las probabilidades en cada estado  $s$ , se deben sumar cada una de las probabilidades de alcanzar el estado  $s$ , dada la acción  $a$ , desde cualquier estado  $s'$  del modelo multiplicada por la probabilidad de encontrarse en  $s'$ . Esta sumatoria se presentan en las líneas 3 y 4. El resultado de la sumatoria es asignado a la nueva distribución  $E$  (Línea 5).

En la línea 6 se normaliza la distribución de estados, con el fin de tener que:

$$\sum_{s \in S} E(s) = 1 \quad (2.19)$$

Es decir que:

$$\beta = \sum_{s \in S} E(s) \quad (2.20)$$

En el Algoritmo 2.4 se hace el cálculo de la ecuación 2.12. Calcula, por cada estado  $s$ , la probabilidad de estar en dicho estado dada una observación  $o$  (líneas 1 y 2). Por último se normaliza la distribución usando el factor  $\beta$ .

1) <b>para</b> $s \leftarrow 0$ <b>hasta</b> $\#(S)-1$ <b>hacer</b> 2) $E[s] \leftarrow O(o s) * E[s];$ 3) $E \leftarrow E / \beta;$
--

Algoritmo 2.4 Distribución de estados dada una observación

**Ejemplo.** Se supone que se tiene el modelo donde se pueden realizar las acciones “derecha” “izquierda” o “parar” expresado en la Figura 2.2. En la figura, los arcos representan las transiciones ocurridas debido a las acciones, colocando en el paréntesis la probabilidad de que realmente ocurra la transición deseada. En caso contrario el modelo realiza la transición contraria. En el caso de realizar la acción “parar”, el modelo queda siempre en el estado donde se realiza dicha acción.



En el ejemplo, la meta es el estado  $s_2$ , y se pueden hacer dos tipos de observaciones: “no meta” (nm) y “meta” (m). Se observa “meta” en el estado  $s_2$  y “no meta” en los estados  $s_0, s_1$  y  $s_3$ .

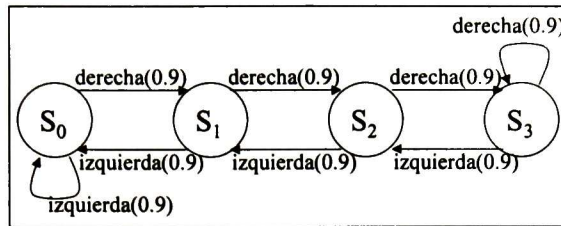


Figura 2.2: Ejemplo.

Se asume que al inicio se puede estar en cualquier estado que no es la meta. Por lo tanto, se tiene la distribución  $E=[0.333 \ 0.333 \ 0.000 \ 0.333]$ .

Si se realiza la acción de ir a la *derecha* y se observa que no se ha llegado a un estado meta (nm), se obtiene una nueva distribución de estados  $E=[0.100 \ 0.450 \ 0.00 \ 0.450]$ .

El cálculo se realiza de la siguiente manera:

$$E^a(s_0) = \frac{1}{\beta} \sum_{s \in S} T(s_0 | s, "derecha") E(s) = \frac{1}{\beta} (0.1E(s_0) + 0.1E(s_1)) = \frac{1}{\beta} 0.2(0.333)$$

$$E^a(s_1) = \frac{1}{\beta} \sum_{s \in S} T(s_1 | s, "derecha") E(s) = \frac{1}{\beta} (0.9E(s_0) + 0.1E(s_2)) = \frac{1}{\beta} 0.9(0.333)$$

$$E^a(s_2) = \frac{1}{\beta} \sum_{s \in S} T(s_2 | s, "derecha") E(s) = \frac{1}{\beta} (0.9E(s_1) + 0.1E(s_3)) = \frac{1}{\beta} (0.333)$$

$$E^a(s_3) = \frac{1}{\beta} \sum_{s \in S} T(s_3 | s, "derecha") E(s) = \frac{1}{\beta} (0.9E(s_2) + 0.9E(s_4)) = \frac{1}{\beta} 0.9(0.333)$$

Ahora bien, se ha realizado la observación de “no meta” (nm), por lo que la distribución queda dada por:

$$E^o(s_0) = \frac{1}{\beta'} O("nm" | s_0) E^a(s_0) = \frac{1}{\beta'} \left( \frac{1}{\beta} 0.2(0.333) \right) = \frac{1}{\beta'} 0.2(0.333)$$

$$E^o(s_1) = \frac{1}{\beta'} O("nm"|s_1)E^a(s_1) = \frac{1}{\beta'} \left( \frac{1}{\beta} 0.9(0.333) \right) = \frac{1}{\beta''} 0.9(0.333)$$

$$E^o(s_2) = \frac{1}{\beta'} O("nm"|s_2)E^a(s_2) = \frac{1}{\beta'} 0(0.333) = 0$$

$$E^o(s_3) = \frac{1}{\beta'} O("nm"|s_3)E^a(s_3) = \frac{1}{\beta'} \left( \frac{1}{\beta} 0.9(0.333) \right) = \frac{1}{\beta''} 0.9(0.333)$$

Si  $\beta=0.666$ , la distribución queda normalizada y se tiene que  $E^o(s_0)=0.1$ ,  $E^o(s_1)=0.450$ ,  $E^o(s_2)=0$  y  $E^o(s_3)=0.450$ .

Realizando el mismo procedimiento, si se realiza la acción de ir a la *derecha*, y se observa que no se ha llegado a un estado meta (*no meta*), se tiene la distribución  $E=[0.100 \ 0.164 \ 0.000 \ 0.736]$ . De tal manera que se va creando una distribución donde existe un estado más probable en que puede estar el sistema.

Se ha mostrado hasta este punto que se puede calcular una política óptima utilizando los MDP a través de los algoritmos 2.1 y 2.2. Asimismo, se ha dividido el problema del cálculo de la distribución de estados, calculándolo cuando a ocurrido una acción o bien cuando ha ocurrido una observación. Esto se realiza con los algoritmos 2.3 y 2.4. Los cuatro algoritmos mostrados convergen en tiempo polinomial y son la base con la que se resuelven los procesos de decisión en la arquitectura que se mostrará en el Capítulo 4.

# Capítulo 3 Modelado del dominio de aplicación

---

El domo de una Central Termo-Eléctrica presenta incertidumbre tanto en los efectos de las acciones como en las observaciones dadas por los instrumentos de medición, durante la operación de la central. En este dominio existe la necesidad de obtener acciones óptimas para el operador.

Intuitivamente, se puede pensar que el dominio es adecuado para que el problema pueda ser resuelto utilizando los MDP y los POMDP.

En este capítulo se describirá y se estudiará el dominio de aplicación. A partir de este estudio, se muestra una metodología para poder modelar el sistema físico a cadenas de Markov. De esta manera se pueden establecer los componentes necesarios para utilizar los MDP y los POMDP

---

## 3.1 Introducción

En trabajos desarrollados en el IIE con relación a planificadores, se ha observado incertidumbre en las transiciones y observaciones realizadas en el dominio que en este capítulo se muestra. En este tipo de dominios, el problema de planificación obliga a buscar nuevas técnicas que hagan factible la solución. Los planificadores se han venido desarrollando para centrales termoeléctricas, y en este sentido es conveniente hacer la distinción entre una Central de Ciclo Combinado (CCC) y una Central Termo-Eléctrica Convencional, a la cual simplemente se le ha llamado CTE en el presente documento. La principal diferencia entre las CCC y las CTE consiste en que las primeras tienen una central *turbogas*, además de la central *termoeléctrica* convencional. Estas dos están interconectadas por un recuperador de *calor generador de vapor*. En la central *turbogas*, la pérdida de calor más importante se presenta en los gases de salida que se tiran a la atmósfera, por lo que se aprovecha esta energía (calorífica) para generar vapor y posteriormente impulsar a una turbina de vapor.

En una CTE, el calor necesario para generar vapor es proporcionado por la combustión de combustible.

Debido a que son de gran importancia para la Comisión Federal de Electricidad (CFE), se ha decidido trabajar sobre este tipo de centrales. Actualmente en México la mayor parte de las centrales son CTE por lo que es necesario trabajar para mejorarlas. Esto no es problema cuando se requiera trasladar la técnica desarrollada en la tesis a una CCC, ya que los componentes de la CTE están en una de CCC; únicamente se deberán modificar algunos valores obtenidos a partir de datos históricos y de la especificación de fabricantes de turbinas. El nivel de abstracción de la técnica es alto, por lo que no es necesario especificar cuál es la fuente de energía que se utiliza para generara el vapor.

En la CTE, se transforma el *agua de alimentación* de su estado líquido a su estado de vapor bajo ciertas condiciones de presión y temperatura, para ser aprovechado por una turbina de vapor para la realización de un trabajo mecánico a partir del cual se genera electricidad. El flujo de líquido-vapor experimenta un ciclo cerrado dentro de la central. Este ciclo, llamado *Ciclo Termodinámico de Rankine* (CTDR), lleva de regreso el líquido

condensado del vapor utilizado en la turbina, para ser evaporada nuevamente. La Figura 3.1 ilustra el ciclo.

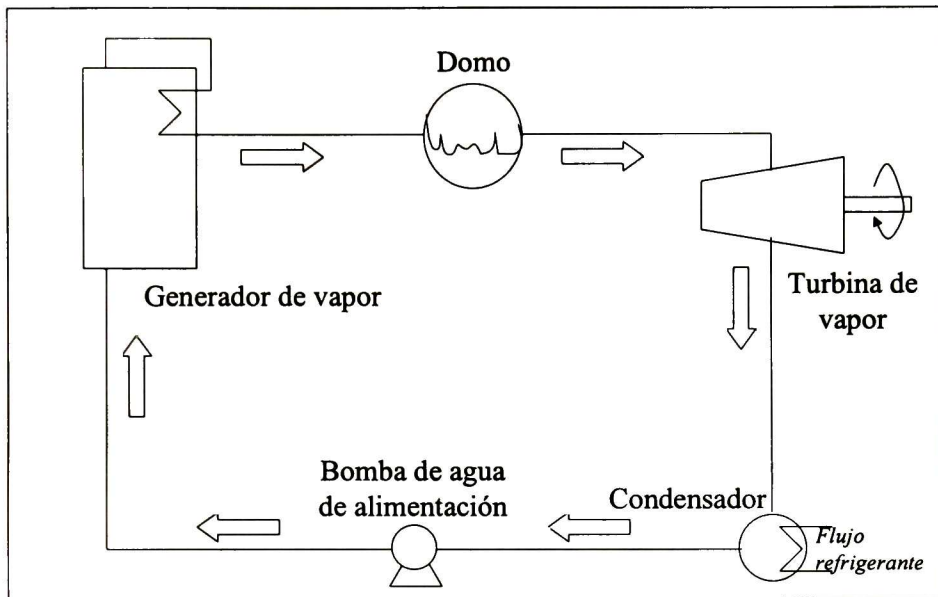


Figura 3.1 Elementos del ciclo Rankine

En el condensador se encuentra un tanque llamado *pozo de condensador*. Ahí se encuentra el agua almacenada en su fase líquida. A través de la *Bomba de agua de alimentación* este líquido es sustraído y alimentado al generador de vapor. Este generador transforma el condensado en vapor en condiciones de presión y temperatura de saturación<sup>2</sup> Este flujo lleva una parte de vapor y una parte líquida, y es separado el vapor dentro de un tanque especial llamado domo. En seguida el vapor pasa por un sobrecalentador para que tenga las condiciones que requiere la turbina de vapor donde las energías interna, de presión y cinética del vapor, generan el trabajo mecánico. Al salir de la turbina, con pérdidas en presión y temperatura, entra al condensador en donde, mediante fluido refrigerante, es condensado; lo que permite ser bombeada e iniciar el ciclo nuevamente.

<sup>2</sup> El agua toma una temperatura que supera la condición de presión, para que el agua hierva. .



Dentro de este ciclo para generar electricidad, existen muchos procesos complicados. El diagrama presentado en la Figura 3.1, es uno muy simplificado. Debido a las condiciones termodinámicas presentes y al desgaste de equipo, es incierto conocer un comportamiento exacto sobre las variables relacionadas con el domo. Esto se discutirá a continuación.

### 3.2 El Domo de una CTE

En el interior del domo se presenta la dinámica más rápida del proceso de generación de vapor, por residir el agua en su interior en dos fases: líquida y vapor. En el domo se debe separar el vapor y el líquido; utilizando el vapor para las turbinas y el agua se vuelve a calentar para generar más vapor. La temperatura de ebullición del agua está

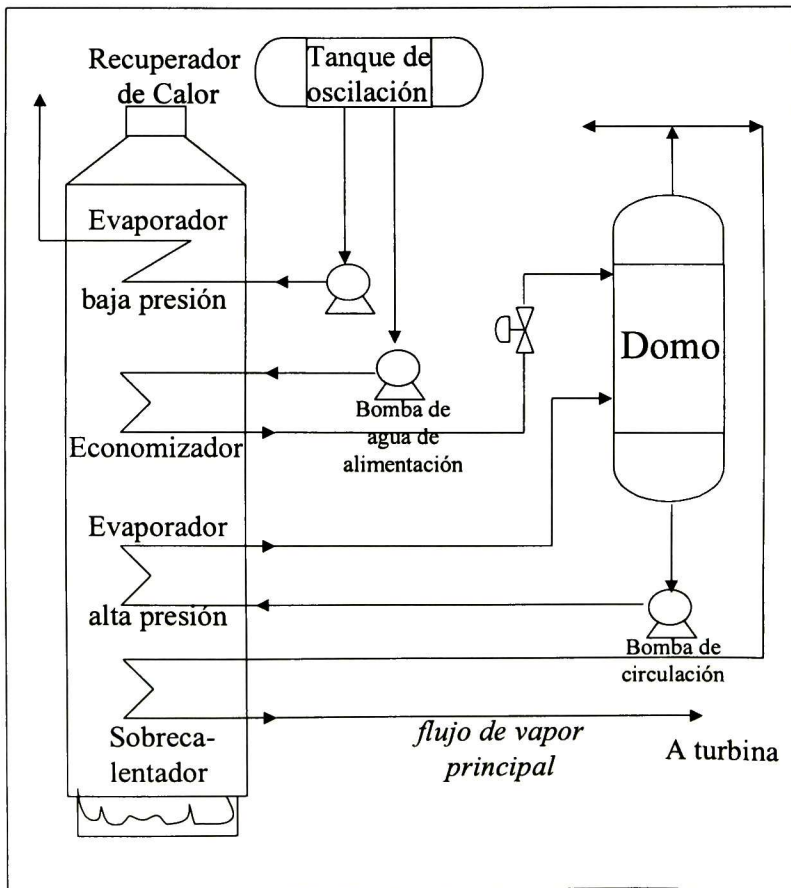


Figura 3.2 Flujo de agua y el domo

determinada por la presión del vapor en el domo (a mayor presión, le corresponde mayor temperatura de ebullición).

Un diagrama simplificado del domo y el flujo que sigue el agua (en estado líquido y vapor) es el mostrado en la Figura 3.2. El objetivo del sistema mostrado en la figura es suministrar el agua precalentada por el *economizador*, donde el calor de la combustión elevará la temperatura del agua hasta el punto de saturación generando así una mezcla de líquido/vapor. Esta mezcla asciende por los tubos de pared, gracias a la diferencia de presión, hasta llegar al domo, donde será separada la mezcla de líquido/vapor. En el *evaporador de alta presión* se calienta el agua para pasarla a su estado de vapor. Cuando se separa el vapor del agua, éste se envía a la turbina, pasando por el *Sobrecalentador* para aumentar la energía del vapor.

El domo de vapor es un recipiente cilíndrico de gran espesor que recibe, almacena y distribuye el agua de alimentación por los tubos del generador de vapor debiendo mantener siempre un nivel de agua seguro para su correcta operación. El domo tiene instaladas cuatro válvulas de seguridad que se abren en secuencia escalonada para desfogar la presión cuando ésta se incrementa muy arriba de su valor nominal de operación (nótese que estas válvulas son de seguridad)[CAOI 98]. En el domo, internamente se encuentra el agua en dos fases, una en forma líquida y la otra en forma de vapor húmedo. La humedad del vapor es eliminada por tres dispositivos: primero por unos separadores ciclónicos, posteriormente por unos secadores de lámina corrugada y finalmente por un tamiz o secador final.

Sin embargo, la cantidad de vapor generada y de líquido en el domo, es difícil de determinar. Esto afecta a las variables de presión y cantidad de flujo de vapor principal. Debido a esto, los desarrolladores de modelos han puesto gran interés en esta parte del proceso. Se puede observar en la Figura 3.3, las entradas y salidas (principales) de agua en el domo. Un recuento de flujos, tomando en cuenta todas las aportaciones y las derivaciones, indica que el total de flujo de vapor ( $G_{VR}$ ) en el domo de recuperador de calor será [Delgadillo 96]:

$$G_{VR} = G_{SD} \times Z_{VDR} + G_{WAD} \times Z_{WAD} - G_{VPEG} \quad (3.1)$$

donde  $G_{SD}$  es la cantidad de flujo que sube del *evaporador de alta presión*,  $G_{WAD}$  es la cantidad de flujo de agua de alimentación,  $G_{VPEG}$  es el flujo de vapor que sale del domo; y

$Z_{VDR}$  y  $Z_{WAD}$  son fracciones de vapor, es decir, el porcentaje del flujo que es convertido (o mantenido) en estado de vapor.

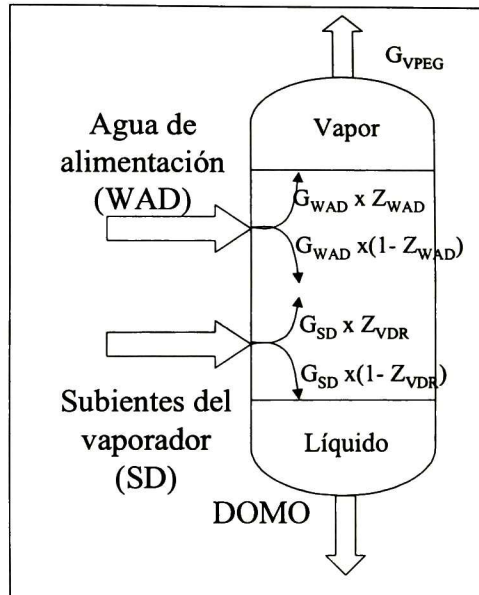


Figura 3.3 Vapor y líquido en el Domo

La ecuación 3.1 es correcta en condiciones estables. Sin embargo, el movimiento combinado de los equipos pertenecientes a las turbinas, por operación o falla, trae consigo cambios intempestivos en la presión. Esto tiene como consecuencia la aparición de fenómenos alternados como el incremento o decremento en el flujo de vapor y la posibilidad de la presencia de burbujas en el interior del líquido almacenado en el domo [Delgadillo 96]. En condiciones estables de operación existe una curva de equilibrio presión-temperatura de ebullición, la cual cambia si se realizan movimientos de carga o velocidad en las turbinas, o cuando se presenta algún tipo de disturbio. Si la presión aumenta, disminuye momentáneamente la rapidez de evaporación. Si la presión disminuye, el líquido en el interior del domo tendrá una entalpía<sup>3</sup> mayor que la que le corresponde para el nuevo punto de ebullición, por lo que el exceso de energía térmica es liberado mediante la evaporación de éste, provocando la presencia de burbujas.

