

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

CAMPUS ESTADO DE MÉXICO



Sistema de planeación de caminos para robots autónomos

**Tesis que para obtener el grado de
Maestría en Ciencias Computacionales Presenta**

Martín Humberto Cabañas Nazuno

BIBLIOTECA

Abril 1996



CONTENIDO

Agradecimientos

Capítulo 1 Antecedentes

1.1	<i>Generalidades</i>	9
1.2	<i>Desarrollo Historico</i>	12
1.3	<i>Evolución de los robots autónomos</i>	15

Capítulo 2 Introducción

2.1	<i>Sistemas Clasicos de planeación</i>	22
2.2	<i>Sistemas de Control Reactivo</i>	26
2.3	<i>Enfoque y Contribuciones de este desarrollo</i>	29

Capítulo 3 Trabajos Previos: Planeación con incertidumbre

3.1	<i>Antecedentes</i>	34
3.2	<i>Generación de planes probabilisticos</i>	34
3.3	<i>Selección de planes usando diagramas de influencia</i>	35
3.4	<i>Planeación mediante el proceso de desición de Markov</i>	36
3.5	<i>Reglas de Control situado</i>	38
3.6	<i>Selección de acciones dinamicas por ampliación de la activación</i>	39
3.7	<i>Sistemas Puramente reactivos</i>	40

Capítulo 4 Modelo Propuesto

4.1	<i>Consideraciones Generales</i>	43
4.2	<i>Un ejemplo de planeación con BURIDAN</i>	46
4.3	<i>Modelo propuesto</i>	47
4.4	<i>Interacción con el mundo</i>	49
4.5	<i>Interpretación del mundo</i>	51
4.6	<i>Algoritmo para la planeación</i>	51
4.7	<i>Implementación de la red</i>	59
4.8	<i>Sumario</i>	68

Capítulo 5 Conclusiones

5.1	<i>Contribuciones del modelo</i>	72
5.2	<i>Comparación con otros desarrollos</i>	73
5.3	<i>Limitaciones y futuros desarrollos</i>	74
	5.3.1 <i>Costo del sensado</i>	74
	5.3.2 <i>Limitaciones de hardware</i>	74
	5.3.3 <i>Aprendizaje de las acciones realizadas</i>	75
5.4	<i>Limitaciones de BURIDAN</i>	75
5.5	<i>Trabajos a futuro</i>	75

Bibliografía

Lista de Figuras

<i>Fig 1.1</i>	<i>Escena de la trilogía filmica Guerra de las Galaxias</i>	<i>20</i>
<i>Fig 2.1</i>	<i>Clasificación en base al tiempo del tipo de desiciones</i>	<i>25</i>
<i>Fig 4.1</i>	<i>El plan encodifica el estado inicial de la distribución y la meta</i>	<i>45</i>
<i>Fig 4.2</i>	<i>Plan para sostener un bloque</i>	<i>47</i>
<i>Fig 4.3</i>	<i>Escenario de ejemplo</i>	<i>53</i>
<i>Fig 4.4</i>	<i>Plan generado por BURIDAN</i>	<i>53</i>
<i>Fig 4.5</i>	<i>Estado hasta la etapa 11 del plan</i>	<i>55</i>
<i>Fig 4.6</i>	<i>Plan de contingencia</i>	<i>56</i>
<i>Fig 4.7</i>	<i>Estado hasta la consecución de la meta</i>	<i>56</i>
<i>Fig 4.8</i>	<i>Diagrama de flujo del algoritmo</i>	<i>58</i>
<i>Fig 4.9</i>	<i>Señal sigmoide</i>	<i>59</i>
<i>Fig 4.10</i>	<i>Distribución de sensores</i>	<i>60</i>
<i>Fig 4.11</i>	<i>Estadística de los patrones de entrenamiento</i>	<i>62</i>
<i>Fig 4.12- 4.15</i>	<i>Comportamiento de los sensores</i>	<i>63</i>
<i>Fig 4.16</i>	<i>Comportamiento de la salida</i>	<i>64</i>
<i>Fig 4.17</i>	<i>Comportamiento de los patrones de entrenamiento</i>	<i>64</i>
<i>Fig 4.18</i>	<i>Estadísticas del entrenamiento</i>	<i>65</i>
<i>Fig 4.19- 4.22</i>	<i>Configuración de las capas de la red</i>	<i>66</i>
<i>Fig 4.23</i>	<i>Resultados despues del entrenamiento</i>	<i>66</i>
<i>Fig 4.24</i>	<i>Grafica del error promedio</i>	<i>67</i>
<i>Fig 4.25</i>	<i>Conjunto de prueba de la red</i>	<i>67</i>
<i>Fig 4.26</i>	<i>Resultados del conjunto de prueba</i>	<i>68</i>

Lista de Tablas

<i>Tabla 4.1</i>	<i>Patrones de entrenamiento de la red neuronal</i>	<i>62</i>
------------------	---	-----------

1 ANTECEDENTES



1.1 Generalidades

Cuando escuchamos la palabra robot la respuesta asociada es generalmente una imagen visual del hardware, es decir los dispositivos mecánicos y electrónicos que componen el aspecto físico del robot, quizás influenciados por series de televisión o películas como la serie de la Guerra de las galaxias; sin embargo cuando más se profundiza en las tareas de un robot más se hace patente que lo que forma un robot es un enlace del software con el hardware.

La palabra robot proviene del vocablo de origen checo *robota* la cual significa trabajo y la definición más apegada a lo que es un robot es la que proporciona "Robot Institute of America " que lo define como *"un manipulador reprogramable multifuncional diseñado para mover materiales, piezas o dispositivos especializados a través de movimientos programados variables para la realización de una diversidad de tareas"*. En suma, un robot es un manipulador reprogramable de uso general con sensores externos que pueden efectuar diferentes tareas. Con esta definición un robot debería de poseer "inteligencia" la cual es otorgada por el software, y es esta "inteligencia" lo que distingue a un robot a otras formas de automatización. Son dos los tipos de robot que básicamente se distinguen los a) De tipo fijo b) Autónomos. El primero se caracteriza por robots industriales de ensamblaje como los utilizados para el armado de automóviles, este tipo de robots debe de operar sólo en un entorno altamente controlado para el cual fueron explícitamente diseñados.

En tanto el robot autónomo se distingue por interactuar con el mundo real; independientemente de algunos modelos muy simples no existen verdaderos robots autónomos, quizás la razón principal de este hecho es que se requiere de la resolución de varios de los más difíciles problemas de la inteligencia artificial como lo son sus esquemas de movimiento; tal vez esta razón es también lo que

hace que un robot autónomo sea tan atractivo para la gente dedicada a la inteligencia artificial.

La comunidad científica dedicada a la robótica ha provisto a sus modelos de capacidades sencillas de actuar y percibir el medio ambiente de una manera predeterminada, a pesar de que resulte obvio que los sistemas biológicos existentes tienen la habilidad de conducirse a través de esas tareas de una manera razonable para alcanzar sus metas. Sin embargo éste enfoque ha sido dejado de lado por la comunidad científica.

Mucho se ha ganado a través del estudio de la neurociencia y la psicología, Pero emerge el cuestionamiento sobre ¿cómo deberíamos usar los densos conocimientos que se tienen acerca del comportamiento de sencillos esquemas biológicos para producir robots verdaderamente "inteligentes"?

La inteligencia artificial frecuentemente evoca el uso de la abstracción como el medio para representar y resolver problemas, pero la abstracción limita el conocimiento que se dispone de los sistemas biológicos. Así que la solución se ve desde un punto de vista diferente y es convertido a una metodología que es más conducente para los sistemas de cómputo, en contraparte se tienen los modelos y herramientas derivados de los sistemas biológicos los cuales pueden ayudar a dotar de inteligencia a un robot.

Hasta el momento el modelo de la robótica han sido los organismos vivos, pero existen dos razones por las cuales éste tipo de modelos no han podido ser emulados eficientemente: primero, la diferencia entre los dos sistemas implicados (computacional y biológico) y segundo el desconocimiento que aun existe sobre la mayoría de los fenómenos que subyacen en las tareas que los organismos vivos efectúan y son los que tratamos de emular.

Hay otro problema que junto con los anteriores limitan el trabajo realizado por la robótica; en el uso que se le da al conocimiento obtenido del estudio de los modelos naturales a emular y más específicamente en el campo de las redes neurales aplicadas a la robótica, está el problema de que muchas veces se confunde el fin (funcional), con los medios (estructural), en el uso que se le da al conocimiento obtenido del estudio.

Valdría la pena reflexionar y hacer una reconsideración acerca de las metas de la simulación orgánica. Desde la antigüedad se experimentó en carne propia el hecho de que la mejor manera de lograr acciones no siempre era la emulación. Bastaría con mencionar al Ícaro cuando quería volar y recordar que la mejor manera de hacerlo no es la imitación fiel a la manera de como lo hacen las aves, si no la creación de mecanismos que basados en esas formas naturales, por lo que tal vez el logro de conductas inteligentes en máquinas quizás no sea logrando una imitación fiel del sistema nervioso humano, sino que el conocimiento profundo de éste permitirá el desarrollo de medios "basados en" y no en medios "a semejanza de " que nos permitan llegar a la realización de estas metas [Ponce95].

Desde Kant hasta nuestros días se ha tomado como base los postulados de teorías y corrientes filosóficas en el desarrollo de modelos más eficientes. La teoría de los esquemas con sus patrones de acción, manejo y control efectivo de la información, es la que ha tenido mayor influencia y nos lleva a la explicación de acciones complejas mediante la aportación de conocimientos de base. Esto permitió que en la década de los ochenta se desarrollaran dentro de la robótica, esquemas de movimiento que nos permiten planificar adecuadamente la acción de los mecanismos y control de los comportamientos y reacciones. Esto se ha hecho posible gracias a la descomposición de tareas, la representación y percepción de la globalidad del entorno y a la adaptación dinámica del medio, hecho que ha llevado al desarrollo de mecanismos que dotan de características similares a las de los organismos vivos, en cuanto a su forma de desenvolvimiento, aprendizaje y reacciones en un ambiente específico.

1.2 Desarrollo Histórico

La palabra robot se introdujo en la lengua inglesa en 1921 con el drama satírico Rossum Universal Robots de Karel Capek . En éste drama los robots son máquinas que se asemejan a los seres humanos, pero que trabajan sin descanso.

La obra presenta a los robots como ayudas para substituir a los operarios humanos, pero posteriormente los robots se vuelven contra sus creadores, aniquilando a toda la raza humana. Dicha obra es el antecedente de algunas de las creencias mantenidas popularmente acerca de los mismos en nuestro tiempo, incluyendo la perfección como máquinas humanoides dotadas con inteligencia y personalidades individuales.

Esta imagen se reforzó con la película "Metropolis", de 1926 con el robot andador eléctrico y su perro << Sparko >> y más recientemente con los robots C3PO y A2D2 protagonistas de la serie de películas de la "Guerra de las Galaxias". Ciertamente los robots industriales modernos parecen primitivos cuando se comparan con las expectativas creadas por los medios de comunicación durante las pasadas décadas.

Los primeros trabajos que condujeron a los robots de hoy en día se remontan al periodo inmediato posterior al fin de la Segunda Guerra Mundial . Durante los años finales de la década de los cuarenta, comenzaron programas de investigación en Oak Ridge y Argonne National Laboratories para desarrollar manipuladores mecánicos controlados en forma remota para el manejo de materiales radioactivos . Estos sistemas eran del tipo <maestro - esclavo>, diseñados para reproducir fielmente los movimientos de mano y brazos realizados por el operario humano. El manipulador maestro era guiado por el usuario a través de una secuencia de movimientos, mientras que el manipulador esclavo duplicaba a la unidad maestra tan fielmente como le era posible. Posteriormente se le añadió la retroalimentación de la fuerza acoplado mecánicamente el movimiento de las unidades maestro y esclavo de forma de que

el operador podía sentir las fuerzas que se desarrollaban entre el manipulador esclavo y su entorno .

A mediados de los años cincuenta, el acoplo mecánico se sustituyó por sistemas eléctricos e hidráulicos en manipuladores tales como el Handyman de General Electric y el Minotaur I construido por General Mills.

El trabajo sobre manipuladores maestro-esclavo fue seguido rápidamente por sistemas más sofisticados capaces de operaciones repetitivas autónomas. A mediados de los años cincuenta George C. Devol desarrolló un dispositivo que él llamó "dispositivo de transferencia programada articulada", un manipulador cuya operación podía ser programada y por lo tanto cambiada y que podía seguir una secuencia de pasos de movimientos determinados por las instrucciones en el programa. Posteriores desarrollos de éste concepto por Devol y Joseph F. Engelberger condujeron al primer robot industrial, introducido por Unimation, Inc en 1959. La clave de éste dispositivo era el uso de una computadora en conjunción con un manipulador para producir una máquina que podía ser "enseñada" para realizar una variedad de tareas de forma automática. Al contrario de las máquinas de automatización de uso dedicado, estos robots se podían reprogramar y cambiar de herramienta a un costo relativamente bajo para efectuar otros trabajos cuando cambiaban las especificaciones de fabricación.

Aunque los robots programados ofrecían una herramienta de fabricación nueva y potente, se hizo patente en los años sesenta que la flexibilidad de estas máquinas podía mejorar significativamente mediante el empleo de una retroalimentación sensorial. Al comienzo de esta década H. A. Earns (1962) publicó el desarrollo de una mano mecánica controlada por computadora con sensores táctiles. Éste dispositivo llamado MH1 podía sentir bloques y usar esta información para controlar su mano de manera que apilaba los bloques sin la ayuda de un operario. Éste trabajo es uno de los primeros ejemplos de un robot capaz de conducta adaptiva en un entorno razonablemente no estructurado.

Dicho programa de investigación posteriormente evolucionó como parte del proyecto MAC, y se le añadió una cámara de televisión para comenzar la investigación sobre la percepción de la máquina. Durante el mismo periodo Tomovic y Boni (1962) desarrollaron una mano prototipo provista con un sensor

de presión que detectaba el objeto y proporcionaba una señal de retroalimentación con entrada a un motor para iniciar uno de los dos modelos de aprehensión, una vez que la mano estaba en contacto con el objeto, se enviaba a una computadora información proporcional a su tamaño y peso mediante estos elementos sensibles a la presión. En 1963 la American Machine and Foundry Company (AMF) introdujo el robot comercial VERSATRAN, comenzando en ese mismo año desarrollos de diversos diseños de brazos manipuladores tales como el brazo Roehampton y el Edinburgh.

A finales de los sesenta McCarthy (1968) y sus colegas del Stanford Artificial Intelligence Laboratory publicaron el desarrollo de una computadora con manos, ojos y oídos (manipuladores, cámaras de TV y micrófonos), demostraron un sistema que reconocía mensajes hablados, "veía" bloques distribuidos sobre una mesa, y los manipulaba de acuerdo a ciertas instrucciones. Durante este periodo, Pieper (1968) estudió el problema cinemático de un manipulador controlado por computadora, mientras que Kahn y Roth (1971) analizaban la dinámica y el control de un brazo restringido utilizando un control de tiempo casi mínimo (bang-bang¹).

En otros países, particularmente Japón se comenzó a ver el potencial de los robots. Ya en 1968 la compañía Kawasaki Heavy Industries negoció una licencia con Unimation para producir sus robots. Uno de los desarrollos poco usuales en robots sucedió en 1969, cuando desarrolló un camión experimental la General Electric para la Armada Americana, en el mismo año se desarrolló el brazo Boston y al año siguiente el brazo Stanford, que estaba equipado con una cámara y controlado por una computadora. Algunos de los trabajos más serios de robótica comenzaron cuando esos brazos se utilizaron como robots manipuladores.

