

Análisis del comportamiento de robots móviles autónomos en base redes neuronales artificiales. Un enfoque desde el paradigma reactivo

Alejandro A. Hossian^{1*}, Gustavo E. Monte^{2*}, Verónica Olivera^{**}

¹ Universidad Tecnológica Nacional, Facultad Regional Neuquén, *alejandrohossian@yahoo.com.ar

² Universidad Tecnológica Nacional, Facultad Regional Neuquén, *gusmonte25@yahoo.com.ar; **verolivera@gmail.com

Fecha de recepción del artículo: 1/6/2011 Fecha de aceptación del artículo: 1/7/2011

Resumen

La navegación robótica es aplicable a múltiples disciplinas y entornos industriales; y en este sentido, la aplicación de la Inteligencia Artificial con tecnologías inteligentes como redes neuronales, algoritmos genéticos y aprendizaje automático entre otras cobra un gran protagonismo dentro del campo de la Robótica Cognitiva para su desarrollo. La capacidad de reacción que posee un robot cuando se encuentra ante situaciones inesperadas, constituye una de sus principales cualidades para desenvolverse eficazmente en el entorno donde este deba operar, lo cual indica el grado de autonomía que este posee.

Las consideraciones expuestas, constituyen la principal fuente de motivación que conduce a este grupo de investigación a realizar una serie de desarrollos tendientes a mejorar el rendimiento de los robots navegadores en entornos estructurados con obstáculos definidos. En esta línea, en el presente artículo se pretende comunicar los primeros desarrollos realizados y cuyo principal aporte se focaliza en evaluar el desempeño del paradigma reactivo bajo la aplicación de la tecnología de las redes neuronales con aprendizaje supervisado. En este sentido, la tecnología utilizada para llevar cabo los diferentes experimentos, permiten obtener conclusiones respecto de las formas más recomendables en cuanto a su implementación en términos de la conducta deseada del robot en su actividad de navegación. La metodología utilizada implementa un entorno de simulación que permite entrenar distintas trayectorias del robot navegador y

evaluar el rendimiento de este en fase de operación. Se exponen y se analizan los principales resultados de los experimentos realizados señalando la importancia de las características de las trayectorias de entrenamiento utilizadas y del entorno de operación del robot. Asimismo, las conclusiones obtenidas destacan las principales bondades y limitaciones del paradigma aplicado, así como también las futuras líneas de investigación que continúa desarrollando este grupo con idea de comunicarlas y ponerlas a consideración en vuestro congreso.

Palabras clave

Mecanismo de Aprendizaje (LearningType), Navegación Autónoma (Autonomous Navigation), Redes Neuronales (Neural Networks), Robótica Cognitiva (Cognitiv Robotics).

Introducción

El objetivo del presente trabajo está focalizado en presentar resultados de investigación en técnicas de navegación robótica basadas en redes neuronales con aprendizaje supervisado, las que, en opinión de gran parte de la comunidad científica de esta área temática, integran las características más potentes del paradigma reactivo concerniente a la navegación autónoma de robots [1], [2] y [3]. El trabajo se encuadra en un campo un poco más amplio y al cual se lo considera una evolución del paradigma reactivo, y que se lo denomina "Aproximación Basada en Comportamientos" [4], entendiéndose que los

comportamientos describen la forma en que se reacciona (en este caso un sistema robot) ante un determinado estado de los sensores, y las actuaciones se determinan de manera más elaborada que una simple consulta a una tabla de correspondencias y realizando cálculos a los efectos de decidir que acción se debe llevar a cabo.

A modo de establecer una pequeña reseña acerca del estado del arte en el tópico, cabe hacer mención a algunos antecedentes históricos que han marcado líneas significativas en lo que respecta a las diferentes formas de comportamiento que presentan los vehículos autónomos, como ser los vehículos de Braitenberg [5], quien diseña una serie de vehículos los cuales mediante una serie de conexiones excitadoras e inhibitoras, y permitiendo un enlace directo entre los sensores y motores correspondientes, se demostraba la capacidad de obtener comportamientos de alta complejidad. También son de destacar las investigaciones llevadas a cabo por Rodney Brooks [6] referidas a las “*Arquitecturas Subsumidas*”, por Pattie Maes [7] quien llevó adelante investigaciones relacionadas con los llamados “*Agentes de Competencia*” y Maja Mataric [8] quien llevó a cabo importantes trabajos de control en una colectividad de agentes o robots que hacen uso de las estrategias de Brooks. En palabras de Pattie Maes, la “*Aproximación Basada en Comportamientos*” no solo es aplicable al campo de la robótica autónoma, sino también para aquellos casos en los cuales un sistema deba realizar varias tareas en forma autónoma dentro de un entorno dinámico, como es el caso de agentes que participan en la planificación de procesos.