<sup>3</sup> Entalpía: Energía interna más el trabajo desarrollado.  $H = E_{interna} + PV$

Los cambios bruscos de flujo, de temperatura o de presión, hacen incierto los efectos de las acciones al domo. Por ejemplo, si la presión disminuye, se provoca burbujeo dentro del domo, y los sensores pueden indicar un nivel alto de agua (no es agua, son burbujas), por lo que se puede pensar en disminuir la cantidad de agua de alimentación. La acción anterior, es contraria a la que se debe realizar.

Tomando en cuenta que existe la necesidad de un planificador en este dominio, es conveniente proponer una técnica para hacer planificación en dominios con incertidumbre. El control en el domo se complica debido a la incertidumbre. Debido a las especificaciones de fabricantes de turbinas, se han tomado las variables de *presión en el domo (pd)* y el *flujo de vapor principal (fvp)* como las variables que tienen un comportamiento incierto; esta idea ha sido apoyada por los expertos en el modelado de estos sistemas. A continuación se mostrará formalmente que se puede obtener la técnica para solucionar el problema en este dominio. Se definirá cada uno de los elementos para los POMDP, presentes en este dominio.

### 3.3 Componentes de los POMDP en el dominio

Como se mencionó en la Definición 2.2.1, los MDP están constituidos por la tupla  $(S,A,T,R)$ . Por otro lado, por la Definición 2.4.1 los POMDP están formados por la tupla  $(S,A,T,R,\Omega,O)$ .

Por lo tanto, para que los MDP y los POMDP puedan ser utilizados en el dominio propuesto, deben contener los seis elementos mencionados [*Agueda e Ibarzüengoytia 01*]:

**Un conjunto finito de estados ( $S$ ).** Este conjunto es obtenido a partir de la discretización de dos variables que presentan incertidumbre en la operación (*flujo de vapor principal y presión en el domo*).

**Un conjunto finito de acciones ( $A$ ).** Este conjunto es formado por la combinación que se tiene de hacer un incremento o decremento en cada variable; o bien, el no moverlas.

**Una función de transición ( $T$ ).** La función es obtenida a partir de datos reales, creando una matriz donde se puede expresar la relación de cambio de estado después de realizar una acción.



***Funciones de recompensas(R).*** Estas funciones son asignadas por operadores humanos expertos con gran experiencia. Ellos asignan valores de recompensa a cada acción dependiendo del estado en que se encuentre el sistema.

***Un conjunto finito de observaciones ( $\Omega$ ).*** Este conjunto es formado por una discretización de las lecturas del par de variables antes mencionadas, dadas por los instrumentos de medición.

***Una función de observación.*** También este elemento puede ser obtenido del dominio. Representa el conocimiento de fallas de los sensores, expresadas en forma de probabilidad de observar un estado incorrecto.

Con estos elementos se deben poder solucionar el problema de planificación en presencia de incertidumbre. Se puede decir que un plan es un conjunto de pasos ordenados que se siguen para alcanzar una meta. Debe existir también una relación causal de tal forma que cuando sucede un evento, se debe saber qué acción llevar a cabo. El plan puede ser general cuando, fuera de línea, se obtienen todos los pasos necesarios. También se pueden calcular nuevos planes o modificar los existentes en tiempo real, esto sucede cuando ocurre algún evento en el sistema que obliga a tomar otras acciones a las planeadas; o bien cuando se requiere cambiar la meta a alcanzar. Se debe entender que siempre se busca que el plan obtenido sea óptimo bajo algún criterio de optimización.

Sin embargo, existe un tipo de problemas en el cual se debe obtener la acción óptima en un instante. En tiempo real, se debe ir obteniendo la acción en cada instante. Esto es a lo que se le conoce como “acting” Este último de problema es el que resuelve el trabajo presentado en la tesis. Se tiene un agente capaz de proporcionar al operador la mejor acción en cada instante. Se le sugerirá una acción de acuerdo al estado actual más probable, para que el operador lo lleve a cabo. Las acciones consisten en modificar los valores de las variables de *fvp* y *pd*; de esta forma se podrán tener valores de éstas al fin de alcanzar un punto que pertenezca al estado meta al que se quiera llegar.



### 3.3.1 Estados

Los MDP necesitan un conjunto finito de estados, por lo que en el dominio se deben identificar algunas variables a través de las cuales se puedan definir los estados del sistema. Las variables aquí estudiadas son el *flujo de vapor principal (fvp)* y la *presión del domo (pd)*. Se considerarán los valores que deben tener en un rango de operación de 50% de carga al 100% de carga de la CTE. De acuerdo a la especificación de fabricantes para una turbina de un solo domo para una CTE, los valores óptimos de estos valores son los que se muestran en la Tabla 3.1.

Presión (Kg /cm2)	Flujo (T/h)	Carga (%)
171.9	531	50
172.8	649	60
174.4	767	70
176.9	885	80
180.3	1003	90
184.5	1120	100

Tabla 3.1 Valores óptimos de operación

Debido, a que los MDP necesitan un conjunto finito de estados, se ha discretizado la cantidad de carga en unidades de 10% y se ha partido de 50% como carga mínima y 100% como carga máxima.

En la tabla se puede observar que existe una relación lineal entre la carga y la cantidad de flujo de vapor principal necesaria. Por cada 10% de porcentaje que se aumenta en la carga, el *fvp* aumenta en 118 (T/h). Por lo tanto, se tiene la siguiente ecuación diferencial:

$$\Delta F_{vp} = 11.8\Delta c \quad (3.2)$$

donde *c* es el porcentaje de carga. Teniendo un punto inicial donde *fvp*=531 (T/h) para una carga de 50 (%), la Ecuación 3.2, se resuelve como sigue:

$$\int_{531}^{F_{vp}} dF_{vp} = \int_{50}^c 11.8dc$$

$$F_{vp} - 531 = 11.8(c - 50) \quad (3.3)$$

$$F_{vp} = 11.8c - 59$$

Por lo tanto, al discretizar los valores de carga en rangos de 10%, se discretizan los valores de  $fvp$  en rangos de 118 unidades.

La variable  $pd$  se ha discretizado en rangos de 2 unidades. De esta forma, se puede graficar la Tabla 3.1 para obtener una curva recomendada de operación y para crear una cuadrícula dada la discretización de las dos variables. La Figura 3.5 muestra la cuadrícula creada. Cada cuadrícula representa un estado del sistema y es indizado de acuerdo al número que está en cada celda.

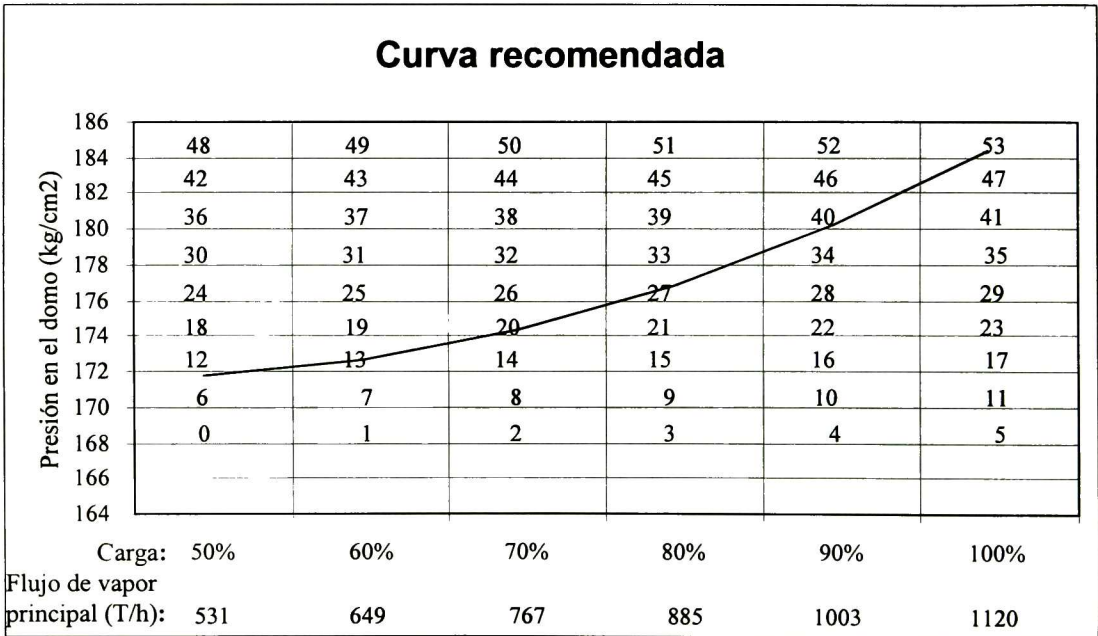


Figura 3.5: Discretización de variables

De tal forma que el conjunto  $S$  está definido por:

$$S = \{s_i \text{ para } i=0,1,\dots,53 \mid s_i \text{ es una celda creada por la discretización de } fvp \text{ y } pd \}$$

donde la discretización es la observada en la Figura 3.5.

### 3.3.2 Acciones

Como se vio en el punto anterior, las variables con que se trabajarán son las de  $fvp$  y  $pd$ . Por lo tanto las acciones que se deben realizar son para modificar los valores de éstas. Para hacer una modificación a alguna de ellas, se deben seguir una serie de pasos (aperturas y cierres de válvulas). Para el presente trabajo, se ha hecho una abstracción de esta serie de pasos.

El nivel de abstracción que se ha tomado permite establecer acciones respecto al incremento o decremento de cada una de las variables, la cual se le ha llamado “delta”( $\Delta$ ), de la variable. Estas acciones son mostradas en la Tabla 3.2.

Acción	$\Delta pd$	$\Delta fvp$
$a_0$	+	0
$a_1$	-	0
$a_2$	0	+
$a_3$	0	-
$a_4$	0	0

Tabla 3.2 Acciones en el dominio

Se dice que cuando  $\Delta pd=+$ , el valor de  $pd$  se debe aumentar (subir) de tal manera que se alcance un estado arriba del estado en el cual se realiza la acción, de acuerdo a la Figura 3.5. Así también se dice que cuando  $\Delta pd=-$ , el valor de  $pd$  se debe decrementar (bajar);  $\Delta fvp=+$  y  $\Delta fvp=-$  indican un incremento (derecha) y un decremento (izquierda) del  $fvp$ , respectivamente. Cuando se tiene que alguna  $\Delta$  es igual a 0, se tiene el caso en que no se hace modificación sobre la variable.

El conjunto de acciones  $A$  está definida por:

$$A=\{a_i \in \Theta \text{ para } i=0,1, \dots ,4\}$$

donde  $\Theta$  son los elementos de la Tabla 3.2, identificadas por la columna de acciones.

### 3.3.3 Función de transición

De acuerdo a la forma en que se han establecido los estados y las acciones, la función de transición representa las probabilidades de cambiar los valores de las variables, de tal manera que se llegue a cualquier celda distinta de donde se realiza la acción dada. Para toda celda de la Figura 3.5 y para cada acción realizada, se establece la probabilidad de alcanzar alguna otra celda.

Como se verá en el capítulo de resultados, esta matriz será obtenida de datos históricos reales. Se deben formar estadísticas con base a los datos reales, de tal forma que se puedan establecer el número de casos en que ocurre cierta transición. La función de transición  $T$  queda definida de la siguiente manera:

$$T: S \times S \times A \rightarrow P=\{x \mid x \geq 0 \text{ y } x \leq 1\}$$

$(s_i, s_j, a) \mapsto p$ , donde  $p$  es la probabilidad de llegar a valores de  $fvp$  y  $pd$  tal que se tenga un punto que esté dentro de la celda con índice  $i$  a partir de la celda con índice  $j$ , cuando se ha dado la acción  $a$ .

### 3.3.4 Función de recompensa

La función de recompensa es dada por un operador experto de CTE. Partiendo de la experiencia de esta persona y de acuerdo a la meta que se quiera alcanzar, se puede dar diferentes valores de recompensa a las acciones dependiendo de los valores de las variables (estado) que tenga el sistema. El conocimiento de los operadores está en forma cualitativa, por lo que se ha propuesto la Tabla 3.3 a manera de poder cuantificar las recompensas o el grado de importancia de las acciones dado un estado.

Cualitativo	Valor
	Recompensa
Exelente	10
Bueno	9,8
Aceptable	7,6,5
No recomendable	4,3
Peligroso	2,1
Nunca	0

Tabla 3.3 Valores de recompensas

La meta, en este caso, consiste en establecer un estado al cual se pretende llegar. Dicho estado debe ser el correspondiente a una celda de la cuadrícula tal que los valores de las variables estén en un punto óptimo de operación respecto a la Tabla 3.1 y de acuerdo a la carga que se requiera. Se han establecido que estas celdas son las indizadas con 6, 13, 20, 27, 40 y 53.

La función de recompensa  $R$  queda definida de la siguiente manera:

$$R: S \times A \rightarrow \{0,1,2,\dots,10\}$$

$(s_i, a) \mapsto r$ , donde  $r$  es el valor de recompensa, de acuerdo a la tabla 3.3, cuando se realiza la acción  $a$  en la celda con índice  $i$ .

### 3.3.5 Observaciones

En el contexto del dominio de aplicación, se ha establecido el conjunto de observaciones como los valores que se pueden registrar de las variables  $fvp$  y  $pd$ ,



discretizándolos al igual como se hizo para obtener los estados. Por lo tanto, en este caso, las observaciones que se pueden realizar es el estado dado por los instrumentos de medición, de acuerdo a la cuadrícula de la Figura 3.5.

Por lo tanto el conjunto de observaciones  $\Omega$  está definido por:

$$\Omega = \{o_i \text{ para } i=0,1,\dots,53 \mid o_i \text{ es una celda creada por la discretización de los valores obtenidos de los instrumentos de medición de } fvp \text{ y } pd \}$$

donde la discretización es la observada en la Figura 1.5.

### 3.3.6 Función de observación

La función de observación representa la probabilidad de que en instrumentos de medición se observe estar en algún estado. Dado un estado, esta función devuelve la probabilidad condicionada de que suceda cada elemento del conjunto de observación  $O$ . Las probabilidades son dadas por la experiencia de operaciones reales de plantas.

La función de observación  $O$  queda definida de la siguiente manera:

$$O: \Omega \times S \rightarrow P = \{x \mid x \geq 0 \text{ y } x \leq 1\}$$

$(o_i, s) \mapsto p$ , donde  $p$  es la probabilidad de observar en el estado  $s$  valores de  $fvp$  y  $pd$  tal que corresponda a un punto que esté dentro de la celda con índice  $i$ .





# Capítulo 4 Arquitectura para generar acciones en dominios con incertidumbre

---

Dados los MDP y los POMDP, se debe crear una arquitectura de un agente que utilice las propiedades de éstos. Dicha arquitectura debe ser capaz de obtener políticas óptimas y sugerir la mejor acción, dado el estado más probable.

Se presentará en este capítulo a AUDA (Arquitectura para generar acciones en Dominios con Incertidumbre). Se describirán los módulos que la componen y la relación entre ellos.

---

## 4.1 Introducción

Como se vio en el Capítulo 1, en el IIE se está desarrollando un planificador para la operación de plantas generadoras de electricidad [Ibargüengoytia y Reyes 01]. Para este tipo de dominio, es conveniente utilizar una plataforma de planificación multiagentes para resolver los problemas complejos de planificación. Cada agente tiene una tarea específica a realizar, la cual le servirá otro agente para alcanzar una meta global. Se le llaman *células* a cada uno de estos agentes [Wilkins y Karem 98]. Puede existir una jerarquía en la forma de interactuar de los agentes. Un agente, para alcanzar su meta, puede utilizar otros agentes para que lleven a cabo subplanes.

La *Arquitectura para generar Acciones en Dominios con Incertidumbre* (AUDA: *Acting on Uncertain Domain Architecture*) se propone en este trabajo, para un agente que aborda el problema de planificación cuando existe incertidumbre sobre los efectos de las acciones y sobre el estado del sistema [Agueda e Ibargüengoytia 01].

## 4.2 Descripción general de AUDA

El agente bajo la AUDA cooperará con otros agentes para llegar a la meta. En AUDA se sugieren acciones y reconocen estados; pero necesita de alguien que verifique los resultados y lleve a cabo las acciones propuestas. A su vez, puede ser utilizado por un agente que no haya considerado la incertidumbre, pero que, en dado caso, la incertidumbre se presente; o bien cuando el agente requiera de un *observador*.

En la Figura 4.1 se ilustra lo anterior. El agente 1, gestiona un nivel de planificación tal que no tiene procesos de decisión para dominios donde existe incertidumbre. Por lo tanto, en un momento dado necesita reconocer el estado actual del sistema y obtener una política óptima para alcanzar la meta que requiera considerando la incertidumbre presente; esto lo realiza utilizando la arquitectura AUDA.

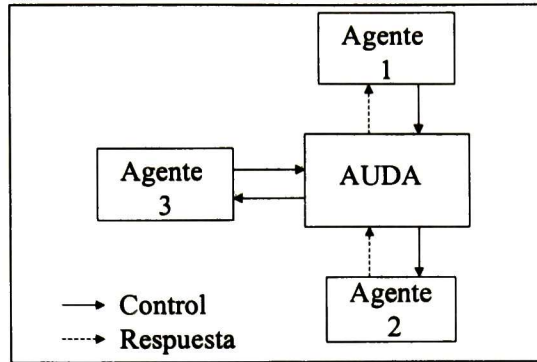


Figura 4.1 Agente AUDA en plataforma multiagente.

Por otro lado, el agente con AUDA, una vez que ha determinado una acción por realizar, no puede ejecutarla. Así que ordena al *Agente2* que ejecute la acción. El *Agente3*, es establece la meta y ordena al agente con AUDA decidir sobre la mejor acción, siendo el *Agente 3* quien llevara a cabo la acción.

En el dominio que se está trabajando, se ha establecido que AUDA interactuará con un agente del tipo *Agente3*. En este caso, este agente tiene la característica de ser un *agente humano*. El factor humano estará presente durante la operación de una Central Termoeléctrica (CTE). En otras aplicaciones, se ha visto la conveniencia de tener arquitecturas de agentes en capas, donde quedan bien definidas las capas tanto de las máquinas como los hombres donde pueden operar; y las relaciones entre estas capas [Bayouth et al. 97].