Un experimento con el brazo Stanford consistía en apilar automáticamente bloques de acuerdo con diversas estrategias. Esto era un trabajo muy sofisticado

¹Los robots no controlados por servos pueden proporcionar una secuencia de movimientos punto a punto, para cada movimiento, los miembros del manipulador se inclinan al máximo hasta alcanzar el límite de desplazamiento. Por ello se denominan robots de golpe a golpe (bang-bang, coger y colocar o robots de secuencia limitada)

para un robot automatizado en aquella época . En 1974 Cincinnati Milacron introdujo su primer robot industrial controlado por computadora, lo llamó "The tomorrow tool " (La herramienta del mañana) o T³ , el cual podía levantar más de 100 libras así como seguir a objetos móviles en una línea de montaje.

Durante los años sesenta se centró un gran esfuerzo de investigación sobre el uso de sensores externos para facilitar las operaciones manipulativas. En Stanford, Bolles y Paul (1973), utilizando retroalimentación tanto visual como de fuerza, demostraron que un brazo Stanford controlado por computadora conectado a una PDP-10, efectuaba el montaje de bombas de agua de automóvil. Hacia la misma época, Will y Grossman (1975) en IBM desarrollaron un manipulador controlado por computadora con sensores de contacto y fuerza para realizar montajes mecánicos en una máquina de escribir de veinte piezas. Inoue(1974), en el Artificial Intelligence Laboratory del MIT, trabajó sobre los aspectos de inteligencia artificial de la retroalimentación de fuerzas. Se utilizó una técnica de búsqueda de aterrizajes, propia de la navegación aérea, para realizar el posicionado inicial de una tarea de montaje precisa. En el Draper Laboratory, Nevins y colaboradores (1974) investigaron técnicas sensoriales basadas en el control coordinado de fuerza y posición. Éste trabajo desarrolló la instrumentación de un dispositivo "remote center compliance" o RCC (centro remoto de control coordinado de fuerza y posición).

1.3 Evolución de los robots autónomos

Los robots autónomos son considerados en éste momento como la última escala en la robótica. Por "autonomía" se entiende la habilidad para tomar decisiones independientes ante los cambios de situaciones, tal habilidad es posible si la inteligencia permite un cierto nivel de independencia y si la meta general de movimiento es formulada por un operador humano, pero las decisiones de un particular movimiento deben de ser tomadas por el robot, sin la intervención del elemento humano.

Esto nos lleva al concepto de grados de autonomía, ya que los robots generalmente son controlados por un humano, mientras que algunas de las operaciones pueden ser planeadas, controladas y ejecutadas sin la participación del ser humano.

Quizás el primer robot autónomo (RA) es el conocido como ELSIE de W. Grey Walter, cientos de juguetes están basados en éste RA, utilizando el principio de "observa la luz y dirígete a ella", la inimaginable e impredecible trayectoria de movimientos del circuito nos sugiere la posibilidad de total libertad e inteligencia independiente. El circuito simplemente reacciona a las condiciones a su alrededor.

Los esfuerzos más serios para el desarrollo de los RA comenzaron poco después de finalizada la II guerra mundial, pero todos los intentos realizados quedaron en eso, sólo intentos. En los años sesenta Stanford Research Institute desarrolló a Shakey el cual poseía una cámara de TV y detectores táctiles, poseía un control interno el cual manejaba las secuencias de movimiento, mientras que la visión y la planeación de tareas se manejaban a través de una computadora la cual se conectaba al robot mediante un transmisor de radio.

Shakey demostró que numerosos principios de la Inteligencia Artificial relacionados con el área de la robótica podían ser realizados por simples circuitos. En su tiempo fue considerado el trabajo más sofisticado en lo que aplicar inteligencia artificial a robots se refiere, el sensor principal era la cámara la cual cubría un amplio campo de visión, con éste sentido de "visión" y con el soporte de su software, fue capaz de navegar, explorar y aprender del mundo; con éste robot se descubrió la importancia de la "visión" de la máquina siendo éste paso fundamental para el desarrollo futuro

La representación del mundo de Shakey consistía de un juego de fórmulas bien formadas del cálculo de predicados, y su nivel de *inteligencia* se caracterizó por su habilidad para determinar la secuencia de operaciones requeridas.

A principios de los años 70 la NASA en cooperación con JPL (Jet Propulsion Laboratory) inició un programa orientado hacia la reducción del apoyo en tierra a las máquinas que realizaban los procesos de lanzamiento y

operaciones orbitales además de las operaciones de sistemas controlados por el hombre en medios ambientes hostiles, el robot resultado de éste programa (Marciano errante) no era del todo autónomo, pero sí era capaz de analizar una escena simple y planear un camino hasta alcanzar su meta en un ambiente simple (en un laboratorio con un limitado número de obstáculos y apropiada iluminación), su modelo del mundo fue dividido en sectores de tamaño conveniente, en donde cada sector representaba un área y las cuales podían ser transitables o no transitables, éste modelo de robot fue empleado también por el departamento de bioingeniería de la Academia de Ciencias de la URSS.

De 1973 a 1981 Hans Moravec realizó un trabajo sobresaliente en el laboratorio de Inteligencia Artificial de la Universidad de Stanford ya que desarrolló un robot que era manejado a control remoto el cual poseía un sistema de cámaras de TV lo que le proporcionaba una visión de 3 dimensiones, que le ayudaba a localizar objetos y podía deducir sus propios movimientos hasta llegar a la meta.

Los principios usados por el robot de Moravec fueron retomados para desarrollarlos y modernizarlos por el Instituto de Robótica de Carnegie-Mellon University. El nuevo prototipo fue un robot cilíndrico el cual poseía 3 ruedas ensambladas de forma independiente lo que le permitió tres grados de libertad de movimiento, fue equipado con una cámara de TV, de rayos infrarrojos y un sonar para detectar proximidad, además de algunos interruptores de contacto. Su concepto de control estaba basado en una estructura jerárquica de tres niveles: Planificador, Ejecutor del plan y Activadores y sensores, además de tener un nivel más en su estructura para el manejo de las comunicaciones. Estas se realizaban por medio de mensajes que se colocaban en un *pizarrón*, y los procesos podían en un momento dado cambiar sus prioridades dependiendo de la relevancia de los mensajes en el mismo.

En la misma época (1979) en Francia se desarrollaron otros conceptos. El primero de ellos utilizaba una computadora IBM 370/168 para su comunicación con el mundo externo, mientras el otro prototipo fue impulsado por el Instituto Nacional de Ciencias Aplicadas de Francia, se trataba de un robot semi-autónomo manipulado a control remoto, y el cual se basó en tres estados principales: Velocidad, Rotación y Traslación, su software contenía dos algoritmos

básicos : Control de velocidad y Control de posición, estos dos prototipos franceses empleaban la estructura mecánica utilizada por Shakey.

Un robot relativamente autónomo con visión ultrasónica y capacidad para sortear obstáculos fue desarrollado por el laboratorio de ingeniería mecánica de MIT el cual estuvo destinado a servir como perro lazarillo a personas invidentes, sus creadores le llamaron Meldog, el cual poseía un control muy preciso de dirección y de velocidad, sus comandos de control eran LEFT, RIGHT, STRAIGHT y STOP. Estos eran transmitidos mediante interruptores de control (semejantes a los actuales carros de juguete manipulados por control remoto); mientras que la comunicación del robot hacia el humano se realizaba mediante choques eléctricos en la mano de la persona que manejaba el robot.

Robart-I es probablemente el primer robot en ser totalmente autónomo además de exhibir un alto nivel de sofisticación, fue construido por la Escuela de Postgraduados de la Naval de los EE.UU. y servía básicamente en tareas de vigilancia y patrullaje que hacía de manera totalmente aleatoria además de tener la capacidad de detectar fuego, humo, inundaciones, gases tóxicos, etc. situaciones para las cuales tomaba la acción preventiva necesaria. Este robot no tenía la capacidad de planear sus actividades, ya que no estaba orientado a las metas.

Su modo de movimiento era totalmente aleatorio, y que escogía de un juego de dieciséis rutinas previamente programadas, incluso en algunas de ellas el robot avanzaba hasta algún punto, y a partir de éste podía detenerse y cambiar de dirección o simplemente continuar con su camino, usualmente su movimiento era hacia adelante a menos que localizara algún objeto extraño, en cuyo caso tomaba las acciones pertinentes.

Robart-I fue seguido por Robart-II el cual fue desarrollado por el Departamento de Ingeniería Mecánica de la Universidad de Pennsylvania en Moore. Fue totalmente autónomo y utilizaba tres microprocesadores 6808 interconectados entre si. El primer procesador controlaba las ruedas y su mecanismo de control, el segundo microprocesador controlaba los sistemas sensoriales del robot, mientras el tercer microprocesador manejaba la planificación además de llevar el control de las intercomunicaciones entre los

microprocesadores. Este sistema multiprocesos permitía a cada uno de los procesadores correr hasta ocho tareas diferentes.

La Universidad Nacional de Australia construyó un robot para utilizarlo como herramienta de desarrollo para la visión de computadoras. Este robot estaba provisto de dos brazos manipuladores, una base móvil además de sensores ultrasónicos. El control se llevaba a cabo mediante una computadora NOVA 2/10, además de tener dos microprocesadores Z80 conectados como esclavos a la computadora los cuales proveían de imágenes en tiempo real, además de permitir el reconocimiento de patrones de lenguaje. Contaba además con una cámara de TV y de sensores que rotaban junto con ésta y le proporcionaba suficiente información acerca del medio ambiente.

Bell and Howell recientemente comenzaron la fabricación de robots que sólo poseen elementos rudimentarios de autonomía y su *inteligencia* tan sólo se resume a la obediencia de un programa. Uno más de los prototipos de Bell & Howell. Dichos robots tienen interconstruido un sistema que puede seguir un camino fluorescente pintado en piso, éste sistema así como el de Weibe System son dedicados totalmente al manejo de materiales sin que estén controlados por el hombre. Estos sistemas pueden ejecutar totalmente una rutina preprogramada o bien en cada estación un operador humano puede enviar mensajes especificando la zona a donde deberán dirigirse.

Existe un buen número de soluciones tecnológicas que constituyen todo un potencial que sirve como base al desarrollo de los robots autónomos. Por ejemplo, un robot desactivador de bombas fue recientemente desarrollado por la policía de la ciudad de Nueva York, al igual que la armada británica que utiliza un robot semejante con el propósito de detectar bombas.

La compañía japonesa Tsugawa Automobile, haciendo énfasis en la utilización de principios de la inteligencia artificial, creó un control de tiempo real para la conducción de un automóvil, los resultados obtenidos con el prototipo son un excelente ejemplo de la aplicación de reconocimiento de patrones.

Hoy en día vemos a la robótica como un campo de trabajo mucho más amplio que el que se tenía hace unos pocos años, tratando con investigación y

desarrolló en una serie de áreas interdisciplinarias, que incluyen cinemática, dinámica, planificación de sistemas, control, sensores, lenguajes de programación e inteligencia de máquina.

Los robots autónomos, por principio de cuentas son mucho más complejos que los robots del tipo industrial, por el simple hecho de que debe ser más "listo", ya que si un robot autónomo pretende operar con éxito en el mundo real, necesitará de varias habilidades que el robot industrial no requiere; por ejemplo, necesitará de sensores que le permitan "oír y ver", debe de comprender el lenguaje natural y lo que significa, un robot con estas capacidades no es una proeza pequeña, además de que el robot debe de ser capaz de resolver problemas, lo cual es tal vez la tarea más complicada de todas.

Esto es necesario para que el robot pueda adaptarse a varias situaciones porque claramente no se puede programar a un robot por adelantado para cada posible suceso. El mayor obstáculo que impide la creación de un robot autónomo es el hecho de que los programadores no hayan desarrollado todavía las técnicas de software necesarias. Queda mucho por hacer en esta área cuya historia apenas ha comenzado.

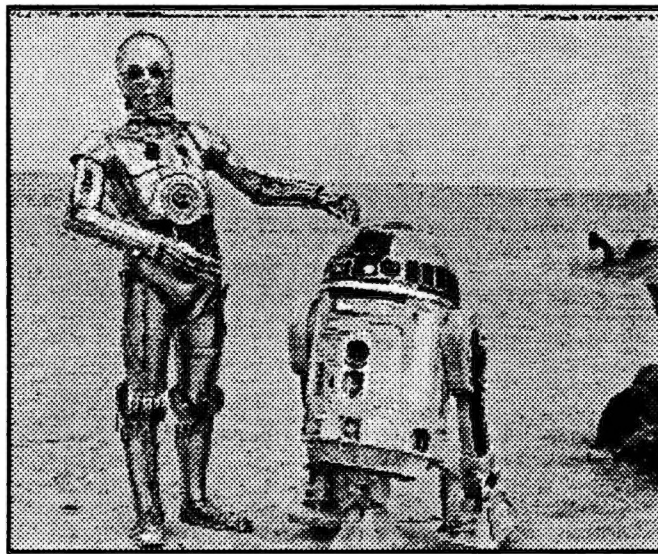


Fig 1.1 Una escena de la trilogía filmica "Guerra de las Galaxias"

BIBLIOTECA



2 INTRODUCCIÓN



2.1 Sistemás Clásicos de Planeación

La mayor parte de los desarrollos de la AI en cuanto a planeación se refiere caen dentro del marco de referencia de la planeación clásica. Un planeador de este tipo produce un plan que cuando se ejecuta nos permite transformar el estado inicial del mundo a un estado final consistente con la(s) meta(s) especificada(s).

Para intentar hacer un problema de planeación tratable, asumimos que el mundo puede representarse tomando < Una foto instantánea > en un determinado momento. Los sistemás que hacen este tipo de consideración son llamados "**sistemás clásicos de planeación**", quedando dentro de está clasificación la mayor parte de los sistemás que se conocen hoy en día. La principal consideración que hace este tipo de planeación es el moverse de un estado o mundo inicial (foto instantánea) a diferentes estados del mundo (estados meta). El término < razonamiento basado en estados > es algunas veces usado para expresar la misma idea.

La descripción de un sistema clásico están definidas por:

- Una conceptualización de un estado inicial.

- Un juego de acciones u operadores los cuales podrían dar lugar a un estado del mundo.

- Una descripción del estado meta

Las salidas de un planeador clásico consiste de una descripción de la secuencia de operaciones, las cuales al aplicarles al estado del mundo actual nos guiarán a un estado del mundo como el descrito en la meta. Cada operador en la secuencia crea un nuevo estado del mundo, sobre el cual el siguiente operador en la secuencia actuará, hasta que la meta sea alcanzada.

El desarrollo del paradigma de la planeación clásica siempre ha sido llevado a ambientes sin incertidumbre, planeadores tales como STRIPS [Nilsson, 1980] y NOAH [Sacerdoti, 1975] dependen del conocimiento previo del mundo (proveen esencialmente al planeador de toda la información del mundo y en el caso particular de STRIPS permiten conocer al plan de todos los efectos de los operadores a priori) para desarrollar un plan estático para una ejecución posterior.

