

Técnica de Navegación Híbrida para Robots Móviles

Adrian. E. Scillato*, Daniel. L. Colón** y Juan. E. Balbuena***

*Estudiantes de 5º año de la carrera de Ingeniería Electrónica,
Universidad Nacional del Comahue. Neuquén, Patagonia Argentina.*

*e-mail: *ascillato@gmail.com **danielluiscolon@yahoo.com.ar ***eduardobalbuena@gmail.com*

Resumen— La navegación robótica es un área de investigación y aplicación muy importante dentro de la Inteligencia Artificial en robótica cognitiva. El objetivo principal de este trabajo es generar una técnica de navegación robótica que esté compuesta de las mejores características de las técnicas existentes y su uso en forma conjunta. Además, se evalúa la performance de estos paradigmas y su impacto en la navegación robótica.

Abstract— Robot navigation is a very important research and application area within Artificial Intelligence on cognitive robotics. The main objective of this paper is to generate a robot navigation technique, which gathers the best characteristics of existing techniques and applies them together. Besides, the performance of these paradigms concerning robot navigation is assessed.

I. INTRODUCCIÓN

Los robots son útiles para reproducir ciertas capacidades de los organismos vivos, como son: la exploración y el transporte para los robots móviles y, la asistencia médica y la automatización de procesos industriales para los robots fijos. Además son utilizados en el control de prótesis, en entretenimiento, etc.

Existen varios paradigmas de Inteligencia Artificial (IA) entre los diferentes tipos de arquitecturas para el control de la autonomía de agentes robóticos móviles [1]. Por un lado está la navegación enmarcada en el contexto de las arquitecturas reactivas, las cuales se basan en el logro de metas de desplazamiento del robot como una consecuencia de acciones estímulo-respuesta. Generalmente estas arquitecturas están fundadas en modelos de procesamiento de información masivamente paralela, tal como es el caso de las redes neuronales artificiales (RNA). Aunque las técnicas reactivas poseen la ventaja de la rapidez de procesamiento de información y la velocidad de reacción, las mismas están limitadas en cuanto a la complejidad de las tareas que el robot pueda abordar debido a que no puede manejar aquellas que requieren planificación.

Por otro lado, se encuentran en el área de sistemas inteligentes, las arquitecturas clásicas, las cuales implican el empleo de técnicas de razonamiento para decidir acciones del agente robótico en base a un modelo del entorno. En este caso, el procesamiento de la información por parte del agente inteligente se realiza en forma secuencial. Así, el robot actúa operando sobre el ambiente de navegación en función de las modificaciones que se producen en el modelo interno que éste posee del entorno. Estas técnicas clásicas implican esquemas de razonamiento en alto nivel

en base a resultados intermedios que emergen de los cambios experimentados en el ambiente de navegación del robot. Este paradigma, si bien permite afrontar requerimientos de desempeño complejos por parte del agente, tiene una velocidad de procesamiento de la información sustancialmente menor que con las técnicas reactivas [3]. Una de las técnicas clásicas ampliamente utilizada es el Aprendizaje Automático (AA) [2].

En base a ello, el presente trabajo consiste en combinar los métodos de procesamiento reactivo y de planificación autónoma de tareas, a efectos de diseñar una técnica de navegación híbrida que optimice el desempeño del robot en su ambiente en términos de evitar colisiones con obstáculos, alta velocidad de respuesta, optimización de las trayectorias y logro de los objetivos.

Para medir la performance de los paradigmas se desarrolló un proceso de simulación donde se debe considerar un modelo tanto del robot como del entorno en el cual pueda interactuar este.

II. MODELO DE ROBOT Y DEL ENTORNO

En el modelo de robot móvil utilizado para la simulación, se lo ha dotado al mismo con cuatro sensores de proximidad (S1, S2, S3, S4), uno en cada una de sus caras, dos sensores internos de posición (P1, P2) y dos motores (M1, M2) que comandan dos ruedas laterales (Fig. 1).