Un diagrama de especificación general de AUDA se puede ver en la Figura 4.2. Se destacan tres módulos importantes. El módulo de los *Procesos de Decisión* es el encargado de estimar el estado actual del sistema, calcular las políticas óptimas y por lo tanto sugerirle una acción al operador (*ACTOR*), que en este caso es un *agente humano*. El módulo de *Conocimiento* es donde se encuentra contenida toda el conocimiento en forma estadística, que serán utilizadas por el MPD. El módulo de *Aprendizaje* tiene como función observar el comportamiento del ambiente y las acciones que realiza el operador, modificando los registros estadísticos contenidos en el módulo de *Conocimiento*.

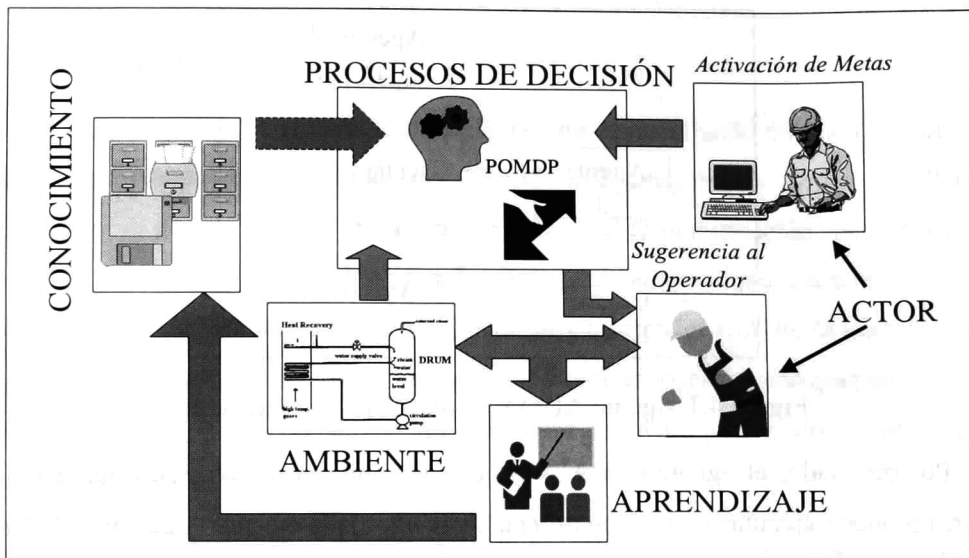


Figura 4.2 Descripción general de AUDA

A través de la interfaz con el agente humano (*ACTOR*) el operador interactúa con el agente bajo AUDA. Activa alguna meta para que AUDA, a través del MPD, obtenga una política óptima y dé la acción de mayor beneficio en cada instante de tiempo. Por otro lado, el agente con AUDA da la respuesta a *ACTOR* para que éste sea quien realice las acciones sobre el sistema. *ACTOR*, de acuerdo a su conocimiento, pudiera realizar alguna acción distinta a la sugerida por AUDA. En este caso a través del MA el agente aprendería del nuevo comportamiento que está tomando *ACTOR* [Agueda e Ibarguengoytia 01 b].

El ambiente también interactúa con AUDA. A través del ambiente, el agente puede cambiar su estado de creencias. *ACTOR*, es quien da acciones al ambiente.

La AUDA está basada en una Arquitectura BDI, por las siglas en *Inglés de Creencias, Deseos e Intenciones* [Bratman 87]. En este tipo de arquitecturas, las “*creencias*” (beliefs) están formadas por todo aquello que el agente conoce del ambiente, es decir el estado en que cree se encuentra el sistema. Los “*deseos*” (desires) representan alguna meta; es decir, lo que el agente quiere hacer. Las “*intenciones*” (intentions) son el conjunto de pasos, tareas o acciones que se deben realizar para cumplir los *deseos*, representan la forma en que se deben cumplir las metas. En AUDA, Las *Creencias* son dadas con base a una distribución de estados obtenida por los POMDP, de la cual se obtiene el estado más probable en que puede estar el sistema. Los *deseos* son dados mediante la



activación de una meta. Las *Intenciones* se obtienen con base a una política óptima calculada por lo MDP. La arquitectura se puede observar en la Figura 4.3

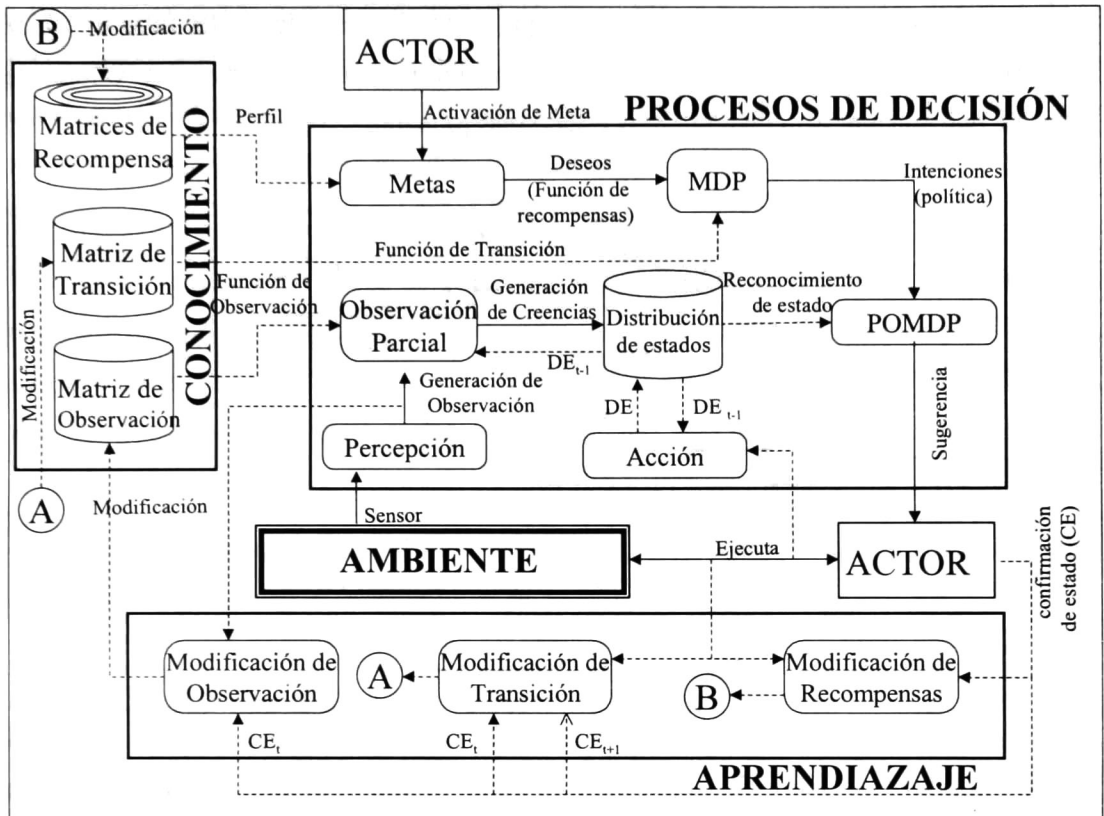


Figura 4.3 Diagrama de AUDA

Otra característica importante de AUDA es que está diseñada con el propósito de trabajar sobre dominios donde existe incertidumbre. Trabaja sobre los POMDP, por lo que puede estimar el estado actual del sistema y obtener la política óptima aún cuando existe incertidumbre. A continuación se describirá más ampliamente cada uno de los módulos de AUDA.

### 4.3 El Ambiente

AUDA requiere observar el ambiente del sistema. El AMBIENTE representa una de las entradas a la arquitectura. Recibe las acciones dadas por el OPERADOR, así mismo es de donde los sensores adquieren las señales necesarias para actualizar los valores de las

variables que se necesitan. Para el dominio de aplicación del presente trabajo, este módulo representa la CTE, en particular, al sistema del *Vapor Sobrecalentado*.

El AMBIENTE recibirá las acciones que afectarán los valores del *flujo de vapor principal (fvp)* y/o la *presión del domo (pd)*; asimismo proporciona los nuevos valores de *fvp* y *pd*, que servirán para determinar el estado más probable.

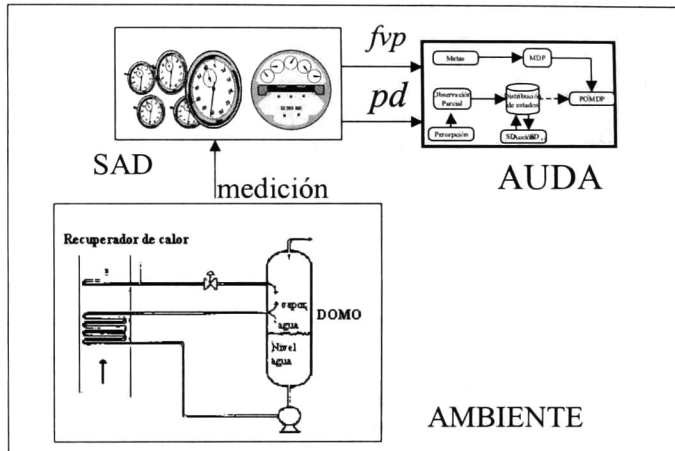


Figura 4.4 Módulo del ambiente

Un Sistema de Adquisición de Datos (SAD) toma mediciones del ambiente. Para el diseño de UADA, la parte de adquisición de datos es una capa de nivel de abstracción inferior a la de la arquitectura. Se considera que la medición se lleva a cabo dentro del *Módulo del Ambiente* y no importa la manera en que esto se haga.

## 4.4 Módulo de operadores (ACTOR)

En este caso, el módulo de operadores (ACTOR) es una interfaz entre el operador de la Central y el agente (programado en un prototipo) con AUDA. A través de esta interfaz, el operador puede introducir alguna meta al agente con AUDA (*activación de meta*), para que ésta calcule la política óptima de acciones. Cada una de las metas posibles a alcanzar están indizadas con un número; la manera en que se activa la meta, es dando el número de índice que le corresponde a la meta deseada. Esta meta representa un porcentaje de carga de la CTE correspondiente a la Tabla 3.1.

El operador de la CTE puede dar acciones al sistema (*ejecuta*). La forma en que se llevan a cabo las acciones queda en un nivel de abstracción debajo de AUDA. Se hace la suposición de que la acción dada por el usuario, es la que ejecutarán los sistemas de la CTE, llámense planificadores (de nivel bajo) [Wilkins y Myers 98] o sistemas de control; también puede ser el mismo operador de manera manual. AUDA es una arquitectura de alto nivel; es decir, da una sugerencia al ejecutor en forma de tareas parciales [Dean y Wellman 91].

Otra opción que tiene el operador es la de confirmar el estado actual de la central (*Confirmación de estado*). Cuando el operador tiene una manera de conocer (o en su caso, suponer) el estado del sistema, puede confirmar el estado actual de tal manera que el módulo de *aprendizaje* pueda tener una mayor solidez. Es decir que el conocimiento adquirido es más cercano a lo que sucede en el sistema físico.

A través de la acción y la confirmación de estado dadas por el operador, el módulo de *aprendizaje* modifica el conocimiento de AUDA.

## 4.5 Módulo de Base de Conocimiento

Aquí se encuentra el conocimiento en forma estadística que será utilizado por los *procesos de decisión*. Los datos que conforman la información están dispuestos en tres tipos de matrices: las que contienen información sobre las recompensas, la que tiene información sobre las probabilidades de las transiciones del sistema y la que contiene la información de las probabilidades sobre las observaciones. Estas matrices son necesarias para los POMDP, con éstas se obtiene el estado actual más probable y las políticas óptimas. El conocimiento dentro de estas matrices será modificado de acuerdo a lo que se obtenga del Módulo de *Aprendizaje*. Las matrices contenidas aquí se describen a continuación.

### 4.5.1 Matrices de recompensas (MR)

Las *Matrices de recompensas* es un conjunto de matrices indizadas de acuerdo al número de meta. A cada meta le corresponde una matriz, a la cual se hace referencia a través del número de índice. Este índice, llamado *perfil*, es único y es el que relaciona al número de matriz con el número de la meta. Cada matriz proporciona las recompensas

adecuadas para la respectiva meta. Se selecciona la MR adecuada (*Selección de perfil*) cuando se activa una meta.

En particular, cada matriz es un arreglo bidimensional de  $n \times m$ , donde  $n$  es el número de estados y  $m$  el número de acciones. Es necesario conocer de cada matriz, cuántos son los datos que conforman la información de las matrices, por lo que se debe tener una forma de agregar el número de muestras realizadas. Los números de los datos muestreados son puestos en una columna aumentada a la derecha de la MR. Esta matriz se le ha renombrado como Matriz de Aprendizaje de Recompensa MAR para identificarla de la MR. Esta matriz tiene los datos a través de los cuales se representa la *Función de Recompensas* para los POMDP.

#### **Definición 4.1 Matriz de Aprendizaje de Recompensas**

Una *Matriz de Aprendizaje de Recompensas* MAR es un arreglo bidimensional donde  $MAR_{i,j} = NR(s_i, a_j)$  = el número de acciones  $a_j$  en el estado  $s_i$ , para  $i=0,1,\dots, \#(S)-1$  y  $j=0,1,\dots, \#(A)-1$ .  $MAR_{i, \#(A)}$  es igual al número total de acciones realizadas en el estado  $s_i$ . Por lo tanto, una MAR queda de la siguiente forma:

$$MAR_p = \begin{bmatrix} NR(s_0, a_0) & NR(s_0, a_1) & \cdots & NR(s_0, a_{\#(A)-1}) & \sum_{i=0}^{\#(A)-1} NR(s_0, a_i) \\ NR(s_1, a_0) & NR(s_1, a_1) & \cdots & NR(s_1, a_{\#(A)-1}) & \sum_{i=0}^{\#(A)-1} NR(s_1, a_i) \\ \vdots & \vdots & & \vdots & \vdots \\ NR(s_{\#(S)-1}, a_0) & NR(s_{\#(S)-1}, a_1) & \cdots & NR(s_{\#(S)-1}, a_{\#(A)-1}) & \sum_{i=0}^{\#(A)-1} NR(s_{\#(S)-1}, a_i) \end{bmatrix}$$

donde  $p$  es el número de meta (o perfil).

La Función de Recompensa se relacionará con la matriz de la siguiente manera:

$$R(s_i, a_j) = \frac{10 \text{ } MAR_{i,j}}{MAR_{i, \#(A)}} \quad (4.1)$$

#### **4.5.2 Matriz de Transición (MT)**

En la *Matriz de Transición* (MT) se encuentra la información referente a las probabilidades de los efectos de una acción. Contiene las probabilidades de llegar a cualquier estado del sistema, dados un estado inicial cualquiera del sistema y una acción.

La MT es de tres dimensiones:  $l \times m \times n$ , donde  $l$  y  $m$  es el número estados y  $n$  es el número de acciones. A esta matriz se le ha agregado una columna a la derecha para colocar el número de muestras realizadas (número de acciones realizadas en el estado correspondiente a la fila), cambiándole el nombre a *Matriz de Aprendizaje de Transición* (MAT).

MAT, contiene los datos que representan la *Función de Transición* de los POMDP, y se define como sigue:

### Definición 4.2 Matriz de Aprendizaje de Transición

Una *Matriz de Aprendizaje de Transición* MAT es un arreglo tridimensional donde  $MAT_{i,i,k} = NT(s_i, s_j, a_k) =$  el número de veces en que se ha alcanzado el estado  $s_i$ , desde  $s_j$ , dada la acción  $a_k$ ; para  $i, j = 0, 1, \dots, \#(S)-1$  y  $k = 0, 1, \dots, \#(A)-1$ .  $MAT_{i, \#(A)}$  es igual al número total de acciones realizadas en el estado  $s_i$ . Por lo tanto, una MAT queda de la siguiente forma:

$$MAT^a = \begin{bmatrix} NT(s_0, s_0, a) & NT(s_0, s_1, a) & \cdots & NT(s_0, s_{\#(S)-1}, a) & \sum_{s' \in S} NT(s_0, s', a) \\ NT(s_1, s_0, a) & NT(s_1, s_1, a) & \cdots & NT(s_1, s_{\#(S)-1}, a) & \sum_{s' \in S} NT(s_1, s', a) \\ \vdots & \vdots & & \vdots & \vdots \\ NT(s_{\#(S)-1}, s_0, a) & NT(s_{\#(S)-1}, s_1, a) & \cdots & NT(s_{\#(S)-1}, s_{\#(S)-1}, a) & \sum_{s' \in S} NT(s_{\#(S)-1}, s', a) \end{bmatrix}$$

dada la acción  $a$ .

La Función de Transición se relacionará con la matriz de la siguiente manera:

$$T(s_i | s_j, a_j) = \frac{MAT^a_{j,i}}{MAT^a_{j, \#(S)}} \quad (4.2)$$

### 4.5.3 Matriz de Observación (MO)

La Matriz de Observación (MO) contiene los datos sobre las observaciones realizadas en cada estado. Esta proporciona las probabilidades de realizar alguna observación en algún estado dado. MO es un arreglo de dos dimensiones,  $m \times n$ , donde  $m$  le corresponde al número de estados, y  $n$  el número de observaciones. A esta matriz se le ha agregado una columna a la derecha para indicar el número de observaciones hechas en el



estado correspondiente a la fila. Esta nueva matriz se le llama Matriz de Aprendizaje de Observación (MAO).

Los datos contenidos en MAO proporcionan la información necesaria para la *Función de Observación* de los POMDP. MAO, se define como sigue:

**Definición 4.3 Matriz de Aprendizaje de Observación**

Una *Matriz de Aprendizaje de Observación* MAO es un arreglo bidimensional donde  $MAO_{i,j} = NO(s_i, o_j) =$  el número de veces en que se ha observado  $o_i$  en el estado  $s_j$  para  $i = 0, 1, \dots, \#(S)-1$  y  $j = 0, 1, \dots, \#(O)-1$ .  $MAO_{i, \#(O)}$  es igual al número total de observaciones llevadas a cabo en el estado  $s_i$ . Por lo tanto, una MAO queda de la siguiente forma:

$$MAO = \begin{bmatrix} NO(s_0, o_0) & NO(s_0, o_1) & \dots & NO(s_0, o_{\#(O)-1}) & \sum_{o' \in O} NO(s_0, o') \\ NO(s_1, o_0) & NO(s_1, o_1) & \dots & NO(s_1, o_{\#(O)-1}) & \sum_{o' \in O} NO(s_1, o') \\ \vdots & \vdots & & \vdots & \vdots \\ P^h(s_{\#(S)-1}, o_0) & P^h(s_{\#(S)-1}, o_1) & \dots & NO(s_{\#(S)-1}, o_{\#(O)-1}) & \sum_{o' \in O} NO(s_{\#(S)-1}, o') \end{bmatrix}$$

La Función de Transición se relacionará con la matriz de la siguiente manera:

$$O(o_i | s_j) = \frac{MAO_{j,i}}{MAO_{j, \#(O)}} \tag{4.3}$$

**4.6 Módulo de Procesos de Decisión (PD)**

En este módulo, AUDA obtiene la mejor acción (probable) a realizar. Se lleva a cabo el proceso para determinar el estado más probable en el que pudiera estar el sistema; a su vez obtiene la política óptima cuando se activa una meta. La manera en que se ha diseñado la arquitectura permite trabajar de forma eficiente con los algoritmos de los POMDP. No se requiere calcular la función de *Suma de Esperada de Recompensas* (Ver la definición 2.5), cada vez que se va a llevar a cabo una decisión. Se utilizan los MDP's para obtener la política óptima, respecto al conocimiento en el momento en que se haya activado una meta. Las extensiones que se realizan a los MDP para formar los POMDP son

utilizadas para crear una distribución sobre los estados posibles en que se puede encontrar el sistema; es decir se utilizan como *observadores*.

