

# Desarrollo de Experimentos de un Robot Navegador con Redes Neuronales en un Ambiente Estructurado: Programación de Conductas con Algoritmo de Retropropagación

Alejandro Hossian, Lilian Cejas, Roberto Carabajal, César Echeverría, Verónica Olivera, Maximiliano Alveal

Proyecto de Investigación FRN - UTN  
Plaza Huinca, Neuquén, Argentina  
alejandrohossian@yahoo.com.ar

**Resumen** — El presente artículo busca comunicar los avances realizados en el análisis y desarrollo de experimentos realizados en computadora para robots navegadores mediante la aplicación de Tecnologías Inteligentes. Para su realización se dispone de software y hardware apropiados a partir de los cuales se diseña un ambiente de navegación con obstáculos definidos en el marco de un entorno industrial. En este contexto, se diseña un ambiente computarizado con el cual el robot interactúa, en el cual se desplaza hasta llegar a un objetivo predeterminado, tratando de no colisionar. Bajo estas premisas y requisitos de usuario, se analiza el desempeño del robot a los efectos de procurar optimizar su performance dentro de este escenario. Para evaluar las potencialidades y limitaciones que presentan las conductas del robot, y en función de la complejidad del ambiente de navegación y de las tareas que éste debe llevar a cabo, se hace uso de las tecnologías de las Redes Neuronales Artificiales (RNA). A tal efecto, una arquitectura de RNA de tipo Backpropagation es la que se utiliza en el presente trabajo.

**Palabras Claves**—Navegación Robótica, Redes Neuronales Artificiales, Paradigma Reactivo, Robótica Autónoma.

## I. INTRODUCCION

El presente artículo forma parte de un proyecto de investigación del grupo de Robótica de la Facultad Regional Neuquén de la Universidad Tecnológica Nacional, el cual consiste en la implementación de las RNA con aprendizaje supervisado para estudiar la performance de un vehículo robótico que navega en un ambiente con obstáculos definidos. Estas técnicas constituyen uno de los tópicos más potentes del paradigma reactivo en lo que concierne a la navegación autónoma de robots [1], [2] y [3].

Se aplica una arquitectura de RNA de Retropropagación, con una capa de entrada, una capa oculta y una capa de salida en su estructura, lo que le permite al robot abordar trayectorias más complejas.

En el proceso de entrenamiento, el robot recibe información del entorno en el cual navega. Esta es volcada en un mapa sensor – motor que esquematiza tal ambiente y se diseña mediante un software que simula su interacción.

Este trabajo se encuadra dentro de un campo más amplio y al cual se lo considera una evolución del paradigma reactivo y que recibe el nombre de “Aproximación Basada en Comportamientos” [4], entendiéndose así, que los comportamientos describen la forma en que reacciona nuestro sistema robot ante un determinado estado de los sensores, y en consecuencia, las actuaciones se determinan de manera más

elaborada que de una simple consulta a una tabla de correspondencias y realizando cálculos a los efectos de decidir qué acción se debe llevar a cabo.

## II. MARCO TEÓRICO

Es importante realizar una breve reseña acerca del estado del arte de estos tópicos, haciendo 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, tales como los vehículos de Braitenberg [5], los cuales mediante ciertas conexiones excitadoras e inhibitoras, permiten un enlace directo entre los sensores y motores correspondientes, demostrando la capacidad de obtener comportamientos de alta complejidad. Además, es dable resaltar los logros obtenidos por Rodney Brooks [6] en relación a las “Arquitecturas Subsumidas”, por Pattie Maes [7] y Maja Mataric [8] quien realizó sobresalientes trabajos de control en una colectividad de agentes o robots que hacen uso de las estrategias de Brooks. Así también lo señaló Pattie Maes, cuando destacó que la “Aproximación Basada en Comportamientos” no sólo 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 este sentido, y siguiendo 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 que se aplica en el robot navegador que, en este trabajo se presenta, es de tipo reactiva, exponiendo a continuación sus aspectos más sobresalientes.