Asumiendo lo anterior, el planeador tiene la habilidad para proyectar su perfecta descripción del mundo conforme a sus operadores en la búsqueda de un estado solución. Pero la tendencia a divergir con la presencia de incertidumbre en el dominio es exacerbada. Cualquier sistema que planea tareas para un mundo real debe de ser capaz de censar el medio ambiente, efectuar cambios en el ambiente y modelar su medio. El marco de la planeación clásica asume que posee perfectas versiones de estas capacidades, pero en realidad un sistema de planeación más realista debe de introducir incertidumbres en por lo menos los siguientes aspectos

- *Error de sensores* el cual ocurre cuando el valor retornado no es un reflejo apropiado del mundo real
- *Error de control* el ocurre cuando un efector actuando en el mundo real no actúa como se espera.
- *Error de modelo* ocurre cuando el modelo del mundo del sistema interno diverge del mundo real actual.

Es claro que los planeadores clásicos no están preparados para tratar con este tipo de incertidumbre, en todo caso orientaran la solución del problema hacia casos promedios inadecuados. Muchos desarrollos para trabajar con micromundos donde es factible el poder buscar soluciones en el espacio entero sin tener que lamentarse por la eficiencia, aunque cuando se requiere trabajar con medio ambientes de tamaño real es necesario el uso de heurísticas para restringir el espacio de búsqueda.

Un buen ejemplo de la planeación clásica adaptado para usarse en el mundo real el planeador SIPE [Wilkins, 91] ya que incorpora dominios

dependientes e independientes y heurísticas para mantener una búsqueda de soluciones dentro de un marco más real.

Por lo que a la planeación de trayectorias se refiere la mayor parte de los desarrollos se han concentrado en la planeación fuera de línea en ambientes estáticos, un plan de movimientos es computado como un camino geométrico, siendo un ejemplo de este paradigma el concepto de espacio de configuración o *C-Space* de el robot. Varios algoritmos planeadores de caminos basados en este concepto cuya principal técnica consiste en explorar una malla uniforme en el espacio de configuración y utilizar por ejemplo el algoritmo de el mejor primero para encontrar la meta o metas. [Latombe1994], [Papadimitriu, 1985] Aunque han sido desarrollados planeadores eficientes bajo este concepto, estos siguen utilizando demasiado tiempo de ejecución (computacionalmente hablando) para ser usados en línea. [Latombe2].

Esta consideración de que un plan puede basarse sobre una "instantánea" del mundo es sólo válida si el mundo no cambia durante el proceso de planeación, este tipo de planeación es útil si se trabaja con procesos continuos, en los cuales se tienen descartadas los cambios bruscos. Pero si se requiere que un sistema pueda reaccionar lo suficientemente rápido a los cambios en el mundo, como para poder replantear y elaborar un nuevo plan en tiempo real, el sistema está descrito por lo que se conoce como "**reactivo**", un sistema de planeación reactivo debe de estar en continua alerta a los cambios en el estado del mundo , cosa que no ocurre en los sistemas clásicos.

Bel [Bel, 1990] categorizó los diferentes tipos de planeación en base a la escala de tiempo que se aplique (figura 2.1). Los sistemas de planeación a largo plazo conciernen principalmente a estrategias y a la inversión; el mediano plazo involucra a decisiones como la planeación de producción con un horizonte de semanas o quizás meses; el corto plazo involucra a la planeación diaria de actividades y recursos, mientras que el control de las decisiones tiene el horizonte de tiempo más corto y debe de ser hecho en tiempo real.

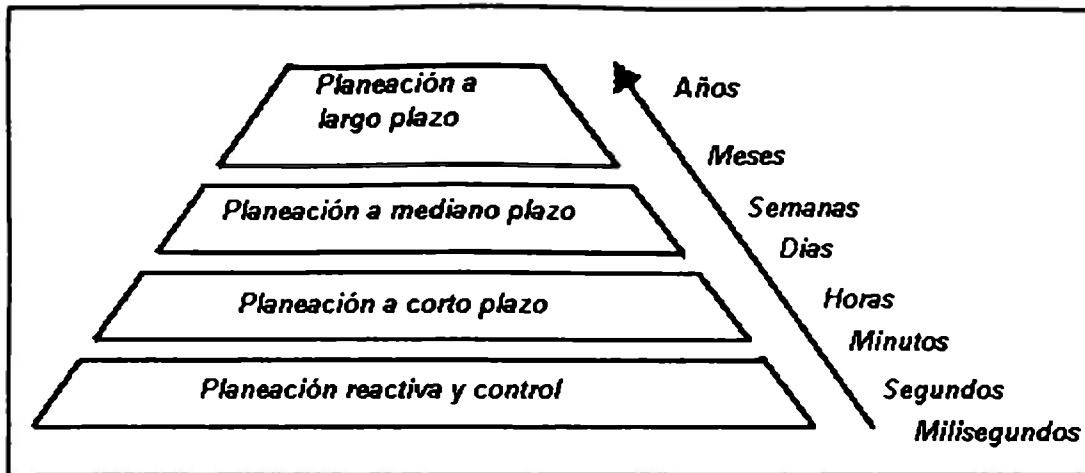


Fig. 2.1 Clasificación en base tiempo del tipo de decisiones

Un problema importante en la construcción de un sistema de planeación es decidir cuáles aspectos de estado del mundo serán alterados por determinada operación (y qué tanto se verá alterado) y cuales no lo serán, inclusive algunas operaciones podrían tener un efecto diferente dependiendo del contexto de algún aspecto del estado el mundo, lo cual es conocido como *el problema de marco*.

2.2 Sistemás de Control Reactivo

Los métodos de construcción de sistemás de robots *inteligentes* utilizando técnicas basadas en la IA han tenido gran dificultad con las aplicaciones del mundo real, tales sistemás requieren de un tremendo poder computacional para tomar decisiones de control, por lo que se ha tenido que sacrificar la habilidad de tomar decisiones en tiempo real, además que deben de asumir muchas consideraciones acerca de la información que está recibiendo de los sensores. Sin embargo cuando estas consideraciones resultan ser inválidas (como ocurre mayormente cuando se interactúa en el mundo real), en la mayoría de las ocasiones es necesario abortar la misión ya que es prácticamente imposible completarla. Basado en el formato de la teoría de los esquemás la teoría del control reactivo ha emergido como un método alternativo a las más tradicionales técnicas de diseño de sistemás de control robótico, mismo que se basa fundamentalmente en :

- La descomposición de las tareas en un una colección de primitivas de bajo nivel
- Se evitan las representaciones globales, se utiliza un acoplamiento directo entre la percepción y la acción motora, por lo que cada módulo extrae únicamente la información que necesita para poder tomar una decisión razonable.
- Realiza un rápido muestreo sensorial del mundo, lo cual permite detectar los cambios dinámicos del medio ambiente.

La navegación de robots autónomos ha servido como dominio primario para probar la metodología basada en los esquemás. Para la construcción de un sistema de robot basado en esquemás se utiliza el procedimiento descrito a continuación.

- Caracterizar el dominio del problema desarrollando y comprendiendo la tarea del robot y las restricciones impuestas por su medio ambiente.
- Enumerar las conductas de movimiento que acompañan a la tarea; descomponer estas conductas en primitivas estímulos-respuestas.
- Formular las conductas de movimiento en los esquemas motores el cual se enclava en un algoritmo específico que responde a una condición de ambiente.
- Probar los esquemas de movimiento en simulacros.
- Desarrollar algoritmos perceptuales que sean capaces de entregar la información del medioambiente directamente a los esquemas de movimiento.

La experimentación y la simulación deben de ser iterativamente aplicadas hasta la obtención de modelos de comportamiento exitosos, siendo la mayor ventaja de la metodología que los esquemas perceptuales y motores son reusables con lo que se tiene un sistema incremental para desarrollo y prueba. En medio ambientes complejos los esquemas pueden ser reconfigurados usando planeadores tradicionales de la IA.

Durante años muchos desarrollos han tratado de conjuntar el aprendizaje biológico con las técnicas de la Inteligencia Artificial [Balkenius 1995], pero en general no se han obtenido resultados satisfactorios, principalmente porque las aplicaciones de la inteligencia artificial se reducen a una simple búsqueda, y se ha perdido de vista que la escuela jerárquica está involucrada con la formulación de planes, mientras que la escuela reactiva debe de encargarse de la realización o ejecución del plan y desembocar en una replaneación dinámica, por lo que debe de buscarse no la superioridad de una escuela sobre la otra sino una integración sintética que nos permita usar el enfoque más apropiado. Los animales nos muestran la evidencia psicológica y neurocientífica para la coexistencia de dos sistemas de planeación-ejecución compatibles: uno que

concierno a la formulación del plan y el otro a la ejecución reflexiva del plan [Arkin 1989].

Estudios psicológicos de [Norman 1986] han mostrado la existencia de dos diferentes modos de comportamiento; el voluntario (controlado) y el automático y ha enumerado las características de las tareas que requieren de control voluntario:

- Planeación (tomar decisiones)
- Problemás
- Nuevas o pobres acciones de aprendizaje
- Peligros o acciones difíciles
- Vencer hábitos o tentaciones

Mientras que otras acciones motoras son típicamente automáticas (reactivas) y pueden ocurrir sin atención (por ejemplo el manejar un automóvil) La mayoría de los comportamientos animales pueden ser explicados en términos de una asociación de estímulo respuesta (E-R). Basados en está creencia numerosos experimentos han sido llevados a cabo con está idea en la mente, el establecer una regla general para la formación de Estimulos-Respuesta , continuando está línea de pensamiento se puede razonablemente entender la situación del aprendizaje, por lo que se puede generalizar que todas las conductas son explicadas en términos de Estímulo-Respuesta. Las reglas utilizadas en los sistemás basados en reglas son muy similares a las asociaciones E-R, cuando una regla es usada para generar una precondition para otra regla. Desde la perspectiva computacional los dos enfoques son en su mayoría idénticos.

Así, un sistema planificador de un robot puede construirse en base a estos principios, donde un comportamiento es un subsistema que responde a algún patrón de acción específica del robot, tal y como lo sugiere [Arkin, 1989] en el modelo integrado de planeación /ejecución (AuRA) el cual es un marco de trabajo donde se integran los mecanismos de planeación y ejecución jerárquicos y reactivos y donde son conducidos experimentos de navegación de robots.

En el contexto de este sistema [Arkin, 1988] se han desarrollado técnicas para la navegación de caminos en la presencia de modelos a priori del mundo, con manejo de incertidumbre, navegación reactiva/reflexiva, replaneación dinámica en medioambientes difíciles. Este paradigma explota algunas formas de la representación del conocimiento: mapas del mundo a priori y modelos de señales, colecciones de conductas motoras y estrategias perceptuales (esquemás) los cuales son seleccionados, parametrizándose instanciados de manera consistente con el conocimiento disponible.

Mucho de este trabajo ha sido y continuará siendo influenciado por estudios psicológicos y neurocientíficos [Arkin, 1989B], aunque se requiere de una integración completa del planeador jerárquico, para una adecuada selección de las conductas y esquemás. Fuertes evidencias existen del éxito de esta integración ya que ambos son usados por la navegación de los animales. La ejecución del plan es altamente apropiado para el control reactivo, mientras que la formulación del plan es apropiado para sistemás jerárquicos los cuales mantienen la habilidad para razonar sobre varios niveles de la representación del mundo.

2.3 Enfoque y contribuciones de este desarrollo

De la revisión anterior podemos concluir que los planeadores jerárquicos tienen una alta fiabilidad en modelos del mundo, se pueden integrar conocimiento de éste, y planeación de navegación en el contexto que puede ser fácilmente manejado con esta técnica. Estas cualidades son cruciales para la formulación de un plan, mientras que los planeadores reactivos permiten el desarrollo modular e incremental además de estar fuertemente acoplados con el uso de datos sensoriales, características también cruciales para la ejecución de un plan.

Los ambientes reales donde actúa un robot autónomo están caracterizados por la incertidumbre, incertidumbre que proviene de muchos factores. Como la falta de conocimiento completos acerca de los efectos de las acciones, cambios dinámicos podrían ocurrir debido a la acción de otros agentes ya sea naturales o artificiales, finalmente los sensores podrían no ser confiables, resultando una diferencia entre el estado real del mundo y el que se está percibiendo.

La planeación con incertidumbre se puede beneficiar por dos características, la primera es la habilidad para usar la experiencia pasada para escoger la más adecuada secuencia de acciones, la segunda es la habilidad para reaccionar ante cambios no esperados y que ocurren en situaciones inciertas.

Por ejemplo, consideremos el siguiente escenario: Un robot que trata de llegar a una meta por su experiencia pasada sabe que es más fácil alcanzar ésta si el camino tiene una pendiente, que si fuera plano, aunque en este momento solo dispone de un camino plano, por lo tanto se genera un plan para utilizar esta ruta. Sin embargo en un momento dado detecta un camino con pendiente, por lo que abandona el camino plano y toma este. Esta conducta aunque parece sencilla, no puede ser alcanzada por la mayoría de los planeadores, aunque estos estén diseñados para manejar incertidumbre, La mayoría de los planeadores clásicos asumen que los efectos de una acción son conocidos con certeza. Con este conocimiento éstos generan una serie de acciones con las cuales pretenden alcanzar las metas propuestas, entonces la confianza en las acciones no es modelada, así como la confianza en el plan seleccionado no es considerada, por lo que se señala como error si el estado observado del mundo no compagina con el estado expectante despues de la ejecución de una acción. Aunque las situaciones no esperadas no son siempre sinónimo de ejecución con errores, en el ejemplo anterior la detección de un camino con pendiente fue inesperada, pero esto no implica que las acciones previas hayan resultado en error, aunque es totalmente correcto hacer cambios en el plan.

Algunos planeadores han sido diseñados para considerar los efectos de acciones probabilísticas [Kushmeric, 1993], donde se compara la efectividad de los planes, pero estos no reaccionan a cambios inesperados en el medio ambiente . Planeadores reactivos[Mitchell, 1994], [Firby, 19889] fueron diseñados

para ejecutar un plan de acción como respuesta al estado actual del mundo, conviviendo la planeación y la ejecución, pero sin usar la información probabilístico con el propósito de polarizar la selección de las acciones que fueron más adecuadas en el pasado, aunque se hace un buen manejo de las situaciones no esperadas, no se utiliza la incertidumbre para escoger un plan de acción más adecuado.

La inhabilidad de los planeadores existentes para manejar situaciones que a menudo ocurren en el mundo real, motiva el desarrollo de esta tesis. El objetivo es el incorporar a un planeador reactivo la habilidad de manejar situaciones no esperadas mediante un planeador probabilístico que basado en la experiencia previa incorpore estas, de forma de probabilidades condicionales, las cuales representan la habilidad de la acción para alcanzar sus efectos, utilizando para ello arquitectura neural, con lo que se tiene un control reactivo, con la capacidad de poder aprender estrategias efectivas de control, para tener la información relevante de cada estado donde posiblemente el sistema podría encontrarse.

En base al objetivo planteado arriba, esta tesis hace contribuciones en sistemas de planeación automáticos en las áreas que a continuación se describen:

- **Mecanismo de planeación ampliado:** El dominio del conocimiento representado como una red jerárquica conteniendo acciones y proposiciones es utilizado para ampliar el mecanismo de activación, el cual es diseñado para incrementar las utilidades de las proposiciones o acciones que afectan directa o indirectamente a las metas, así, como proyectar la probabilidad de futuros estados a partir del estado presente (lo cual sólo puede ser conocido probabilísticamente). Este mecanismo permite al sistema seleccionar acciones para planeación.
- **Ampliación de la base de conocimientos por medio de observación y experiencia:** A partir de las especificaciones de conducta previamente establecidas, el sistema puede predecir probabilidades futuras las cuales se utilizan para resolver las tareas asignadas.