De acuerdo a lo que se ha visto en el capítulo dos, la descomposición anterior es correcta. La distribución  $E_0^a$  utilizada para el cálculo de políticas con los POMDP, corresponde al estado de creencias en el momento de calcular la política. Los POMDP permiten calcular la distribución de estados dada una observación o una acción por separado. Por lo tanto se utilizan en AUDA únicamente para determinar el estado más probable. Por otro lado, los MDP se utilizaron para calcular políticas, independientemente de la distribución de estados. Este tipo de descomposición ya se ha realizado para navegación de robots [Nikovski y Nourbakhsh].

En la Figura 4.5 se ilustra cómo se lleva a cabo la decisión para sugerir la acción óptima. Cuando se activa la meta los MDP obtienen la acción que se debe realizar en cada estado. En la figura, cada cuadrícula de la *Política* representa un estado y las flechas, la acción que se debe realizar en su respectivo estado.

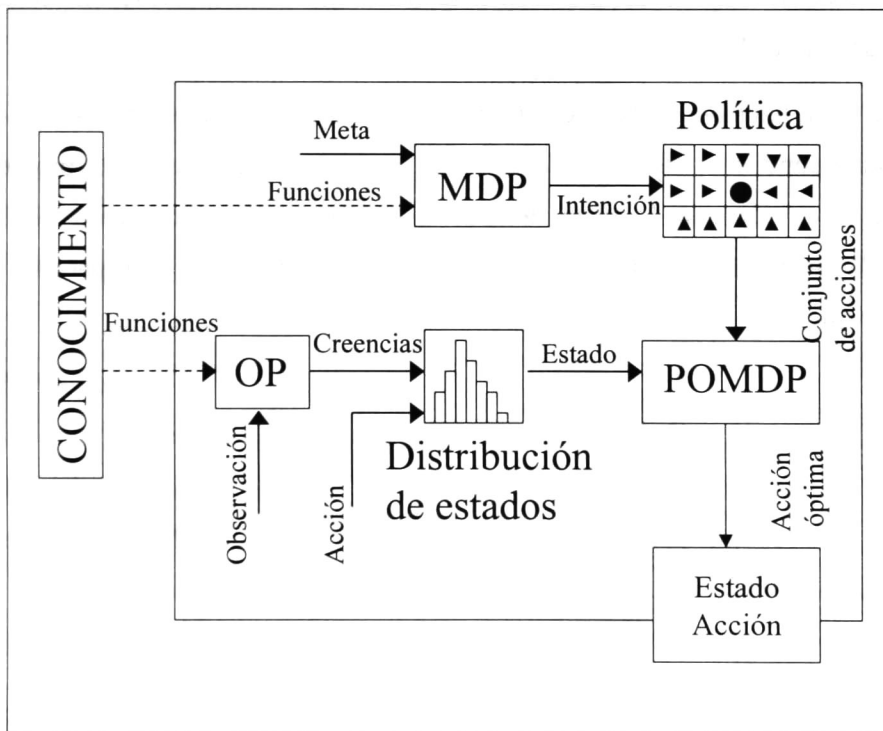


Figura 4.5 Los MDP y los POMDP en AUDA

La política sería poco útil si no se conociera el estado del sistema. Es por eso que el módulo de *Observación Parcial (OP)* es utilizado para crear una situación de creencias a través de una distribución de estados. Dicha distribución también es modificada cuando se realiza una acción; es decir, las creencias cambian después de realizar una acción. Cuando se ejecuta una acción, se espera que el estado actual cambie. Con la distribución se puede conocer cuál es el estado con mayor probabilidad y se toma a éste como el estado actual. Con el estado encontrado, se puede hacer una relación dentro de la política obtenida anteriormente para encontrar la acción óptima. De esta forma se hace la sugerencia al operador a través de ACTOR.

A continuación se irán describiendo las partes que conforman el módulo de los *Procesos de Decisión (PD)* y cómo interactúan.

### 4.6.1 Metas

A través de ACTOR, el operador activa una meta. La *Activación de Meta* consiste en indicar el número correspondiente a la meta. Con este número se hace una búsqueda en las *Matrices de Recompensas*, para obtener la matriz que contiene las recompensas para la meta que se quiere alcanzar.

Se le ha llamado *Perfil* al conjunto de los valores de recompensas obtenidas de la matriz correspondiente, debido a que es un tipo de comportamiento propio de los operadores para alcanzar la meta deseada. Los valores de recompensa de la matriz describen la función de recompensa que será utilizada por los MDP. La Figura 4.6 ilustra lo que se ha explicado.

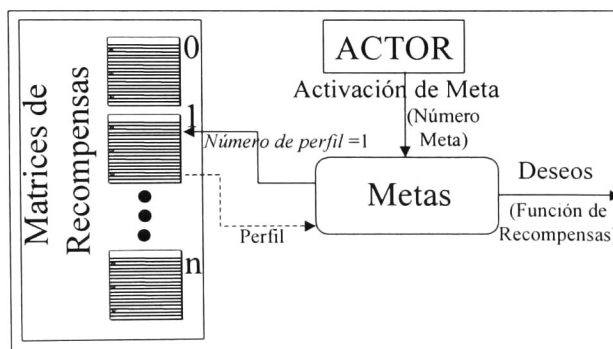


Figura 4.6 Metas

Una vez determinada una meta, se obtiene la función de recompensas asociada a dicha meta.

### 4.6.2 Módulo de los MDP

La función de recompensas **R** es la parte de deseos de una arquitectura BDI. Esta función es una de las entradas para los MDP. Otra entrada que se necesita es la función de transición **T**. Teniendo como entradas a **R** y a **T**, se pueden llevar al cabo los *Proceso de Decisión de Markov*, esto se puede ver en la Figura 4.7. El conjunto de estados **S** y el de acciones **A**, se establecieron desde el momento en que se definieron las matrices del modelo; por lo tanto, se tiene la tupla (S,A,T,R) que forman un MDP. Al resolver el MDP, se obtiene como resultado una política óptima.

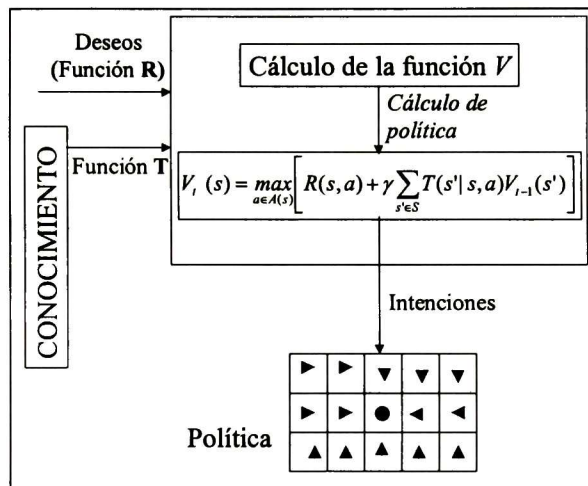


Figura 4.7 Los MDP

### 4.6.3 Percepción

Se encarga de obtener del SAD los valores necesarios para definir el estado observado del sistema y de esta forma, generar una observación (*Generación de Observación*). Con la medición proveniente del ambiente, se obtienen los valores actuales de *fvp* y *pd*. Estos valores se utilizan para determinar el estado correspondiente a esos valores. El estado actual se define como el estado *s* tal que, la coordenada  $c = (\lceil fvp \rceil, \lfloor pd \rfloor)$  está dentro de la región acotada por *s*. El número de estado del sistema se calcula con la siguiente función:

SF:  $\mathcal{R} \times \mathcal{R} \rightarrow \mathcal{N}$

$$(fvp, pd) \mapsto SF(fvp, pd) = 6 \left\lceil \frac{fvp - 531}{118} \right\rceil + \left\lceil \frac{pd - 169}{2} \right\rceil$$

Por lo tanto, el estado observado, dadas las mediciones de  $fvp$  y  $pd$  es el estado  $s_{SF(fvp, pd)}$ . El cual se le dará como la observación generada al módulo de OP.

#### 4.6.4 Observación Parcial (OP)

Esta parte tiene como entradas la *función de observación*  $\mathbf{O}$  a través de la *matriz de observación*; y la *observación generada* por el módulo de *Percepción*.

Dado un estado anterior de **creencias**  $E'$  ( $DE_{t-1}$ ) y una *observación generada*, se calcula una nueva distribución de estados de acuerdo a lo que se vio en la Sección 2.5.2.2. El resultado obtenido es un nuevo estado de **creencia**  $E^o$ . Este módulo es una parte encargada de generar las creencias para la arquitectura.

#### 4.6.5 Acción

Se supone que cuando se realiza una acción sobre el sistema, éste cambia su estado. Es en esta parte donde se consideran estos cambios para generar, al igual que OP, un nuevo estado de **creencias**.

Una de sus entradas es una acción sugerida por los PD, de las definidas en el conjunto  $\mathbf{A}$ , ejecutadas (ejecuta) por el ACTOR. La otra entrada es el estado de creencias en tiempo anterior al actual ( $DS_{t-1}$ ). Con éstas, calcula la distribución actual de estados dada una acción de acuerdo al punto 2.5.2.1.

#### 4.6.6 Distribución de estados (DE)

La Distribución de Estados (DE) consiste en un vector que representa la distribución de probabilidad de encontrarse en algún estado del sistema. Esta distribución es modificada cuando se ha llevado al cabo una observación o cuando se ha realizado una acción sobre el ambiente del sistema. De esta distribución, se calcula el estado que tiene mayor probabilidad de encontrarse el sistema (reconocimiento de estado) y se da como resultado al módulo de los POMDP. La Figura 4.8 ilustra lo anterior.



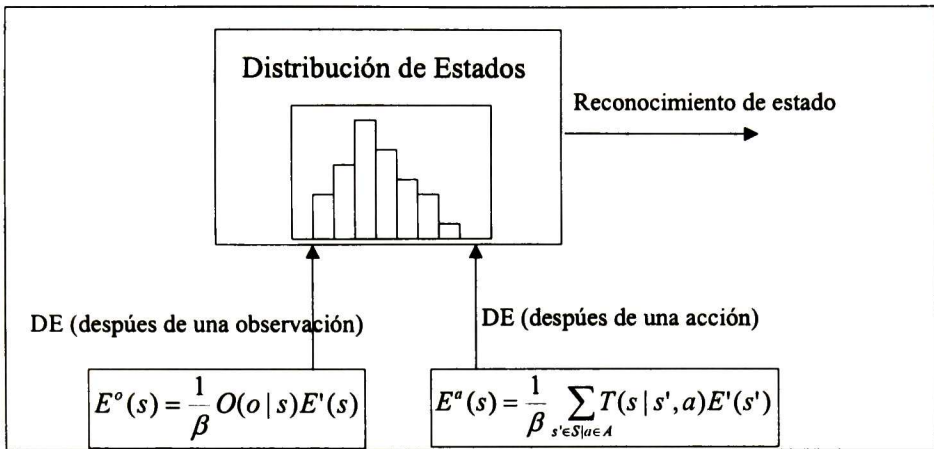


Figura 4.8 Distribución de Estados

### 4.6.7 Módulo de los POMDP

Una de las entradas a este módulo es el estado más probable dado una distribución de estados (*distribución de estados*). Esta distribución de estados se lleva a cabo a través del módulo OP y a través del de *Acción*. La distribución es obtenida a través de las ecuaciones 2.11 y 2.12, por lo que se utilizaron las funciones de transición y las de observación dadas por las matrices correspondientes. La otra entrada son las **intenciones** (política de acciones) dada por MDP. Como ya pudimos ver, en el MDP se cuenta con la tupla (S, A, T, R). Ahora bien, en éste módulo se tiene la tupla (S, A, T, R, O,  $\Omega$ ), la cual es necesaria para los POMDP.

El POMDP, da una sugerencia al operador a través de ACTOR. Indica el estado más probable del sistema y su probabilidad; por consiguiente, la acción óptima a realizar en ese momento. El operador, en el momento de poder realizar la acción, puede consultar la distribución de probabilidad para verificar cuáles son las probabilidades de encontrarse en otro estado. Es aquí donde se juntan las **creencias** y las **intenciones**; es decir, se tiene un Proceso de Decisión Parcialmente Observable (Ver la Figura 4.5).

## 4.7 Módulo de Aprendizaje (MA)

Este hace posible que un agente con AUDA aprenda del comportamiento del ambiente y del operador. El aprendizaje se va obteniendo de las acciones dadas por el operador en cada estado de acuerdo a la meta activada. Los POMDP basan su conocimiento

en datos históricos, información estadística respecto a las observaciones, acciones y transiciones; además de un conjunto de penalizaciones o recompensas al realizar acciones. Por esto, en esta parte se podrán identificar las acciones que realiza el Actor dada una meta y un estado, durante todo el tiempo de operación del sistema. El módulo puede tener información referente a las observaciones que se generan en cada estado. De esta forma, el módulo va adquiriendo más datos para modificar las matrices que se están utilizando, conforme va creciendo la historia del sistema.

#### 4.7.1 Modificación de Recompensas

Este módulo tiene como entradas por parte del actor la *confirmación del estado* (CE) y la acción (*ejecuta*) realizada sobre el sistema. Con dichos valores de entrada, la matriz de recompensas puede ser modificada a fin de definir de nuevo la función de recompensas. La adquisición de conocimiento es a través de la forma de actuar del operador ante algún estado del sistema. El módulo se basa en la “clonación” del operador; por lo que AUDA, a través del tiempo, será tan experto como lo sea el operador [Urbancic y Bratko 94].

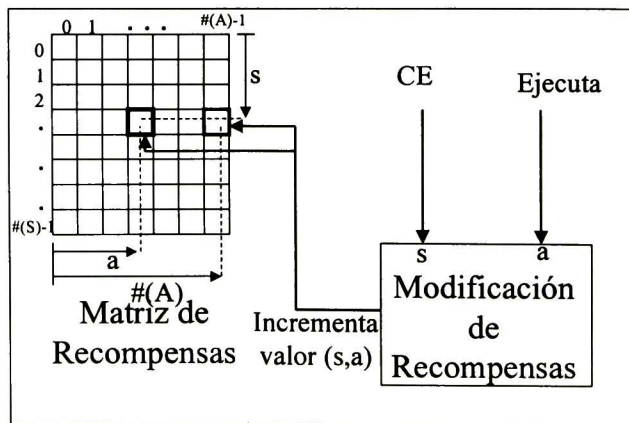


Figura 4.9 Modificación de Recompensas

De acuerdo al formato de la matriz de recompensas, se debe incrementar el valor de la acción ejecutada por el operador dentro de la fila correspondiente al estado confirmado, además de incrementar el número de muestras para dicho estado. Esto se puede ver en la Figura 4.9.

### 4.7.2 Modificación de Transición

Este módulo se encarga de modificar la matriz de transición del modelo. Permitirá adquirir conocimiento más preciso sobre el comportamiento del sistema. La forma en que adquiere el conocimiento es modificando los datos estadísticos de la matriz de transición de acuerdo a lo que ocurre en el sistema. Dada una acción  $a$ , en un tiempo  $t$  y asumiendo que se conoce el estado del sistema  $s'$  en el momento de realizar la acción (en el tiempo  $t-1$ ), se observa cuál es el estado  $s$  al que se ha llegado en el tiempo  $t$  (dado por el operador). La tupla  $(s, s', a)$  es agregada a los datos históricos del modelo.

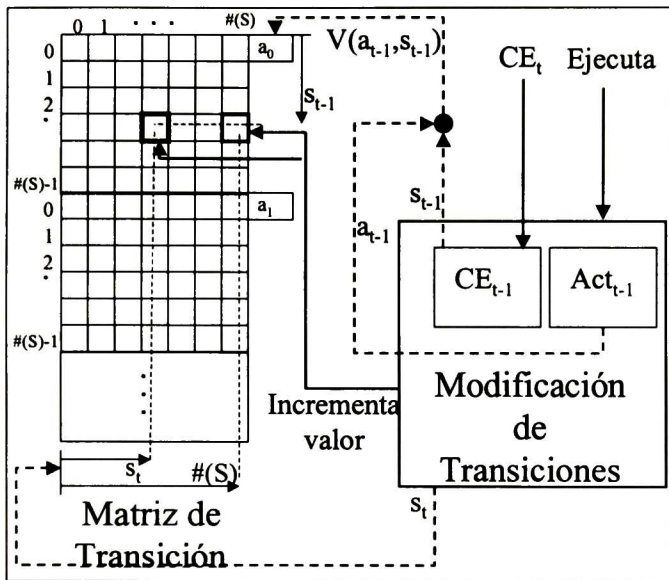


Figura 4.10 Modificación de Transición

La Figura 4.10 ilustra la forma en que se determina qué lugares de la matriz se deben modificar y de que manera. El operador confirma el estado  $s_{t-1}$  anterior en que estaba el sistema, antes del estado en que se va a modificar la matriz. Este estado es almacenado dentro del módulo, dentro de una localidad de memoria, de un solo lugar, y se le nombra  $CE_{t-1}$ . En ese mismo instante, el operador ejecuta una acción  $a_{t-1}$ , la cual es almacenada en una localidad de memoria de un valor, y se le conoce como  $Act_{t-1}$ . En el momento de confirmar el nuevo estado  $s_t$  (CE), el módulo modifica la matriz.

La modificación se hace incrementando un caso más a la matriz en el elemento  $mat_{s_{t-1}, s_t, a}$ , además del valor de número de muestra. Para encontrar los lugares dentro de la

matriz correspondientes a los elementos mencionados, se crea la dirección virtual  $V$  para localizar la fila correspondiente al estado donde se van a realizar las modificaciones. Esta dirección virtual se obtiene de la siguiente manera:

$$V: \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$$

$$(a, s) \mapsto V(a, s) = a(\#(A)) + s$$

Una vez localizada la fila, se hacen los desplazamientos dentro de ella, la cuales son dados por  $s_t$  y  $\#(S)$ .