### A. Aspectos introductorios

Las arquitecturas reactivas están basadas en modelos biológicos que buscan explicar el comportamiento observado en distintos organismos vivos e implementar estrategias de control, los cuales responden a una colección de pares “condiciones-acciones”. La respuesta a estímulos es reflexiva, en la cual no se advierten procesos de carácter deliberativos de ningún tipo y los movimientos del robot se guían solamente a partir de la información que le brinda en ese momento sus

sensores. En consecuencia, las acciones del robot se basan en un acoplamiento directo entre sensores y actuadores mediante bucles rápidos de realimentación.

Este tipo de arquitecturas se caracteriza por su mayor velocidad en el procesamiento de información, a diferencia de las arquitecturas cuyo enfoque es el basado en conocimiento; no obstante, la gran velocidad de respuesta que poseen también conlleva notables limitaciones cuando el robot debe abordar tareas que requieren planificación. Estas arquitecturas se basan en modelos de procesamiento de información “masivamente paralela”, siendo un ejemplo de ellas las redes neuronales artificiales (RNA).

Siguiendo este razonamiento, puede inferirse que el robot percibe el mundo y ejecuta la acción que le parezca más adecuada; es decir que, la información percibida por el robot no se integra en ningún modelo del ambiente, o bien, el ambiente de operación es su mejor ambiente. A continuación, la figura 1 ilustra las arquitecturas de control reactivas.

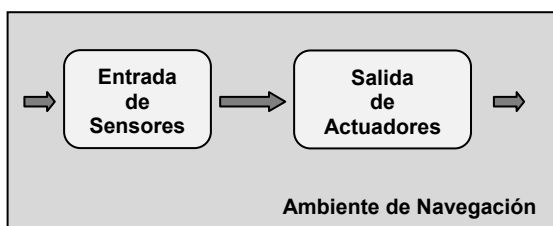


Fig. 1 Arquitectura de Control Reactiva

### B. Red Neuronal de tipo Perceptron

En la fase inicial de trabajo dentro del marco de este proyecto se ha utilizado una red de tipo Perceptron para evaluar el rendimiento del robot móvil en un ambiente estructurado; considerando que este tipo de red reúne las premisas que caracterizan a las arquitecturas reactivas. Su arquitectura puede observarse en la figura 2. Este modelo de red consta de dos capas, una de entrada con seis neuronas y una de salida dos neuronas, las que representarían a los sensores y a los actuadores o motores respectivamente.

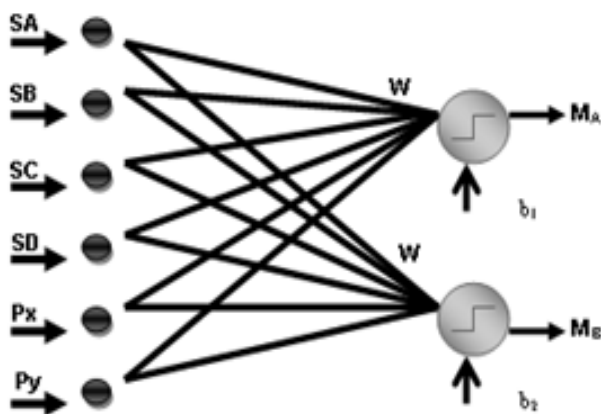


Fig. 2 Arquitectura de Control Reactiva

### C. Red Neuronal de tipo Backpropagation

En virtud del objetivo, el cual busca evaluar el rendimiento del robot móvil en un ambiente de simulación computarizado y bajo las premisas que caracterizan al paradigma reactivo, en esta instancia se emplea un robot navegador que desarrolla un aprendizaje del entorno en base a la tecnología de RNA; en

particular, se considera una configuración de red correspondiente a un modelo de Backpropagation con una capa de entrada, una capa oculta y una capa de salida, cuya arquitectura se puede observar en la siguiente figura [9].