En esta línea introductoria, el sistema robot que se utiliza para simular la actividad de navegación en el entorno propuesto, debe poseer algún tipo de arquitectura cognitiva para su control que le permita establecer vinculaciones entre su sistema sensorial y las acciones que toma sobre dicho entorno. La arquitectura empleada en el robot navegador que se presenta en esta propuesta es de tipo reactiva, cuyos principales aspectos se exponen en la siguiente subsección.

Arquitecturas reactivas

Las arquitecturas de este tipo (también llamadas *puramente reactivas* en la literatura) se basan en modelos biológicos para explicar el comportamiento observado en distintos organismos vivos e implementan estrategias de control soportadas por una colección de pares *condiciones – acciones*. La respuesta a estímulos es reflexiva, no regulada por procesos de carácter deliberativos de ningún tipo y los movimientos del robot se guían únicamente a partir de la información que se encuentra presente en ese momento en los sensores. Por consiguiente, las acciones del robot se basan en un acoplamiento directo entre sensores y actuadores mediante bucles rápidos de realimentación. Las características principales que presentan estas arquitecturas de control son las siguientes:

Estas arquitecturas presentan mayor velocidad en el procesamiento de información que las correspondientes al enfoque basado en conocimiento; no obstante la gran velocidad de respuesta que poseen estas arquitecturas, las mismas presentan importantes limitaciones cuando el robot debe abordar tareas que requieren planificación. Estas arquitecturas se basan en modelos de procesamiento de información “*masivamente paralela*”, como por ejemplo, las redes neuronales artificiales (RNA).

En función de estas características, se puede inferir que el robot percibe el mundo y ejecuta la acción que le parezca más adecuada; en otros términos, la información percibida por el robot no se integra en ningún modelo del ambiente. Es decir, el ambiente de operación es su mejor ambiente.

La Figura 1 ilustra las arquitecturas de control de tipo reactiva.

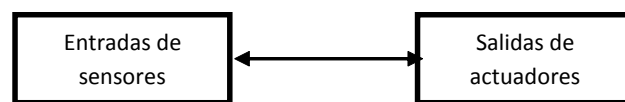


Figura 1. Arquitectura reactiva.

Arquitectura de la Red Neuronal propuesta y tipo de robot

A los efectos de evaluar el rendimiento del robot móvil en un ambiente de simulación computarizado y bajo las premisas que caracterizan al paradigma reactivo, se considera un robot navegador que desarrolla un aprendizaje del entorno en base a la tecnología de redes neuronales artificiales; en particular, se considera una configuración de red correspondiente a un modelo de perceptrón simple con una capa de entrada y una capa de salida, cuya arquitectura se puede observar en la Figura 2 [9].

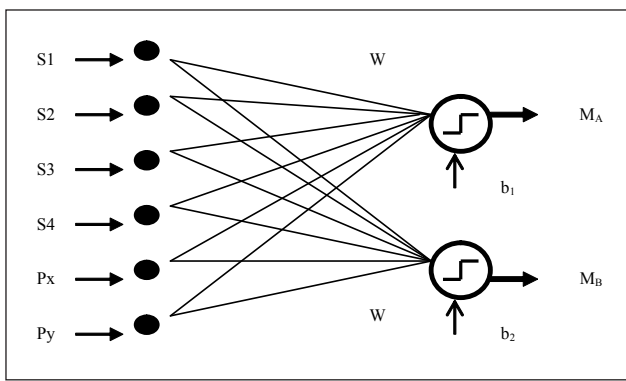


Figura 2. Arquitectura de la red neuronal propuesta como cerebro del robot

Esta arquitectura está conformada una capa de entrada y una de salida; la capa de entrada está compuesta por seis neuronas: cuatro de ellas corresponden a sensores de proximidad (S1, S2, S3 y S4), y dos a neuronas que hacen referencia a la posición según el eje x y según el eje y que ocupa el robot en el ambiente en ese momento (Px y Py). Por su parte, la capa de salida está conformada por dos neuronas, correspondientes a los dos motores que comandan las dos ruedas laterales (M_A y M_B). En la próxima sección se explica con mayor grado de detalle el esquema “*sensor – motor*” del robot y como interactúa este con el entorno.