El modelo del ambiente se representa en la Fig. 2 y en el mismo los casilleros llenos denotan obstáculos y no pueden ser ocupados por el robot. Los casilleros libres o en blanco denotan posibles posiciones para recorrer. Las posiciones quedan identificadas mediante el sistema de coordenadas que se indica en dicha figura. Como hipótesis, los límites

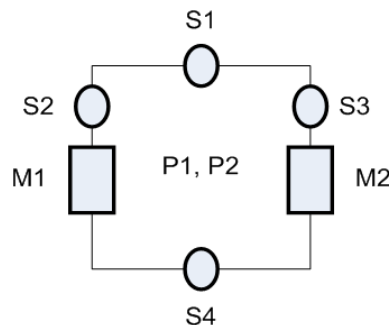


Fig. 1. Esquema Sensorial del Robot

0,0	1,0	2,0	3,0	...	n,0
0,1	1,1	2,1	3,1	...	n,1
0,2	1,2	2,2	3,2	...	n,2
0,3	1,3	2,3	3,3	...	n,3
0,4	1,4	2,4	3,4	...	n,4
...
0,m	1,m	2,m	3,m	...	n,m

Fig. 2. Esquema general del ambiente

físicos del entorno se consideran también obstáculos para el robot.

Internamente el robot posee dos sensores de posición P1 y P2, los cuales contienen las coordenadas horizontal y vertical respectivamente, correspondientes a la posición del robot en el ambiente. Los sensores S1, S2, S3 y S4 tienen como función detectar la proximidad de objetos. Sus posibles lecturas son

- 1: hay proximidad de un objeto. Se considera que un objeto está próximo cuando está a “dos casilleros” o menos respecto de la ubicación del sensor.
- -1: no hay proximidad de objetos. Se considera que un objeto no está próximo cuando está a más de dos casilleros (o posiciones) respecto del sensor.

En cuanto a los motores, los valores asignados a los mismos tienen como efecto considerar en principio la posibilidades de: “avanzar”, “Girar a derecha”, “Girar a Izquierda” y “Retroceder”, sin descartar en un desarrollo posterior los demás grados de giros intermedios a estos [4] [5].

La estructura del robot con los sensores de proximidad por ultrasonido puede verse en la Fig. 3.

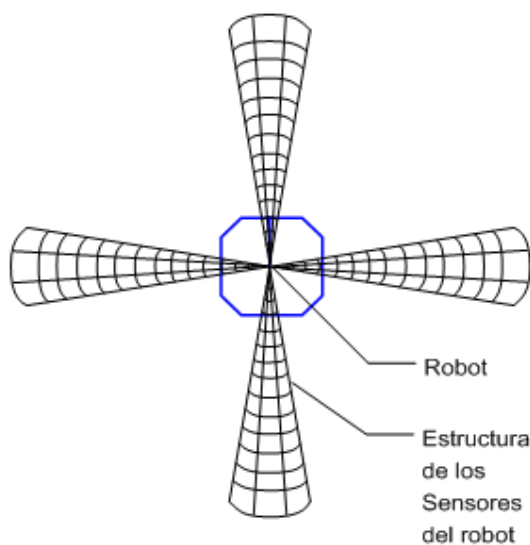


Fig. 3. Estructura de los sensores de proximidad del robot

III. EXPERIMENTACIÓN DE TÉCNICAS EXISTENTES BÁSICAS

A. Introducción

Como punto de partida se experimentaron las dos técnicas más conocidas dentro de la IA, como lo son las RNA y el AA. Así, en función de los resultados obtenidos por cada técnica en forma independiente, se pudieron apreciar las mejores características de cada una de ellas, que sirvieron como base para la técnica híbrida desarrollada.

B. Diseño de un Entorno de Simulación

Para facilitar la experimentación y la comprensión del comportamiento de las diferentes técnicas a utilizar, se diseñó un entorno simulado de movimiento del robot (Fig. 4).