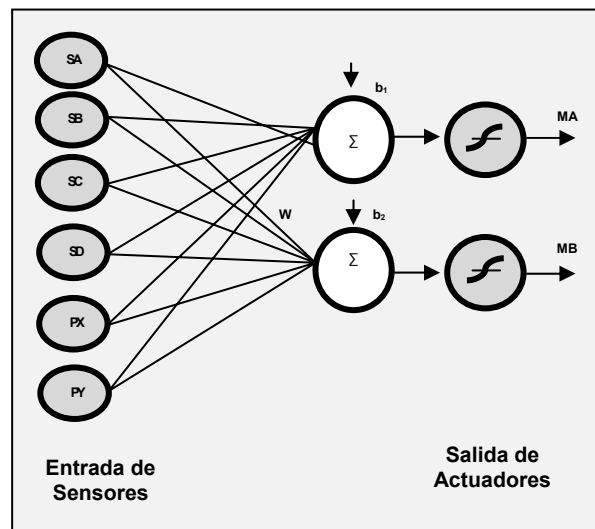


Fig. 3 Modelo de Red Neuronal Backpropagation

La arquitectura que muestra la figura se compone de una capa de entrada, una oculta y una de salida; la capa de entrada está formada por seis neuronas: cuatro de ellas corresponden a sensores de proximidad (SA, SB, SC, SD), 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), tal como en el caso del Perceptron. Por su parte, la capa de salida está conformada por dos neuronas, correspondientes a los dos motores que comandan las dos ruedas laterales (MA y MB), como ocurre con el Perceptron. Más adelante se explica en detalle la elaboración del esquema “sensor-motor” del robot y su interacción con el entorno.

## III. EXPERIMENTACIÓN

Tal como se expuso en la sección previa, los autores tienen como objetivo central simular en computadora distintas rutas de navegación del robot para diferentes entornos de operación. La combinación de rutas definidas para diferentes configuraciones del ambiente de navegación, constituyen las distintas conductas de navegación que exhibe el robot. Con este motivo se utilizó hardware Pentium 5 y el paquete de software MATLAB (versión 12.0) con el respaldo de su Toolbox de RNA. Mediante estos recursos 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. La figura 4 describe un esquema de la organización del ambiente donde navega el robot, como una grilla cuyas coordenadas (x, y) se encuadran dentro de un eje cartesiano. Este entorno visual brinda la información referida a la localización del robot (proporcionada por las coordenadas X e Y, que en este caso corresponden a 0 según X y a 9 según Y), su orientación (dada por Norte, Sur, Este y Oeste), el ambiente propiamente dicho, 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.

Los espacios de la grilla ocupados por una cruz corresponden a posiciones con “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. Es importante señalar que, los límites de la grilla también representan un obstáculo para el mismo. Cada casillero puede individualizarse mediante el mencionado sistema de coordenadas, cuyos valores se indican en los márgenes de la grilla.

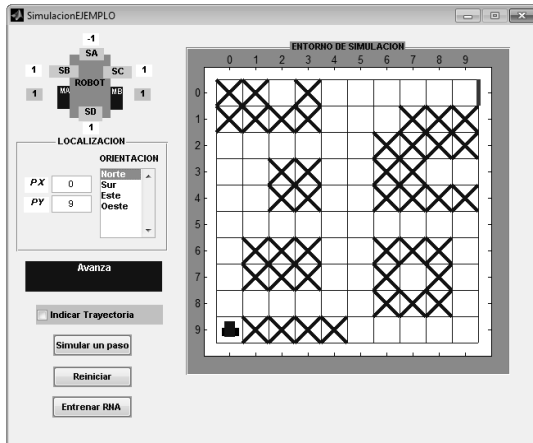


Fig. 4 Esquema de simulación del entorno

El robot empleado en este trabajo 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, como puede observarse en el extremo superior izquierdo de la figura 4.

PX y PY son los sensores de posición y los que contienen las coordenadas horizontal y vertical respectivamente, correspondientes a la posición del robot en su ambiente de navegación. Mientras que los sensores SA, SB, SC y SD, son los encargados de detectar la proximidad de objetos u obstáculos. Estos sensores pueden adoptar, por ejemplo, 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).

Asimismo, los motores (MA, MB) adquieren un par ordenado de valores que ponen de manifiesto un efecto determinado, esto es: (-1 -1) si el robot debe desplazarse una posición hacia atrás, (-1 1) si el robot debe desplazarse una posición hacia la izquierda, (1 -1) si el robot debe desplazarse una posición hacia la derecha y (1 1) si el robot debe desplazarse una posición hacia delante.

En este sentido, el método de experimentación consiste en la confección de una trayectoria de entrenamiento a partir de la elaboración de un mapa sensor- motor que incluya dicha trayectoria. Posteriormente, se entrena la red para esa trayectoria, y ya en fase de operación se verifica 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.