### 4.7.3 Modificación de Observación

A través de este módulo se modifica el conocimiento respecto a la función de observación contenida en la Matriz de Observación. Las entradas que necesita para hacer la modificación a la matriz son la observación  $o$  que se generó y el estado  $s$ , que ha confirmado el Actor, en que se encuentra el sistema.

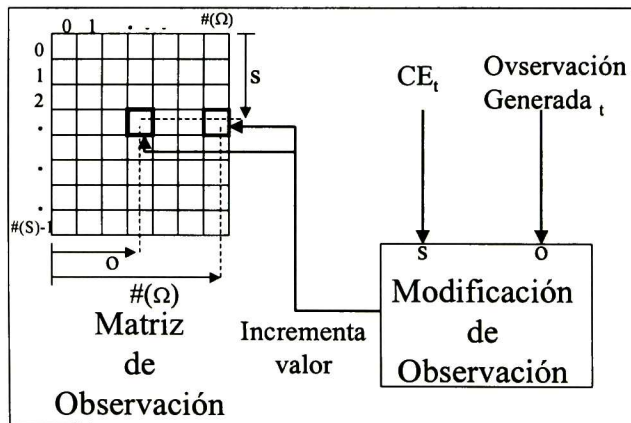


Figura 4.11 Modificación de Observación

La Figura 4.11 muestra cómo se hace la búsqueda de los elementos a modificar. Estos elementos incrementan su valor, modificando así las estadísticas. La modificación se hace en el caso observado y en el correspondiente al número de muestras realizadas hasta el momento.

## Capítulo 5 Experimentos y resultados

---

En este capítulo se darán a conocer algunos experimentos realizados con un prototipo que utiliza la arquitectura propuesta. Se mostrarán los resultados referentes a la obtención de políticas óptimas, al cálculo de estados del sistema y al aprendizaje del agente con AUDA.

---



## 5.1 Introducción

En este capítulo se presentarán los experimentos realizados en un prototipo de la arquitectura que se propone. Se explicará de qué manera se han cubierto los objetivos de la tesis a) al obtener las políticas óptimas para alcanzar las metas; b) al poder estimar el estado más probable en que puede estar el sistema, a través del cálculo de una distribución de probabilidad sobre los estados; y c) al lograr que la arquitectura presentada aprende a comportarse de acuerdo a lo que el operador realiza.

Cabe mencionar que los experimentos se realizaron simulando el comportamiento del sistema utilizando datos históricos de una Central Termo Eléctrica y con base al conocimiento experto de operadores. Lo realizado ha servido para demostrar la solidez de la arquitectura. Esta arquitectura y el prototipo utilizado para los experimentos son la base para asegurar trabajos futuros en el dominio del tipo estudiado para la presente tesis (ver Capítulo 3).

## 5.2 Prototipo

Cada uno de los módulos descritos en la arquitectura del capítulo cuatro se ha programado en el prototipo que se explicará a continuación. Este prototipo permite integrar cada módulo a fin de que juntos puedan dar sugerencias de acciones a los operadores. El prototipo calculará una política de acciones dada una meta por el operador, y obtendrá una distribución de probabilidad sobre los estados que permite estimar cuál es el estado más probable de estar en un determinado tiempo; de esta forma se le puede decir al usuario cuál es el estado actual más probable y, por lo tanto, cuál es la acción óptima para realizar. La Figura 5.1 muestra la interfaz del prototipo.

En la interfaz se observa el comportamiento del ambiente del sistema (environment), proporcionando el estado real del sistema. Lo anterior se hace en la zona marcada como "A". Aquí se muestra el índice correspondiente al estado actual del sistema, que puede ir de 0 a 53.

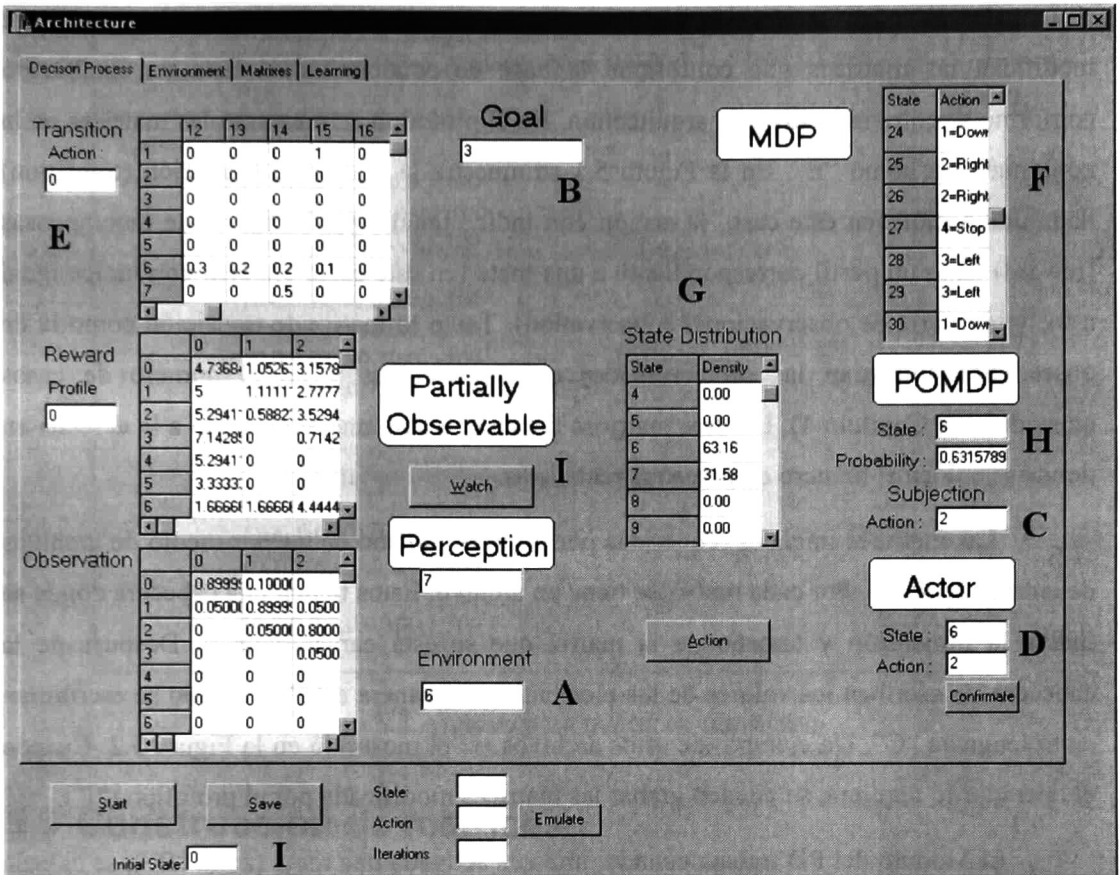


Figura 5.1 Prototipo deAUDA

El operador y los módulos restantes de la arquitectura interactúan cuando el operador activa alguna meta, cuando los de Procesos de Decisión (PD) le sugieren la acción óptima, o cuando el operador realiza una acción sobre el sistema. La activación de la meta se realiza a través de la zona marcada como “B”; donde escribe el número (perfil) de la meta a alcanzar. Los PD le sugieren la acción óptima en la zona “C”. En la zona “D”, el operador escribe la acción que va a realizar y si tiene la forma de determinarlo sin utilizar el prototipo, puede dar una confirmación sobre el estado en que está el sistema antes de realizar la acción. Esto lo realiza escribiendo el número de estado y presionando el botón de confirmar (confirmate). Por último el usuario le da la acción al sistema de control presionando el botón de “acción” (action).

Con la confirmación y la acción dada por el operador, los módulos de aprendizaje modifican las matrices que conforman la base de conocimientos. Ésta va cambiando conforme se entrena (utiliza) la arquitectura. En el prototipo se muestran las matrices en la zona marcada como “E” En la Figura 5.1 se muestra la matriz de transición (transition) dada una acción (en este caso, la acción con índice igual a 0); la matriz de recompensas (reward) dado un perfil correspondiente a una meta (en este caso, el perfil con índice igual a 0); y la matriz de observaciones (observation). Tanto la matriz de transición como la de observación muestran las probabilidades correspondientes y no el número de casos ocurridos(ver Capítulo 4). Las tres matrices tienen una columna aumentada a la derecha en donde se coloca el número de muestras realizadas.

Las matrices iniciales requeridas por el prototipo son dadas por medio de archivos de entrada (ASCII). Por cada matriz se tiene un archivo. Éstos tienen una cabecera donde se indica la dimensión y tamaño de la matriz que se está caracterizando. Después de la cabecera, se escriben los valores de los elementos, de manera similar a como se escribirían en el lenguaje “C” Un ejemplo de estos archivos es el mostrado en la Figura 5.2. Cuando el operador lo requiera, se pueden grabar las matrices modificada por el prototipo (“I”).

El Módulo del PD trabaja cuando, una vez activada una meta (zona “B”), se calcula una política óptima. Esta política se muestra en la tabla marcada con “F”. Esta política sirve para sugerir al operador una acción cuando se ha determinado el estado más probable del sistema. Para estimar el estado, se cuenta con la Distribución de Estados (DE) marcado con “G” la que proporciona un reconocimiento de estado que consiste en indicar el estado más probable y el valor de dicha probabilidad (“H”). La DE es modificada cuando se realiza una acción (“D”) o bien cuando se hace una observación. Para realizar la observación, la interfaz cuenta con el botón de “ver” (watch), que proporciona una percepción del ambiente (“I”) en un tiempo determinado.

```

// Matiz para la navegación de robot
// Dimensión (Núm. Acciones, Núm Estados)
3, 8
// acción I
{
{100,0,0,0,0,0,0,0},100
{90,10,0,0,0,0,0,0},100
{0,90,10,0,0,0,0,0},100
{0,0,90,10,0,0,0,0},100
{0,0,0,90,10,0,0,0},100
{0,0,0,0,90,10,0,0},100
{0,0,0,0,0,90,10,0},100
{0,0,0,0,0,0,90,10},100
},
...
// acción III
{
{100,0,0,0,0,0,0,0},100
{0,100,0,0,0,0,0,0},100
...
{0,0,0,0,0,0,0,100},100
}, END

```

Figura 5.2 Archivo para matriz de transición

### 5.3 Construcción de matrices

Una parte medular para el funcionamiento de los POMDP son las funciones de *recompensa*, de *transiciones* y de *observación* (v. cap 2). En la arquitectura están expresadas por medio de matrices (v. cap. 4). Los valores de las matrices pudieran tener valores no confiables, o dados de forma determinista (con 1 y 0 de probabilidades); el agente con la arquitectura, a través del módulo de aprendizaje, modificaría los valores de las matrices, de tal forma que se tenga un comportamiento adecuado del sistema. Sin embargo, se han dado matrices con valores cercanos a lo que ocurre en la realidad.

Las matrices necesarias para el funcionamiento del prototipo fueron creadas de dos formas:

- 1) utilizando datos históricos (transiciones) y
- 2) con el conocimiento de expertos en la operación de centrales (recompensas y observaciones)

A continuación se describirá cómo se construyeron cada una de las matrices.



### 5.3.1 Los datos históricos

En el IIE se cuenta con información de las variables necesarias para el estudio en el dominio y sus valores muestreados. Para hacer los experimentos se utilizó información de valores obtenidos de una Central Termoeléctrica Convencional en México. Los datos corresponden a una Unidad de Generación Termoeléctrica de 300 MW. Los datos que se utilizaron representan información histórica sobre el comportamiento de la central. Se utilizaron 736 horas de historia, descritas en variables muestreadas en intervalos de una hora. Las variables proporcionadas en los diarios corresponden a los sistemas de *calor sobrecalentado*; *vapor recalentado*; *calentadores baja presión*; *extracciones de baja presión*; *calentadores alta presión*; *agua de condensado y alimentación*; *condensador*; *turbina y generador*; *aire de combustión*; *combustible y quemadores*; y *gases*.

Dentro del conjunto de variables dadas en estos sistemas, se cuentan las de *flujo de vapor principal (fvp)* y la *presión en el domo (pd)*, ubicadas en el sistema de *vapor sobrecalentado*. Con estas dos variables, se pueden estudiar los efectos producidos cuando cambia el valor de alguna de ellas. A continuación, en la Tabla 5.1 se muestra el ejemplo de un diario en el sistema, de donde se pueden estudiar las variables.

Como se puede observar, las variables necesarias son las nombradas como 02ABF314 y 02AAP227, correspondientes al *fvp* y a la *pd*, respectivamente. Por lo tanto, se tomaron únicamente las columnas correspondientes a estas variables y se hizo el análisis sobre éstas.

### 5.3.2 Matriz de transición

La matriz de transición se presentó en la sección 4.5.2. Ésta representa la función de transición para los MDP. Por lo tanto, para construir la matriz de transición se necesita conocer, por cada estado  $s$ , la probabilidad de alcanzar un estado  $s'$  dada una acción. Las acciones que se pueden realizar son las descritas en la Tabla 3.2.

Sea  $MT$  una matriz de transición a utilizar en el prototipo, entonces  $MT$  es una matriz de  $54 \times 54 \times 5$ . Es decir, considera los estados  $s_0$  hasta el  $s_{53}$  (Figura 3.5) y 5 acciones. Para conocer las transiciones se utilizaron los datos históricos.



MEDIDAS PERIÓDICAS[ 1] : REPORTE.DIA1				VAPOR SOBRECALENTADO				
02ABF314	FLUJ VAP PRINC			02AEF141	FLUJ AG ATEM A SOBRECA			
02ABP303	PRES VAP SOBRECAL			02AET1420	TEMP SAL AG CAL 7			
02AEF144	FLUJ AG DE ALIM AL ECON			02AAL224	NIV DOMO DE VAP			
02AAP227	PRES DEL DOMO DE VAP			02ABT3031	TEMP VAP SOBRECAL			
09/07/99	02ABF314	02AEF141	02ABP303	02AET1420	02AEF144	02AAL224	02AAP227	02ABT3031
	T/H	T/H	BAR	CTG	T/H	MM	BAR	CTG
01:00:00	632.02	27.37	169.54	219.09	673.24	-3.4	173.47	535.62
02:00:00	629.44	25.3	169.49	219.81	619.38	15.44	173.5	535.53
03:00:00	624.66	32.55	169.48	219.26	635.99	-9.87	173.48	540.78
04:00:00	616.5	31.29	169.47	218.82	605.18	10.24	172.97	540.41
05:00:00	630.05	14.47	168.98	220.31	661.82	-9.3	172.56	539.41
06:00:00	650.67	25.01	169.05	221.78	662.4	9.68	173.1	543.16
07:00:00	649.36	31.88	169.47	222.25	652.83	2.51	173.61	538.84
08:00:00	653.06	19.21	169.09	222.73	640.67	24.37	173.1	532.16
09:00:00	751.69	34.68	169.05	228.89	765.82	9.71	174.7	542.41
10:00:00	787.78	25.49	169.06	230.19	767.14	17.66	175.24	538.34
11:00:00	792.61	23.93	169.48	230.16	825.49	10.56	175.74	532.25
12:00:00	987.47	25.66	168.68	241.6	979.83	24.76	177.83	535.16
13:00:00	1043.06	31.58	167.82	243.63	1064.01	18.05	177.3	531.66
14:00:00	1011.56	27.5	167.44	241.69	1022.51	20.96	176.77	535.84
15:00:00	984.37	33.17	167.86	241.08	1004.54	7.82	176.77	533.25
16:00:00	802.41	15.25	168.72	232	873.1	-8.14	175.19	529.47
17:00:00	801.56	22.75	167.92	232.43	806.25	24.9	174.15	535.37
18:00:00	932.81	26.4	167.5	237.77	949.56	14.18	175.68	535.25
19:00:00	888.37	29.56	167.48	236.21	888.38	22.37	175.18	532.19
20:00:00	787.36	16.3	168.26	230.78	839.21	-3.43	174.67	529.53
21:00:00	985.5	27.58	167.73	241.02	1022.12	3.53	176.76	534.94
22:00:00	926.06	26.61	167.74	239.24	965.23	-0.52	175.74	531.25
23:00:00	987.75	40.1	167.76	241.22	1011.08	8.45	176.73	533.5
00:00:00	879.09	27.49	168.2	237.28	916.75	-1.78	175.66	530.09

Tabla 5.1 Diario

### 5.3.2.1 Explicación del método.

Los elementos  $mt$  de la matriz MT se calculan con la siguiente expresión:

$$mt_{i,j,k} = \frac{\sum_{t=1}^{\#(H)} C(t, s_i, s_j, a_k)}{\sum_{s' \in S} \sum_{t=1}^{\#(H)} C(t, s_i, s', a_k)} \quad (5.1)$$

donde  $\#(H)$  es el número de muestras de los datos históricos y

$C: \mathcal{N} \times \mathcal{S} \times \mathcal{S} \times \mathcal{A} \rightarrow \{1, 0\}$

$(t, s, s', a) \mapsto C(t, s, s', a) = 1$  si en la muestra número  $t$ , el estado inicial =  $s$ , estado final =  $s'$ , y se cumple que:

$$a = a_0 \Rightarrow \Delta pd = +;$$

$$a = a_1 \Rightarrow \Delta pd = -;$$

$$a = a_2 \Rightarrow \Delta fvp = +;$$

$$a = a_3 \Rightarrow \Delta fvp = -;$$

0 en otro caso.

Se tomó primero la variable  $pd$  (convertida a  $\text{kgf/cm}^2$ ) como independiente y la  $fvp$  como dependiente de la primera. Considerando esto, las acciones que se pueden estudiar son  $a_1 =$  incremento de la “delta” de presión ( $\Delta pd = +$ ) o bien  $a_2 =$  decremento de la “delta” de presión ( $\Delta pd = -$ ); observando lo que pasa con la “delta” del  $fvp$ . Así que se puede conocer el estado inicial y el estado alcanzado después de la acción dada ( $a_1$  o  $a_2$ ).

Posteriormente, se toma la variable  $fvp$  como independiente, verificando los efectos de las acciones,  $a_3 =$  incremento de la “delta” del flujo de vapor ( $\Delta fvp = +$ ) o bien  $a_4 =$  decremento de la “delta” del flujo de vapor ( $\Delta fvp = -$ ). La matriz obtenida se puede observar en el apéndice de este documento.