Este entorno permite una amplia variedad de tipos de experimentación y posee fácil adaptabilidad al tipo de técnica que se utilice. Fue desarrollado mediante el programa Matlab.

El objetivo propuesto para los diferentes experimentos fue que el robot alcance un determinado punto del mapa evitando los obstáculos que se le presenten en el entorno. De este modo son manejadas las situaciones de colisión y de lazo cerrado, entendiéndose a esta última como una situación en donde el robot ejecuta una serie de acciones que lo llevan a una trayectoria de circuito cerrado, en la cual no es posible alcanzar el objetivo planteado sin la utilización de un plan de contingencia para tal situación.

C. Función de Aptitud

Para poseer un parámetro directo de la performance alcanzada por cada paradigma se utiliza una función de aptitud (AP) que tenga en cuenta los diferentes parámetros que influyen en el adecuado funcionamiento del robot.

$$Ap = \frac{10 \cdot N_{exi}}{N_{exp}} - \left(\frac{10 \cdot N_{choq} + N_{lc}}{10 \cdot N_{exp}} \right) \quad (1)$$

donde

- N_{exp} : Número de experimentos.
- N_{exi} : Número de éxitos, es decir, veces que el robot llegó a la meta.
- N_{choq} : Número de choques.
- N_{lc} : Número de situaciones de lazo cerrado.

La función de aptitud diseñada (1), para este caso como una función de costos, indica que a mayor valor “Ap” (cuyo valor máximo es diez), mejor será la performance del paradigma que se ha utilizado. En ella se da mayor peso al número de ocasiones en que fue alcanzada la meta con éxito, ya que este es el objetivo principal del problema planteado en la simulación. Además, todos los factores se han ponderado con el número total de experimentos realizados, siendo mejor el paradigma que maximice a dicha función.

Se generaron 30 posiciones aleatorias dentro de las posibles, es decir, del conjunto de casillas que no sean

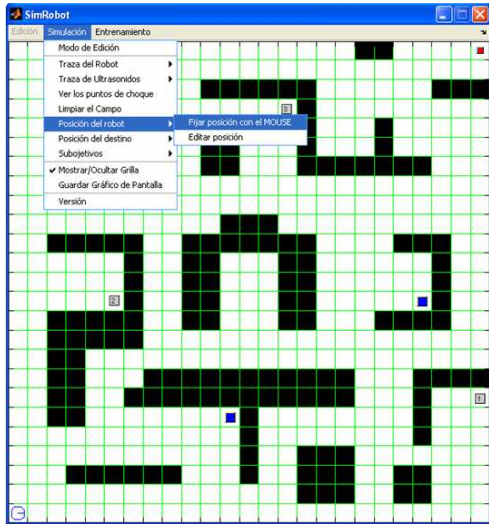


Fig. 4. Simulador diseñado en Matlab

obstáculos (utilizadas luego para todos los experimentos). De este modo se cuenta un conjunto de 30 posiciones iniciales, desde las cuales se iniciará la simulación para medir la capacidad de los paradigmas de cumplir su objetivo.

D. Planes de Contingencia

Para todos los casos en los cuales la salida generada por el paradigma utilizado indique al robot una acción tal que este produzca situaciones de colisión o lazo cerrado se utiliza un algoritmo que permita al agente robótico encontrar una alternativa adecuada para continuar la tarea objetivo fijada. Estos algoritmos son denominados como “Planes de Contingencia”.

Luego de la ejecución del Plan de Contingencia, se debe utilizar esta información para las situaciones futuras en las que el robot pase nuevamente por esa posición. Así, en dicha posición se recurre directamente a la solución del plan de contingencia, almacenada en la memoria, descartándose la salida presentada por el paradigma utilizado.