Siguiendo este razonamiento se diseña mediante lenguaje de programación un modelo de red Backpropagation que reúna las características antes descritas para luego

incorporarlas en la inteligencia del robot a fin de evaluar su performance. Esto puede observarse a continuación en el siguiente código:

```

% Experimento Prueba para definir una trayectoria o
% camino con BACKPROPAGATION
% -----
% Definición de las variables para el entrenamiento:
% Sensores de Entrada: SA, SB, SC, SD, PX, PY

SENSORES=
[-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 1 -1 -1 1 1 1 1 -1 -1 -1 -1 -1 -1 -1 1 -1 -1 -1 -1;
 0 0 0 1 2 3 3 3 3 3 4 5 5 5 5 5 5 5 5 6 7 8 9;
 9 8 8 8 8 8 8 7 6 6 6 6 6 5 4 3 2 1 0 0 0 0 0];

% Salidas para los motores
% MA, MB
TRAYECTORIA=
[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];

% La función NEWFF permite crear una red
% Backpropagation
% Este comando NEWFF crea un objeto del tipo red y
% también inicializa los pesos y % el bias de la
% red, usando por default el comando (initnw). Por
% lo tanto la red % está lista para ser entrenada.

% Creación de la red:
net=newff(minmax(SENSORES),[6,2],{'tansig','purelin'});
'traingd');

% La Función TRAINGD es un Algoritmo de pasos
% descendientes, que actualiza pesos y ganancias
% variándolos en la dirección negativa del gradiente
% de la función del error. Es un algoritmo de
% aprendizaje muy lento, que requiere de la
% siguiente sintaxis:

% Intervalo de visualización de los resultados
net.trainParam.show = 50;

% Tasa de aprendizaje
net.trainParam.lr = 0.05;

% Máximo número de iteraciones para obtener
% convergencia
net.trainParam.epochs = 3000;

% Error máximo permitido
net.trainParam.goal = 0.01;

[net, tr]=train(net,SENSORES,TRAYECTORIA);

%Se verifican las salidas y se comparan con la
% tabla:
b = sim(net,SENSORES)
TRAYECTORIA

```

Presentado el método experimental basado en la aplicación de la tecnología de las RNA en función del entorno, del tipo de robot y como interactúa este con aquel; se presenta a modo ilustrativo en la siguiente figura una traza gruesa que indica la trayectoria utilizada 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 (0,9) con orientación NORTE y se lo entrena intentando que alcance la posición (9,0). Esta trayectoria es recorrida exitosamente tal como se observa en la figura 5.

A continuación se ilustran algunas de las respuestas obtenidas cada vez que se evalúa la performance del robot.

En la figura 6 se puede observar al robot partiendo de una posición (9, 5) con orientación Oeste y tras recorrer cierto camino encuentra la trayectoria con la cual fue entrenado, alcanzando de manera correcta el objetivo, aunque el trayecto que se observa es más extenso.

