

Agentes Robóticos Autónomos Basados en Redes Neuronales Artificiales: Análisis de Resultados Experimentales en Conductas de Navegación

¹Enrique Sierra, Alejandro Hossian, Gustavo Ramos, Roberto Carabajal Cesar Echeverria, Gustavo Monte

Grupo de Inteligencia Artificial
Regional Académica Confluencia
Universidad Tecnológica Nacional
Pedro Rotter s/n – (8317) – Plaza Huincul
Neuquen – Argentina – Telefax: 54-299-4963292
Contacto: ¹enriquesie@yahoo.com.ar, ¹hossi@jetband.com.ar
Paper received on 11/08/08, accepted on 10/09/08.

Resumen. La navegación robótica es un área muy importante de investigación y aplicación de la Inteligencia Artificial en el campo de la Robótica Cognitiva. El objetivo principal de este artículo es presentar resultados de investigación en técnicas de navegación robótica basadas en redes neuronales, las cuales concentran las mejores características del paradigma reactivo concerniente a navegación autónoma de robots. En el presente artículo se evalúa la efectividad y el desempeño de este paradigma, obteniendo conclusiones respecto de las formas más recomendables en cuanto a su implementación en términos de la conducta deseada del robot.

1 Aspectos Generales de la Navegación Robótica

Un paradigma está constituido por los supuestos teóricos generales, que mediante su aplicación caracterizan el modo de resolver problemas en robótica. Normalmente, el paradigma constituye un enfoque, una forma de abordaje de un determinado problema, el cual está caracterizado por la utilización de una serie de técnicas orientadas a su solución. Básicamente, las técnicas son los procedimientos o conjuntos de éstos usados para resolver diferentes clases de problemas que tienen como objetivo obtener un comportamiento determinado para el robot bajo cierto paradigma. Los paradigmas existentes para la navegación robótica son esencialmente [1]:

- Deliberativo: Basado en conocimiento
- Reactivo: Basado en comportamiento
- Híbrido: Deliberativo-Reactivo

1.1 Enfoque Deliberativo

Es el primer paradigma propuesto para la navegación robótica (1967) [2]. Este se basa en una visión introspectiva (es decir, en una observación interior de los propios actos o estados de ánimo o de conciencia) de sobre cómo pensamos los humanos y tomamos decisiones. Por ejemplo: si se ve una puerta, se decide pasar a través de ella.

Las funciones de un robot pueden ser divididas en tres categorías generales, tal como se ilustra en la Figura 1:

- *Sensado*: Esta función adquiere la información proveniente de los sensores y produce una salida filtrada que resultará útil a las otras funciones (lo que ayuda a la constitución del modelo del mundo).
- *Planeado*: Partiendo de la salida de la función previa y agregando información propia, tiene por objeto generar una o más tareas que serán ejecutadas por el robot (esto sería por ejemplo, un plan de acción).
- *Actuación*: Esta función es la que envía comandos de salida a los actuadores y efectores del robot, de acuerdo al plan previamente elaborado.

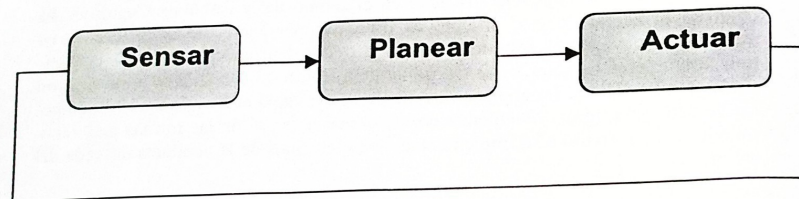


Fig. 1. Enfoque Deliberativo en navegación robótica

El robot percibe el mundo, luego planea la acción a realizar y ejecuta la misma. La información que percibe se representa en un modelo global del mundo.

1.2 Enfoque Reactivo

Este enfoque nace en oposición al enfoque deliberativo (1986) [3] y se basa en modelos biológicos y psicológicos para explicar el comportamiento visto en diversos organismos vivos. En base a un comportamiento complejo se encuentran conductas básicas del tipo "Estímulo \rightarrow Respuesta" [4].

El funcionamiento de un robot, en este caso, se organiza como un conjunto de módulos o comportamientos, cada uno de los cuales está ligado a los sensores y a los actuadores del robot. Las funciones de sensado y actuación se encuentran descentralizadas en una cantidad de comportamientos. Se considera que los comportamientos son procesos que dependen del sensado local, tal como se muestra en la Figura 2.