Materiales y métodos

Para llevar a cabo las investigaciones que se presentan en este trabajo, se utilizó hardware Pentium 5 y el paquete de software MATLAB (versión 7.0) con

soporte del correspondiente toolbox de redes neuronales. Con los insumos a los que se hizo referencia, se procedió a implementar un método de trabajo para llevar a cabo la investigación, el cual consistió en el desarrollo de un entorno de navegación con obstáculos definidos en el mismo, y en el marco de un ambiente de desarrollo presentado en forma de menú que permite entrenar la red neuronal propuesta para luego colocar al robot en fase de operación. En figura 3 se observa un esquema correspondiente al entorno de simulación en donde navega el robot, en el cual se puede observar la información referida a la localización del robot (proporcionada por las coordenadas X e Y que en este caso corresponden a 5 según X y 5 según Y), su orientación (dada por **Norte**, **Sur**, **Este** y **Oeste**), el entorno en sí con la ubicación de sus obstáculos y las opciones para entrenar la red, reiniciar el proceso cuando sea necesario y simular los pasos de la trayectoria seguida por el robot.

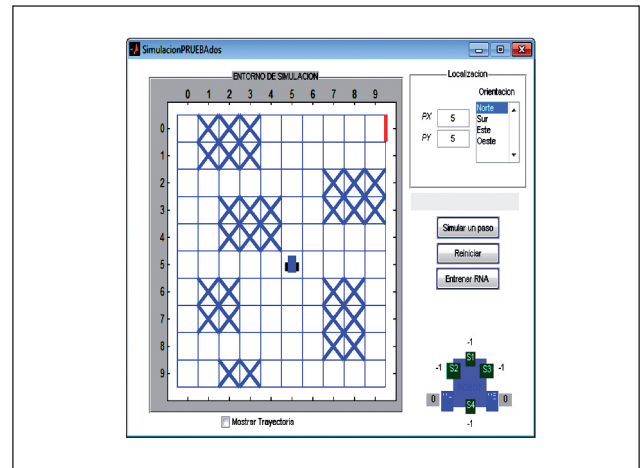


Figura 3. Esquema del entorno de simulación donde navega el robot

Tal como se puede observar, los espacios de la grilla ocupados por una cruz azul corresponden a lugares considerados obstáculos, los cuales no pueden ser ocupados por el robot, y por consiguiente, deben ser evitados por el mismo para no colisionar. Mientras que el resto de los casilleros, considerados libres, son aptos para su desplazamiento. Cabe destacar que, los límites de la grilla también representan un obstáculo para el mismo. Cada casillero puede individualizarse

mediante un sistema de coordenadas definido por los números indicados en los márgenes de la grilla (tanto izquierdo como superior, en correlación el eje x y el eje y del sistema de coordenadas convencional). El robot empleado en este ejemplo que se ilustra en figura 4 consta de cuatro sensores, uno en cada cara, dos sensores internos de posición y posee, además, dos motores, cada uno de los cuales ordena el movimiento de las ruedas laterales.