$fvp$ (T/h)	$pd$ ( $\text{kgf/cm}^2$ )	Estado inicial	Estado final	$\Delta fvp$	$\Delta pd$
933.7	178.5	33	40	+	+
976.9	181.2	40	46	0	+
1013.6	182.2	46	33	-	-
928.1	179.1	33	34	+	0
960.1	179.3	34	26	-	-
754.9	177.0	26	21	+	-

Tabla 5.2 Tabla de valores

**Ejemplo.** Tomando los datos de la Tabla 5.2, el elemento  $mt_{33,40,3}$  se calcula de la siguiente manera:

Se está calculando un elemento correspondiente a la acción  $a_3$ . Se puede observar que en la Tabla 5.2, se encuentra 3 veces la condición de dicha acción, es decir,  $\Delta fvp = +$

$$\sum_{s' \in S, s \in S} \sum_{t=1}^{\#(H)} C(t, s, s', a_3) = 3 \quad (5.2)$$

La Tabla 5.2 queda reducida a la siguiente:

$fvp$ (T/h)	$pd$ (kgf/cm <sup>2</sup> )	Estado inicial	Estado final	$\Delta fvp$	$\Delta pd$
933.7	178.5	33	40	+	+
928.1	179.1	33	34	+	0
754.9	177.0	26	21	+	-

Tabla 5.3 Tuplas que cumplen  $\Delta fvp=+$

Ahora bien, se está haciendo el cálculo para un elemento que tiene estado inicial  $s=s_{33}$ . Se observa que sólo hay 2 tuplas que tienen estado inicial en  $s_{33}$ ; por lo que se tiene el valor del denominador de la ecuación 5.1,

$$\sum_{s' \in S} \sum_{t=1}^{\#(H)} C(t, s_{33}, s', a_3) = 2 \quad (5.3)$$

lo cual representa el número de veces en que se ha realizado la acción  $a_3$  en el estado  $s=s_{33}$ , reduciendo de esta manera la tabla en:

$fvp$ (T/h)	$pd$ (kgf/cm <sup>2</sup> )	Estado inicial	Estado final	$\Delta fvp$	$\Delta pd$
933.7	178.5	33	40	+	+
928.1	179.1	33	34	+	0

Tabla 5.4 Tuplas que cumplen  $\Delta fvp=+$  y *Estado Inicial*=33

De las dos veces, sólo una vez se ha llegado al estado  $s_{40}$ . Esto corresponde al elemento que se está calculando. Por lo tanto, se cuenta con el valor del numerador de la ecuación 5.1:

$$\sum_{t=1}^{\#(H)} C(t, s_{33}, s_{40}, a_3) = 1 \quad (5.4)$$

Sustituyendo 5.3 y 5.4 en 5.1, se obtiene la probabilidad de alcanzar el estado  $s_{40}$  dada la acción  $a_3$  en el estado  $s_{33}$ ; la cual es igual al valor del elemento  $mt_{33,34,3}$  de la matriz de transición MT.

$$mt_{33,40,3} = \frac{\sum_{t=1}^{\#(H)} C(t, s_{33}, s_{40}, a_3)}{\sum_{s' \in S} \sum_{t=1}^{\#(H)} C(t, s_{33}, s', a_3)} = \frac{1}{2} = 0.5 \quad (5.5)$$

Debido a que el módulo de aprendizaje así lo requiere, las matrices de transición se les ha dado otro formato (v. cap. 4), llamándola matriz de transición de aprendizaje (MTA), donde los elementos de la matriz  $mta_{i,j,k}$ , son el número de veces de ocurrencias de la transición  $s_i a_k s_j$ . Se agrega una columna a la derecha de la matriz donde se coloca el número de veces que se ha realizado la acción  $a_k$  en el estado  $s_i$  donde  $i$  corresponde al número de la fila en la matriz. Así pues, se tienen matrices de la siguiente forma:

$$MTA_a = \left[ \begin{array}{cccc} \sum_{t=1}^{\#(H)} C(t, s_0, s_0, a) & \sum_{t=1}^{\#(H)} C(t, s_0, s_1, a) & \cdots & \sum_{t=1}^{\#(H)} C(t, s_0, s_{53}, a) & \sum_{s' \in S} \sum_{t=1}^{\#(H)} C(t, s_0, s', a) \\ \sum_{t=1}^{\#(H)} C(t, s_1, s_0, a) & \sum_{t=1}^{\#(H)} C(t, s_1, s_1, a) & \cdots & \sum_{t=1}^{\#(H)} C(t, s_1, s_{53}, a) & \sum_{s' \in S} \sum_{t=1}^{\#(H)} C(t, s_1, s', a) \\ \vdots & \vdots & & \vdots & \vdots \\ \sum_{t=1}^{\#(H)} C(t, s_{53}, s_0, a) & \sum_{t=1}^{\#(H)} C(t, s_{53}, s_1, a) & \cdots & \sum_{t=1}^{\#(H)} C(t, s_{53}, s_{53}, a) & \sum_{s' \in S} \sum_{t=1}^{\#(H)} C(t, s_{53}, s', a) \end{array} \right] \quad (5.6)$$

donde  $MTA_a$  es la matriz de transición de aprendizaje dada la acción  $a$ . En el ejemplo dado anteriormente, la MTA contendrá los elementos:

$mta_{33,40,1} = 1$	$mta_{33,54,1} = 1$
$mta_{40,46,1} = 1$	$mta_{40,54,1} = 1$
$mta_{46,33,2} = 1$	$mta_{46,54,2} = 1$
$mta_{34,26,2} = 1$	$mta_{34,54,2} = 1$
$mta_{33,34,3} = 1, mta_{33,40,a3} = 1$	$mta_{33,54,3} = 2$
$mta_{46,33,4} = 1$	$mta_{46,54,4} = 1$
$mta_{34,26,4} = 1$	$mta_{34,54,4} = 1$

### 5.3.2.2 Utilización de SQL

Obviamente, por el número de muestras que se tiene (736), es conveniente automatizar el método para construir las matrices. Esta automatización se logra poniendo los datos históricos en una tabla DBF (Data Base File), de tal manera que se pueda hacer



consultas utilizando consultas (queries) de SQL (Structured Query Language). Estas consultas forman parte del algoritmo para construir las matrices de forma automática. Los algoritmos son de la siguiente forma:

```

1)      para INI = s0 hasta s53 hacer
2)          para FIN = s0 hasta s53 hacer
3)              sele count( ) from ARCHIVO where
                  EstadoInicial= s0
                  EstdoFinal= s33
                   $\Delta pd = +$ 
                  into CONTADOR
4)              Escribe (CONTADOR)
5)              sele count( ) from ARCHIVO where
                  EstadoInicial= s0
                   $\Delta pd = +$ 
                  into CONTADOR2
6)      Escribe( CONTADOR2 + “retorno de carro”)

```

Algoritmo 5.1

En este algoritmo se está construyendo la parte de la matriz que le corresponde a la acción  $a_1$ , es decir se está verificando que  $\Delta pd = +$ ; lo cual queda expresada en la condición dada en las consultas 3 y 5. En las líneas 1 y 2 se manifiesta que se está construyendo una matriz de 54x54. La línea 3, corresponde a la parte del numerador de la ecuación 5.1; mientras que la línea 6 obtiene el valor de denominador. En 4 y 6 se escriben los valores obtenidos en la matriz, de tal forma que se construye de la forma mostrada en 5.6.

### 5.3.3 Matriz de recompensas

Los valores de los elementos de la matriz de recompensa fueron establecidos con base a heurísticas dadas por expertos en la operación de centrales. Estos valores representan la recompensa al realizar una acción en un estado. Uno de los criterios básicos proporcionados, dado una meta, es la de dar mayor recompensa a las acciones que permiten reducir la distancia entre el estado donde se realizará la acción y el estado final de la meta; dando baja recompensa a las acciones que llevan a estados sobre la curva. El permanecer en estados sobre la curva, causa daños en el equipo; si existe poco flujo de vapor y un exceso de presión en el domo, las tuberías y las paredes del domo pueden sufrir daños. Por otro lado, se prefieren aquellos estados bajo la curva, debido a que en esos no hay daños en el equipo, aunque se está trabajando de una forma poco eficiente.



Sea  $MR$  una matriz de recompensa, entonces  $MR$  una matriz de  $54 \times 5 \times 6$ . Se debe construir una matriz para 54 estados, 5 acciones y 6 metas. Los elementos  $mr$  de  $MR$  tienen los siguientes valores:

$$mr_{i,j,k} = R^h(s_i, a_j, k)$$

donde  $R^h$  es una heurística dada por un experto:

$$R^h: S \times A \times \mathcal{K} \rightarrow \{x / x \in \mathcal{R} \text{ y } 0 \leq x \leq 10\}$$

$(s, a, p) \mapsto R^h(s, a, p)$  = la recompensa, según el experto, de realizar la acción  $a$  en el  $s$ , para alcanzar la meta caracterizada por el perfil  $p$ .

El módulo de aprendizaje utiliza un formato específico para la  $MR$ . Esta otra matriz llamada matriz de recompensa de aprendizaje (MRA) tiene elementos  $mra_{i,j,k}$  que representan el número de veces que el operador ha realizado la acción  $a_j \in A$ , en el estado  $s_i \in S$ , bajo la meta descrita en el perfil  $k$ . Se ha agregado una columna a la derecha de la matriz que indica el número de veces que se ha realizado alguna acción en el estado correspondiente a cada renglón de la matriz (v. cap. 4). Para adaptar la matriz dada por el experto a la requerida por el módulo de aprendizaje, se ha puesto la suma de cada renglón en la columna aumentada. La matriz de recompensa para un perfil  $p$ , queda de la siguiente forma:

$$MRA_p = \begin{bmatrix} R^h(s_0, a_0, p) & R^h(s_0, a_1, p) & \cdots & R^h(s_0, a_4, p) & \sum_{i=0}^4 R^h(s_0, a_i, p) \\ R^h(s_1, a_0, p) & R^h(s_1, a_1, p) & \cdots & R^h(s_1, a_4, p) & \sum_{i=0}^4 R^h(s_1, a_i, p) \\ \vdots & \vdots & & \vdots & \vdots \\ R^h(s_{53}, a_0, p) & R^h(s_{53}, a_1, p) & \cdots & R^h(s_{53}, a_4, p) & \sum_{i=0}^4 R^h(s_{53}, a_i, p) \end{bmatrix} \quad (5.7)$$

### 5.3.4 Matriz de observación

Como ya se mencionó en el capítulo 4, la matriz de observación describe la función de observación necesaria para los POMDP. En la investigación de campo realizada no se encontró información cuantificada de errores debido a los dispositivos de medición de  $fvp$  y  $pd$ . De tal manera que no se pudo establecer una función de observación por medio de datos

históricos. No se contaba con datos que pudieran servir para obtener una estadística de las veces que se hace una observación correcta o incorrecta (en su caso, qué se está observando), por cada estado.

La forma en que se obtuvo esta matriz fue por heurísticas proporcionadas por los expertos. Éstos, apoyados en sus años de experiencias, pueden dar una aproximación de lo que en la realidad ocurre. Sin embargo, esta matriz puede ir mejorándose y acercándose mejor a la realidad debido a las modificaciones realizadas por el módulo de aprendizaje.

Sea  $MO$  una matriz de observación requerida por el prototipo, entonces  $MO$  es una matriz de  $54 \times 54$ . Se está considerando cada uno de los estados, y la probabilidad de observar cualquier estado del sistema. Es decir, desde un estado  $s \in S$ , se debe expresar cuál es la probabilidad de observar como estado actual a  $s' \in S$ .

Los elementos  $mo$  de la matriz  $MO$  se calcula de la siguiente manera:

$$mo_{i,j} = P^h(s_i, o_j) \quad (5.8)$$

donde  $P^h$  es una heurística dada por un experto:

$$P^h: S \times O \rightarrow \{x / x \in \mathcal{R} \text{ y } 0 \leq x \leq 1\}$$

$(s, o) \mapsto P^h(s, o) =$  la probabilidad dada por el experto de hacer una observación  $o$ , en el estado  $s$ .

El módulo de aprendizaje utiliza un formato específico para esta matriz. Esta matriz  $MOA$ , tiene elementos  $moa_{ij}$  que representan el número de veces que se ha hecho una observación  $o_j \in O$ , en el estado  $s_i$ . Se ha agregado una columna a la derecha de la matriz que indica el número de observaciones realizadas en el estado correspondiente a cada renglón de la matriz (v. cap. 4). Para adaptar la matriz dada por el experto a la requerida por el módulo de aprendizaje, se ha dado el valor de 1 a la columna aumentada. Por lo que se tiene una matriz de la siguiente forma:

$$MOA = \begin{bmatrix} P^h(s_0, o_0) & P^h(s_0, o_1) & \cdots & P^h(s_0, o_{53}) & 1 \\ P^h(s_1, o_0) & P^h(s_1, o_1) & \cdots & P^h(s_1, o_{53}) & 1 \\ \vdots & \vdots & & \vdots & \vdots \\ P^h(s_{53}, o_0) & P^h(s_{53}, o_1) & \cdots & P^h(s_{53}, o_{53}) & 1 \end{bmatrix} \quad (5.9)$$

## 5.4 Obtención de políticas

Las matrices obtenidas se utilizan en el prototipo para probar la arquitectura. Se presentarán a continuación resultados obtenidos de estas pruebas. Se muestra que la arquitectura obtiene la política óptima de acciones para alcanzar una meta. Esto se realiza con los MDP funcionado de manera correcta en este dominio.

### 5.4.1 Consideraciones

Para el experimento que se mostrará, se consideran inoperantes las partes de la arquitectura que no están relacionadas con la obtención de la política, v.g. el módulo de los POMDP.

Las matrices de transición y de recompensas fueron dadas de acuerdo a lo se mostró en las secciones 5.3.2 y 5.3.3.

Se presenta un experimento en el que se activó una meta que consistía en llegar a un nivel de carga del 80%. El objetivo fue el estado  $s_{27}$ .

Se consideró como **hipótesis** que una política producida por los MDP es óptima [White 93], dado que las heurísticas proporcionadas por los expertos son correctas.

### 5.4.2 Desarrollo del experimento

Al momento de iniciar la ejecución del prototipo, todas las matrices son cargadas. Para el cálculo de la política, se utilizó una  $\gamma=0.9$  y un *error Bellman*  $\epsilon=0.001$ . Cuando el operador activa el perfil correspondiente a la meta (estado  $s_{27}$ ), el prototipo muestra en la tabla correspondiente al módulo de los MDP la política calculada (Ver la Figura 5.3).

También se hicieron experimentos variando los parámetros  $\gamma$  y  $\epsilon$ . Se observó lo que pasaba con el número de iteraciones y con el valor de un barómetro [Brassard y Bratley 97] puesto en la línea 8 del Algoritmo 2.1.

### 5.4.3 Resultados

La política obtenida para alcanzar el estado  $s_{27}$  es óptima, ya que es obtenida por los MDP. Esto fue aprobado por los expertos del IIE. La política se puede observar en la Figura 5.4

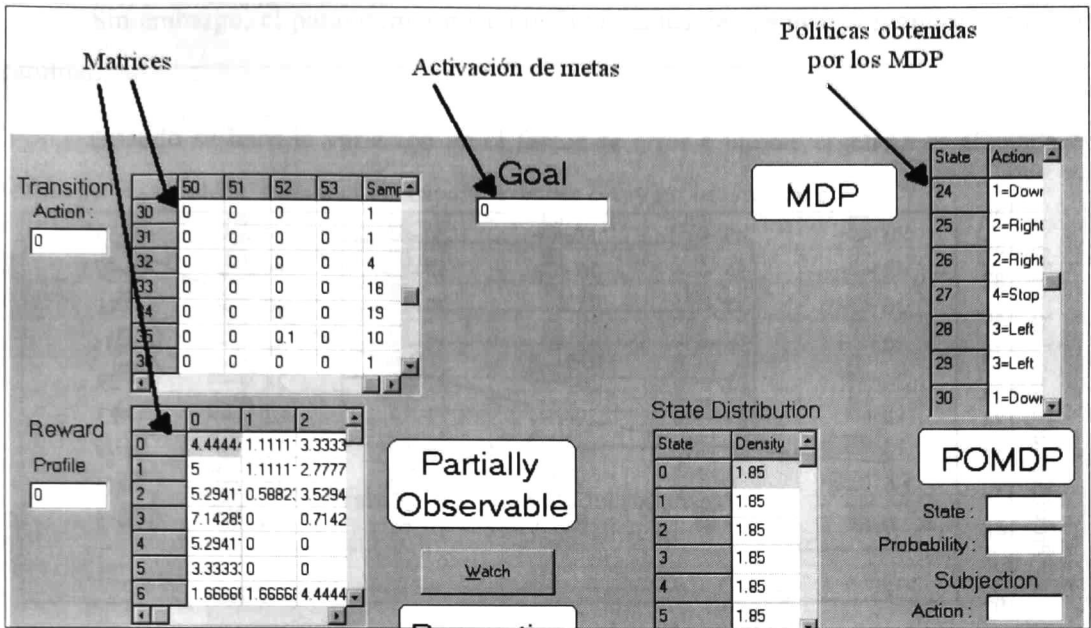


Figura 5.3 Política calculada

De acuerdo a la variación de parámetros hechos en  $\gamma$  y  $\varepsilon$  se obtuvo otra política a ésta, similar a esta aunque con tres variantes: la acción en los estados  $s_{16}$  y  $s_{17}$  es “decremento de flujo de vapor”; y en el estado  $s_{25}$ , “decremento de presión” Esta otra política se obtuvo cuando se modificó el valor de  $\gamma$  a 0.1.

$\gamma$	$\varepsilon = 0,001$	
	Iteraciones	Barómetro
0.1	5	72900
0.3	9	131220
0.5	14	204120
0.7	25	364500
0.9	82	1195560

Tabla 5.5 Variación del parámetro  $\gamma$

Se pudo observar también que, con estas variaciones en  $\gamma$  y  $\varepsilon$ , cuando  $\gamma$  tiende a 1 (por la izquierda), el algoritmo realiza más iteraciones para converger. Cuando tiende a 0, el algoritmo converge más rápido, aunque el grado de profundidad en el análisis de incertidumbre se reduce [Madani et al. 99]. Esto se muestra en la Tabla 5.5 y en la Gráfica 5.1.