El robot percibe el mundo y ejecuta, en respuesta a ello, la acción que parezca más adecuada. En este caso, la información percibida no se integra en ningún modelo del mundo. Para el robot, el mundo es su mejor modelo.

1.3 Enfoque Híbrido

Surge a partir de la hibridación de los paradigmas deliberativo y reactivo (1990) [5], ya que estos presentaban por separado diversas falencias. Se intenta tomar así lo mejor de cada paradigma:

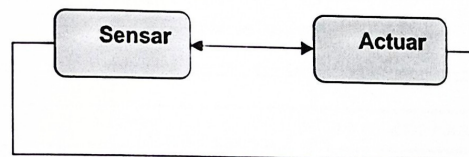


Fig. 2. Enfoque Reactivo

- Comportamiento deliberativo, orientado a metas
- Comportamiento reactivo, de rápida respuesta y adaptable a los cambios del mundo.

Se mezclan las reglas deliberativas y reactivas en dos capas de control ejecutándose en paralelo a diferentes velocidades:

Una capa **reactiva** encargada del control de las acciones de base (como, avanzar, evitar obstáculos, etc.) y una capa **deliberativa** encargada de planificar acciones más elaboradas (como, por ejemplo, dirigirse a un sitio preciso, reconocer visualmente marcas y objetos).

La información de los sensores es accesible a los módulos de comportamiento, pero también al planificador para la construcción de un modelo del mundo orientado a tareas, tal como se muestra en la Figura 3.

Un planificador traza las acciones del robot, en función de sus objetivos globales. Los planes son secuencias de acciones ejecutables por módulos programados según el esquema reactivo. Ambas capas se comunican para informarse sobre las acciones a realizar y sobre su ejecución.

2 Características del entorno de simulación

Para medir la performance del agente robótico se consideró un entorno de simulación, creado mediante la utilidad GUI de MATLAB con el que pueda interactuar un robot. Este ambiente se representa en la Figura 4. Las áreas marcadas en color

gris denotan obstáculos y no pueden ser ocupados por un 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 la misma Figura 4. Los límites del mundo se consideran también obstáculos para el robot.

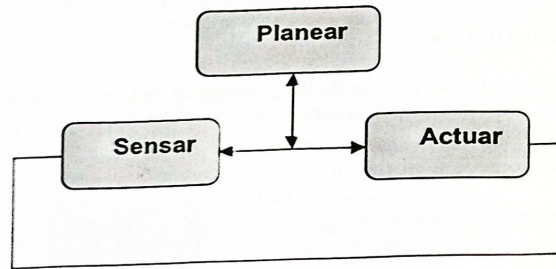


Fig. 3. Enfoque Híbrido

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. 4. Esquema General del ambiente

El agente robótico móvil utilizado para la simulación se muestra en la Figura 5. Está dotado de un sensor de proximidad en cada una de sus caras, dos sensores internos de posición y dos motores que comandan dos ruedas laterales.

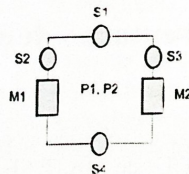


Fig. 5. Esquema Sensorial del robot

Internamente el robot posee dos sensores de posición P1 y P2, los cuales contendrán 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 dos posiciones) o menos respecto de la ubicación del sensor, en el mundo del robot.
- -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 el mundo del robot.

En cuanto a los motores, los valores asignados a los mismos tienen el siguiente efecto, ilustrado en la Tabla 1:

Tabla 1. Efecto de los valores asignados al Motor

M1	M2	EFEECTO
-1	-1	Desplazamiento del robot una posición hacia atrás
-1	1	Desplazamiento del robot una posición hacia la izquierda
1	-1	Desplazamiento del robot una posición hacia la derecha
1	1	Desplazamiento del robot una posición hacia delante

3 Experimentación con redes neuronales artificiales

Para la experimentación empleando el paradigma reactivo se analiza como primer ambiente de simulación el de la Figura 6. Los casilleros o posiciones cruzados por la línea componen la trayectoria del robot que se emplea para encontrar los patrones de entrenamiento de la red neuronal que actúa como "cerebro" del robot. En esta trayectoria, el robot parte de la posición (0,9) y llega a la posición (9,0). La orientación del robot cuando parte de la posición (0,9) es la indicada en la Figura 7, mientras que la red neuronal utilizada para programar el comportamiento reactivo del robot puede apreciarse en la Figura 8.