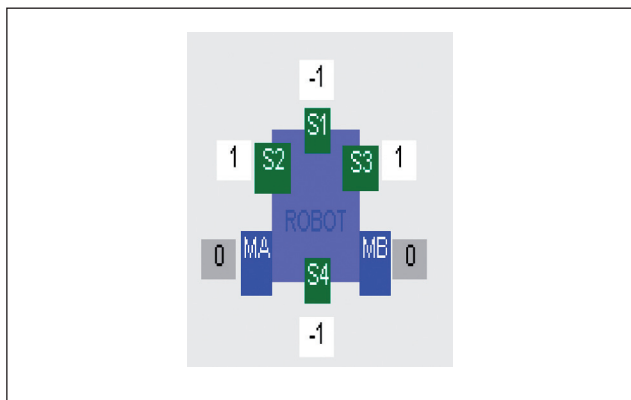


Figura 4. Esquema Sensor/Motor del Robot

Los sensores de posición Px y Py son los que contienen las coordenadas horizontal y vertical respectivamente, correspondientes a la posición del robot en el entorno. Los sensores S1, S2, S3 y S4, son los que deben detectar la proximidad de objetos u obstáculos, adoptando el valor 1 ante la presencia cercana de un objeto (considerando cercano un objeto cuando está a dos casilleros o dos posiciones o menos respecto de la ubicación del sensor, dentro del entorno de operación); o el valor -1 ante la no cercanía de objetos (considerando que un objeto no es cercano cuando está a más de dos casilleros o posiciones respecto del sensor, dentro del entorno de operación).

De igual forma, los motores (MA, MB) adquieren un par ordenado de valores que ponen de manifiesto un efecto determinado, a saber:

- (-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
- Por consiguiente, el método de experimentación consiste en la confección de una determinada trayectoria de entrenamiento a partir de la elaboración de un mapa sensor motor del robot de acuerdo a dicha trayectoria, luego entrenar la red para esa trayectoria, y ya en fase de operación verificar si la misma fue aprendida por el robot. Como paso final del procedimiento, se procede a presentarle al robot otras trayectorias diferentes de las que aprendió para evaluar su performance.

Resultados

Explicado el método experimental con la aplicación de la tecnología de las Redes Neuronales en función del entorno, del tipo de robot y como interactúa este con aquel; en esta sección se presentan una serie de figuras que ilustran los resultados a los que se arribó como consecuencia de los experimentos realizados. La figura 5 muestra una traza roja que constituye la trayectoria del robot que se utilizó para encontrar los patrones de entrenamiento de la red neuronal que actuará como “cerebro” del robot. En este caso, el robot parte de la posición (5,5) con orientación NORTE y se lo entrena haciéndolo llegar a la posición (9,0).

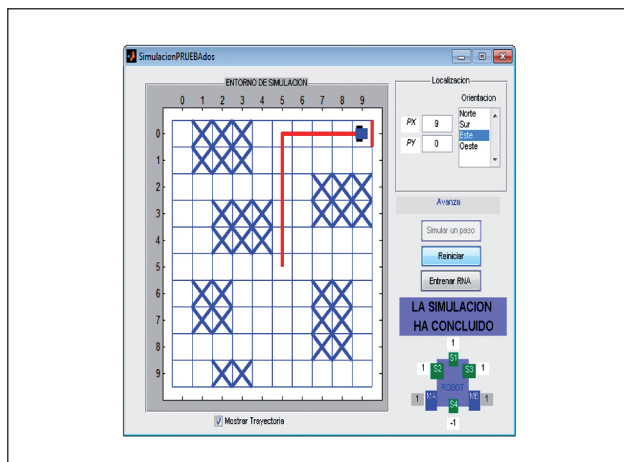


Figura 5. Trayectoria de entrenamiento partiendo de (5,8) y con orientación NORTE.

El proceso de la experimentación

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

Haciendo uso de los recursos del entorno propuesto, se entrena un perceptron que consta de seis entradas y dos neuronas de salida, condicionando el límite de entrenamiento con un tope de 2000 pasos.

