

ESTUDIO EXPERIMENTAL SOBRE COMPORTAMIENTOS REACTIVOS-EVOLUTIVOS EN NAVEGACION DE ROBOTS MOVILES

J. A. Fernández León¹⁻³, M. Tosini¹, G. G. Acosta²⁻³, H. N. Acosta¹

¹ Instituto de Investigación INTIA – Departamento de Computación y Sistemas
Facultad de Ciencias Exactas - Universidad Nacional del Centro de la Provincia de Buenos Aires
(7000) Tandil - Buenos Aires – Argentina

² Grupo INTELYMEC – Departamento de Ingeniería Electromecánica
Facultad de Ingeniería - Universidad Nacional del Centro de la Provincia de Buenos Aires
(B7400JWI) Olavarría - Buenos Aires – Argentina

³ CONICET – Consejo Nacional de Investigaciones Científicas y Técnicas, Argentina

{jleon, mtosini, nacosta} @ exa.unicen.edu.ar

gerardo.acosta @ ieee.org

Resumen

En este trabajo se analiza la navegación y la evasión de obstáculos para robots móviles en un ambiente no conocido, estático y simulado. A partir de la lectura de los sensores de proximidad, los controladores basados en Redes Neuronales Artificiales (RNA) establecen la trayectoria deseada entre la posición actual y la posición objetivo. Algoritmos Evolutivos son usados en la selección del mejor controlador. Esta metodología de trabajo, es conocida como Robótica Evolutiva (RE), comúnmente utilizando simples arquitecturas de redes neuronales. A pesar de que los controladores desarrollados dentro de RE generalmente presentan procesamiento temporal, la mayoría no considera la experiencia obtenida en el proceso evolutivo del controlador. Por lo tanto, el presente trabajo, se refiere a la especificación y testeo de controladores neuronales, realizando mutaciones genéticas entre generaciones de controladores en base a la experiencia adquirida. Controladores basados en Redes Neuronales de Tiempo Discreto (TRNN) fueron desarrollados con dos variantes: Redes Neuronales Plásticas (PNN) y redes del tipo Feed-Forward (FFNN). Este trabajo demuestra que la mutación controlada no presenta mayores ventajas respecto de la no controlada, mostrando que la diversidad es más poderosa que la adaptación controlada.

Palabras Clave: Robótica Evolutiva; Redes Neuronales Evolutivas; Adaptación Robótica; Agentes Robóticos Simulados

1. Introducción

Según Hebb [1], el comportamiento es la adaptación inicial al entorno considerando la percepción sensorial. Esto permite que el organismo se aleje de los eventos perjudiciales y se acerque hacia los favorables, o que realice cambios en su entorno local para su beneficio en cuanto a su supervivencia. En este contexto, Cliff [9] y Nolfi [10] indican que la manera más apropiada de obtener un comportamiento adaptable, es mediante la Robótica Evolutiva (RE). En RE el comportamiento surge de la interacción directa con el entorno, limitando la intervención del hombre a la definición de los objetivos a cumplir por el comportamiento requerido. De este modo, es necesario una regla que determine cuantitativamente el grado que un determinado comportamiento se aproxima al comportamiento deseado.

En este artículo, la RE es utilizada para obtener controladores que sean capaces de adaptar el comportamiento de un robot según las entradas de los sensores. El trabajo planteado reviste de interés como un paso previo al estudio de *escalabilidad* dentro de la RE [19, 23]. Particularmente, el trabajo se orienta al estudio de la *adaptabilidad* de neuro-controladores monolíticos desde un plano evolutivo y adaptativo durante la vida del controlador. Dicho trabajo busca entender algunos de los mecanismos artificiales que permiten adaptación de neuro-controladores.

1.1. Robótica Evolutiva

Robótica Evolutiva es un subcampo del área denominada *Behavioral Robotics*, y está referida a la aplicación de métodos computacionales evolutivos en sistemas autónomos de control robótico. Uno de los objetivos principales de la RE es el desarrollo automatizado de métodos que puedan ser usados en la evolución de estrategias de control de comportamientos complejos [2].

Diferentes trabajos señalan que uno de los intereses principales en RE es el desarrollo de sistemas de control basados en inteligencia computacional o, resumiendo, sistemas de control inteligentes [2, 3, 4, 5, 6, 7]. Principalmente, dichos trabajos describen pruebas experimentales vinculadas a evasión de obstáculos, navegación en laberintos, aprendizaje robótico, y controladores adaptativos mediante el uso de pequeños robots móviles o modelos computacionales de ellos (enfoque simulado).