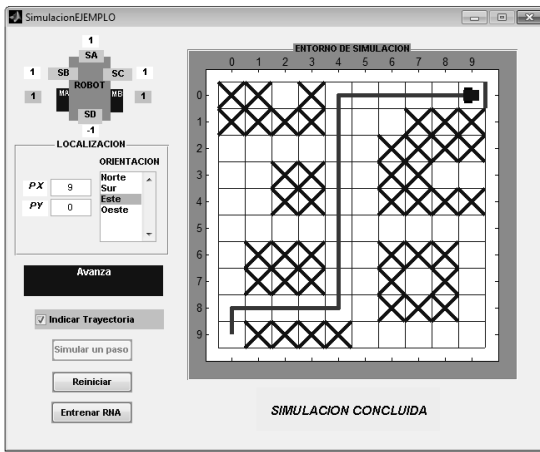


Fig. 5 Trayectoria de entrenamiento

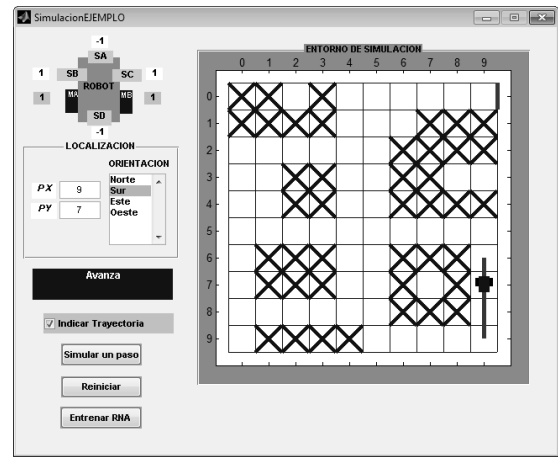


Fig. 8 El robot ingresa en un bucle cuando parte desde (8,5).

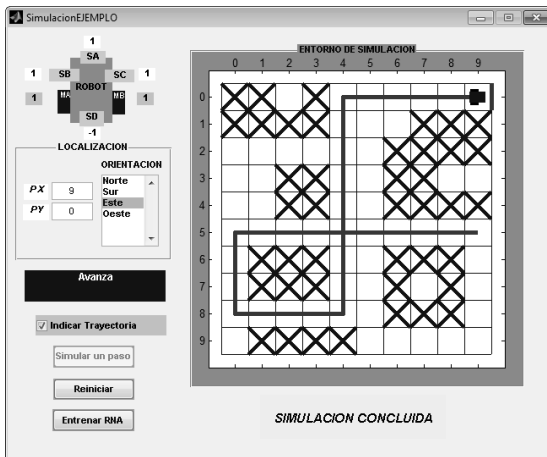


Fig. 6 Trayectoria exitosa desde otro punto de partida.

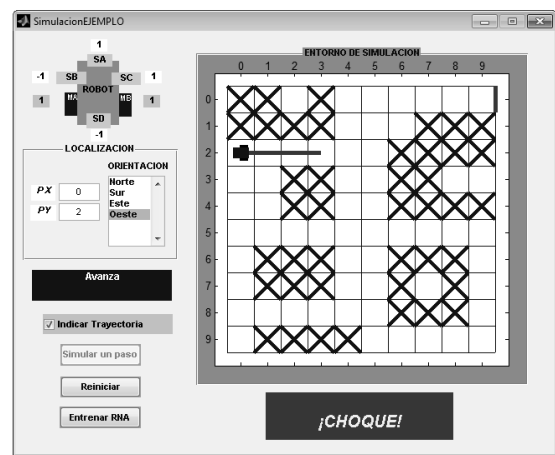


Fig. 9 El robot colisiona en algunos casos

En la figura 7, puede observarse al robot partiendo desde la posición (1,4). En este caso, el robot recorre distintos caminos hasta hallar aquel que le resulta familiar logrando llegar a destino.

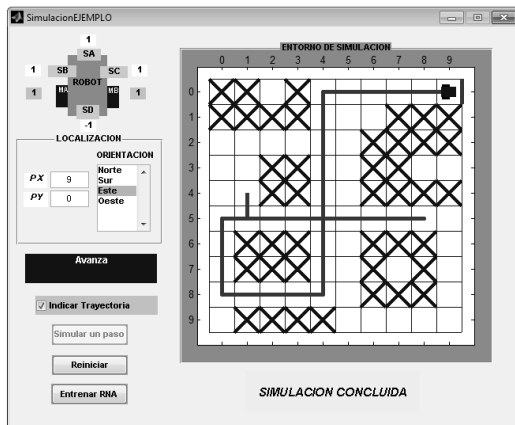


Fig. 7 Trayectoria exitosa desde otro punto de partida.

Profundizando la etapa de experimentación, se somete al robot a otros puntos de partida y se observa que en algunos casos no logra resolver adecuadamente el objetivo. Tal es el caso de la figura 8, donde el mismo ingresa en un bucle del cual no puede salir cuando parte desde la posición (8,5) con orientación Oeste.

O bien como ocurre en el caso de la figura 9, cuando parte de una posición (3,2) con orientación Oeste, la que le resulta totalmente ajena, termina colisionando.

#### IV. CONCLUSION

Este trabajo es una continuación del trabajo iniciado con las RNA de tipo Perceptron. Señalando que sus limitaciones marcaron la necesidad de profundizar en un modelo de red que pudiera resolver problemas de mayor complejidad.