Para la realización de los objetivos anteriores se utiliza el siguiente marco de referencia: Un robot situado en un medioambiente con una variedad de obstáculos, como lo son sillas, mesas, muebles, etc. Trabajando con dos diferentes tareas que el robot debe de aprender.

- **Evitar la colisión con obstáculos**
- **Exploración del medio ambiente**

Antes de pasar a describir el trabajo mencionado, se hará una breve revisión del trabajo previo realizado en lo que respecta a la planeación bajo incertidumbre, desde los planeadores probabilísticos a los sistemas reactivos que se presenta en el próximo capítulo.

3 TRABAJOS PREVIOS: *Planeación con incertidumbre*



3.1 Antecedentes

Existen diversos trabajos que tratan con la planeación bajo dominios inciertos; en primera instancia están aquellos en los que la generación del plan se basa en una representación explícita de la incertidumbre y aquellos en los que la selección reactiva de acciones basadas en la situación presente, con un implícito desconocimiento de la incertidumbre. Con la representación explícita de la incertidumbre se han desarrollado la mayoría de los trabajos sobre el área, en esta sección se hará una revisión de algunos de ellos [Bagchi, 1994], [Latombe, 1994].

3.2 Generación de Planes Probabilísticos

BURIDAN [Kushmeric, 1993], es un ejemplo de planeador clásico que toma ventaja del conocimiento probabilístico, como es un planeador clásico solo está interesado en la generación del plan, más no en la ejecución, maneja la incertidumbre creando planes que tienen aceptables oportunidades de éxito, pero no reacciona a situaciones inesperadas causadas por la misma incertidumbre.

BURIDAN genera planes cuyas probabilidades de éxito excede a un umbral predefinido, la forma en que representa la incertidumbre es asignando probabilidades a proposiciones booleanas que son afectadas por la acción, por ejemplo la probabilidad de escoger un camino con pendiente podría ser 0.5 si la proposición caminos-sinuosos es falso y 0.95 si caminos-sinuosos es verdadero.

Al representar el dominio del conocimiento de esta manera, el algoritmo sintetiza una secuencia de acciones, que alcanzan la proposición meta con una probabilidad más alta que el umbral τ asignado por el usuario, la planeación comienza con un plan nulo, el proceso involucra dos pasos; el primero es el *aseguramiento del plan* donde la probabilidad con el cual se alcanza la meta es calculada si es más alta que τ , la planeación finaliza, si esta no es superada el segundo paso, *refinamiento del plan* se ejecuta y termina hasta que un plan es generado, aunque este es insatisfactorio en ambientes bajo ambientes inciertos los cuales pueden cambiar e cualquier momento y alterar la probabilidad de éxito del plan generado.

3.3 Selección de planes usando diagramas de influencia

Los diagramas de influencia han sido usados en el desarrollo de operaciones como medio para explicar la dependencia probabilística entre decisiones y variables aleatorias que representan información acerca del medio ambiente, se utiliza una formulación gráfica donde las decisiones y las variables aleatorias son representadas como nodos, las relaciones probabilísticas entre los nodos son representados por arcos dirigidos, un tercer tipo de nodo, llamado nodo valor representa la utilidad del nodo de la variable aleatoria o del nodo de decisión al cual es ligado.

La construcción del diagrama es un proceso fuera de línea, una vez construido, los nodos pueden ser removidos por un proceso transformación llamado reducción del valor preservado, hasta solo tener el valor de un nodo. En este punto las utilidades expectantes para las diferentes decisiones son

conocidas, y la decisión con el máximo valor de expectativa será la que se escoja

Básicamente, antes de correr el sistema se generan muchos planes como sean posibles en un tiempo determinado, entonces durante el tiempo de corrida se escoge el plan con el valor más alto de utilidad expectante basados en el estado presente del medio ambiente, el problema con este desarrollo es que si no se es capaz de tratar con la incertidumbre, los valores de utilidad expectante en el diagrama de influencia cambiaran y el plan existente ya no será valido, y el sistema tendrá que generar nuevos planes fuera de línea por lo que resulta caro en el incremento del tiempo de computo ya que no se disponen de planes precompilados.

3.4 Planeación mediante el proceso de decisión de Markov

El proceso de decisión de Markov provee una útil representación para la planeación en dominios inciertos [Papadimitriou, 1984], se utilizan métodos de programación iterativos, para computar el optimo, lo cual preescribe la mejor acción a tomar en cada estado del medio ambiente.

Tenemos un conjunto de estados S , siendo un estado una combinación de proposiciones que describen el medio ambiente del agente. En teoría, el número de estados, $|S|$ es exponencial al numero de proposiciones. Una acción toma un agente de un estado al otro con alguna probabilidad, por ejemplo, $P(s_1, a, s_2)$ donde se denota la probabilidad de que el estado s_2 sea alcanzado si la acción a es ejecutada en le estado s_1 ¹

¹el efecto de una acción depende solo del estado presente y es independiente de las acciones previas que causaron este estado.

Dependiendo de la meta, a ciertos estados se les asigna una compensación $R(s)$, por ejemplo si la meta es hacer a cierta proposición verdadera una compensación es asignada a todos los estados que se encuentran en ese caso. El valor de un estado denota que tan deseable es estar en ese estado si el planeador esta siguiendo una política π , la política es un factor que determina la secuencia de acciones a ser tomadas desde el estado actual hasta alcanzar el estado meta.

El valor del estado s cuando se ejecuta la política π esta dado por:

$$\pi(s) = R(s) + \gamma \sum_{s' \in S} P(s, \pi(s), s') V \pi(s') \quad 3.1$$

donde $\pi(s)$ es la acción a ser tomada desde el estado s , prescrito por la política π . γ es un factor de descuento entre 0 y 1, el cual controla la magnitud de la compensación deberá influenciar el valor del estado s .

La ecuación 3.1 hace posible el describir el algoritmo de la política de iteración para encontrar la optima política $\hat{\pi}$ tal que $V_{\hat{\pi}}(s) \geq V_{\pi}(s)$ para toda $s \in S$ y política π . Brevemente, el algoritmo resuelve las ecuaciones lineales $|S|$ dadas por la ecuación 3.1 para obtener $V_{\pi}(s)$ para toda $s \in S$ y una política π la cual es inicialmente seleccionada aleatoriamente, enseguida el algoritmo hace una escalada de colinas creando nuevas políticas para substituir acciones alternadas en la política actual con el objetivo de encontrar la política más optima, cuando la escalada de colinas falla en generar una mejor política, la política presente es retornada como una política optima.

3.5 Generación de reglas de control situado

En el inciso anterior se describió la generación de una política para ser usada durante la ejecución de un plan mediante el proceso de decisión de Markov, el presente apartado define la transición hacia la planeación reactiva, donde el objetivo de la planeación es el generar una tabla de búsquedas, o un conjunto de reglas que especifican la acción a tomar desde cualquier estado e incluso reaccionar asimismo.

[Drummond, 1989] describió el proceso de proyección sintética² para generar lo que ellos llaman reglas de control situado para los componentes reactivos de el agente en uso, la representación es similar al proceso de decisión de Markov excepto que aquí no existe la compensación asociada con las metas, y vez de restricciones de ambiente define una combinación lógica de proposiciones que deben de ser alcanzadas o mantenidas.

El algoritmo de proyección comienza desde el estado inicial y busca hacia adelante aplicando transiciones probabilísticas a otros estados llevado por la ejecución de acciones hipotéticas. Las ramificaciones de esta búsqueda hacia adelante es limitada por la eliminación de transiciones con probabilidad baja además de usar una heurística que estima el esfuerzo requerido para alcanzar la meta desde un estado dado. La búsqueda termina cuando un estado satisface las condiciones.

²La proyección sintética es diferente a la proyección analítica realizada con los diagramas de influencia para la evaluación de los planes existentes, porque la síntesis de el plan también es ejecutada.

3.6 Selección de acciones dinámicas por ampliación de la activación

Este paradigma describe el como alcanzar el rendimiento de las ganancias; combinando las ventajas de la planeación y la reacción usando dos módulos separados pero cooperativos.

Utiliza un mecanismo que puede hacer un cambio en el ambiente de planeación del reactivo al deliberativo, para lo cual se utiliza una red que representa el dominio del conocimiento con las acciones representadas como nodos, tres tipos de ligas dirigidas son posibles entre nodos:

- Ligas de *sucesores*: Conecta a un nodo cuyos efectos incluyen una proposición a un segundo nodo el cual tiene la misma proposición como precondition.
- Por cada liga sucesora hay una liga hacia su correspondiente *predecesor* que apunta del segundo nodo al primero.
- Un *conflicto* liga un nodo cuya precondition es amenazada por el nodo cuyos efectos afectan a este

Con el dominio del conocimiento representado de esta manera, un mecanismo de ampliación de la activación es usado para seleccionar las acciones relevantes. Los nodos son activados como ampliación potencial de las condiciones externas validas y por las condiciones de meta que deben de ser

satisfechas, cuando el potencial de un nodo excede el umbral, la acción es ejecutada si todas las precondiciones son satisfechas. Este es un proceso continuo: Los cambios en el medio ambiente o metas inmediatamente afecta a la ampliación del potencial de los nodos. Si el umbral para disparar un nodo es elevado el proceso de ampliación de la activación es forzado a continuar por más tiempo, teniendo como resultado una gran cantidad de tiempo computacional en búsquedas y deliberación, en cambio si el umbral es disminuido, los nodos se disparan más rápido y más seguido.

3.7 Sistemas puramente reactivos

En muchos dominios es posible realizar la planeación y deliberación juntos y tener un sistema reactivo cuyas fases se ejecuten al mismo tiempo, influenciados por la conducta animal propuesta como marco de trabajo [Brooks, 1991] el termino sistemas basados en la conducta, donde cada conducta puede ser observada como una acción la cual se ejecuta cuando un estímulo es presentado. Un mecanismo de adjudicación media entre las acciones cuando existen muchas respuestas para el mismo estímulo.

Algunos desarrollos [Brooks, 1987] sugieren que el camino a seguir en dominios con incertidumbre, es no planear el todo, basado en la forma en que los seres vivos actúan, combinando la experiencia previa y reaccionando a la presencia de estímulos.

La explicación más trivial [Balkenius, 1994] que los seres vivos tienen que aprender de la asociación estímulo - respuesta . En un laberinto cada sitio de este esta asociado con una respuesta correcta a dar en dicho sitio, el problema con este concepto es que las conductas generadas son inestables, ya que las

reacciones ejecutadas son orientadas hacia el estímulo, en vez de por el estímulo, por lo que una respuesta Ro ejecutada como resultado de observar un estímulo. Esto podría dar origen a diferentes movimientos en el espacio dependiendo de la posición inicial del animal, por lo que las asociaciones estímulo - respuesta son divergentes, por lo que si una secuencia de acciones E - R es ejecutada el error se ira acumulando hasta estar a la deriva.

4 *MODELO PROPUESTO*



4.1 Consideraciones generales

La habilidad de un sistema de planeación para reaccionar ante situaciones no esperadas es una de las cualidades que un buen planeador debe de tener, además es deseable que el planeador también tenga las siguientes cualidades:

- i) La capacidad para reaccionar cuando una acción falla al no tener un efecto esperado.
- ii) Poder reaccionar ante los cambios no esperados en el medio ambiente, lo cual causa un intercambio en los planes.
- iii) Responder a metas cambiadas repentinamente por el usuario, por lo que el mecanismo de activación es inmediatamente afectado por los cambios en la situación presente.

Un planeador es efectivo si es que puede manejar estas situaciones no esperadas.

Consideremos la situación donde una acción no alcanza el efecto deseado, el planeador ahora tiene la responsabilidad de escoger la siguiente acción, la cual podría ser la misma que fallo o una nueva, pero esto no significa de que se disponga de un plan alternativo, ya que solo se aplica el criterio de escoger la acción con la máxima utilidad expectante.

El medio ambiente es por naturaleza incierto y no es posible incorporar todos los detalles relevantes acerca de este en el dominio del conocimiento para resolver algún problema. En el presente capítulo las razones de la incertidumbre serán discutidas e incorporadas a un modelo usando probabilidades, y en base a este modelo, se propone un sistema de planeación que incorpore las relaciones

probabilísticas entre las condiciones sensadas que definen el estado del medio ambiente y las acciones propuestas.

Los planeadores clásicos asumen que se tiene una información completa acerca del estado del mundo y los efectos de sus acciones, pero estas consideraciones son inapropiadas para muchos dominios [Kushmerick, 1994]: por ejemplo; todas las mañanas que tomamos nuestro automóvil para dirigirnos a nuestros sitios de trabajo, tenemos trazado un plan acerca de la ruta que debemos seguir, para llegar a nuestro destino, es decir tenemos un bosquejo del plan a seguir, pero no sabemos nada acerca del tráfico, de las condiciones de la vía, etc., por lo que estamos planeando con información incompleta, de hecho así es el mundo real.

Para propósitos del desarrollo de este trabajo se utiliza como base el planeador conocido como BURIDAN¹ [Kushmerick, 1994] el cual no depende de las consideraciones de una información completa y determinista, ya que utiliza una distribución de probabilidades sobre posibles estados del mundo para poder modelar la información imperfecta el estado inicial del mundo, por lo que las acciones se modelan utilizando una distribución de probabilidades condicionales sobre cambios en el estado del mundo.

El adoptar un modelo probabilístico complica la definición que se tiene de un plan exitoso, ya que la planeación termina cuando se alcanza un plan lo suficientemente exitoso, para el cual su probabilidad de éxito sea mayor o igual a la del umbral, siempre que el plan exista, pero siempre hay la probabilidad de que exista un plan con mayor probabilidad que el generado.

BURIDAN² amplía el standard STRIPS con una representación que permite condicionales y probabilidades, en STRIPS una acción es activada si las precondiciones son satisfechas cuando la acción es ejecutada, en cuyo caso la acción tiene un efecto determinístico, en cambio si las precondiciones no se dan

¹ Puede obtenerse una copia de BURIDAN enviando un mail a bug-buridan@cs.washington.edu para obtener información sobre como adquirir BURIDAN mediante un FTP anonimo.

²Para Una mayor información acerca sobre como trabaja y la implementación de BURIDAN, una copia del paper original puede conseguirse mediante mail a {nick, hanks, weld}@cs.washington.edu.

la acción es desechada. Este modelo por su simplicidad no es suficiente para la representación de acciones con múltiples consecuencias.

El trabajo entonces de un algoritmo de planeación probabilístico es el construir una secuencia de acciones que se ejecuten por turno y que garantice que cuando se alcance el estado meta este tenga al menos una probabilidad de éxito igual a la del umbral inicialmente propuesto, cada plan consiste de un conjunto de acciones con una relación de orden temporal sobre las acciones, un conjunto de ligas causales, y un conjunto de metas.

La planeación comienza con un plan nulo; el cual consiste de tan solo dos acciones inicial y meta, con la restricción de que la inicial debe de ejecutarse primero y la meta al ultimo. Estas acciones especiales encodifican la distribución de la probabilidad sobre el estado inicial del mundo y la meta respectivamente, por ejemplo consideremos el mundo de un bloque, supóngase que inicialmente el bloque se encuentra en el piso y la pinza que servirá de sujeción para el mismo se encuentra seca con una probabilidad de 0.7. La acción inicial que corresponde a esta distribución se muestra en la figura 4.1 Cada consecuencia del inicio describe uno de estos dos posibles estados del mundo.