Puede verse que en el caso de una situación de posible colisión en la posición (X_i, Y_j) , siendo la dirección prop-

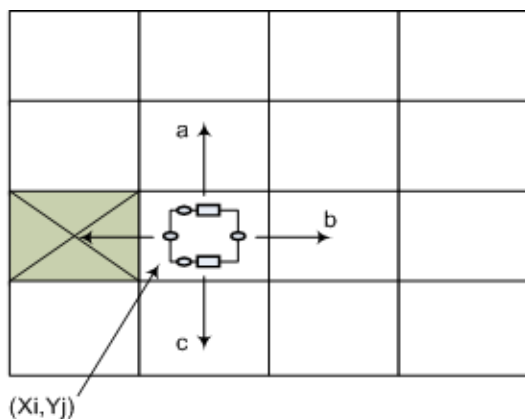


Fig. 5. Alternativas en caso de situación de colisión

uesta por el paradigma utilizado la indicada en la Fig. 5, el Plan de Contingencias debería seleccionar bajo determinado criterio una de las tres posibles alternativas (a) (b) o (c). De este modo el Plan de Contingencias resolvería esta situación en un caso futuro en base a la información almacenada en la memoria.

E. Experimentación con Redes Neuronales Artificiales

La RNA utilizada es una red multicapa con 12 neuronas en la capa oculta y 2 en la capa de salida, como puede verse en la Fig. 6. La misma utiliza el algoritmo de retropropagación (Backpropagation).

Para alcanzar una arquitectura de red adecuada a las exigencias del problema, se realizaron sucesivas modificaciones tanto en la cantidad de capas ocultas como en el número de neuronas de las mismas hasta encontrar una de las mejores estructuras de red (entre las probadas), sin descartar la posibilidad de la existencia de una mejor.

Esta red es entrenada con una serie de trayectorias que unen un punto inicial (de partida) y un punto final (de llegada). La red es entrenada y simulada para cada una de las trayectorias de entrenamiento por separado.

Además, se realizaron experimentos para observar la capacidad de entrenamiento de la RNA. En ellos se utilizaron trayectorias múltiples. Como resultado de esta experimentación pudo verse la dificultad de convergencia que tienen las RNA, mostrando así las limitaciones de las mismas cuando deben aprender una gran cantidad de información. La RNA no soporta el entrenamiento con trayectorias múltiples, es decir, trayectorias compuestas de dos o más trayectorias simples, dado que pueden presentarse, por ejemplo, patrones de entrenamiento con información contradictoria

F. Experimentación con Aprendizaje Automático

Se puede decir que el AA es un campo que se dedica al desarrollo de métodos computacionales para los procesos de aprendizaje, y a la aplicación de los sistemas informáticos de aprendizaje a problemas prácticos [6].

En este caso por medio del AA se genera un árbol de decisión, en el cual, en función de los estados presentados por los diferentes sensores se obtiene la “clase” a la cual pertenece dicho estado tomándose así la acción que deberán ejecutar los motores del robot.

Para generar un árbol de decisión, generalmente se utiliza el algoritmo ID3 de Quinlan [2]. Éste solo permite clasificar datos pertenecientes o no a una clase o categoría (Verdadero, Falso), es decir genera árboles binarios. En nuestro caso, requerimos de varias clases de salida. Para estos casos Quinlan recomienda el uso de un número de árboles en paralelo igual a la cantidad de clases que se requieran [2] [9]. Así, cada árbol indica si dicho dato pertenece o no a su clase. Este tipo de estructura posee varias falencias como lo son el alto consumo computacional y las contradicciones en caso de que dos o más árboles indiquen que el dato pertenece a su clase. En la navegación robótica no pueden existir contradicciones del tipo “Girar a la izquierda” y al mismo tiempo “Girar a la derecha”. Debido a esto, el ID3 ha sido modificado por los autores