3.1 Procedimiento para la experimentación

Para la construcción de un mapa sensor-motor [6] para el robot con la trayectoria indicada, se arma la matriz P (que contiene los datos de los sensores y la posición), y la matriz T (que contiene la salida asignada al estado sensorial almacenado en la respectiva fila de P).

Para facilitar la experimentación y la comprensión del comportamiento de la red, se diseñó un programa de simulación para el tipo de robot cuya descripción se efectuó anteriormente en forma conjunta con su ambiente. El programa se diseñó mediante el recurso GUIDE del software MATLAB versión 7.0. En la Figura 9 puede observarse un aspecto general del entorno con su interfase de usuario y en la Figura 10 la respuesta general del robot cuando se le coloca en su trayectoria de entrenamiento.

$$P = \begin{bmatrix} S1 \\ S2 \\ S3 \\ S4 \\ P1 \\ P2 \end{bmatrix} = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & -1 & 1 & 1 & -1 & -1 & -1 & -1 & -1 & -1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 2 & 3 & 4 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 6 & 7 & 8 & 9 \\ 9 & 9 & 9 & 9 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (1)$$

$$T = \begin{bmatrix} M1 \\ M2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & -1 & 1 & 1 & 1 & 1 \end{bmatrix} \quad (2)$$

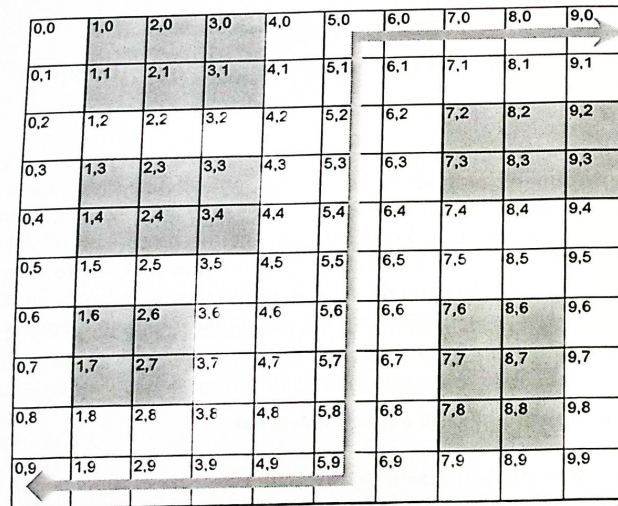


Fig. 6. Esquema del ambiente utilizado para simular la respuesta del Robot

En el caso de que la red de tipo Perceptrón obtenida como resultado del entrenamiento verifique correctamente el patrón utilizado, se procede a la simulación del

comportamiento del robot haciendo que éste inicie su recorrido en diferentes posiciones. Así, luego se grafican las trayectorias obtenidas. La primera posición de inicio para simular es la (9,9), estando el robot con la orientación ilustrada en la Figura 11. En las Figura 12 se muestra el resultado de la simulación y en la Figura 13, el resultado de colocarlo en la posición (0,0) como punto de partida inicial.

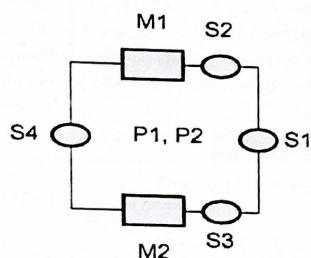


Fig. 7. Posición inicial del robot

A partir de la observación de la conducta del robot en las situaciones descritas por las Figuras 12 y 13 puede observarse que el robot busca la trayectoria de entrenamiento y una vez que la encuentra, la sigue hasta la posición-objetivo. Por lo tanto puede asumirse que en estos dos casos analizados, el comportamiento del robot es el esperado. Sin embargo, en la Figura 14 puede observarse que al colocarlo en la posición de partida (4,2), el robot colisiona.

3.2 Análisis de los resultados del experimento

Como se ha podido observar el patrón de entrenamiento es suficiente para algunas de las posibles trayectorias, pero en general es poco representativo del medio bidimensional en el cual se mueve: en 18 de sus 20 movimientos la salida es avanzar. Así finalmente, partiendo de la posición 4.2 el robot colisiona.