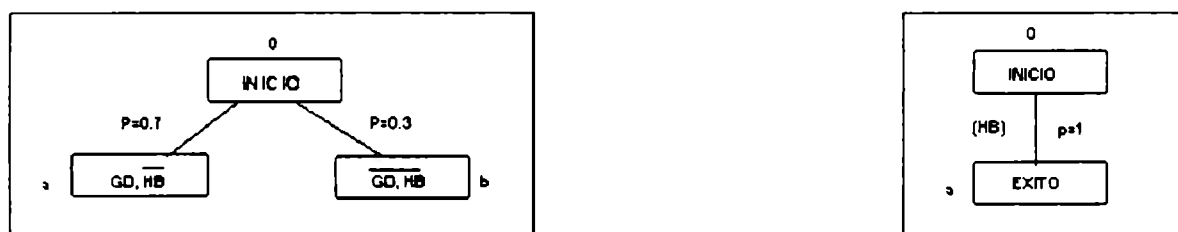


fig. 4.1 El plan encodifica el estado inicial de la distribución y la meta

Supóngase que la meta es el sostener el bloque HB en el lado derecho de la figura la acción meta para esta expresión. La meta tiene tan solo tiene un consecuente que produce el éxito, disparado por la expresión de la meta.

BURIDAN busca en el espacio planes parciales ejecutando en cada visita a los nodos dos operaciones:

Aseguramiento del plan: Para determinar la probabilidad de que el plan actual alcance la meta con al menos la probabilidad del umbral, y terminando satisfactoriamente.

Refinamiento del plan: Con lo que trata de incrementar la probabilidad de la satisfacción de la meta mediante la refinación del plan actual

- Escogiendo una submeta $p@A_j$, y agregando una liga a A_j desde alguna nuevo o ya existente acción A_i ; tal que está pueda producir p , en un intento de incrementar la probabilidad de p cuando A_j es ejecutada.
- Escogiendo un liga existente y resolviendo para está liga

Una diferencia fundamental entre BURIDAN y los planeadores clásicos es la que concierne a la relación entre las submetas de BURIDAN y la agenda de metas de los planeadores clásicos. En el planeador clásico cada entrada en la agenda debe de ser verdadera antes para que el plan pueda ser considerado como solución, mientras que en el caso de BURIDAN este no considera todas las submetas para formular un plan que alcance la meta con suficiente probabilidad.

4.2 Un ejemplo de planeación con BURIDAN

Retomando el ejemplo del robot cuya pinza la cual está probablemente mojada (con probabilidad de 0.3) y necesita sostener un bloque, la acción **tomar** no es realizable especialmente cuando la pinza está mojada, supóngase también que existe una acción **seco**, la cual regularmente sucede (con probabilidad de 0.8). Y se le pide a BURIDAN una probabilidad mínima de 0.9.

El refinamiento del plan comienza con un plan igual al de la figura 4.1, por lo que el sostener el bloque (**HB**) no es verdad en ninguno de los estados iniciales de la distribución, BURIDAN agrega una instancia de la acción **tomar**, ya que solo está acción puede hacer que **HB** sea cierto. BURIDAN desde **tomar**, consecuente con la acción meta, está liga permite concluir al planeador que **HB** será verdadera porque **tomar** es verdad, tan pronto como las condiciones bajo las cuales **tomar** produce **HB** y no así las acciones intermedias que producen $\neg\text{HB}$.

BURIDAN debe entonces de obtener las circunstancias por las cuales se cause tomar para producir HB. En general BURIDAN no puede garantizar que una acción tenga una consecuencia en particular, aunque el planeador puede agregar mayores refinamientos, para hacer que la consecuencia deseada sea *más a modo*, en el ejemplo esto significa que se debe de tratar de hacer que la pinza este seca para cuando **tomar** sea ejecutada. BURIDAN puede hacer que la pinza este seca (GD) de dos formás, inicialmente GD es cierto de dos formas. GD inicialmente está seca con una probabilidad de 0.7, por lo que la primera acción es agregar una liga desde el inicio.

Por lo que se tiene un plan sencillo el cual consiste de una acción tomar que tiene una probabilidad de 0.815, la cual es menor al umbral (0.90), por lo que BURIDAN entonces agrega un soporte adicional para GD agregando una liga desde la acción seco. La figura 4.2 muestra el resultado del plan alcanza HB con una probabilidad de 0.923 la cual excede el umbral de 0.9, por lo que BURIDAN ha encontrado una solución exitosa.

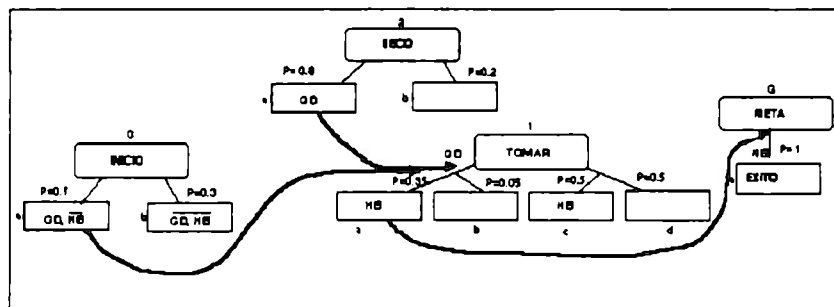


Fig. 4.2 Este plan para sostener el bloque HB trabaja con al menos 90% de probabilidades de éxito, aunque el factor GD (que la pinza este seca no se garantiza).

4.3 Modelo Propuesto

Aunque BURIDAN es un planeador que incorpora la incertidumbre en base a probabilidades de éxito-fracaso de un plan, este no nos asegura el éxito si las condiciones del mundo cambian, es decir solo se trabaja con el estado inicial del

mundo <una foto instantánea>, se consideran probabilidades del éxito del plan o planes generados, pero una vez seleccionado un plan este no puede cambiarse, por lo que si el estado del mundo cambia de tal manera que haga imposible la consecución del plan trazado, este fracasara sin remedio, por lo que si se desea tener un planeador exitoso este deberá de estar sensando continuamente la situación del mundo para obtener información que en un momento dado pueda llevar a tomar la decisión de cambiar de plan, por lo tanto el planeador debe de estar ejecutándose en tiempo real para poder reaccionar ante la incertidumbre precisamente en tiempo real, de modo semejante a como lo realizan los seres humanos; por ejemplo cuando un persona toma su automóvil para dirigirse a su trabajo puede planear la ruta a seguir, pero no sabe a ciencia cierta si la ruta está congestionada o no y si en un momento dado deba de tomar alguna vía alterna. De hecho conoce el principio, la meta y un esbozo del camino a seguir, pero si en el trayecto encuentra una contingencia reaccionara de manera inmediata a está, para poder llegar con éxito a su destino final.

En consecuencia de lo anterior, es recomendable el generar paso a paso los movimientos a seguir dados dos puntos fundamentales, el inicio y la meta, y a partir de sensores que nos permitan conocer en tiempo real el estado del mundo reaccionar a los cambios que pudiera tener este.

El alcance de este trabajo es el implementar en base a BURIDAN un planeador que pueda reaccionar a los cambios del medio ambiente que rodea al robot, y cuya ejecución se realice en tiempo real.

A semejanza de como lo haría el ser humano, retomando el ejemplo del conductor del automóvil, el cual tiene un esbozo del plan a seguir para llegar a su destino, pero si al estar interactuando con el mundo real, es necesario tomar una vía alterna a causa de diferentes motivos, de hecho el plan se modifica.

Una de las inconveniencias de los planeadores clásicos y de BURIDAN aunque este incorpore modelos probabilísticos e incertidumbre, es que una vez que un plan está ejecución no es posible modificar este, por lo que este trabajo incorpora en base a BURIDAN una forma de poder contrastar lo planeado, contra lo que ocurre en el mundo real, y que por si alguna circunstancia es necesario abortar el plan en ejecución se realice está operación y se tome un plan

alternativo, que de acuerdo a las condiciones del mundo nos garantice con éxito el llegar a la meta deseada.

4.4 Interacción con el mundo

Aunque no es el objetivo de este trabajo los sistemas de visión para poder sentir e interactuar con el mundo real es conveniente hacer una revisión de los trabajos realizados en el área, como ya se mencionó también en este trabajo se presupone que tomamos la información del mundo real por medio de una cámara de televisión, aunque en la práctica esto no es tan sencillo, el tomar información de esta forma es una tarea poco más que complicada, el tomar la información de la cámara significa que se debe disponer de un medio para digitalizarla, la tarjeta para digitalizar una imagen en blanco y negro más rápida disponible en el mercado hoy en día tarda 22ms. solamente en digitalizar además hay que agregar el tiempo que se tarda el algoritmo para poder analizar la imagen y poder obtener información útil de ella, por lo que si se desea tomar decisiones en tiempo real este tiempo resulta inoperante, por lo que es necesario pensar en otras alternativas.

La primera de ellas sería el pensar poder tener un modo de saber que tanto cambia un cuadro de una imagen de otro, generalmente una escena no cambia mucho con respecto a otra (a menos que estemos hablando de una calle con mucho tráfico) por lo que podría buscarse una alternativa para analizar la información de cada 3 cuadros por ejemplo.

Además de que no se necesita conocer de modo exacto la textura de los objetos por lo que quizás un algoritmo que nos permita detectar aristas sea suficiente [Zabih 1995] lo que además nos permitiría realizar una clasificación e inclusive detectar rompimiento de escenas .

Incluso ya que se trata de situaciones en las que hay movimientos implícitos y dado que se tiene conocimiento del mundo pueden usarse heurísticas

para la decodificación de imágenes [Kam 1993] o bien el uso de probabilidades de presencia de objetos usando para tal efecto la medición de ángulos y promedios de longitudes para guiar el proceso de reconocimiento [Shimshoni 1995], [Carlsson 1994], [Plantinga 1988].

El campo de la visión computacional y el reconocimiento de patrones es otra área de la robótica en la cual queda también mucho por realizar y lo anteriormente expuesto es solamente un esbozo de lo que podría ser el sistema de visión para un robot.

En el caso de este trabajo, la forma en que interactúa el planeador con el mundo es mediante sensores (a manera de radar) con los cuales se detecta la presencia de obstáculos, se cuenta con cuatro sensores localizados de tal forma que permita una detección periférica, lo cual se explica con detalle más adelante.

4.5 Interpretación del mundo

Un robot autónomo debe de ser capaz de poder analizar un escenario en tiempo real y en tiempo real tomar las decisiones que tengan lugar, es decir debe de analizar las imágenes por lo que debe saber interpretar distancias y profundidades en la escena que está observando, dependiendo del tipo de robot quizás deba de reconocer también texturas, aunque como se explico anteriormente por el tiempo que tardan estas tareas, lo más recomendable es solo detectar aristas.

Pero en lo que se refiere en la generación de un plan y conocer entre que puntos moverse, al planeador hay que darle las posibles alternativas por donde navegar y que el decida por una opción, haciendo una analogía con el ejemplo del automovilista supongamos que deseamos ir una calle de la ciudad la cual no sabemos donde está, lo que comúnmente hacemos es tomar la guía roji y ver donde se encuentra el sitio donde deseamos ir, la cual localizamos en un plano cartesiano, y de esa manera proponer un plan para dirigirnos hacia nuestra meta, dado el nivel de abstracción en donde estamos trabajando la planeación se hace en 2D.

Lo mismo es lo que se hace para trabajar con BURIDAN, se tiene un plano cartesiano del escenario y se le dan el espacio de estados, así como la localización inicial, la meta y la probabilidad de umbral, por lo que toda la planeación se hace en 2D.

4.6 Algoritmo para la planeación

De alguna manera el robot debe de tener contacto con el mundo real en este caso en particular la información del mundo es suministrada al planeador por

cuatro sensores (se escogen cuatro, con el propósito de poder tener en cierta forma dominio periférico del escenario), cuya función es el dar un valor entre el cero y el uno en función de que tanto está despejado el camino en que se está navegando, estos sensores pueden ser de diversos tipos; como cámaras de televisión, retinas en base a perceptrones, sonares, etc. Dado que el objetivo del proyecto no es el diseño de estos sensores, para efectos prácticos solo nos limitaremos al valor que entregue a su salida en función de lo que está observado, y haciendo la consideración de que el sensor entregara su máximo valor de salida si se tiene el camino despejado a 2 metros de distancia.

El modelo de planeación que se propone se basa en los sistemas basados en comportamiento [Arkin, 1992] los cuales han sido exitosamente probados en un amplio rango de tareas de navegación, fundamentando su éxito en la forma de construir su representación del mundo, con una continua interacción con este, de la cual obtiene información para poder actuar.

A semejanza del ser humano, el robot debe de tener un plan previamente concebido sobre el cual basarse, y contra el cual comparar la situación presente del mundo, por lo que en primera instancia debe de tenerse un plan de acción el cual debe de ser generado por BURIDAN ya que este incorpora la incertidumbre del mundo en forma de probabilidades. Pero toda vez que el robot empieza la ejecución del plan se debe de sensar el mundo para poder estar contrastando el medio ambiente actual contra el que se planeo, es decir ejecutar paso a paso la trayectoria y a cada paso comparar el plan contra lo real. Ejemplificando lo anterior; se tiene un escenario con obstáculos fijos y móviles (fig. 4.3) en dos dimensiones y dividiéndola como plano cartesiano, como lo pudiera ser una planta de producción o la casa de cualquier persona, en primera instancia el robot está colocado en la referencia (A, A') y tiene que llegar a su meta colocada en (L, K') .

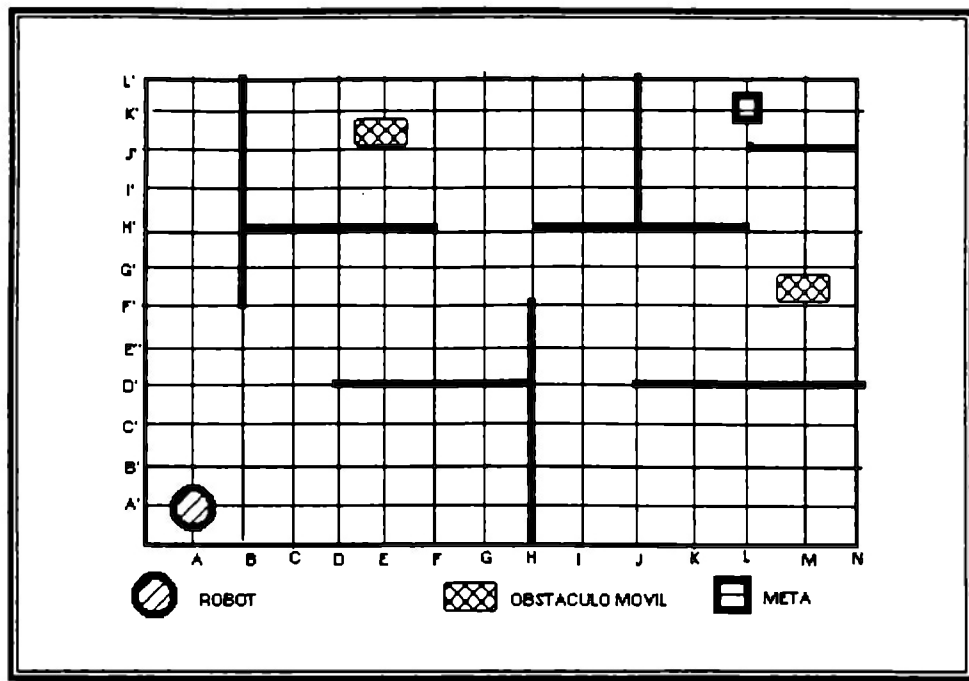


Fig. 4.3 Escenario de ejemplo

Para que BURIDAN produzca un plan solo habría que adicionar que se requiere una probabilidad de éxito del 90 %, con lo que el plan que se genera es el que se muestra a continuación (fig. 4.4).