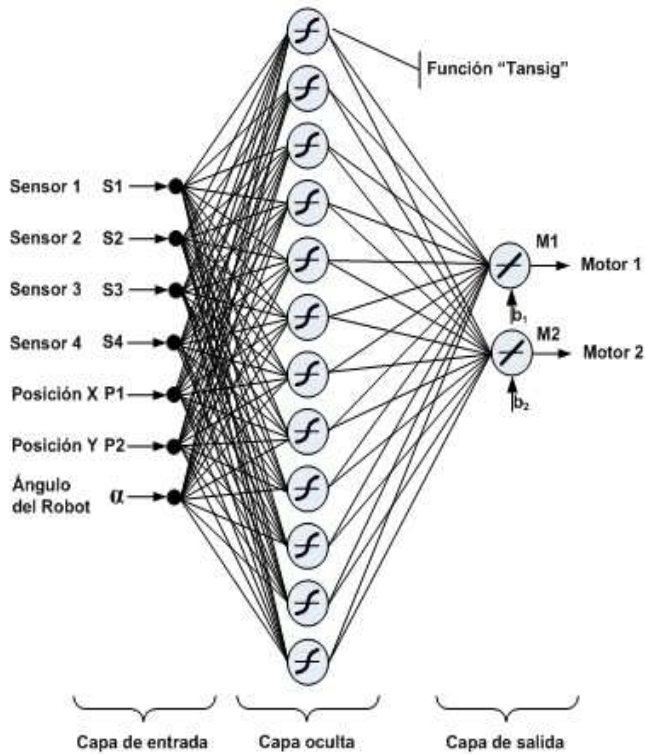


Fig. 6. Estructura de RNA utilizada

para que se puedan clasificar adecuadamente los datos en varias clases de salida y con la utilización de un solo árbol de decisión. Este nuevo algoritmo ha sido llamado ID3M.

De manera similar a como se entrena a las RNA se deben utilizar un conjunto de patrones de entrenamiento (mediante una trayectoria o un conjunto de ellas) para realizar la etapa de entrenamiento del AA.

G. Análisis Comparativo entre RNA y AA

Los resultados presentados por la RNA y el AA entrenados con 4 trayectorias simples, y el AA con una trayectoria múltiple (compuesta por las 4 trayectorias simples) pueden verse en la Tabla I.

TABLA I
RESULTADOS PRESENTADOS POR LA RNA Y EL AA

	Números de exitos	Números de choques	Números de lazos cerrados	Ap
RNA promedio	25	36	524	5,36
AA promedio	27	66	20	6,58
AA con multiples trayectoria	30	43	9	8,54
Cantidad de experimentos por trayectoria				30

Queda expresada en los resultados la superioridad del paradigma de AA frente a la RNA, básicamente el AA, cuenta con una navegación que posee una cantidad significativamente menor situaciones de lazo cerrado que la RNA y una cantidad mayor de éxitos. Esto implica que se tiene un robot cuya navegación resulta más fluida y exploradora por el ambiente. Pese a esto, es importante observar que aún el número de colisiones y situaciones de lazo cerrado siguen siendo elevados en relación al número de ocasiones en que fue alcanzada la meta con éxito.

IV. TÉCNICA HÍBRIDA

A. Introducción

Por medio de los paradigmas actuales y como se ha venido desarrollando, no se alcanzan niveles de performance adecuados para el desempeño del robot en su entorno. Es por tales motivos, que se busca una técnica que utilice los diferentes tipos de arquitecturas para el control de la autonomía de agentes robóticos móviles de un modo tal, que permita obtener un desempeño superior al de cualquiera de las técnicas por separado.

La técnica híbrida desarrollada utiliza un esquema de trabajo "modular", con un modulo central, que puede estar dominado por un paradigma de Aprendizaje Automático, y un conjunto de módulos que trabajan bajo el paradigma que mejor aptitud presente para el funcionamiento característico de cada uno de ellos [7]. Cada módulo tendrá asignado un comportamiento o función dentro de las diferentes posibilidades que deba desarrollar el robot en el medio.