1.2. Agentes Artificiales

Según lo propuesto por Russell y Norvig [8], “un agente es algo que puede percibir su entorno por medio de sensores y desenvolverse en él a través de actuadores”. Además, en [8] se expresa que “un agente ideal” es “uno que siempre toma la acción que se espera para maximizar su desempeño o performance, dada una determinada secuencia de percepciones realizadas”. Según esto, un agente que maximiza su desempeño o medida de *fitness* (expresado por una función matemática) se considerará apropiado para una determinada tarea en un entorno específico. El presente trabajo tiene como meta principal alcanzar la construcción de un agente que sea capaz de adaptarse en forma apropiada manteniendo y respondiendo a la información proveniente de su entorno.

1.3. Robótica Evolutiva y Adaptación en Sistemas Artificiales

En un contexto real, los animales pueden ser considerados dentro de la definición de agentes dada anteriormente. Efectivamente, los animales no sólo se adaptan a los cambios ambientales, ellos también pueden acumular adaptaciones.

Los animales son capaces de almacenar “conocimiento” sobre entornos previamente encontrados y usar tal conocimiento para modificar su comportamiento cuando se encuentren nuevamente con dicho ámbito. Este proceso es llamado *aprendizaje* cuando ocurre en un tiempo acotado, y *evolución* cuando ocurre en una escala generacional [13]. En otras palabras, un agente se adapta mediante la evolución a escala generacional (etapa de desarrollo del controlador) de diferentes comportamientos útiles para distintos entornos, y mediante aprendizaje (etapa de evaluación del controlador).

Sin embargo, es importante señalar que la definición de “características apropiadas” para un ambiente en particular, está definida explícitamente y sin ambigüedad en la función de fitness.

En suma, en [13] no se considera la interacción directa entre evolución y aprendizaje, pero en dicho trabajo se realiza la suposición que todas las generaciones previas han sido expuestas al mismo conjunto de entornos. Esta suposición es demasiado restrictiva y por ende poco aplicable a un caso real.

A pesar de ello, puede ser considerado que el agente haya sido capaz de acumular adaptaciones con pequeños cambios en la experiencia obtenida en una generación. En este caso en particular, la experiencia obtenida por un neuro-controlador se refiere a las características que pueden ser desarrolladas en una generación (por ej., los pesos sinápticos establecidos luego de un proceso evolutivo).

Uno de los desafíos principales en RE es descubrir y modelar distintos mecanismos de adaptación. Muchos de los trabajos en RE [3, 10, 11] consideran a la evolución artificial de neuro-controladores como un mecanismo de adaptación válido. También, se propone que la RE es una metodología viable para desarrollar agentes autónomos que puedan presentar habilidades con algún grado de conciencia. La evolución artificial difiere de otros esquemas de aprendizaje debido a que trabaja sobre una población de diferentes individuos y está basada en un enfoque seleccionista más que en uno meta-dirigido [12]. Este enfoque de la RE es el adoptado en este trabajo.

1.4. Redes Neuronales Artificiales en Robótica Evolutiva

Las Redes Neuronales Artificiales (RNA) han sido la estructura computacional de excelencia adoptada en RE. Los controladores basados en RNA han sido implementados para llevar a cabo distintas funciones con robots móviles [2, 4, 14, 15]. La mayoría de tales aplicaciones usan arquitecturas simples de RNA, las cuales son capaces de realizar procesamiento temporal. Ejemplos típicos son las Redes Neuronales Recurrentes de Tiempo Discreto (TRNN- Discrete Time Recurrent Neural Networks) con dos variantes: Redes Neuronales Plásticas (PNN - Plastic Neural Networks) utilizadas en [16, 17], y una variante de las redes tipo Feed-Forward (FFNN) descritas en [18, 19]. Este tipo de controladores es capaz de presentar un comportamiento apropiado, recordando las habilidades adquiridas y pasándolas a las generaciones venideras. Las mismas permiten almacenar experiencia previa y utilizarla para modificar su comportamiento actual, así como el comportamiento de sus descendientes cuando se presente un entorno determinado (o situación). Esta adaptación es más rápida a medida que se incrementa el valor de la función de *fitness*.