Con todos estos datos necesarios para la experimentación se creó un Programa en MATLAB tipo Script, llamado “ExperimentoPRUEBAdos.m”, cuyo texto se transcribe a continuación a modo de ilustración:

```
% *****  
% Experimento PRUEBA dos  
% *****  
% Se definen las variables requeridas para el  
entrenamiento  
% Las entradas están dadas por el valor de los  
sensores y la posición  
% Sens1  
% Sens2  
% Sens3  
% Sens4  
% px  
% py  
  
SP=[-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;  
5 5 5 5 5 5 5 6 7 8 9;  
5 4 3 2 1 0 0 0 0 0 0];  
  
% Las salidas están dadas por la matriz que generan  
los motores  
% MA
```

```
% MB  
  
M=[ 1 1 1 1 1 1 1 1 1 1 1;  
1 1 1 1 1 -1 1 1 1 1 1];  
  
net=newp([-1 1;-1 1;-1 1;-1 1;0 9;0 9],2,'hardlims');  
  
% Syntax  
% net = newp(pr,s,tf,lf)  
% NET = NEWP(PR,S,TF,LF) takes these inputs,  
% PR - Rx2 matrix of min and max values for R  
input elements.  
% S - Number of neurons.  
% TF - Transfer function, default = 'hardlim'.  
% LF - Learning function, default = 'learnp'.  
% Returns a new perceptron.  
% The transfer function TF can be HARDLIM or  
HARDLIMS.  
  
% Cantidad de Epocas  
net.adaptParam.passes = 2000;  
[net,a,e]=adapt(net,SP,M);  
  
%Se verifica luego las salidas y se las compara con  
la tabla “control”:  
control = sim(net,SP)  
  
La ejecución del mismo a través de MATLAB, arroja  
como resultado la salida de la red para este patrón  
de entrenamiento:  
>> ExperimentoPRUEBAdos  
control =  
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  
>>
```

Lo que permite verificar si el perceptrón resolvió correctamente la trayectoria indicada, comparando la similitud de la salida “control” con la matriz M. Es decir, si la salida de la red es igual a la salida deseada del patrón de entrenamiento. En consecuencia, para experimentar este aprendizaje del robot, se analiza la salida de la red en otras posiciones, utilizando el comando “control = sim(net,SP)”, que utiliza un parámetro SP, que denota el estado de los sensores y la posición en alguna zona elegida del mapa. Para la comprensión de esta propuesta se diseñó un programa de simulación adecuada al tipo de robot del ejemplo estudiado en el presente artículo. El programa mencionado se compone de dos archivos simulacionPRUEBAdos.m y simulacionPRUEBAdos.fig y se ha diseñado mediante el GUIDE de MATLAB. De manera que, para visualizar su funcionamiento, ambos archivos deben copiarse en el work de MATLAB y escribir en el Command Window del mismo el nombre del archivo simulacionPRUEBAdos.

Experimentación con otras trayectorias

De igual manera se realizan otros experimentos, tratando de complejizar la trayectoria realizada por el robot, puede verse en las siguientes imágenes, donde se observan los resultados de los robots en “SimulacionPRUEBAtres.m y SimulacionPRUEBAcuatro.m”. En tal sentido, se entrena el robot con una trayectoria que parte del punto (5, 8) con orientación norte y se lo hace llegar a la posición (9, 0), la cual se ve en la Figura 6.

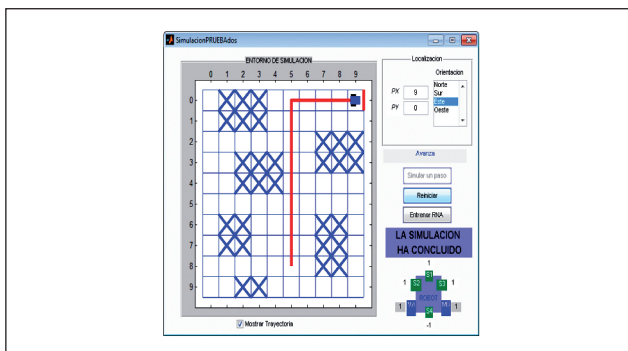


Figura 6. Trayectoria de entrenamiento partiendo de (5,8) y con orientación NORTE.

La respuesta del robot es satisfactoria cuando realiza la trayectoria de entrenamiento en fase de operación; por lo que se opta por verificar si el robot llega al punto (9, 0), partiendo de la posición (3,8) con orientación este. Esto se ve en la Figura 7.

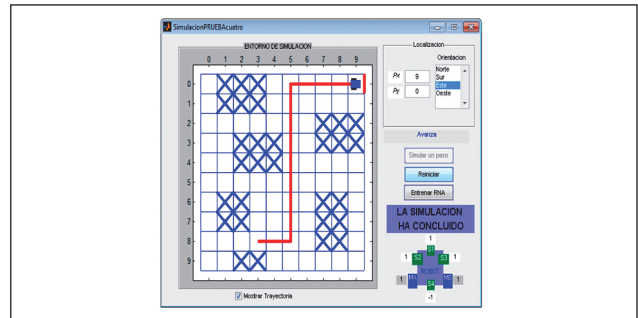


Figura 7. Robot en operación partiendo de (3,8) con orientación ESTE y entrenado con la trayectoria de la Figura 6.