En este enfoque la idea fundamental es separar eficientemente los comportamientos de navegación, teniendo así, una capa de conductas básicas y otra de conductas más abstractas [8], dado que esta estructura es la que se encuentra presente en la naturaleza y en particular en el ser humano. Correspondiendo así, los comportamientos más básicos a la RNA y los superiores a la capa de AA (Fig. 7) serie de casos en donde debe corregir su curso en base a una desviación entregada por la capa superior (a modo de orientación) sin colisionar en el trayecto, como se detallará más adelante.

Con esta estructura se separa la información proveniente del entorno para disminuir la complejidad de los datos, observándose un incremento sustancial en la velocidad de convergencia de la RNA y en la generación del árbol de AA con respecto a las técnicas por separado.

B. Estructura para el entrenamiento

El entrenamiento de este enfoque se realiza mediante dos etapas.

1. *Entrenamiento de la primera capa*, para cumplir la conducta básica de "Corregir la trayectoria en función de la desviación y evitar las colisiones". Se genera un árbol de inducción temporal que debe poseer la información necesaria para la orientación que se utilizará para entrenar la RNA, como se puede ver en la Fig. 8.

Luego la RNA debe ser entrenada mediante una serie de trayectorias básicas tal que, pese a la orientación del ángulo indicada por el AA, el robot posea la capacidad de

TABLA II
FUNCIÓN DE APTITUD DE TÉCNICA HÍBRIDA USANDO UNA
MULTITRAYECTORIA

Número de éxitos	Número de choques	Número de lazos cerrados	Ap
30	0	386	8,71
Cantidad de experimentos por trayectoria			30

evitar colisionar contra los obstáculos y llegar al final de dichas trayectorias. Como se puede ver en la Fig. 9, las trayectorias de entrenamiento de la RNA están en concordancia con las orientaciones (Fig. 8), pero difiriendo en el caso en que la orientación indique avanzar y exista un obstáculo. En esta situación la RNA no indicará avanzar, y elegirá un camino alternativo para el robot en el cual exista una desviación mínima de la orientación objetivo. Este entrenamiento es independiente del mapa en donde interactúe el robot en la etapa de funcionamiento.

2. Posteriormente es entrenada la *segunda capa de la IA* mediante un conjunto de trayectorias a almacenar, para alcanzar los objetivos principales en el entorno donde interactuará el robot. Se conservará para ello, solamente la RNA entrenada anteriormente y se generará un nuevo árbol de inducción para el AA con la/s trayectoria/s a utilizar en la etapa de funcionamiento, las cuales dependerán del mapa. Como es el ejemplo mostrado en la Fig. 10

C. Resultados Obtenidos con la Técnica Híbrida

Los resultados presentados por la Técnica Híbrida pueden verse en la Tabla II.

De estos resultados se desprende una performance superior a la mostrada por cualquiera de las técnicas básicas por separado.