Es así que, explicado el método experimental con la aplicación de la tecnología de las RNA de tipo Backpropagation en función del entorno, del tipo de robot y como interactúa este con aquel, se ha podido advertir las diferentes conductas del robot cuando parte desde diferentes puntos con la misma trayectoria de entrenamiento, o cuando se lo hace partir desde el mismo punto con distinta orientación para una misma trayectoria de entrenamiento.

A partir de las investigaciones y experimentos realizados se obtuvieron las siguientes conclusiones:

- Se puede ver que el robot se desempeña bien dependiendo del lugar de salida e intentando siempre hallar la trayectoria de entrenamiento.
- El patrón de entrenamiento ha sido suficiente para algunas trayectorias, pero en ciertos casos dicho patrón no resulta ser representativo del entorno en el cual opera el robot, ya que la mayoría de las salidas es avanzar y por ese motivo es que choca partiendo desde otras posiciones. O bien, ingresa en un bucle sin poder resolver esa situación, quedando inmóvil.
- Para atenuar este problema, se aumenta el número de patrones de entrenamiento utilizando varias trayectorias

o más largas, vigilando siempre la convergencia de la red.

- Se modifica la arquitectura de la red neuronal (se agrega una capa oculta) para obtener: 1) Convergencia para los patrones de entrenamiento y 2) Máxima generalización, para las nuevas posibles situaciones.

En síntesis, se asume que el uso de las RNA 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 [9]. 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 o bucles 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.

La idea que subyace detrás de este objetivo central, es que dichas mejoras puedan verse reflejadas en términos de evitación de obstáculos, velocidad de respuesta, optimización de las trayectorias y logro de los objetivos [4].

Concluyendo, las arquitecturas híbridas, a través de la combinación de ambos enfoques (basado en conocimiento y basado en comportamiento), permiten la reconfiguración de los sistemas de control reactivos por medio de su habilidad para razonar sobre los comportamientos fundamentales.

En virtud de lo expuesto cabe considerar la consiguiente investigación sobre el objetivo de mejorar el desempeño del robot en lo que se refiere a sus conductas de navegación.

Esto es, 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.

Asimismo, y dado el objetivo planteado por los autores, se considera la aplicación de algoritmos de búsqueda como nuevas técnicas para la resolución de este caso de estudio. Esto es posible, pues el robot se desempeña en un ambiente estructurado, es decir invariante en el tiempo. Con lo cual y bajo estas circunstancias se plantean alternativas como algoritmos de búsqueda a ciegas, algoritmos de búsqueda de la trayectoria más corta y algoritmos de búsqueda con costos, como posibles tópicos de análisis.

#### 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. In Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 490-495 (2002)
- [4] Ollero Baturone, A. “Robótica Manipuladores y robots móviles”. Ed. Alfaomega – Marcombo, Barcelona España, 2007.
- [5] Braitenberg, V. “Vehicles. Experiments in Synthetic Psychology”. The MIT Press. EEUU (1987)
- [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.: Behavior-Based System: Main Properties and Implications, Proceedings of IEEE International Conference on Robotics and Automation, Workshop on Architectures for Intelligence Control Systems, Nice, France, pp. 46-54, (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] Hossian, A., Cejas, L., Sapag, L., Olivera, V. 11° Congreso Interamericano de Computación Aplicada a la Industria de Procesos. “Estudio del Comportamiento de Robots Móviles Autónomos en Entornos de Navegación Estructurados Mediante la Aplicación de Tecnologías Inteligentes”. ISBN: 978-612-4206-04-7. Perú, 2013.



**Alejandro Armando Hossian.** Es responsable de formar recursos humanos en el área de investigación y desarrollo dirigiendo tesis de doctorado y maestría vinculadas a sus áreas de especialización. Es Profesor Titular Regular de la Facultad Regional Neuquén, Universidad Tecnológica Nacional (UTN - FRN). Es Docente Investigador, Director del grupo de investigación

de robótica cognitiva y Director de Posgrado de la misma casa de altos estudios. Es Docente Investigador Categoría III del Ministerio de Educación y Par Evaluador de la Comisión Nacional de Evaluación y Acreditación Universitaria (CONEAU), Republica Argentina. Acredita estudios universitarios de Ingeniero Civil por la Universidad Católica Argentina, es Especialista en Ingeniería de Sistemas de Expertos por el Instituto Técnico Buenos Aires (ITBA), es Máster en Ingeniería de Software por la Universidad Politécnica de Madrid (UPN), es Magister en Ingeniería de Software por el Instituto Técnico Buenos Aires (ITBA) y es Doctor en Ciencias Informáticas por la Facultad de Informática de la Universidad Nacional de La Plata (UNLP).