Tal como se observa, el robot se pone en movimiento desde la posición (3,8) con orientación este y llega hasta el objetivo (punto (9, 0)) al encontrar la trayectoria de entrenamiento que, en este caso, la halla rápidamente y con un exiguo esfuerzo de navegación si se tiene en cuenta la cercanía del punto de partida. Ahora bien, también resulta de interés observar la conducta del robot cuando se lo coloca en otra posición, como la (3, 7) y con orientación norte – sur, se orienta hacia el este en (3, 7), pareciendo ser que encontraría la trayectoria de entrenamiento, pero resulta ser que se pasa de esta y colisiona en la posición (6, 7) originando la no convergencia de la red. La Figura 8 ilustra la situación expuesta.

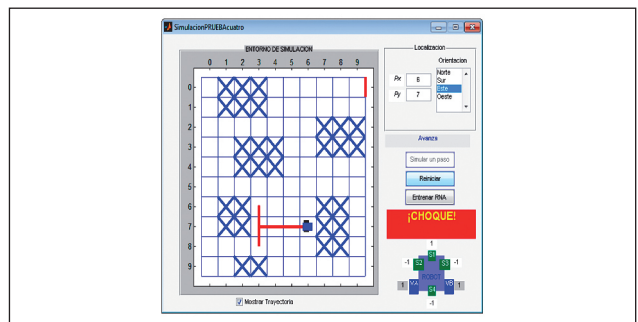


Figura 8. Robot en operación partiendo de (3,7) con orientación NORTE y entrenado con la trayectoria de la Figura 6.

Por último, se entrenó al robot con la trayectoria de figura 9 partiendo de la posición (4, 9) y haciéndolo llegar al mismo punto objetivo (9, 0). Es de interés destacar, que en este caso el robot no aprendió la trayectoria con la que se lo entrenó poniendo de manifiesto la limitación de la arquitectura propuesta. En la Figura 9 se presenta la trayectoria de entrenamiento más compleja.

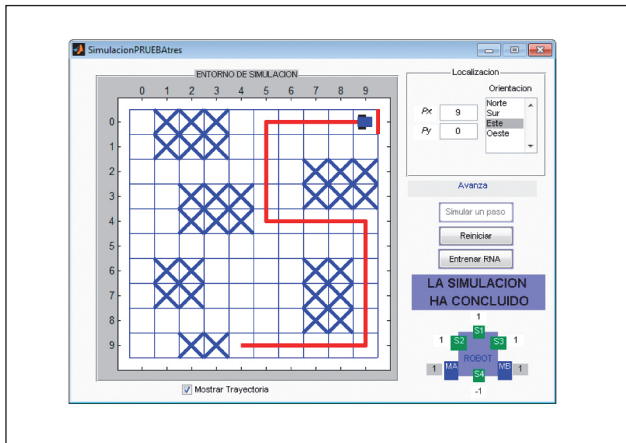


Figura 9. Trayectoria de entrenamiento partiendo de (4,9) y con orientación ESTE.

Discusión

Una de las primeras inferencias significativas que se puede hacer en función de los resultados que arrojaron los experimentos realizados, es que el patrón de entrenamiento ha sido suficiente para algunas trayectorias que entrenó el robot, pero en líneas generales, este dicho patrón no resulta ser muy representativo del entorno bidimensional en el cual este se desplaza [9], habida cuenta de que la mayoría de las salidas que manifiestan sus motores consiste en avanzar y por esa razón es que colisiona partiendo desde otras posiciones y/o otra orientación, tal como se muestra en la figura 8. Para atenuar este problema, se considera aumentar el número de patrones de entrenamiento utilizando trayectorias más largas, vigilando siempre la convergencia de la red. Conforme a lo expresado, es de hacer notar que, por ejemplo, cuando se considera una trayectoria de entrenamiento como la de la Figura 9, la cual está dotada de mayor complejidad en lo que hace a la mayor cantidad de maniobras que debe realizar

el robot en su conducta de navegación, se observa que el mismo no puede aprenderla ante la no convergencia de la red propuesta.