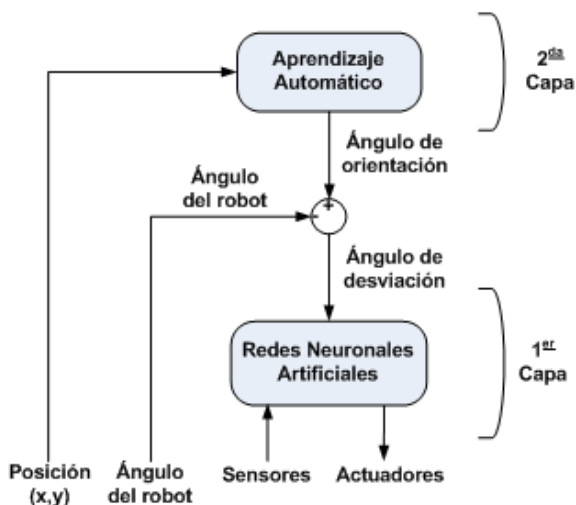


Fig. 7. Esquema para la hibridación

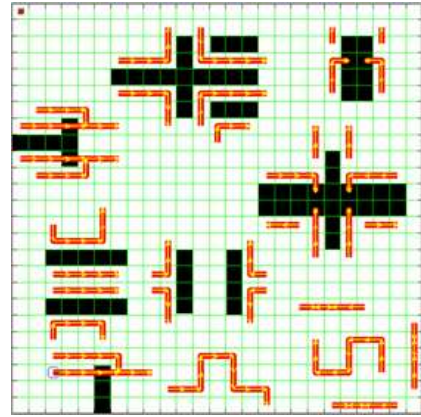


Fig. 8. Orientación que recibirá la RNA

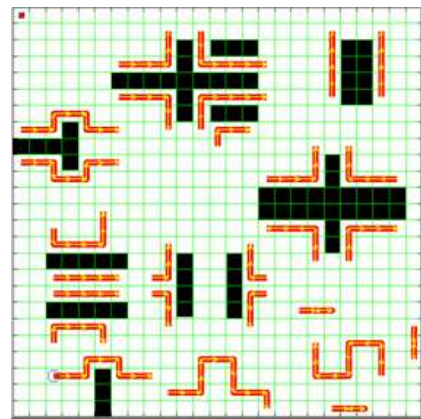


Fig. 9. Trayectorias ejemplo que recibe la RNA

D. Comparación General de las Técnicas

Se tuvieron en cuenta para la comparación los siguientes parámetros,

Precisión: Se mide en Porcentaje de llegadas, cantidad de lazo cerrado y de colisiones, en base a una función de aptitud desarrollada para tal motivo.

Eficiencia: Se mide en función a la cantidad de tiempo y recursos que un sistema necesita para alcanzar un buen desempeño. Así, debemos evaluar los requerimientos computacionales. Se tienen en cuenta para ello:

- Velocidad de convergencia
- Uso de la memoria (a menor uso de la misma mayor eficiencia)
- Tamaño de código (líneas de programa)

Comprensión: Este parámetro de comparación se refiere al entendimiento que se puede obtener sobre su funcionamiento interno y conceptos aprehendidos.

Robustez: Se refiere sobre su capacidad de manejar situaciones no previstas. Se tienen en cuenta para ello:

- Complejidad de la información manejada
- Capacidad de múltiples trayectorias

De lo expuesto en la Tabla III puede verse la superioridad de la Técnica Híbrida desarrollada teniendo en cuenta los diferentes parámetros que afectan el desempeño general del robot.

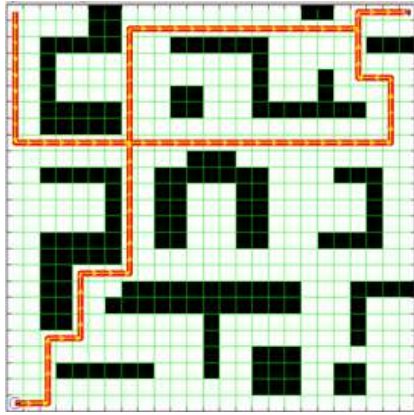


Fig. 10. Ejemplo de dos trayectorias para generar el árbol de AA de la 2ª Capa

TABLA III

COMPARACIÓN GENERAL DE LAS TÉCNICAS UTILIZADAS

	Robot con RNA	Robot con AA	Robot con Técnica Híbrida
Precisión	Baja	Media	Alta
Eficiencia	Media	Media	Buena
Comprensión	Baja	Media	Media/alta
Robustez	Baja	Media	Alta

La técnica híbrida presenta la información que debe manipular el robot para cumplir sus funciones separada en diferentes módulos, lo cual genera mayores velocidades de convergencia en la etapa de entrenamiento que las alcanzadas por las RNA y AA.

De forma global, la cantidad de información que debe procesar cada módulo es una fracción significativamente menor que las que debería procesar cada técnica básica por separado. Lo que proporciona mejor capacidad de generalización de las diferentes conductas que se esperan que el robot pueda manejar.