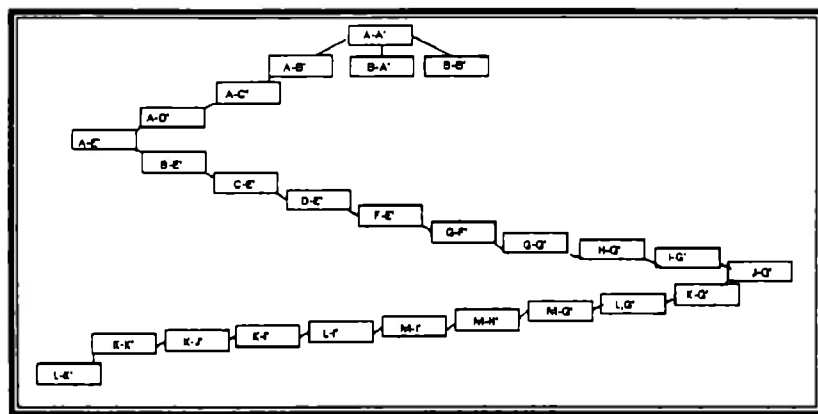


fig. 4.4 Plan ejemplo generado por BURIDAN

En la generación de su plan BURIDAN toma en cuenta los obstáculos móviles incorporándolos como incertidumbre, pero como ya se ha mencionado, básicamente la planeación se llevo a cabo tomando una fotografía instantánea del mundo, tomo en cuenta a los obstáculos, pero ignora si después esos obstáculos puedan impedir la consecución del plan.

El modelo que se propone usa como base este plan de BURIDAN y con el uso de una cola la cual inicialmente estará llena con la información que genero BURIDAN y de la cual se irán eliminando los componentes toda vez que se halla alcanzado la submeta en cuestión, con lo que se tiene una estructura que nos permitirá controlar la ejecución y generar planes alternativos eventualmente, como se explica más adelante.

Al iniciar su tarea el robot, también comienzan a moverse los obstáculos móviles el primero de ellos entre las coordenadas $(C, J') \rightarrow (I, J')$ y el segundo entre $(I, G') \rightarrow (M, G')$, la ejecución del plan puede presentarse sin problemas hasta la coordenada (H, G') , pero analicemos con más detenimiento que ha pasado hasta esta etapa, al iniciar su tarea el robot se mueve hacia el punto A,B' pero los sensores del robot deben de proveer información acerca del estado del mundo para poder validar la información y permitir entonces que se ejecute esta etapa de la tarea, la forma de validar esta información es mediante el empleo de una red neuronal, lo cual se explica con mayor detalle en la sección 4.4. Al momento de ejecutarse el primer paso del plan se elimina de la cola la coordenada (A, B') , con lo que la próxima etapa del plan a ejecutarse es la coordenada (A, C') , de la misma forma, antes de alcanzar esta etapa debe de tenerse información del mundo real, y validar de nueva cuenta la información que se tiene, para saber si por lo menos se alcanza la probabilidad de umbral de éxito del plan (90%) al ejecutar ese paso, la ejecución del plan en este caso no presentara ningún contratiempo hasta alcanzar el punto (H, G') .

En este caso hipotéticamente, solo hubo necesidad de cambiar de plan una sola vez, pero en caso de que se detectase otra posibilidad de fracaso del plan se hubiese tenido que generar tantos planes de contingencia como veces sea necesario, hasta que se alcance con éxito la meta, o hasta que BURIDAN indique que es imposible encontrar un plan con una probabilidad de éxito por lo menos igual a la del umbral.

Es obvio decir que el robot, de alguna manera debe de saber el sitio donde se encuentra inicialmente, dado que conoce la meta, debe de ir midiendo que tanto ha avanzado con ayuda de sus sensores y conocer su localización exacta, en un modelo complejo el sistema de visión es el que debe de proporcionar esta información en el caso de este trabajo lo que se hace es moverse a través de un plano cartesiano en donde se tienen perfectamente localizados todos los puntos, por lo que el simulador no se pierde.

Este modelo de planeación que se propone, es capaz entonces de poder reaccionar a los cambios en el estado del mundo a diferencia de los planeadores clásicos, donde como se menciono anteriormente la planeación se basa en una "foto instantánea del mundo" sin considerar los cambios que pueden haber en este y de los planeadores puramente reactivos, donde no se tiene un plan preconcebido sobre el cual estar comparando continuamente.

Por lo que el modelo propuesto se asemeja más a la forma en que los seres humanos planean y ejecutan las acciones, ya que previamente se tiene un plan general sobre el cual se trabaja y que si en dado caso se requiere modificar este por las condiciones del mundo se realizan sobre la marcha.

El diagrama de flujo del proceso algoritmico del planeador propuesto es el que se muestra en la figura 4.8

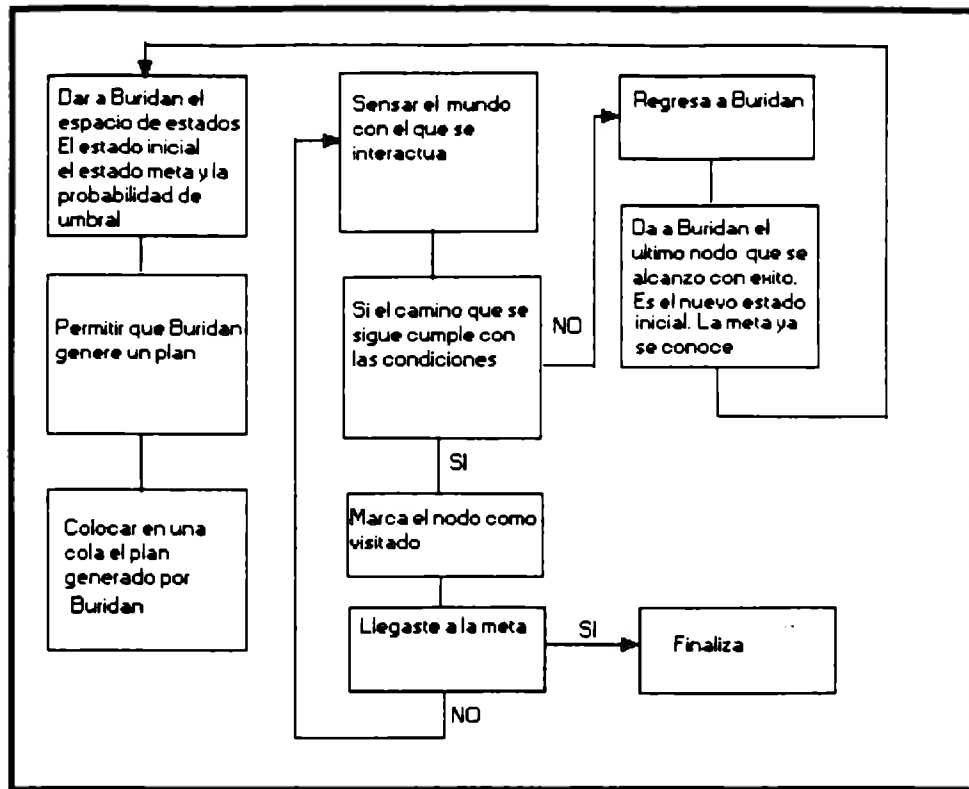


Fig 4.8 Diagrama de flujo del proceso algorítmico del planeador propuesto

En adición a BURIDAN solo es necesario el contar con una estructura de datos (cola) con el suficiente espacio para guardar tantas etapas del plan como sea necesario, sobra mencionar entonces que debe de tratarse de una estructura que maneje la memoria en forma dinámica, mientras que para interpretar la información que llega a través de los sensores, se utiliza una red neuronal utilizando el modelo de retropropagación el cual fue formalizado por Werbos [Werbos, 1974], y posteriormente por Rumelhart [Rumelhart, 1986] esta arquitectura neuronal permite a la red aprender un conjunto predefinido de entradas-salidas usados como conjunto de entrenamiento, además de entrenar a los nodos de la capas intermedias tal que puedan reconocer diferentes patrones, aunque se encuentren incompletos o con la presencia de ruido [Freeman 1992].

Para fines de la implementación y entrenamiento de la red neuronal se utiliza el simulador Neuroshell2 que corre bajo Windows®, que entre sus ventajas tiene la interfaz gráfica, además de la facilidad que se tiene para poder importar-exportar datos de otros paquetes que corren bajo esta plataforma, al igual de la disponibilidad inmediata de gráficos relacionados al comportamiento

de la red. Además de implementar la red en lenguaje C, con el fin de poder correr está en ambiente UNIX.

4.7 Implementación de la red

Quizás los problemas principales cuando se trata de entrenar una red neuronal del tipo que sea es el número de neuronas a utilizar, así como los patrones de entrenamiento, desgraciadamente no existe una "regla de oro" que permita el poder afirmar o calcular con más o menos grado de certeza las estimaciones anteriormente mencionadas, por lo que todas las apreciaciones se hacen en base al ensayo-error, y hasta encontrar una combinación tal que en base a los patrones de entrada se obtengan resultados dentro de un rango esperado, y donde el error sea despreciable para propósitos prácticos.

Aprovechando las características de la señal sigmoideal (fig. 4.9), presente a la salida de esta arquitectura de red neuronal, nos permite el definir los criterios y establecer cuales son los patrones de comportamiento de la red, y los cuales van a depender de los valores que se obtengan de los sensores.

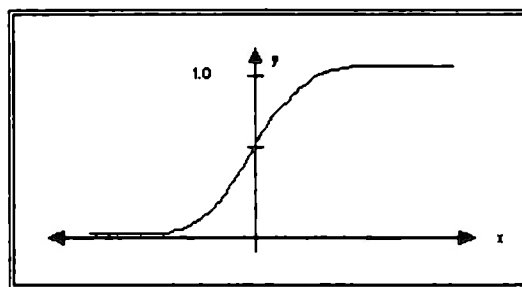


Fig. 4.9 Señal característica de la función sigmoide

A partir del hecho de que el móvil posee cuatro sensores para que estos formen un par de retinas, con lo cual se puede tener una visión periférica del mundo, (la distribución de estos es la que se muestra en la figura 4.10) con lo cual se puede tomar las

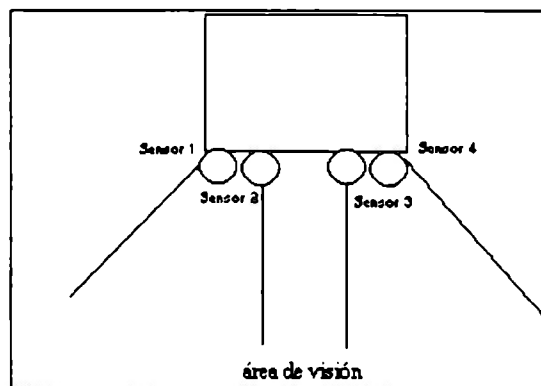


Fig. 4.10 Distribución de los sensores del móvil

siguientes consideraciones: Si el valor de la salida de los sensores 2 y 3 es uno o un valor muy cercano significa que el camino que se sigue está libre, pero una disminución o aumento en el valor de cualquiera de ellos equivale a que el camino comienza a volverse menos probable (quizás una puerta semiabierta o un objeto que obstruye parcialmente el camino), si el valor de los sensores 1 y 4 es cercano al cero o cero y el valor de los sensores centrales es uno o un valor muy cercano el valor de la salida de la red neuronal debe de tender a 0.5 o en la vecindad de este, pensemos por ejemplo que el robot se desplaza por un pasillo estrecho, ya que es el camino que tiene más probabilidades de éxito.

La disminución en el valor de salida de los sensores centrales por regla general debe de ir acompañada por un aumento en cualquiera de los sensores de los extremos lo cual en un momento dado va a significar un cambio en la trayectoria, ya que el camino original decaerá por abajo de la probabilidad de

umbral y alguno de los caminos alternativos (izquierda o derecha) será una mejor opción, aunque por otro lado podría ser el caso de que todos los sensores tuvieran una salida máxima o en la vecindad de este, lo que equivale a moverse en un espacio sin obstáculos y en el cual sin duda la mejor opción es moverse siempre hacia el frente ya que mientras la probabilidad del camino que se sigue, no caiga por abajo de la probabilidad definida por el umbral, no deberá de tomarse la decisión de cambiar de trayectoria.

Por el contrario en el momento en que se detecte una caída en el valor de la probabilidad por abajo de la del umbral debe de abortarse el plan y seleccionar otro.

En el momento de elegir otro plan BURIDAN deberá de mirar entonces el valor de la probabilidad de éxito de los caminos alternativos, tocando a este, escoger un plan alternativo que asegure un probabilidad de éxito al menos igual a la del umbral, en el caso de que ninguno cumpla con está condición, entonces BURIDAN deberá de abandonar la operación ya que no se encontró un plan que satisfaga los requerimientos mínimos.

Para el entrenamiento de la red neuronal, se escogieron 26 patrones los cuales engloban los casos más representativos y los cuales se muestran en la tabla 4.1

Sensor 1	Sensor 2	Sensor 3	Sensor 4	Salida
1	1	1	1	0.5
0	1	1	1	0.5
1	1	1	0	0.5
0.5	1	1	0.5	0.5
0.6	1	1	0.6	0.5
0.7	1	1	0.7	0.5
0.8	1	1	0.8	0.5
0.9	1	1	0.9	0.5
0.4	1	1	0.4	0.45
0.3	1	1	0.3	0.55
0.2	1	1	0.2	0.45
0.1	1	1	0.1	0.45
0.5	0.5	1	1	0.7
0.25	0.25	1	1	0.6
0	0	1	1	1
0	0	0.5	1	0.8
0	0.1	0.75	0.8	0.95
0	0.2	0.75	0.8	0.75
0	0.3	0.5	0.75	0.6
0	0.4	0.5	0.8	0.4
1	1	0	0	0
1	1	0.1	0.2	0
1	0.8	0.4	0.4	0.20
0.8	0.8	0.5	0.5	0.25
0.7	0.9	0.8	0.4	0.30
0.6	0.25	0.25	0.25	0.35

Tabla 4.1 Patrones de entrenamiento de la red neuronal

Cuyas estadísticas se muestran en la tabla de la fig. 4.11

Variable Name	Sensor1	Sensor2	Sensor3	Sensor4	Salida
Variable Type					
Min	0	0	0	0	0
Max	1	1	1	1	1
Mean	.4711539	.7115384	.7942300	.5923077	.4853846
Std. Deviation	.3883743	.3795544	.3014389	.3431304	.2623621

Fig. 4.11 Estadísticas de los patrones de entrenamiento de la red neuronal

Mientras que en las gráficas de las figuras 4.12, 4.13, 4.14, 4.15 se observan los comportamientos del valor de cada una de las entradas, con respecto al número de muestra de entrenamiento, en la figura 4.16 se muestra el

patrón de conducta de la salida, mientras que en la gráfica de la figura 4.17 contiene la sobreposición del comportamiento de los cuatro sensores junto con el patrón salida.