Por otra parte, otro detalle importante a considerar en el desarrollo bajo este paradigma, es la constante búsqueda del robot por hallar su trayectoria de entrenamiento; lo cual constituye una conclusión lógica es en conformidad con las características del paradigma reactivo que representan las redes neuronales [10], pero cabe señalar que en situaciones reales, esta situación puede conllevar a un desgaste excesivo del sistema robot en su accionar navegador, así como también a un alto consumo de combustible.

En síntesis, se asume que el uso de las Redes Neuronales Artificiales como técnica de navegación de características reactivas proporciona resultados satisfactorios para ciertas trayectorias en la fase de operación, tanto más en la medida que estas trayectorias presenten mayor similitud con las que desarrolló en la fase de entrenamiento; así es que se tendrá por caso, que el robot buscará girar más para el lado que lo hace en la trayectoria de entrenamiento que para el otro. A medida que las trayectorias que se le proponen al robot son tanto más complejas que la que este entrenó, este paradigma exhibe sus limitaciones haciendo que la red no converja y se produzcan situaciones de colisión en el ambiente de navegación. En otros términos, se produce una incorrecta generalización de la red neuronal para las nuevas situaciones que debe afrontar el robot, las cuales no se encontraban presentes en las trayectorias de entrenamiento [11].

En virtud de lo expuesto cabe considerar las siguientes cuestiones sobre las que este grupo de investigación se encuentra trabajando con el objetivo de mejorar el desempeño del robot en lo que se refiere a sus conductas de navegación:

- Trabajar con modelos de redes neuronales de arquitecturas más complejas (agregando capas ocultas) para lograr una máxima generalización de la red y de esta manera, poder implementar

trayectorias más representativas del ambiente de navegación del robot, teniendo en cuenta que a medida que aumenta la cantidad y complejidad de la información que usa la red es más difícil que esta converja.

- Complementando el concepto expresado en el punto anterior, considerar la aplicación de técnicas de razonamiento de alto nivel de tipo deliberativas (aprendizaje automático y planificación autónoma de tareas entre otras) como complemento de las técnicas reactivas; las cuales, si bien son de menor velocidad de reacción que estas, también le permiten al robot afrontar requerimientos de desempeño más complejos.

Referencias

1. Santos, J., Duro, R.: *Evolución Artificial y Robótica Autónoma*, Ed. Alfaomega – Ra-Ma, México (2005).
2. Harvey, I.: *Artificial Evolution and Real Robots*, Proceedings of International Symposium on Artificial Life and Robotics (AROB), Masanori Sugisaka (Ed), Beppu, Japan, pp. 138-141, (1996).
3. Amalia F. Foka and Panos E. Trahanias. Predictive autonomous robotnavigation. 2002. In In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 490-495.
4. Ollero Baturone, A.: *Robótica Manipuladores y robots móviles*, Ed. Alfaomega – Marcombo, Barcelona España (2007).
5. Valentino Braitenberg, *Vehicles. Experiments in Synthetic Psychology*, 1987, The MIT Press. EEUU.
6. Brooks, R.: *Achieving Artificial Intelligence through Building Robots*, A.I. Memo 898, MIT, AI Lab (1986).
7. Maes, P.: *A Bottom-up Mechanism for Behavior Selection in an Artificial Creature*, Proceedings of the First International Conference on Simulation of Adaptive Behavior (SAB90), The MIT Press, pp. 238-246, (1991).
8. Mataric, M.J.: *Designing Emergent Behavior: From Local Interactions to Collective Intelligence, From Animals to Animat 2*, Proceedings of the Second International Conference on Simulation of Adaptive Behavior (SAB92), J.-A. Meyer, H. Roitblat, and S. Wilson (Eds.), MIT Press, Cambridge, MA, pp. 432-441, 1992.
9. Adrian. E. Scillato, Daniel. L. Colón y Juan. E. Balbuena: *Técnicas de Navegación Híbrida para Navegación de Robots Móviles*. Ed. Rama de Estudiantes del IEEE. Tesis de grado para obtener el grado de Ingeniero Electrónico. Universidad Nacional del Comahue.
10. 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.
11. García Martínez R., Servente M., Pasquini D., (2003). *Sistemas Inteligentes*. Ed: Nueva Librería. Buenos Aires.