V. CONCLUSIONES

En este trabajo se han implementado en primer lugar las técnicas existentes básicas (dentro de la amplia gama de paradigmas de IA) más conocidas como lo son: Las “Redes Neuronales Artificiales” y el “Aprendizaje Automático”. A partir de los resultados obtenidos en su aplicación se diseñó una técnica híbrida, la cual combina las mejores características de desempeño de las técnicas básicas.

Como hipótesis simplificada fuerte para el entrenamiento del robot se supone el conocimiento de parte de la información del mapa. En el caso real, esta información se obtendría luego de dejar al robot explorando durante un lapso de tiempo, y a partir de ello, en base a diferentes técnicas de búsqueda hallar trayectorias adecuadas para el entrenamiento.

Para poder evaluar la performance de las técnicas diseñadas respecto a las técnicas experimentadas ante-

riormente, se realizaron una serie de experimentos que corroboraron un mejor desempeño de la técnica híbrida realizadas por sobre las otras técnicas.

Al separar la IA en capas, se logra que cada uno de los módulos tenga que manejar objetivos en diferentes niveles de conducta, simplificando así la cantidad de información que debe procesar cada módulo por separado. Se parte desde las conductas básicas, como “evitar colisión” función de la primer capa, hacia conductas más complejas como en el caso de la segunda capa es buscar la orientación necesaria para alcanzar el objetivo fijado, así como también, el plan de contingencia que se debe tomar en caso de ser necesario.

A partir de lo expuesto y dado que todas las experimentaciones fueron realizadas bajo las mismas hipótesis, la técnica híbrida, superó a las otras técnicas utilizadas, en el cumplimiento de los objetivos propuestos.

VI. FUTUROS TRABAJOS

Como una posterior mejora se propone una ampliación de la técnica híbrida desarrollada, mediante el agregado de diferentes complementos para que el robot pueda resolver situaciones más complejas y con mayor eficiencia. Se propone el agregado de una tercer capa que sea la encargada de administrar los objetivos a cumplir. Así esta capa le informará a la segunda capa a que zona debe ir. También será la encargada de controlar el abastecimiento de energía y el cumplimiento de subobjetivos. Éstos pueden ser, por ejemplo, el itinerario que debe resolver el robot dentro de una fábrica en forma cíclica.

Además se plantea la mejora de los planes de contingencia mediante la expansión del algoritmo actual.

También se han analizado nuevos submódulos que serán incluidos dentro de las capas de IA del Robot tales como un generador de nuevas trayectorias, un actualizador del estado del mapa (proporcionándole al robot una mayor capacidad de desenvolverse en ambientes dinámicos), entre otros.

REFERENCIAS

- [1] John J. Braig, “Introduction to robotics”, Mechanics and control. Addison-Wesley, 1986.
- [2] Tom M. Mitchell, “Machine Learning,” McGraw-Hill Science/Engineering/Math, ISBN: 0070428077, March 1977.
- [3] Russell, Norvig, “Inteligencia Artificial, un enfoque moderno”, Prentice Hall, 1996.
- [4] K. S Fu, R. C. González, G. S. G. Lee, “Robótica. Control, detección, visión e inteligencia”, McGraw-Hill, 1988.
- [5] R. A. Brooks, “A Robust Layered Control System for a mobile Robot”, A. I. Memo 864. MIT, AI Lab. Setiembre 1985.
- [6] R. A. Brooks, “Achieved Artificial Intelligence”, A. I. Memo 899. MIT, AI Lab. Setiembre 1986.
- [7] J. H. Connel, “SSS: A Hybrid Architecture Applied to Robot Navigation”. Conf. on Robotics and Automation (ICRA-92), pp. 2719-2724. Proc. of the 1992
- [8] Ronald C. Arkin, “Behavior-Based Robotics”, The MIT Press, 2000.
- [9] J. R. Quinlan, “Decision trees and multi-valued attributes”, En J. E. Hayes, D. Michie y J. Richards, Machine Intelligence, Vol. II, pp. 305-318. Oxford University Press, Oxford, UK, 1988.