**Lilian Cejas.** Posee título de grado universitario de Ingeniera Química, una Especialización en Docencia Universitaria y Maestría en Docencia Universitaria. Se ha capacitado en distintos cursos afines a su desempeño profesional enfocado principalmente al área educacional superior universitario. Se ha desempeñado desde hace más de veinticinco años en la Universidad Tecnológica Nacional, Argentina. Es docente-investigadora en la modalidad presencial y desde unos diez años en la modalidad virtual, como docente, investigadora. Es Directora de Departamento de Ciencias Básica y Directora de Carrera de la Licenciatura en Tecnología Educativa, modalidad a distancia. Participa como docente invitada en la modalidad virtual de la Universidad Autónoma del estado de Hidalgo, México. Ha participado en numerosos congresos y reuniones científicas nacionales e internacionales, tanto con

presentación de trabajos de investigación como organizadora, como así también es miembro de comisiones científicas y tecnológicas.



**Roberto Carabajal.** Egresado como Ingeniero Electricista orientación Automática, se ha especializado en software para la administración y control en la industria, con amplia experiencia en redes, servidores y bases de datos. Ha trabajado como proveedor de servicios en el área del mantenimiento electrónico para informática. Fue profesor en la carrera corta Técnico Superior en

Programación de la Facultad Regional del Neuquén, y actualmente se desempeña como profesor en ingeniería química y electrónica de la misma facultad. Está categorizado con Categoría "5": Programa de Incentivos a docentes e investigadores de Universidades Nacionales y con Categoría "E": Carrera de docente investigador de UTN. Ha participado en diversas publicaciones científicas y en el grupo de investigación Desarrollo de Métodos y modelos no convencionales para la programación de comportamientos en robots. Es codirector de la obra de diseño y puesta en marcha de la planta Planta Piloto- Unidad de Destilación de Dos Componentes de la F.R.N.



**Carlos César Echeverría.** Ingeniero Electrónico por la Universidad Nacional de La Plata. Docente de la Universidad Tecnológica Nacional Facultad Regional de Neuquén en las cátedras de Teoría de Circuitos 2, Electrónica Aplicada 1 y Medidas Electrónicas 1. Categorizado con Categoría "5" del Programa de Incentivos a docentes e investigadores de Universidades Nacionales y con Categoría "E"

de la Carrera de docente investigador de UTN.

Profesor en Disciplinas Industriales (PDI) - Profesor en Docencia Superior (PDS). Diplomatura Superior en Enseñanza de las Ciencias. Especialización en "Tecnología de antenas para uso Satelital". Ha participado en diversas publicaciones científicas y en el grupo de investigación Desarrollo de Métodos y modelos no convencionales para la programación de comportamientos en robots. Ha desarrollado varios proyectos sobre la implementación de tecnología en las aulas de enseñanza de Ingeniería y ciencias duras como la Física y la Matemática.



**Lorena Verónica Olivera.** Trabaja en el grupo de Investigación de Robótica de la Facultad Regional del Neuquén de la Universidad Tecnológica Nacional (UTN-FRN). Docente Investigador por la Universidad Tecnológica Nacional. Ha participado en la elaboración de artículos sobre automatización y Robótica Cognitiva. Es Ayudante de la Cátedra de "Tecnología de redes neuronales" en UTN -

FRN. Es Estudiante avanzado de Ingeniería Electrónica en la misma casa de altos Estudios.



**Emanuel Maximiliano Alveal** – Becario de investigación que desarrolla sus actividades en el marco del Grupo de Investigación en Robótica de Manipuladores de la Universidad Tecnológica Nacional – Facultad Regional del Neuquén. Estudiante avanzado de la Carrera de Ingeniería Electrónica de la misma casa de altos estudios.