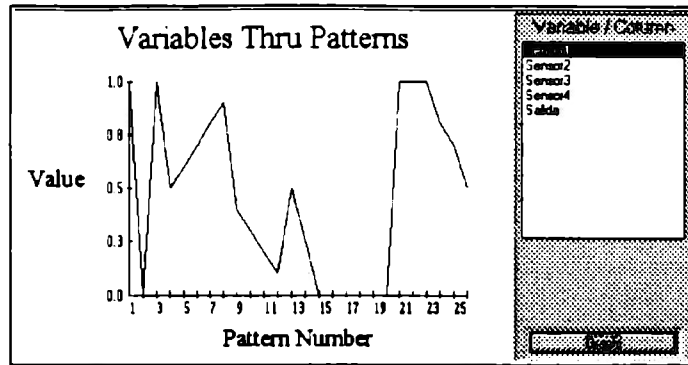


Fig. 4.12 Gráfica del comportamiento del sensor1

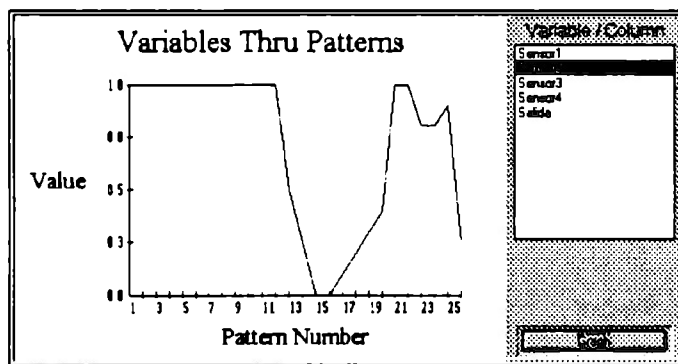


Figura 4.13 Gráfica del comportamiento del sensor 2

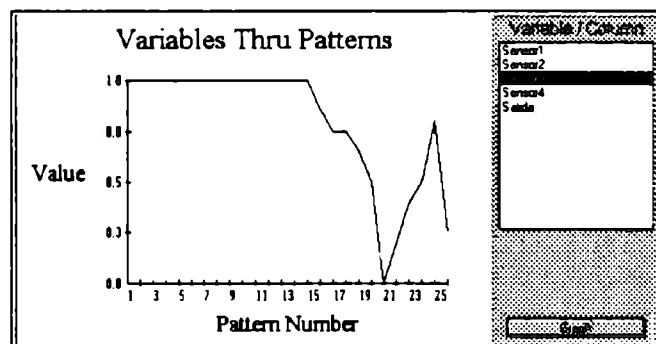


Fig 4.14 Gráfica del comportamiento del sensor 3

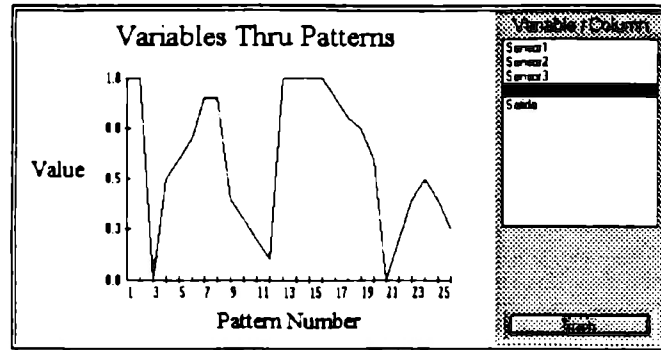


Fig. 4.15 Gráfica del comportamiento del sensor 4

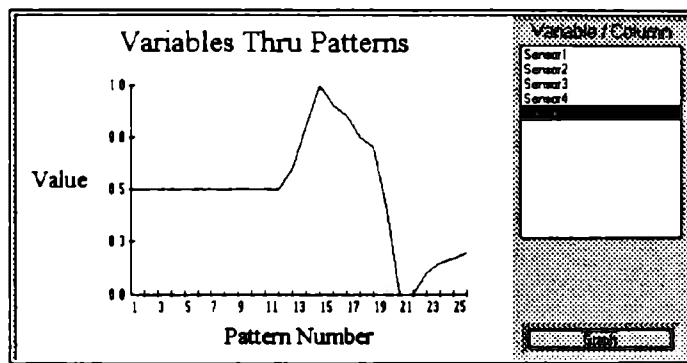


Fig. 4.16 Gráfica del comportamiento de la salida

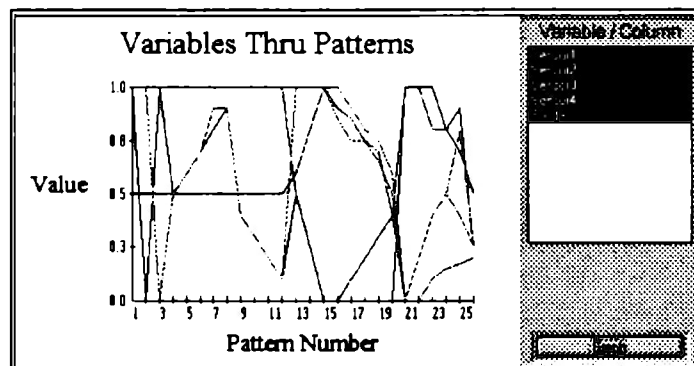


Fig. 4.17 Gráfica del comportamiento de los patrones de entrenamiento de la red neuronal

Los parámetros bajo los cuales se rigió el entrenamiento de la red neuronal se muestran en la figura 4.18.

<p>Stop training when the one of these is true about the training set.</p> <p><input type="checkbox"/> average error < <input type="text" value="0.002"/></p> <p><input checked="" type="checkbox"/> epochs since min. avg. error > <input type="text" value="2000"/></p> <p><input type="checkbox"/> largest error < <input type="text"/></p> <p><input type="checkbox"/> learning epochs > <input type="text"/></p> <p><input checked="" type="checkbox"/> Train with Training Set in memory</p>	<p>Stop training when one of these is true about the test set & Max-Patience test interval is > 0.</p> <p><input type="checkbox"/> average error < <input type="text"/></p> <p><input type="checkbox"/> epochs since min. avg. error > <input type="text" value="20000"/></p> <p><input type="checkbox"/> largest error < <input type="text"/></p> <p>Max-Patience test interval (epochs): <input type="text" value="0"/></p> <p><input checked="" type="checkbox"/> Train with Test Set in memory</p>
<p>Consider Missing Values to be:</p> <p><input type="radio"/> none <input type="radio"/> minimum values <input type="radio"/> maximum values <input type="radio"/> average values <input checked="" type="radio"/> error conditions</p>	

Fig. 4.18 Parámetros de entrenamiento de la red neuronal

Una de las consideraciones importantes cuando se trabaja con el modelo de retropropagación, son los valores iniciales de los pesos y de las ligas, estos fueron asignados de manera aleatoria, por el simulador y son los que se muestran las figuras 4.19, 4.20, 4.21 y 4.22

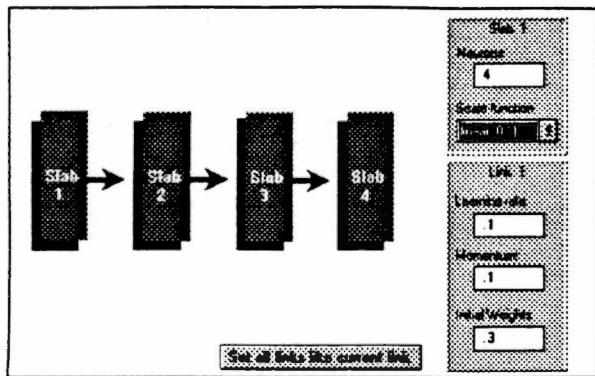


Fig 4.19 Configuración inicial de la capa de entrada

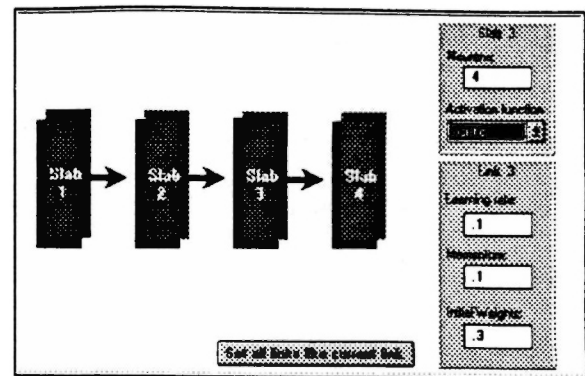


Fig. 4.21 Configuración inicial de la capa oculta 2

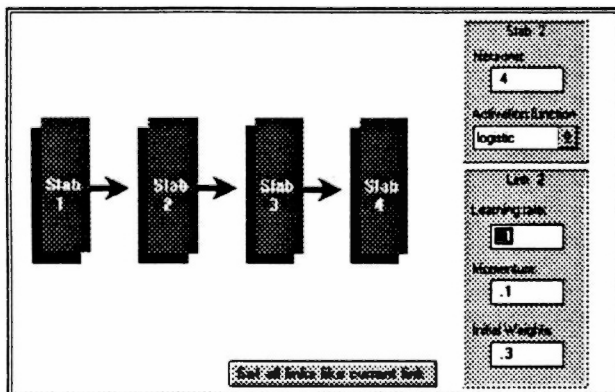


Fig. 4.20 Configuración inicial de la capa oculta 1

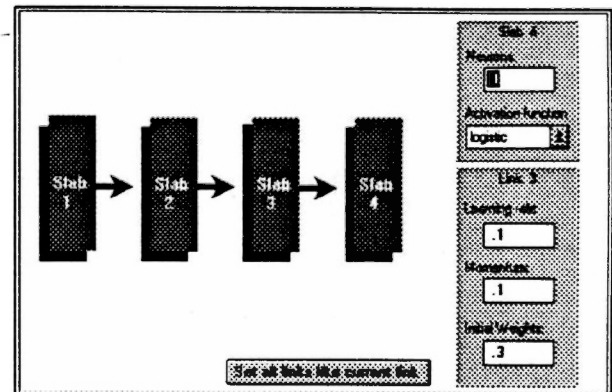


Fig. 4.22 Configuración inicial de la capa de salida

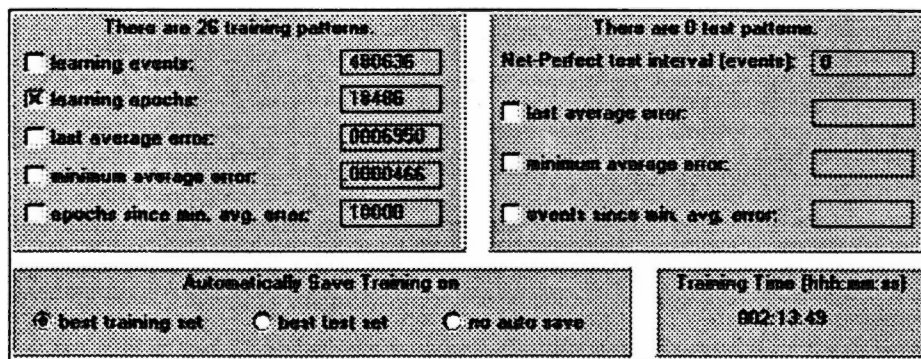


Fig. 4.23 Resultados presentados después del entrenamiento de la red neuronal

En la figura 4.23 se muestran los resultados obtenidos después del entrenamiento de la red neuronal, donde el error mínimo promedio es de 0.0000460. mientras que en la figura 4.24 se muestra la gráfica del promedio de error del conjunto de entrenamiento

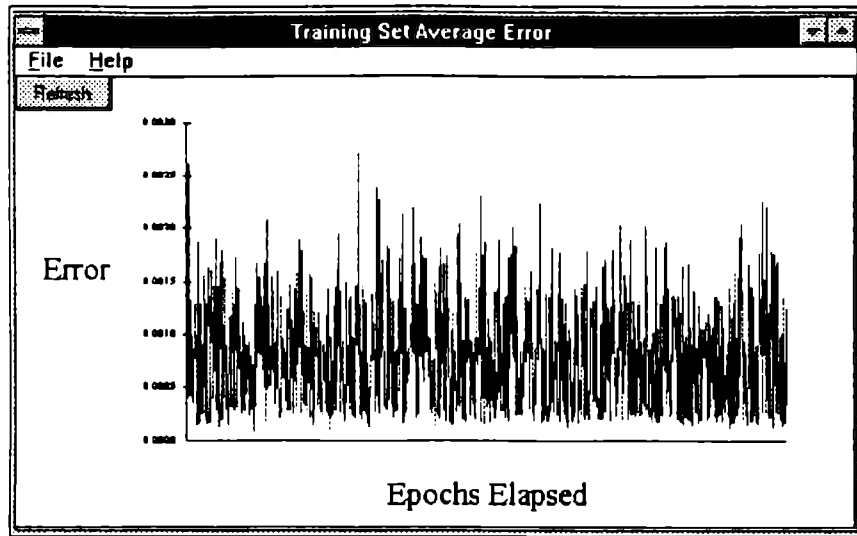


Fig. 4.24 Gráfica del error promedio de los patrones de entrenamiento en función de la épocas de entrenamiento transcurridas

Toda vez que la red neuronal fue entrenada se procedió a realizar pruebas mediante un conjunto entrenamiento, el cual se muestra en la tabla de la figura 4.25, las columnas A, B, C, D corresponden a los sensores 1, 2, 3, y 4 respectivamente mientras la columna E es la salida esperada para cada conjunto de prueba

	A	B	C	D	E
1	0.90	0.90	0.90	0.90	0.45
2	0.10	0.90	0.90	0.90	0.46
3	0.90	0.90	0.92	0.10	0.44
4	0.45	0.90	0.90	0.45	0.44
5	0.50	0.90	1.00	0.60	0.53
6	0.60	1.00	1.00	0.70	0.49
7	0.80	0.85	0.80	0.90	0.38
8	0.90	0.90	0.85	0.75	0.40
9	0.40	0.94	0.96	0.40	0.48
10	0.30	0.89	0.89	0.30	0.45

Fig. 4.25 Conjunto de prueba de la red neuronal

En la tabla de la figura 4.26 muestra la comparación de resultados; el esperado contra el obtenido por la red (columna A resultado esperado, columna B resultado obtenido) mostrando en la columna C el error resultante

	A	B	C
1	0.45	0.45	0.00
2	0.46	0.46	0.00
3	0.44	0.44	0.00
4	0.44	0.44	0.00
5	0.53	0.53	0.00
6	0.49	0.49	0.00
7	0.38	0.38	0.00
8	0.40	0.40	0.00
9	0.48	0.48	0.00
10	0.45	0.45	0.00

Fig. 4.26 Comparación de los resultados obtenidos con el conjunto de prueba

De los resultados mostrados en la figura 4.26 podemos extrapolar que el entrenamiento fue realizado adecuadamente y que los resultados que nos brinde serán lo suficientemente confiables, de hecho para todos los conjuntos de prueba el error resultante fue cero.

4.8 Sumario

Esta sección mostró la forma en que trabaja el sistema de planeación propuesto y la interrelación que existe entre el plan conceptual y la ejecución de este en el mundo real. Y la fuerte interacción que existe entre las etapas de planeación y navegación de un robot, de hecho no puede existir una separación entre ambas abstracciones ya que no podemos olvidarnos del plan durante la ejecución del mismo.

Muchos factores influyen en el rendimiento de la planeación-ejecución como son la velocidad, la robustez del cálculo de probabilidades, en realidad todo el entorno que interactúa el robot, por lo que se debe tener especial cuidado en los patrones de entrenamiento con los que se ejercita a la red neuronal y en el número de estos sobre todo ya que son una fuente de

conocimiento, y porque es la forma en que se va interpretar la situación actual del mundo información con la que se van a tomar decisiones.