Además, se observó un inconveniente del entrenamiento dado que como el robot no tiene una medición de su *orientación angular* respecto del entorno, existen casos en algunas trayectorias posibles de entrenamiento, en donde se llega a patrones de entrenamiento que son contradictorios. Esto puede ser visualizado en la trayectoria de entrenamiento de la Figura 15. El punto conflictivo es el (5,5).

En la Figura 15 se puede apreciar que la única diferencia en el punto (5,5) es la orientación angular del robot respecto del entorno. Así, el robot tiene los mismos estados del sensor y de la posición con la diferencia de su orientación. Pero como la orientación no es tomada en cuenta en los patrones de entrenamiento, se ve que para

el mismo estado sensorial y la misma posición el entrenamiento asigna dos salidas deseadas mutuamente excluyentes. Los patrones de entrenamiento tomados de la misma trayectoria que son contradictorios dan como resultado que la red no pueda converger. Este hecho se debe a que el algoritmo utilizado para hallar los pesos que unen las neuronas de la red funciona ajustándolos para que la red obtenga un valor de salida (con ese patrón de entrada), que en un paso puede ser "Girar Izquierda" y en el próximo paso es "Avanzar" (ver Figura 16), con lo cual el error de convergencia se mantiene constante y la red finalmente no converge [7]. Por lo tanto, simplemente con agregar la orientación a los datos de entrada de red, se asegura la separabilidad lineal [8] de la red para las trayectorias a entrenar.

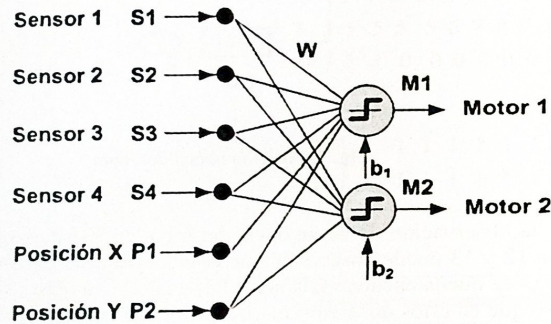


Fig. 8. Red Neuronal empleada como cerebro

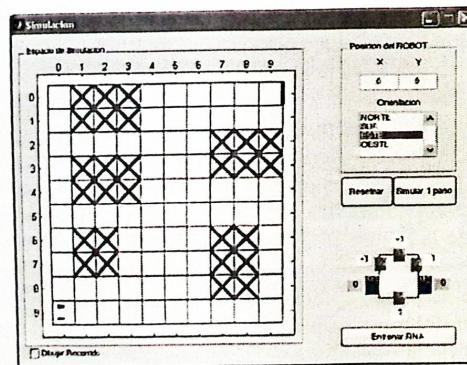


Fig. 9. Entorno de simulación

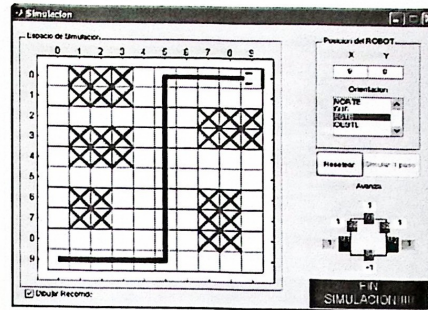


Fig. 10. Entorno de simulación

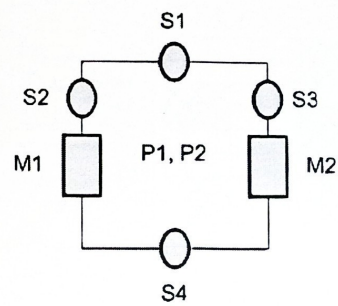


Fig. 11. Orientación Norte del robot

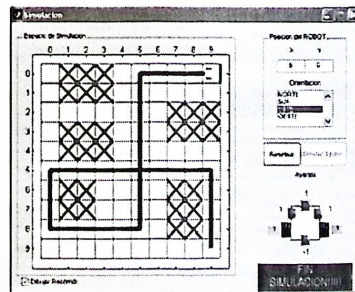


Fig. 12. Simulación partiendo de posición (9,9)