En este artículo, se examina un sistema evolutivo de control robótico dentro de un entorno simulado mediante la generación de neuro-controladores en un proceso evolutivo artificial. La descripción del entorno de simulación, los neuro-controladores implementados y la evolución de los mismos mediante el uso de algoritmos genéticos, es presentada en las siguientes secciones. El presente artículo también expone resultados y conclusiones preliminares en el uso de controladores basados en RNA en RE, ventajas y desventajas de adaptaciones controladas, y la influencia de la velocidad de las adaptaciones en el comportamiento general del robot simulado.

2. El objeto de estudio

2.1. Descripción del robot

El robot empleado en este estudio es el Khepera [26], un sencillo modelo de robot móvil con gran flexibilidad que se emplea frecuentemente en estudios de este tipo. El robot posee sensores de proximidad para determinar la presencia de obstáculos y fotosensores que dan cuenta de la presencia de fuentes de luz. Sus actuadores son dos ruedas independientes movidas por sendos motores paso a paso. La disposición de estos elementos del sistema de control puede apreciarse en la Figura 1.

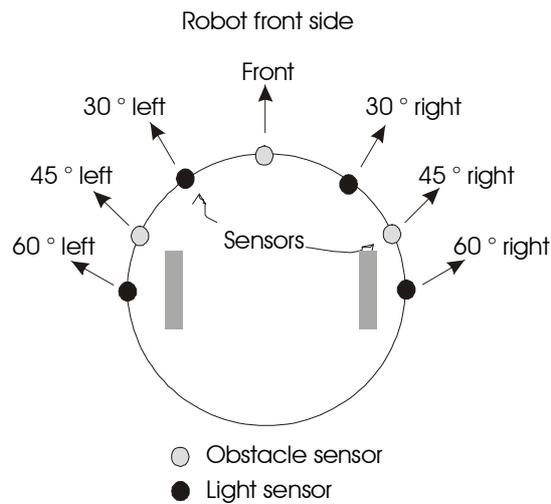


Fig. 1 – Distribución de los sensores en el cuerpo del robot

Aquí se aprecian tres sensores de distancia dispuestos en el frente del robot, separados entre sí por 45° , y cuatro fotosensores dispuestos equidistantemente también en el frente del robot. Los sensores de distancia envían un valor positivo inversamente proporcional a la distancia entre el sensor y algún obstáculo y su alcance es de 15 cm. La salida de los sensores de luz también es inversamente proporcional a la distancia y al ángulo entre la fuente y el sensor.

2.2. Neuro-controladores artificiales

Los controladores basados en RNA utilizados en este trabajo, se vinculan con el paradigma de agentes descrito previamente. Las RNA seleccionadas para realizar las distintas pruebas corresponden a redes neuronales de tiempo discreto (Discrete-Time Neural Networks) del tipo PNN y FFNN. Se realizaron dos tipos de pruebas: con redes neuronales no recurrentes y recurrentes. En las pruebas del primer tipo, se usaron redes neuronales del tipo PNN y FFNN, mientras que para el segundo tipo, se agrega recurrencia en la capa oculta de la anterior red. La red FFNN no recurrente fue usada como una red simple similar a la desarrollada para el vehículo de Braintenberg [21], la cual produce una señal de salida en respuesta directa al rango actual de la lectura de los sensores. Las redes recurrentes previamente enunciadas, posibilitan realizar procesamiento temporal. Las conexiones recurrentes permiten que la red recuerde la acción tomada en etapas temporales previas. Según [2], los controladores que permiten el uso de información temporal poseen el potencial de superar a los controladores reactivos que simplemente utilizan los sensores del robot.

La topología de la RNA empleada para implementar el controlador puede apreciarse en la figura 2. En ella se ve que dos neuronas con funciones de activación del tipo sigmoide fueron utilizadas en la capa oculta de la red. La capa de salida consiste de una neurona con función de activación también sigmoide.