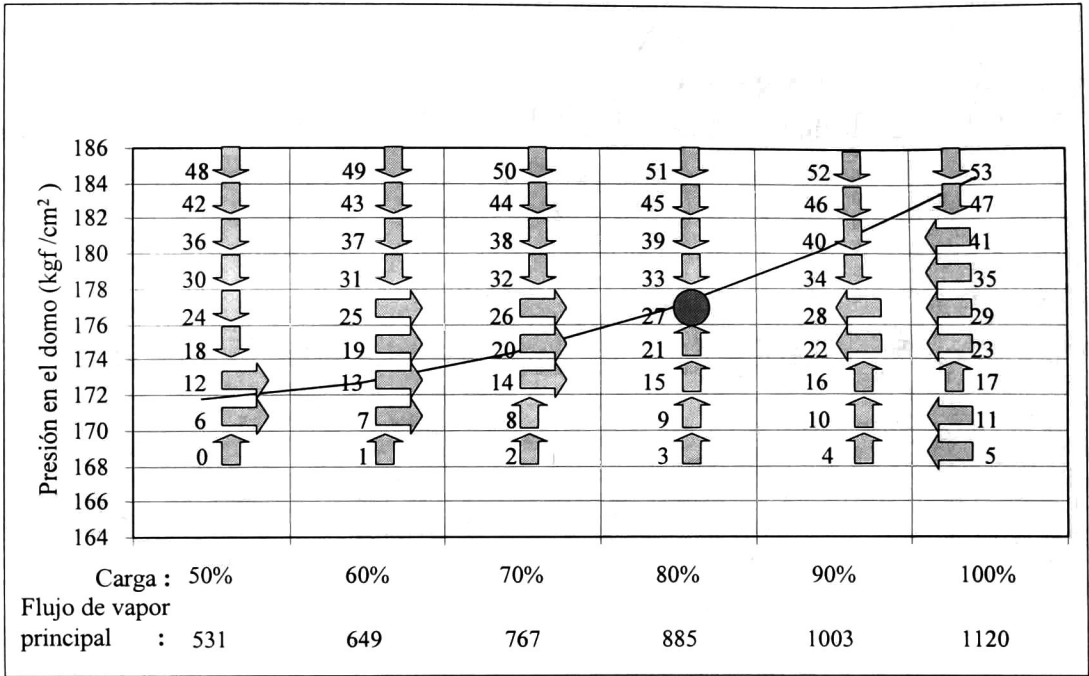
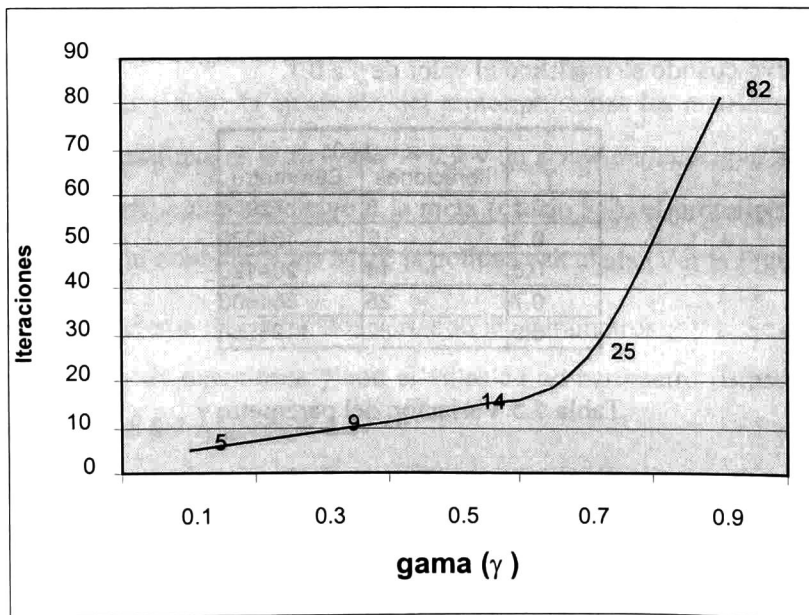


Figura 5.4 Política calculada



Gráfica 5.1 Complejidad de la convergencia variando a  $\gamma$

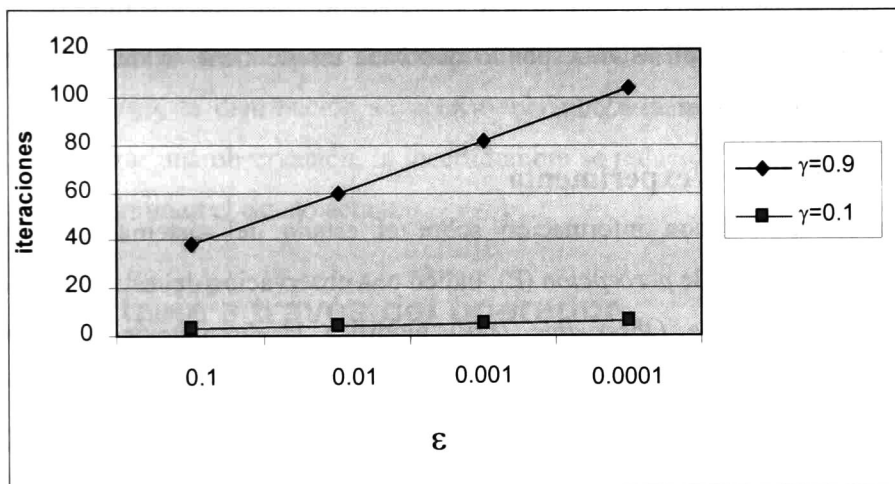


Sin embargo, el parámetro  $\gamma$  queda definida antes de ejecutar el monitor y no debe cambiar.

Cuando se hace la variación en el factor de error  $\epsilon$  la convergencia es afectada en forma polinomial. Se puede observar en la Tabla 5.6 y en la Gráfica 5.2.

$\epsilon$	$\gamma$	
	0.9	0.1
0.1	38	3
0.01	60	4
0.001	82	5
0.0001	104	6

Tabla 5.6 Variación del parámetro  $\epsilon$



Gráfica 5.2 Complejidad de la convergencia variando  $\epsilon$

Si se considera a  $\epsilon=(0.1)^n$ , la complejidad del algoritmo queda en el orden de  $n$ .

## 5.5 Los POMDP

Esta prueba consistió en demostrar que la arquitectura puede determinar, en forma probabilista, el estado actual del sistema cuando en presencia de incertidumbre. La determinación del estado más probable se de acuerdo a una distribución de probabilidad sobre los estados, la cual es modificada cuando se realiza una operación o bien una observación.

### 5.5.1 Consideraciones.

Debido a que no se está llevando el prototipo al sistema real, se simuló el comportamiento utilizando la misma matriz de transición obtenida en el punto 5.3. La matriz de observación también se utilizó para emular los errores que ocurrirían en los sensores. Debido a que en los datos históricos no se cubrieron todas las transiciones; se consideró como **hipótesis** que eran deterministas aquellas no conocidas. Por ejemplo, como no se cuenta información sobre lo que pasa cuando se aumenta la presión en el estado  $s_8$ , se considera que siempre llegará al estado  $s_{14}$ .

Las acciones que se llevaron a cabo fueron las correspondientes a la política obtenida, y mostrada en la sección 0 (ver Figura 5.4). Se estableció el estado  $s_{15}$  como el estado inicial del sistema, y se asumió que el modelo no tiene ninguna información previa sobre el estado inicial del sistema; por lo que cada estado tiene la misma probabilidad (0.0185); para toda  $s$ , tal que  $s \in S$ ,  $E(s)=0.0185$ .

### 5.5.2 Desarrollo del experimento

Por no contar con información sobre el estado del sistema, se realizó una observación. El módulo de *percepción* (P), indicó una observación del estado  $s_{15}$ ; por lo que el módulo *Parcialmente Observable* (MP) modifica la distribución a  $E(9)=0.1739$ ,  $E(14)=0.0435$ ,  $E(15)=0.6087$  y  $E(16)=E(21)=0.087$ . Los POMDP determinaron entonces que el sistema estaba en  $s_{15}$  con una probabilidad de 0.6087 y sugirieron la acción de  $\Delta p = +$  ( $a_0$ ).

Después de la acción, el sistema alcanzó el estado  $s_{21}$ , teniendo que  $E(20)=E(22)=E(28)=0.0105$ ,  $E(21)=0.3789$ ,  $E(27)=0.0526$ ,  $E(29)=0.1579$  y  $E(35)=0.3789$ . Los POMDP encontraron que hay dos estados más probables en que puede estar el sistema,  $s_{21}$  y  $s_{35}$ ; por lo que no hizo sugerencia al operador. Por casos como el anterior, se recomienda realizar una observación para determinar el estado. En esta observación, se percibió el estado  $s_{21}$ , por lo que se tiene que  $E(20)=0.0002$ ,  $E(21)=0.9844$ ,  $E(22)=0.0039$  y  $E(27)=0.0098$ . En este caso se sugirió la acción  $\Delta p = +$  en el estado  $s_{21}$ .

Una vez realizada la acción, se alcanzó el estado  $s_{27}$  y se tuvo  $E(27)=0.4922$ ,  $E(28)=E(34)=E(35)=0.0007$ ,  $E(29)=0.4961$  y  $E(33)=E(41)=0.0049$ . Los POMDP determinaron estar en el estado  $s_{29}$ ; esto no era así. Sin embargo, las probabilidades de

haber estado en  $s_{29}$  o  $s_{27}$  son muy parecidas. Lo anterior es otra razón más por la que se recomienda realizar una observación antes de ejecutar una acción. Al realizar la observación, se percibió el estado  $s_{27}$  con la probabilidad  $E(27)=0.9993$  mientras que se tiene que  $E(28)=0.0001$  y  $E(33)=0.0006$ . Los POMDP sugirieron entonces, no hacer cambios en las variables, porque se había alcanzado la meta.

### 5.5.3 Resultados

Como resultado se tuvo que los POMDP funcionan correctamente. Se observa que el módulo PO modifica la distribución de estados dada una acción o una observación. A través de esta distribución de estados, los POMDP determinan el estado actual del sistema y sugieren al operador la acción óptima a realizar de acuerdo a la política obtenida por los MDP. En el experimento se pudo observar que la distribución de estados, en algunos casos, presenta cierta ambigüedad después de realizar una acción, debido a la incertidumbre que existe. Sin embargo, la distribución se acerca a lo que está sucediendo en la realidad. Después de realizar una observación, la incertidumbre se reduce notablemente, por lo que los POMDP determinan el estado actual.

## 5.6 Aprendizaje a través del operador

Se mostrará a continuación que la arquitectura aprende del comportamiento tanto del ambiente del sistema, como de las acciones del operador. Con lo aprendido, se pueden crear políticas más apropiadas al comportamiento real del sistema y de acuerdo a como se comporta el operador, clonando la operación de éste.

### 5.6.1 Consideraciones

Para mostrar los resultados obtenidos por el módulo de aprendizaje, se utilizaron matrices especiales de transición, observación y recompensas. Para el aprendizaje del ambiente del sistema (*Modificación de Transición y Modificación de Observación*), las matrices de observación y transición no contenían datos de muestra. Para el aprendizaje de recompensas (*Modificación de Recompensas*), la matriz de recompensas solamente indicaba la acción de parar ( $a_4$ ). Se consideró, que la meta era alcanzar el estado  $s_{27}$ .

Para emular el comportamiento del sistema, se hicieron las mismas consideraciones del experimento mostrado en la sección 5.5.

## 5.6.2 Desarrollo del experimento

Iniciando en el estado  $s_0$ , en cada estado alcanzado se realiza una acción  $a \in A$  en forma aleatoria. Se hicieron pruebas con 1000, 5000 y 10000 acciones. Por cada experimento realizado se grabaron las matrices obtenidas y se compararon con las matrices que dan el comportamiento del sistema (transiciones y observaciones).

Para la prueba de aprendizaje del comportamiento del operador, se estableció un comportamiento específico del operador, el cual consistió en bajar la presión si el sistema se encontraba en un estado sobre la curva; en caso contrario, aumentar el  $f_v$  si éste es menor que el requerido para llegar a la meta, o disminuirlo si era mayor.

## 5.6.3 Resultados

Se mostró que los módulos de aprendizaje modifican las matrices de la arquitectura clonando el comportamiento tanto del sistema como el del operador. Como ejemplo, se muestra cómo se fue adquiriendo el conocimiento sobre lo que ocurría en el estado  $s_{35}$  cuando se realizaba la acción  $a_0$ . Esto se puede ver en la Tabla 5.6. Se observa que se va acercando a lo que se estableció en las matrices de comportamiento del sistema, cuyas transiciones son  $MT_{35,41,0} = 0.8$ ,  $MT_{35,47,0} = 0.1$  y  $MT_{35,52,0} = 0.1$ . En el apéndice se muestran las matrices completas.

acciones totales	acciones $a_0$ en $s_{35}$	estados alcanzados			probabilidades		
		s41	s47	s52	$P(s_{41} s_{35},a_0)$	$P(s_{47} s_{35},a_0)$	$P(s_{52} s_{35},a_0)$
1000	17	16	1	0	0.94	0.06	0.00
5000	62	50	7	5	0.81	0.11	0.08
10000	124	101	12	11	0.81	0.10	0.09

Tabla 5.7 Transición en el estado  $s_{35}$

Referente a los sensores, la arquitectura obtuvo información de las fallas en las observaciones. En la Tabla 5.7 se presenta cómo fueron las modificaciones en la matriz de observación para el estado  $s_{35}$ . Se puede ver que se acerca mucho a lo que se estableció en la matriz del comportamiento del sistema, que tiene las probabilidades de observación:  $O_{35,29} = 0.15$ ,  $O_{35,34} = 0.05$ ,  $O_{35,35} = 0.6$  y  $O_{35,41} = 0.2$ . La matriz completa se puede observar en el apéndice.



observaciones totales	observaciones en $s_{35}$	observaciones				Probabilidades			
		$o_{29}$	$o_{34}$	$o_{35}$	$o_{41}$	$P(o_{29} s_{35})$	$P(o_{34} s_{35})$	$P(o_{35} s_{35})$	$P(o_{41} s_{35})$
1000	67	10	4	41	12	0.149	0.060	0.612	0.179
5000	343	57	20	205	61	0.166	0.058	0.598	0.178
10000	621	95	28	398	100	0.153	0.045	0.641	0.161

Tabla 5.8 Observación en el estado  $s_{35}$

El comportamiento del operador también fue clonado por la arquitectura modificando la matriz de recompensas. Cuando el operador realizaba con más frecuencia una acción en un estado, la matriz quedaba con mayor recompensa para esa acción en dicho estado. Ejemplo, como la matriz de recompensa de entrada,  $MT_0$ , indica solamente la acción de “parar” se tuvo que  $MT_0[0] = [0 \ 0 \ 0 \ 0 \ 1]$ . Cuando en el estado  $s_0$  se ejecuta por primera vez la acción  $a_2$ , se tuvo que  $MT_0[0] = [0 \ 0 \ 1 \ 0 \ 1]$ , por lo que parar o aumentar  $f_v$  tendrían la misma recompensa en  $s_0$ . Cuando se volvió a ejecutar  $a_2$  en  $s_0$ ,  $MT_0[0] = [0 \ 0 \ 2 \ 0 \ 1]$ , por lo que en ese momento la acción  $a_2$  es la de mayor recompensa en el estado; por lo que los MDP en ese momento sugieren realizar  $a_2$  en dicho estado.

Con el experimento se obtuvo la matriz de recompensa mostrada en el apéndice XX, obteniendo la política de la Figura 5.5. Se puede observar que se recomienda una política similar al comportamiento del operador. En el estado  $s_{25}$  se sugiere una acción distinta a la que el operador realiza, esto debido a que aún con las acciones que ha realizado el operador, los MDP calculan mayor recompensa esperada para la acción propuesta en la política.



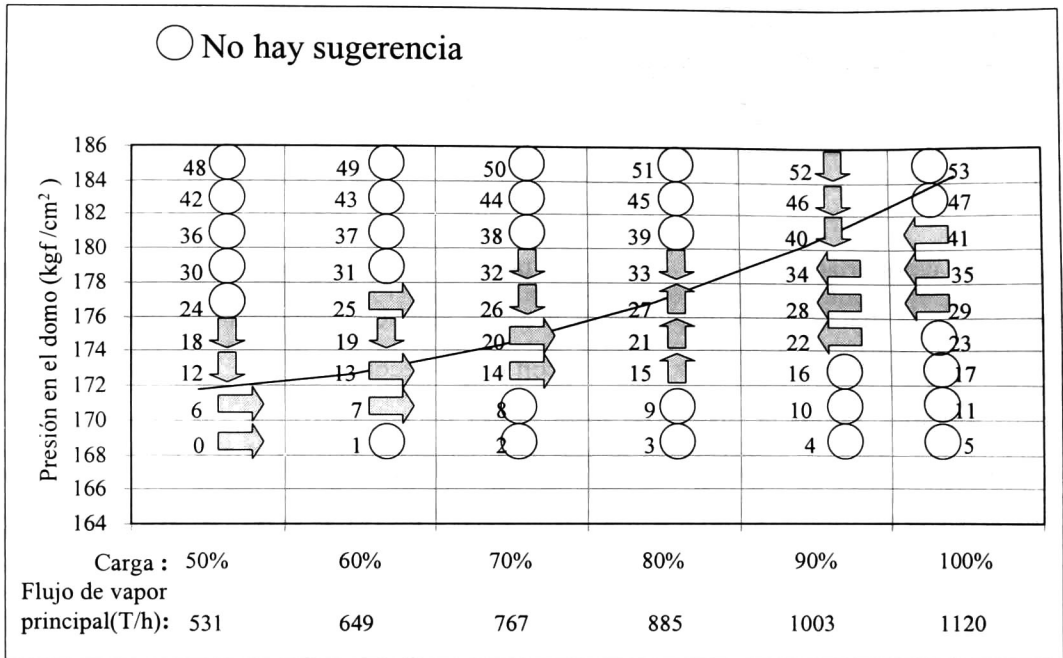


Figura 5.5 Política dado un aprendizaje.

## Capítulo 6 Conclusiones y trabajos futuros

---

En este capítulo se presentan las conclusiones a partir de los resultados expuestos en la tesis. Una solución al problema relacionado con la planificación en dominios con incertidumbre es mostrada de tal manera que quedan satisfechos los objetivos propuestos al inicio del trabajo. Finalmente, se mencionan algunos puntos que pueden ser desarrollados en trabajos futuros.

---

## 6.1 Conclusión general

El problema de planificación en dominios donde existe incertidumbre, tanto en los efectos de las acciones, como en las lecturas de los sensores, puede tener una solución a través de una arquitectura de ayuda a la toma de decisiones. En la generación de electricidad, existen maniobras y procedimientos de operación que requieren tomar decisiones de una manera rápida y efectiva. La planificación inteligente representa un método que ha mostrado efectividad en la generación de ayudas para los operadores. Sin embargo, algunos de esos procedimientos presentan altos índices de incertidumbre. Los Procesos de Decisión de Markov (MDP) y los MDP Parcialmente Observables (POMDP) son herramientas de Inteligencia Artificial que permiten abordar el problema. La solución se presenta a través de una arquitectura, basada en los MDP y POMDP, cuya base de conocimientos está sustentada con el conocimiento de expertos; la cual cambia a través de un módulo que aprende del comportamiento del sistema y de las acciones que se realiza el operador. El problema fue abordado a través de una técnica para la generación de ayuda para la operación del domo de una central termoeléctrica.

Como las principales aportaciones de esta tesis se pueden destacar las siguientes:

- a. desarrollo de una arquitectura para sugerir acciones en presencia de incertidumbre, la cual utiliza los MDP y POMDP, además de un módulo de aprendizaje que adapta el conocimiento experto a las características particulares del proceso [Agueda e Ibargüengoytia 01 b]
- b. utilización de las técnicas MDP y POMDP en aplicaciones reales [Agueda e Ibargüengoytia 01 a]
- c. desarrollo de un sistema de soporte en la operación del (control de) domo de un generador de vapor de una CTE.