5 CONCLUSIONES



El afirmar que existe una dicotomía entre los sistemas de planeación cognitivos y los reactivos es falso, ya que existe una fuerte evidencia por la forma en que son usados juntos en la navegación de animales, ya que la planeación propiamente dicha es asociada con los procesos cognitivos e inferencias lógicas realizadas sobre los niveles de representación del mundo y la ejecución del plan es altamente apropiado para el control reactivo.

La planeación para un robot autónomo puede ser visto desde ambos puntos de vista pero no es posible tomarlos aislados uno del otro si se quiere tener un sistema inteligente robusto y flexible.

Considerar la planeación de caminos sin tomar en cuenta la ejecución del plan, equivale a restringir el dominio del problema, ya que un robot debe de tener la habilidad de responder rápida y efectivamente a los cambios que están ocurriendo en su mundo, si un sistema intenta el modelar o el estar preparado con un plan para cualquier eventualidad que pudiera ocurrir, pudiera ser que el proceso de planeación nunca termine, y por otra parte si no es seguro para un robot el tomar una decisión o suposición acerca del mundo, entonces este no refleja la naturaleza dinámica que este debe de exhibir.

Existe en la actualidad en la comunidad científica dos campos opuestos de desarrollo en lo que a planeación para la navegación de robots se refiere, cada uno por su parte ha tratado de demostrar su superioridad, lo cual ha sido un factor que ha influido en el desarrollo del área ya que cada paradigma ataca desde su punto de vista los mismos problemas, cuando en realidad es que la escuela jerárquica está más involucrada con la formulación del plan, mientras que la escuela reactiva trata con la ejecución compartiendo ambos la replaneación dinámica.

El sistema que aquí propone incorpora lo que hace mejor cada uno de los paradigmas, por un lado se utiliza el enfoque jerárquico para realizar la planeación usando para ello un modelo probabilístico, con lo que se incorpora la incertidumbre del medio, dado que el mundo real es incierto por naturaleza, y por el otro lado la ejecución del plan se realiza desde el punto de vista del paradigma de la teoría reactiva, con el uso de estrategias perceptuales con lo que se

parametriza y se realizan las instancias de manera consistente con el conocimiento disponible.

La escuela cognitiva habla de la planeación y la navegación de un robot como tareas independientes, pero es imposible olvidarse de la planeación durante la ejecución de un plan, ya por más que se trate de integrar la incertidumbre inherente del mundo real al planeador, jamás vamos a tener un plan con el 100% de confianza de éxito, por lo que no podemos dejar este a la deriva si no que es necesario retroalimentar continuamente durante el proceso de navegación al planeador, por lo tanto no deberían de considerarse como etapas separadas si no como una sola etapa.

En este capítulo se hace una comparación con otros desarrollos realizados antes, de las aportaciones realizadas y de las limitaciones, lo que sirve de base motivacional para el desarrollo de futuros trabajos en el área.

5.1 Contribuciones del modelo

El haber tomado como base un planeador probabilístico, permitió manejar la incertidumbre desde el estado inicial del mundo, (por medio de una distribución de probabilidades sobre todos los estados del mundo) con lo que la ejecución de una acción causa una transición de la distribución de un estado a otro, con lo que no se limita a planear sobre una foto instantánea del mundo, si no que por el contrario se considera que el mundo real es incierto por naturaleza.

El incorporar al modelo un control para la ejecución el cual incluye una forma de sensado del mundo para reaccionar ante los cambios que se presentan en este, ya que si realmente se desea tener un robot autónomo este debe ser capaz de tomar sus propias decisiones, y al estar interactuando continuamente

con el medio se tiene una forma de medir el grado de éxito con el que se está llevando a cabo la ejecución del plan formulado y permite el poder reaccionar ante eventualidades, con lo que inmediatamente se pueden generar cuantos planes alternativos sean necesarios para llevar al robot a su meta, siempre tomando en cuenta la incertidumbre propia del medio .

5.2 Comparación con otros desarrollos

A partir de 1989 año en que Ronald Arkin [Arkin, 1989] habla del modelo AuRA (Arquitectura para robots autónomos) han surgido varios desarrollos que incorporan a los paradigmas cognitivo y reactivo para la planeación/ejecución de sistemas de planeación, lo cuales se basan en planeadores que realizan su labor con conocimientos a priori del mundo, es decir sin tomar en cuenta la naturaleza incierta del medio, incorporando estructuras de control que permiten al robot sensor al mundo y de ser necesario modificar el plan general cuantas veces se requiera, siendo la principal diferencia entre el modelo que aquí se propone y los desarrollados con anterioridad.

5.3 Limitaciones y Futuros Desarrollos

Durante el desarrollo de este modelo se detectaron algunas limitaciones y desventajas, las cuales amplían el panorama para futuros trabajos en base a esta propuesta.

5.3.1 Costo del sensado

Se ha considerado que todos los cambios no esperados en el medio ambiente pueden ser inmediatamente procesados por el sistema, aunque en realidad no se puede asumir que resulte lo suficientemente barato (computacionalmente hablando) el hecho de estar sensando al mundo de una forma continua, extendiendo un poco el concepto deberían de asignarse costos y utilidades a los sensores, la cual podría ser derivada de las metas y la importancia de la proposición a ser sensada para alcanzar la meta.

5.3.2 Limitaciones de Hardware

Para la implementación del modelo se requiere de un hardware especializado, en el cual se tengan varios procesadores trabajando en paralelo, porque una vez generado un plan debe de monitorearse al mismo tiempo el estado del mundo, estar comparando contra el plan original la información que se está llegando y estar actualizando el estado de la ejecución, para fines de prueba se utilizó la programación concurrente, con lo que se tiene únicamente la emulación en software de lo que se debe de tener en hardware. Para hacer las pruebas del modelo propuesto se trabajó en una estación de trabajo RISC 6000 programando en C para UNIX, y comunicando los diferentes procesos mediante sockets del modo conectado, en realidad y lo más conveniente sería contar con un procesador dedicado para cada tarea (planeación, ejecución, sensado).

5.3.3 Aprendizaje de las acciones realizadas

Al seguir tomando como base la teoría reactiva en cuanto a la ejecución del plan sería deseable que el sistema pudiera aprender a moverse en determinados entornos, para que pueda tomar decisiones de una manera menos cara, sería adecuado el tener asociada una segunda red neuronal que permitiera el entrenamiento acerca de los movimientos de robot, y su reacción ante las diferentes situaciones.

5.4 Limitaciones de BURIDAN

El utilizar a BURIDAN como base de la planeación, funcionalmente nos está limitando a una representación preposicional, y a la generación de un plan que probablemente no sea el mejor, por que se basa en la búsqueda del primer plan que cumpla con la probabilidad de umbral requerida, por lo que sería necesario el explorar otros métodos algorítmicos para controlar la búsqueda de mejores planes, o la posibilidad del uso de planeadores basados en frames involucrando conceptos de lógica difusa para la realización de razonamientos. La principal desventaja de BURIDAN, es que nos entrega el primer plan que cumple con las condiciones dadas, por lo que corre en un tiempo polinomial, pero en grafos densos (un plano lo suficientemente grande) pudiera degenerar en una búsqueda de fuerza bruta. Por lo que se vuelve lento para tener decisiones en tiempo real, cuando replanea y quizás la alternativa en ese caso es solamente generar un bosquejo de la ruta alternativa.

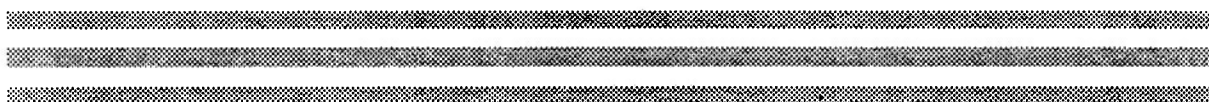
5.5 Trabajos a futuro

Las limitaciones anteriores más que ser una barrera del modelo representan áreas de desarrollo futuro muy interesantes, con las cuales poder ampliar las perspectivas del alcance del modelo, quizás el siguiente paso es el de dotar al

modelo de una red que permita al planeador aprender de las decisiones tomadas, a si como el poder incorporar frames en el modelo de planeación con lo que se podría tener un marco de trabajo más flexible y más acorde con las condiciones del mundo real.

Este modelo propone una alternativa, pero a la cual es necesario incorporarle un sistema de visión y reconocimiento de patrones para que el robot pueda saber exactamente donde se encuentra y dejar de guiarse por radar. La puerta está abierta en realidad la investigación en el área apenas comienza y quizás no este lejano el día en cada uno de nosotros pueda tener en casa su propio robot, tal y como lo hemos visto en las películas y el cual es uno de los sueños que todavía no puede realizar el hombre.

BIBLIOGRAFIA



- **[Abrams, 94]** Abrams Steven, Allen Peter. Dynamic Sensor Planning, DARPA DACA-76-92-C-007, 1994.
- **[Arbid, 1964]** Arbid Michael, Brains Machines and Mathematics, McGraw Hill, NY 1964.
- **[Arkin, 1989]** Arkin Ronald, Motor Schema Based Mobile Robot Navigation, International Journal of Robotics Research , 1989.
- **[Arkin, 90]** Arkin Ronald, Autonomous Navigation in a Manufacturing Environment, IEEE Transactions on Robotics and Automation, 1990.
- **[Arkin, 90]** Arkin Ronald. Integrating Behavioral, Perceptual and Word Knowledge in Reactive Navigation en [Maes 90] 1990.
- **[Arkin, 89B]** Arkin Ronald., Neuroscience in motion: The Application of Schema Theory to Mobile Robotics, Plenum Press, 1989B.
- **[Arkin,87]** Arkin R., AuRA: An Architecture for vision-based Robot Navigation, Proceedings of the 1987 DARPA Image Understanding Workshop, Los Angeles, Ca.

- **[Arkin, 92]** Arkin R., Towards the Unification of Navigational Planning and Reactive Control Notes of the AAAI Spring Symposium on Robot Navigation Stanford University, 1992.
- **[Bagchi, 94]** Bagchi S. Planning under Uncertainty by Spreading Activation Through an Adaptive Probabilistic Network, CIS 94-03, Universidad Vanderbilt, 1994
- **[Balkenius, 94]** Balkenius C., Biological Learning and Artificial Intelligence, Lund University Cognitive Science, 1994.
- **[Brooks, 86]** Brooks Rodney, A Robust Layered Control System for a Mobile Robot, IEEE Journal of Robotics and Automation 1986.
- **[Brooks, 91]** Brooks Rodney, Integrated Systems Based on Behaviors SIGART of Integrated Intelligent Systems, 1991.
- **[Brooks, 91]**. Brooks R. Intelligence without representation. Artificial Intelligence 47:139-159, 1991.
- **[Brooks, 91]** Brooks A. Rodney. Intelligence Without Reason, MIT, Artificial Intelligence Laboratory, 1991.

- **[Brooks, 90]** Brooks Rodney. Learning to Coordinate Behaviors, AAAI-90 Boston Ma. 1990.
- **[Brooks, 89]** Brooks Rodney. Robots Being , IEEE/RSJ International Workshop on Intelligent Robots and Systems, Tsukuba Japan, 1989.
- **[Cohen, 89]** Cohen Paul, Feigenbaum Edward. "The Handbook of Artificial Intelligence Vol I II, III, IV, Addison-Wesley, 1989.
- **[Donald, 92]** Donald B. The complexity of planar compliant motion planning under uncertainty, Computer Science Department, Cornell University, 1992.
- **[Freeman, 92]** Freeman J. Neural Networks, Addison-Wesley Publishing Company, 1992.
- **[Fu, 1990]** K.S. Fu. "Robotics: Control, Sensing, Vision, and Intelligence"n McGraw Hill, 1990.
- **[Kam, 93]** Kam Anthony, Kopec Gary. "Heuristics Image Decoding Using Separable Models" Xerox Palo Alto Research Center, 1993.

- **[Khatib, 86]** O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots" Int J. of Robotics Research 5 1986.
- **[Kushmerick, 93]** Kushmerick, N. ; Hanks, S.; Weld, D. An Algorithm for probabilistic planning. Technical Report 93-06-04, University of Washington, Seattle, 1993.
- **[Kushmerick, 93]** Kushmerick, K. An algorithm for probabilistic planning. Technical Report 93-06-04, Universidad de Washington, 1993
- **[Latombe, 91]** Latombe Jean C. Robot Motion Planning, Kluwer, Boston Ma, 1991.
- **[Latombe, 94]** Latombe J.C. Probabilistic Roadmaps for Path Planning in high-Dimensional Configuration Spaces, ARPA, N00014-92-J-1809, 1994
- **[Latombe, 94]** Latombe Jean C. Randomized Query Processing in Robot Motion Planning, DARPA, Stanford University, 1994.
- **[Lumelsky, 86]** Lumelsky V. Continuous Motion Planning in Unknown Environment for a 3D Cartesian Robot Arm, IEEE International Conference on Robotics and Automation, 1986.

- **[Maes, 90]** Maes Pattie. Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back. MIT Press. 1990.
- **[Meistel, 91]** Meistel A. "Autonomous Mobile Robots: With cognitive control" Word Scientific, 1991.
- **[Movarec, 81]** Movarec Hans. Robot Rober Visual Navigation, UMI Research Press, Ann Arbor MI 1981.
- **[Nilsson, 80]** Nilsson N.J. Principles of Artificial Intelligence. Morgan Kaufmann, 1980.
- **[Nilsson, 84]** Nilsson Nils. Shakey the Robot, SRI. AI. Center Technical Note 323 1984.
- **[Norman, 93]** Norman D. Attention to action: Willed and Automatic Control of Behavior, from Consciousness and Self regulation, Advances in Research and Theory, eds, 1993.
- **[Papadimitriou, 85]** Papadimitriou Christos. "Games Against Nature", Journal of Computer and System Sciences 31, 288-301, 1985.

- **[Papadimitriou, 87]** Papadimitriou Christos. "The Complexity of Markov Decision Processes", *Mathematics of Operations Research*, Vol 12 , No3, 1987.
- **[Plantiga, 88]** Plantiga William, "A Continuous, Viewer-Centered Object Representation for Computer Vision", University of Wisconsin-Madison, 1988
- **[Ponce, 95]** Ponce Antonio, "Inteligencia Artificial y Educación", SINECTICA No 5 ITESO Mex. Segundo Semestre 1995.
- **[Rumelhart, 86]** Rumelhart D. *Parallel Distributed Processing* Vol 1 , 2 MIT press, Cambridge, MA, 1986.
- **[Sacerdoti, 74]** Sacerdoti, E.D. Planning in a hierarchy of abstraction spaces. *Artificial Intelligence* 5: 115-135, 1974
- **[Shimshoni, 95]** Shimshoni Ilan, Probabilistic 3D Object Recognition, Department of Computer Science and Beckman Institute, University of Illinois, Urbana Il. 1995.
- **[Simmons, 94]** Simmons R. Probabilistic Navigation in Partially Observable Environments, School of Computer Science, Carnegie Mellon University. 1994

- **[Werbos, 74]** Werbos P. Beyond Regression: New tools for prediction and Analysis in the Behavioral Sciences, PhD thesis , Harvard, Cambridge, MA, August 1974.
- **[Wilkins, 79]** Wilkins David E. Using patterns and Plans to Solve Problems and Control Search, Stanford AI Memo 329, 1979.
- **[Winston, 84]** Winston Patrick H. Artificial Intelligence, Second Edition, Addison-Wesley, Reading, MA, 1984.
- **[Zabih, 95]** Zabih Ramin, "A Feature-Based Algorithm for Detecting and Clssifying Scene Breaks" Computer Science Department Cornell University, 1995.

Q24 -