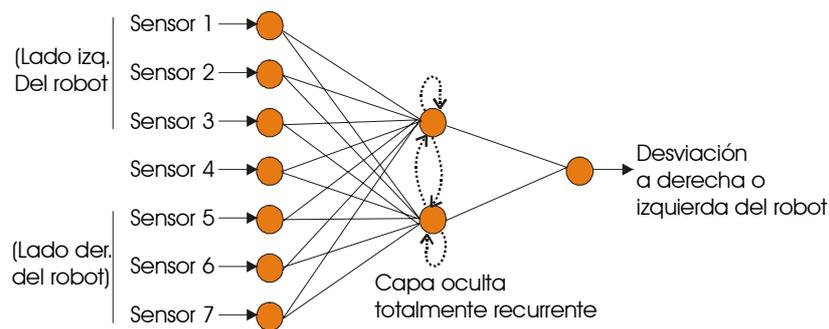


Fig. 2 – Topología de la red neuronal. El nivel de entradas consiste de siete receptores totalmente conectados a dos neuronas ocultas. Un conjunto de conexiones recurrentes son incorporadas a la capa oculta solo para implementaciones recurrentes. Las neuronas ocultas no están conectadas a los motores-actuadores del robot, en cambio están conectadas a una neurona de salida la cual modela el ángulo de desviación del robot.

Puede observarse que cuando los pesos sinápticos recurrentes (flechas de punto) se inicializan con el valor cero, es posible remover las conexiones recurrentes para implementar redes no recurrentes. En base a esta topología, se realizaron modificaciones para obtener las siguientes configuraciones:

FFNN:

- Una red FFNN simple sin recurrencia en la capa oculta contra una FFNN con recurrencia en la capa oculta
- Inicialización aleatoria de los pesos sinápticos (RANDOM-INIT) contra una inicialización de los signos/pesos (PRESEL-INIT) para una red FFNN sin recurrencia
- Mutaciones de pesos aleatorias (TOTAL-MUTATION) contra mutaciones controladas parciales (PARTIAL-MUTATION) para FFNN con recurrencia.

PNN:

- Inicialización de signo aleatoria
- Inicialización de signo definida a priori

2.3. Ambiente de Simulación

Para realizar las pruebas se emplearon dos ambientes de simulación: uno para el aprendizaje y otro para la validación. Ambas representan un robot como el de la figura 1, con sensores de proximidad y de luz.

El ambiente de simulación de aprendizaje estuvo compuesto por objetos bidimensionales (x, y) de forma rectangular que ocupa un lugar del ambiente. El número de objetos para las pruebas se fijó en 10 y su posición dentro del área de trabajo fue aleatoria. En cada generación, los miembros de la población (diferentes instancias de neuro-controladores para el robot) también fueron inicializados en posiciones aleatorias dentro del ambiente. Además de los obstáculos, el ambiente contó con una fuente de luz dispuesta en una posición al azar para cada corrida. Esta luz representó la meta final a ser alcanzada por el robot. Cada ambiente, que permite la simulación de un único robot por corrida, fue especificado como una estructura conteniendo la posición actual del robot y la posición de los objetos (obstáculos y luces). En suma, tanto la estructura de cada neuro-controlador como los algoritmos evolutivos son asociados a cada instancia de robot generada para el aprendizaje. En este ambiente de simulación no se utiliza un modelo del robot que tenga en cuenta dinámicas ni no linealidades debido a los actuadores motores y ruedas. Sólo se tiene en cuenta en cada paso de

simulación el ángulo de desviación del robot respecto de su objetivo, reduciendo el problema a la simulación cinemática solamente.

Una vez concluida la etapa de aprendizaje, la mejor instancia de robot se seleccionó y evaluó en otro ambiente de simulación, el simulador YAKS [22], dado que este nuevo ambiente de simulación presenta un modelo del entorno más realista. Los resultados obtenidos en tal simulación, son cualitativamente similares a los alcanzados en el ambiente de aprendizaje. La decisión de no usar un único ambiente de simulación obedeció a que la mayor complejidad del YAKS (y por ende sus resultados más realistas) implican un retardo significativo en la obtención de las distintas generaciones de neuro-controladores para el robot. Esto se prevé unificar en un futuro próximo.

2.4. El algoritmo evolutivo

El algoritmo evolutivo fue empleado en la etapa de aprendizaje de los neuro-controladores, con las particularidades de cada caso. En efecto, en el caso de controladores tipo PNN se determinaron las reglas Hebbianas tal como en [7, 18], mientras que para redes tipo FFNN las conexiones sinápticas se inicializaron con valores aleatorios en el rango real $[-2, 2]$, para la configuración RANDOM-INIT; o valores aleatorios en el rango real $[0, 2]$ con preselección de signos, para la configuración PRESEL-INIT.