En la elaboración de la tesis se cumplió con:

### 1) El estudio del dominio de aplicación

En el capítulo 3, dada la necesidad de contar con planificadores en ciertos dominios de las Centrales Termo-Eléctricas (CTE), en especial para auxiliar en el control del domo,

se hizo un estudio sobre la incertidumbre presentada en el domo. De este análisis, se pudieron obtener los elementos suficientes que permiten plantear una solución al problema de planificación para el domo.

## **2) La Definición de la herramienta de Inteligencia Artificial utilizada**

En el capítulo dos se mostró que los MDP y los POMDP se pueden utilizar como herramientas una solución al problema de planificación en dominios estocásticos. Estas herramientas han sido de gran interés por la comunidad científica, por lo que se han obtenido algoritmos cada vez más eficientes para resolver los procesos de decisión. Los MDP obtienen la política óptima de acciones cuando existe incertidumbre en los efectos de las acciones. Los POMDP además determinar el estado más probable en que puede estar el sistema real dada la incertidumbre de las lecturas de los sensores.

## **3) La especificación de los requerimientos para utilizar la herramienta**

En el capítulo tres se mostró la correspondencia entre algunos elementos del dominio y los elementos necesarios para desarrollar un modelo MDP y POMDP. Con esto se demostró la factibilidad de la utilización de la herramienta del capítulo dos, para solucionar el problema descrito en el capítulo uno.

## **4) El desarrollo de una arquitectura**

En el capítulo cuatro se presentó AUDA, una arquitectura basada en las propiedades de los MPD y los POMDP. La arquitectura está diseñada para sugerir la acción óptima cuando algún agente (en este caso el operador) la requiera. Además, AUDA adquiere conocimientos durante su ejecución a través de un módulo de aprendizaje, el cuál actualiza los valores de las matrices utilizadas para resolver los procesos de decisión.

## **5) El Desarrollo de un prototipo de AUDA**

En el capítulo cinco se presenta AUDA programada en un prototipo para el Instituto de Investigaciones Eléctricas. Con este prototipo se llevaron a cabo los experimentos presentados en esta tesis.

## **6) La realización de pruebas con el prototipo**

En el capítulo cinco se presentaron los experimentos realizados con AUDA. Con las matrices de los MDP construidas con datos reales, se obtuvieron políticas óptimas correspondientes a las metas dadas; así mismo con los POMDP se determinaron los estados actuales del sistema. Con esto, se pudieron dar sugerencias al operador. El módulo de aprendizaje también se probó, demostrando que las matrices son modificadas de acuerdo al comportamiento del sistema. Los resultados demuestran que los módulos de la arquitectura funcionan correctamente. Con las pruebas se ha logrado mostrar que el problema de planificación bajo incertidumbre puede ser abordado.

## **6.2 Discusiones**

A pesar de los objetivos alcanzados, en el trabajo presentado contiene puntos que pueden presentarse a discusión:

### **1) Autonomía de la arquitectura**

La arquitectura no es autónoma. No se realizan acciones directas sobre el ambiente, sino que se presentan al operador para que éste confirme la acción. La arquitectura desde un principio fue diseñada para que un operador humano llevara a cabo la acción; sin embargo éste puede ser sustituido por algún otro agente inteligente autónomo adecuado que no sea humano.

AUDA es una arquitectura que interactúa con el ser humano, ya que de éste toma gran parte de su conocimiento. Esto pudiera ser un importante punto de discusión debido a que si el operador es torpe en su desenvolvimiento, el agente inteligente con arquitectura AUDA se comportará de la misma forma. Sin embargo, se puede considerar un periodo de entrenamiento para la arquitectura, donde serán operadores expertos y hábiles los que trabajarán con el agente.

### **2) El conjunto de estados**

El conjunto de estados que se estableció en el presente trabajo, puede considerarse pequeño para la complejidad del problema real. Sin embargo, se considera un conjunto



suficiente para hacer la aplicación real, además los algoritmos son eficientes (polinomiales) respecto al número de estados del sistema.

### **3) Herramienta de IA utilizada**

Se puede pensar que los MDP y los POMDP no son los únicos que se pueden utilizar para solucionar el problema. Sin embargo son los más aceptados para resolver el problema de planificación en presencia de incertidumbre. La mayoría de los trabajos enviados al congreso de incertidumbre en la inteligencia artificial (UAI), presentan estas herramientas. Ninguna de estas herramientas se ha utilizado en una aplicación real, por lo que una aproximación a una aplicación real es de gran interés por la comunidad.

En este punto, cabe mencionar que se utilizaron por separado los MDP y los POMDP. Los primeros, son utilizados exclusivamente para obtener las políticas; mientras que los segundos se utilizaron para determinar el estado más probable de acuerdo a las observaciones que se realizan del ambiente. En el desarrollo del documento se ha justificado esta idea.

## **6.3 Trabajos futuros**

Debido a los resultados obtenidos durante la tesis y a la aceptación que se ha observado, tanto en el Instituto de Investigaciones Eléctricas como en la comunidad de Inteligencia Artificial, se ha despertado gran entusiasmo para seguir trabajando sobre la línea trazada en la tesis. De manera que no queda duda que esta investigación no tiene el carácter de concluyente. A continuación se muestran algunos trabajos que pueden ser desarrollados a partir de éste.

### **1) Ampliación de los datos históricos**

A pesar de haber utilizado 736 horas de datos históricos, no se abarcó todo el espacio del dominio. Se considera conveniente obtener más datos históricos para crear matrices más confiables. Se puede también utilizar el prototipo mostrado en la tesis para acoplarlo a una central de tal forma que a través del módulo de aprendizaje de AUDA, se vayan adquiriendo información de los sucesos percibidos.

## **2) Proceso de obtención de matrices**

Debido al formato en el cual se presentaron los datos históricos utilizados, ha sido difícil establecer exactamente la relación causal entre las acciones realizadas y los efectos observados en el sistema. En el capítulo cinco, se muestra cómo se obtuvieron las matrices. En particular, la matriz de transición se pueden obtener de una forma más formal, si los datos a utilizar están en el formato adecuado. Por tanto, se sugiere, seguir haciendo minería de datos para encontrar una forma más adecuada de crear la matriz de transición. Asimismo, se pueden buscar datos históricos sobre fallas de sensores para crear la matriz de observación con base histórica y no únicamente con la experiencia de operadores.

## **3) Implantación del prototipo a una Central Termo Eléctrica**

Aunque la técnica desarrollada fue estudiada a partir de un problema real y para un dominio real, aún no se puede presentar como una herramienta acoplada en un ambiente real. Se debe trabajar sobre módulos que permitan la aplicación real, con el fin de presentar el prototipo como una herramienta de ayuda a un operador en planta. Estos módulos podrían ser módulos de adquisición de datos, de validación de información, de consulta y de operación.

## **4) Aplicación en otro dominio.**

Los dominios que presentan incertidumbre en los efectos producidos por una acción y sobre las observaciones percibidas del sistema, no son propios únicamente del domo de vapor de una CTE. Existen otros dominios reales con el problema de calcular planes de acción bajo estas incertidumbres. Aunque todo el desarrollo presentado en la tesis fue basado en el problema del domo, la aplicación de la técnica y la arquitectura quedan abiertas para ser utilizadas en otro dominio real como: visión por computadora, consulta de base de datos distribuidos, mercadotecnia, diagnóstico médico, etc.

# Referencias

- [Agueda e Ibargüengoytia 01 a] Mario E. Agueda y Pablo Ibargüengoytia. *Control of a Power Plant Using MDP and POMDP*. 3er. Encuentro Internacional de Ciencias de la Computación. Septiembre 2001. (Aceptado)
- [Agueda e Ibargüengoytia 01 b] Mario E. Agueda y Pablo Ibargüengoytia. *An Architecture for planning in Uncertain Domains*. The Thirteenth IEEE International Conference on Tools with Artificial Intelligence. Dallas, Texas, 2001. (Sometido).
- [Arroyo et al. 98] G. Arroyo-Figueroa, E. Solís, A. Villavicencio y E. Súcar. *SADEP –a fuzzy diagnostic system shell- an application to fossil power plant operation*. Experts Systems with Applications (14):43-52. Elsevier Science Ltd.1998.
- [Bayouth et al. 97] Michelle Bayouth, Illah Nourbakhsh and Charles Thorpe. *A Hybrid Human-Computer Autonomous Vehicle Architecture*. In Proceedings, Third ECPD International Conference on Advanced Robotics, Intelligent Automation and Control. 1997
- [Bellman 57] R. E. Bellman. *Dydamic Programming*. Princenton University Press, Princeton, New Jersey. 1957
- [Boutilier y Poole 96] C. Boutilier y D. Poole. *Computing optimal policies for partially observable decision processes using compact representations*. In Proceeding of the Thirteenth National Conference on Artificial Intelligence, 1168-1175. 1996.
- [Brassard y Bratley 97] G. Brassard y P. Bratley. *Fundamentos de Algoritmia*. Prentice Hall, Madrid, 1997. p. 118
- [Bratman 87] Michael Bratman. *Intentions, Plans and Practical*. Harvard University Press. Cambridge, MA, USA. 1987.
- [CAOI 95] Centro de Adiestramiento de Operadores Ixtapantongo. *Documento S007: "Sistema Agua de Alimentación"*. Comisión Federal de Electricidad. México. 1 de Septiembre de 1995
- [Cassandra et al. 94] Anthony R. Cassandra, Leslie Pack Kealbling and Michael L. Littman. *Acting Optimally in Partially Observable Stochastic Domains*. In proceeding of the twelfth National Conference on Artificial Intelligence, Seattle, WA, 1994.
- [Cassandra et al. 97] Anthony Cassandra, Michael L. Littman and Nevin L. Zhang. *Incremental Pruning: A Simple, Fast, Exact Method for Partially Observable Markov Decision Processes*. Uncertainty in Artificial Intelligence (UAI). 1997.



- [Cassandra 98] Anthony R. Cassandra. *Exact and Approximate Algorithms for Partially Observable Markov Decision Processes*. Ph. D. Thesis. Brown University, Department of Computer Science, Providence, RI, 1998.
- [Cassandras 94] Christos G Cassandras y Stéphane Lafortene. *Introduction to Discrete Event Systems*. Kluwer Academic Publishers.
- [Dean et al. 95] Thomas Dean, Leslie Pack Kelbling, Jack Kirman y Ann Nicholson. *Planning under Time Constraints in Stochastic Domains*". Artificial Intelligence, Vol 76:35-74. 1995.
- [Dean y Wellman 91] Thomas L. Dean y Michael P. Wellman. *Planning and control*. Morgan Kaufmann Publishers. San Mateo, CA, USA. 1991.
- [Delgadillo 96] Miguel Delgadillo Valencia. *Recuperadores de calor*. Documento final del emulador de Central Termoeléctrica de Ciclo Combinado, tomo I. IIE, Proyecto IC5117. Departamento de control e instrumentación. Cuernavaca. 1996.
- [Fikes y Nilsson 71] R. E. Fikes y N. J. Nilsson. *STRIPS: a new Approach to the Application of Theorem Proving to Problem Solving*. Artificial Intelligence, 2(3-4):189-208. 1971.
- [Green 69] C. Green. *Theorem-proving by resolution as a basis for question-answering systems*. In Meltzer, B, Michie, D. and Swann, M., editors, Machine Intelligence (4):183-205. Edinburgh University Press, Edinburgh, Scotland. 1969.
- [Ibargüengoytia et al. 99] Pablo Ibargüengoytia, Dionisio Suárez y Alejandro Villavicencio, *Sistema Inteligente para Ayuda en el arranque de Unidades Termoeléctricas*. Boletín IIE, Año 23, (23):186-192. 1999.
- [Ibargüengoytia y Reyes 01] Pablo Ibargüengoytia G. y Alberto Reyes. *Continuos Planning for the Operation of Power Plants*. Encuentro Internacional de Computación. México 2001.
- [IIE 98] Boletín IIE, año 22, Septiembre – Octubre, 1998. Vol. 22, núm. 5, pp 210,211.
- [Kealbling et al. 98 ] Leslie Pack Kealbling, Michael L. Littman y Anthony R. Cassandra. *Planning and Acting in Partially Observable Stochastic Domains*. Artificial Intelligence, 101(1-2), 99-134.
- [Littman et al. 95 ] Michael L. Littman, Thomas L. Dean, and Leslie Pack Kaelbling. *On the Complexity of Solving Markov Decision Problems*. International Conference on Uncertainty in Artificial Intelligence, 1995.
- [Lusena et al. 01 ] Christopher Lusena, Judy Goldsmith and Martin Mundhenk. *Nonapproximability Results for Partially Observable Markov Decision Processes*. Artificial Intelligence Research 14: 83-103. 2001.
- [Madani et al. 99 ] O. Madani, S. Hank y A. Condon. *On the undecidability of probabilistic planning and infinite-horizon partially observable Markov decision problems*. In Proceeding of the 16<sup>th</sup> National Conference on Artificial Intelligencc (pp. 541-548).

- [Nikovski y Nourbakhsh] Daniel Nikovski y Illah Nourbakhsh. *Learning Probabilistic Models for Decision-Theoretic Navigation of Mobile Robots*. Proceedings of the International Conference on Machine Learning 2000, San Francisco, CA, U.S.A. Morgan Kaufmann. 2000.
- [Puterman 94 ] Martin L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. A Wiley-Interscience publication, Manchester, 1994.
- [Russell y Norvig 94] Stuart J. Russell y Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Englewood Cliffs, New York, 1994.
- [Sawaki y Ichikawa 78] K. Sawaki y A. Ichikawa. *Optimal control for partially observable Markov decision processes over an infinite horizon*. Journal of the Operations research Society of Japan 21(1):1-14. 1978.
- [Urbancic y Bratko] Tanja Urbancic e Ivan Bratko. *Reconstructing Human Skill with Machine Learning*. ECAI 94. 11<sup>th</sup> European Conference on Artificial Intelligence . A. Cohn Published. p.p.498-502. 1994.
- [Valdez 01] José G. Valdez. *Desarrollo de un Monitor de Ejecución de un Planificador Inteligente*. Proyecto de Investigación para obtener el título de Ingeniero en Sistemas Computacionales. Instituto Tecnológico de Tuxtla Gutiérrez. Chiapas, México. Mayo 2001.
- [White 92] Douglas John White. *Markov decision processes*. John Wiley & Sons Ltd, New York, 1992.
- [Wilkins y Myers 98] David E. Wilkins y Karem L. Myers. *A Multiagent Planning Architecture*. International Conference on AI Planning Systems, p.p. 154-162. 1998.
- [Zhang y Liu 92] N.L. Zhang y W. Liu. *Planning in stochastic domain: Problem characteristics and approximation*. Technical Report HKUST-CS96-31, Department of Computer Science, Hong Kong University of Science and Technology. 1992.



# Apéndice

Se presenta en este apéndice las matrices utilizadas para el prototipo y algunas obtenidas en los experimentos.

Las matrices A.1 a la A 4 son las correspondientes a las de transición dadas las acciones  $a_0$ , a la  $a_3$ . Para la acción  $a_4$ , los elementos  $mt_{i,j}$  de la matriz de transición MT, se definen como  $mt_{i,j} = 1$  si  $i=j$ , 0 en otro caso.

La matriz A.4 es la que se utilizó como *Matriz de Recompensas* para el experimento mostrado en el capítulo 5. Es la correspondiente a la meta de alcanzar el estado  $s_{27}$ .

La matriz A.5 es la que se utilizó en el experimento como *Matriz de Observación*.

Las matrices de la A.6 a la A.10 son las matrices de transición que se obtuvieron a través del módulo de aprendizaje de AUDA después de 1000 iteraciones. El experimento se puede leer en el capítulo 5.

Por último la matriz A.11 es la matriz de observación obtenida por el módulo de aprendizaje de AUDA en el experimento mostrado en el capítulo 5.

En las matrices de aprendizaje (A.6 a A.11) tienen una columna aumentada con título "NM" Esta columna indica el número de muestras que se realizaron en la fila correspondiente. Se debe recordar la forma en que fueron definidas las matrices utilizada por el módulo de aprendizaje.















	0	1	2	3	4
0	8	2	6	0	2
1	9	2	5	1	1
2	9	1	6	1	0
3	10	0	1	3	0
4	9	0	0	8	0
5	5	0	0	10	0
6	3	3	8	2	2
7	6	4	8	3	2
8	9	1	9	2	1
9	10	0	1	3	1
10	9	0	0	8	0
11	3	0	0	10	0
12	3	5	9	1	2
13	4	6	8	3	3
14	8	3	8	4	1
15	10	0	3	5	2
16	5	0	1	8	1
17	2	0	0	10	0
18	2	9	7	3	0
19	3	9	9	1	0
20	2	7	9	3	0
21	10	4	2	5	4
22	6	0	1	9	1
23	3	0	0	10	0
24	0	9	6	0	0
25	0	9	6	0	0
26	0	8	10	1	0
27	2	4	3	2	10
28	7	2	1	9	4
29	3	1	0	9	2
30	0	10	5	0	0
31	0	10	5	0	0
32	0	9	7	0	0
33	0	10	7	2	3
34	3	8	1	4	5
35	1	1	0	9	3
36	0	8	8	0	0
37	0	8	8	0	0
38	0	9	8	0	0
39	0	10	8	0	1
40	1	10	5	3	3
41	3	8	0	9	0
42	0	10	9	0	0
43	0	10	9	0	0
44	0	10	9	0	0
45	0	10	8	0	0
46	0	10	7	1	0
47	1	9	0	6	2
48	0	10	9	0	0
49	0	10	9	0	0
50	0	10	9	0	0
51	0	10	7	0	0
52	0	10	7	3	0
53	0	10	0	5	1

Matriz A.4 Matriz de recompensa para el perfil 4 (meta =  $s_{27}$ )













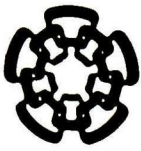












**CENTRO DE INVESTIGACION Y DE ESTUDIOS AVANZADOS DEL IPN  
UNIDAD GUADALAJARA**

El Jurado designado por la Unidad Guadalajara del Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, aprobó la tesis:

Arquitectura para ayuda en la toma de decisiones en presencia de incertidumbre

del C. Mario Enrique Agueda Herrera  
el día 14 de Diciembre del 2001.

Dr. Ofelia Begovich Mendoza  
Investigador Cinvestav 3 A  
CINVESTAV DEL IPN  
Guadalajara

Dr. Luis Ernesto López Mellado  
Investigador Cinvestav 3 A  
CINVESTAV DEL IPN  
Guadalajara

Dr. Antonio Ramírez Treviño  
Investigador Cinvestav 2 A  
CINVESTAV DEL IPN  
Guadalajara

Félix Francisco Ramos Corchado  
Investigador Cinvestav 2 A  
CINVESTAV DEL IPN  
Guadalajara



CINVESTAV  
BIBLIOTECA CENTRAL



SSIT000003885