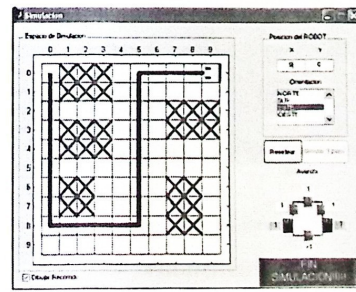


Fig. 13. Simulación partiendo de posición (0,0)

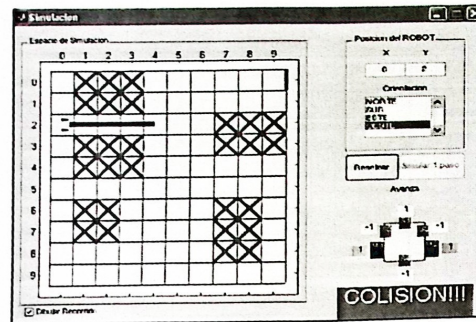


Fig. 14. Posición de inicio en (4,2): hay colisión

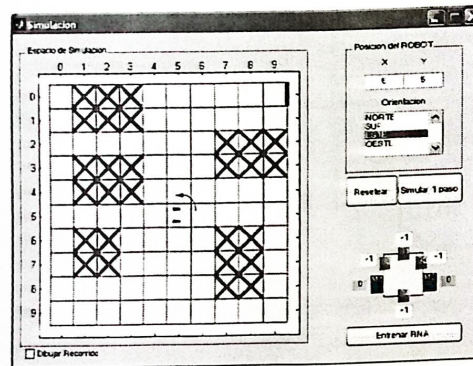


Fig. 15. Ambigüedad en el estado de los sensores

Caso	Patrón de entrada de la RNA						Salida deseada de la RNA
	S1	S2	S3	S4	X	Y	Acción a Realizar
Figura 18	-1	-1	-1	-1	5	5	Girar Izquierda
Figura 17	-1	-1	-1	-1	5	5	Avanzar

Fig. 16. Patrones de entrenamiento contradictorios

4 Conclusiones

En base a la experimentación realizada, puede concluirse que si se desea obtener una mejor performance en la navegación del robot ante las variadas condiciones de partida, cabrían las siguientes estrategias:

- 1- Agregar a los patrones de entrenamiento la orientación que el robot posee respecto al entorno para lograr la separabilidad lineal de la red para las trayectorias a entrenar.
- 2- Considerar una trayectoria de mayor complejidad, donde por ejemplo, el robot deba realizar una mayor cantidad de movimientos dentro de todas las posibilidades que este presenta.
- 3- Aumentar el número de patrones de entrenamiento (usando una trayectoria más larga y/o varias trayectorias). Cabe aclarar, que esto dependerá de las limitaciones del paradigma utilizado, ya que en el caso de las redes neuronales, puede dificultarse su convergencia.
- 4- Modificar la arquitectura de la red neuronal, para obtener:
 - a. Convergencia para los patrones de entrenamiento.
 - b. Máxima generalización, para las nuevas posibles situaciones.
- 5- Utilizar técnicas de contingencias. como accesorio a lo anterior, fundamentalmente para evitar las colisiones y lazos cerrados, es decir recorridos recurrentes en la navegación del robot.

Referencias

1. García Martínez R., Servente M., Pasquini D (2003). *Sistemas Inteligentes*. Nueva Librería. Buenos Aires.
2. Santos J., Duro R. J.(2005) *Evolución Artificial y Robótica Autónoma*. Editorial RAMA. Madrid.
3. Brooks. R. A.(1986) *Achieving Artificial Intelligence through Building Robots*. AI Memo 898. MIT, Artificial Intelligence Laboratory.
4. Brooks. R. A.(1990) *Elephants don't play chess*. In *Designing Autonomous Agents: Theory and Practice from Biology to Engineering and Back* Patti Maes (Ed.) MIT Press. Cambridge, MA
5. Mataric, M. J. (1992) *Behavior Based Systems: Main Properties and Implications* Proceedings of IEEE International Conference on Robotics and Automation.
6. Mataric, M. J.(1992) *Designing Emergent Behaviors: From Local Interactions to Collective Intelligence*. Proceedings of the Third International Conference on Simulation of Adaptive Behavior (SAB-92). MIT Press, Cambridge, MA.
7. Sierra, E., García-Martínez, R., Hossian, A., Britos, P. y Balbuena, E. 2006. *Providing Intelligent User-Adapted Control Strategies in Building Environments*. Research in Computing Science Journal, 19: 235-241.