La estructura de datos del cromosoma (Tabla 1) representa al neuro-controlador de la Figura 2. Para controladores genéticamente determinados (como el FFNN), un signo y una magnitud de peso para cada sinapsis; y para controladores de sinapsis adaptables (como el PNN), un signo, una regla Hebbiana específica, y una tasa de aprendizaje).

Codificación del genotipo	Valor para una sinápsis		
	FFNN	Signo	Peso sináptico
PNN	Signo	Regla Hebbiana	Tasa

Tabla 1 – Codificación genética de los parámetros sinápticos. Para controladores FFNN, un peso con signo para cada sinápsis. Para controladores PNN, un signo, una (de cuatro) regla Hebbiana y una tasa de mutación [18].

En el ámbito de las mutaciones de neuro-controladores en una escala evolutiva, la diversidad se refiere a la libre elección aleatoria de los valores genéticos. Por el contrario, el criterio de mutación se refiere a una selección controlada de los parámetros de mutación genética (p. ej. mutación de pesos sinápticos y su tasa de mutación). Como se ve en [20], la adaptación ignora la diversidad.

En este trabajo, los pesos fueron mutados con una tasa fija de 50%. El criterio de mutación adoptado para este trabajo fue el siguiente. Cada magnitud de peso para una conexión sináptica en una red FFNN (controlador genéticamente determinado) depende de la adaptación acumulada durante el proceso evolutivo [11], y es afectado por una tasa de mutación aleatoria en el rango $[-2; 2]$, para la configuración TOTAL-MUTATION; o una variación en el rango $[-0.25, 0.25]$ del peso original mutado, para la configuración PARTIAL-MUTATION válida solamente para FFNN. Esta pequeña tasa de adaptación provoca mínimas mutaciones del neuro-controlador en una escala evolutiva.

En el caso de FFNN, los pesos (w) fueron mutados usando la siguiente ecuación:

$$w = w + u * R \tag{ecuación 1}$$

donde u es la tasa de mutación en el rango $[0; 1]$ y R es un valor aleatorio en el rango $[-0.25; 0.25]$. El efecto de la ecuación de mutación (1) es una pequeña variación del controlador de una generación a la siguiente. Por el contrario, Para redes PNN el criterio de mutación se refiere solamente a cambios en el signo, mientras que las variaciones en magnitud de los pesos son determinadas por reglas Hebbianas durante el proceso de evaluación del controlador. Este criterio de mutación permite que un neuro-controlador con buen fitness esté junto con varios de sus descendientes en la generación siguiente en el proceso de mutación genética. Como resultado de esto, los descendientes estarán también apropiadamente adaptados al medioambiente, tomando ventaja de la experiencia adquirida.

2.5. La Función de Fitness

La evaluación del desempeño de cada controlador está basada en una variante de la función de fitness mostrada en [2]. Básicamente, el fitness de un controlador representa la cantidad de movimientos exitosos respecto del total de movimientos que son posibles de realizar. En este trabajo se consideraron dos parámetros asociados al incremento de la función de fitness para cada instante de tiempo. Estos son descritos a continuación:

$$F(c_i) = \begin{cases} F(c_i) + I & \text{si } k_1 \wedge k_2 \\ F(c_i) & \text{otro} \end{cases} \quad (\text{ecuación 2})$$

dónde k_1 refleja el bloqueo de un robot cerca de un objeto, por lo que aquellos controladores que presenten mayor cantidad de situaciones de bloqueo, recibirán un menor nivel de fitness que aquellos que presenten acciones de evasión de obstáculos; y k_2 privilegia a aquellos controladores que presenten movimientos positivos de avance hacia el objetivo (obtenida mediante los valores de los sensores de luz inferiores a un umbral), dándole un incremento a su performance en cada acción exitosa. Cabe destacar que el valor de fitness señalado, está normalizado respecto del número máximo de posibles acciones exitosas durante la vida de un controlador (iteraciones), para cada uno de los escenarios (corridas) en los que es probado tal controlador. La Figura 3 muestra la evolución de la función de fitness normalizada en el rango $[0; 1]$.

Un robot que no puede evadir objetos terminará inmovilizado al bloquearse su recorrido obteniendo, en consecuencia, un bajo resultado de fitness.

3. Resultados y Discusión

Las capacidades de evasión de obstáculos y navegación son utilizadas en diversas pruebas en la literatura de RE [3]. Consisten en el entrenamiento evolutivo de controladores neuronales específicos a fin de adquirir capacidades de evasión de objetos en un medioambiente determinado o navegar hacia un punto específico en un área cerrada [2, 19].

En nuestros trabajos previos (p.e., [24][25]) y en [2], se encontró que redes neuronales del tipo Feed-Forward con una sola capa oculta eran capaces de controlar agentes robóticos en un ambiente simulado, similar al usado en este trabajo. Esto es posible cuando los sensores del robot son aproximaciones simples de sensores reales (de comportamientos no lineales, generalmente). Efectivamente, se pudo implementar un simple vehículo tipo Braitenberg para tareas de evasión de obstáculos con, razonablemente, buenos resultados en el ambiente simulado. El controlador Braitenberg fue seleccionado ya que no necesita información de estados pasados para realizar las

tareas de percepción. Estos controladores, por lo tanto, carecen de capacidad de procesamiento temporal.

Para las pruebas presentadas en este trabajo, siete alternativas de redes FFNN y PNN fueron entrenadas con 30 genotipos (de 20 cromosomas cada uno) sobre 200 iteraciones de 10 corridas cada una para cada neuro-controlador (fenotipo). Los controladores obtenidos muestran diferentes características dependiendo de las alternativas de inicialización de pesos y de mutación utilizadas. En particular, neuro-controladores con pesos o signos preseleccionados a priori y/o mutación controlada evolucionaron mejor que otros con inicialización y mutación aleatoria. Los agentes robóticos usados en cada uno de las pruebas enunciadas fueron entrenados evolutivamente con un criterio de mutación estándar, para PNN y con uno generacional, para FFNN.

Se encontró que la configuración TRNN_PRESEL_MUTPAR sobre FFNN produjo los mejores resultados en menor tiempo de simulación (generaciones). Las configuraciones de redes descritas en la sección 2.1 fueron probadas con pesos sinápticos fijos, y con pesos inicializados aleatoriamente (Fig. 3). De acuerdo a los resultados obtenidos, los controladores con signos/pesos preseleccionados y/o mutación controlada de pesos evolucionaron mejor que otros que no trabajaban con esas características.

La Figura 3 muestra el *fitness* promedio generado durante una corrida típica para las configuraciones de redes FFNN analizadas. En cada generación se obtuvo el mejor y peor *fitness* y el *fitness* promedio de cada neuro-controlador.

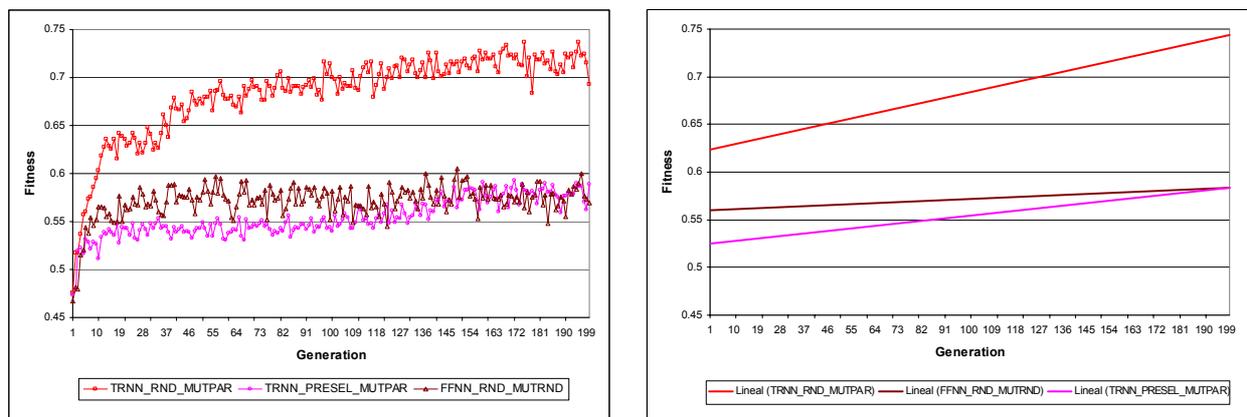


Fig. 3 – Resultados de tres generaciones de evolución. (a) Muestra la función de *fitness* de todas las generaciones para el mejor (TRNN_PRESEL_MUTPAR), intermedio (FFNN_RND_MUTRND), y peor (TRNN_RND_MUTPAR) controlador estudiado; (b) muestra la tendencia lineal de los controladores mostrados en (a). Los controladores evidencian las diferencias de performance relacionadas con las distintas alternativas de inicialización y mutación.

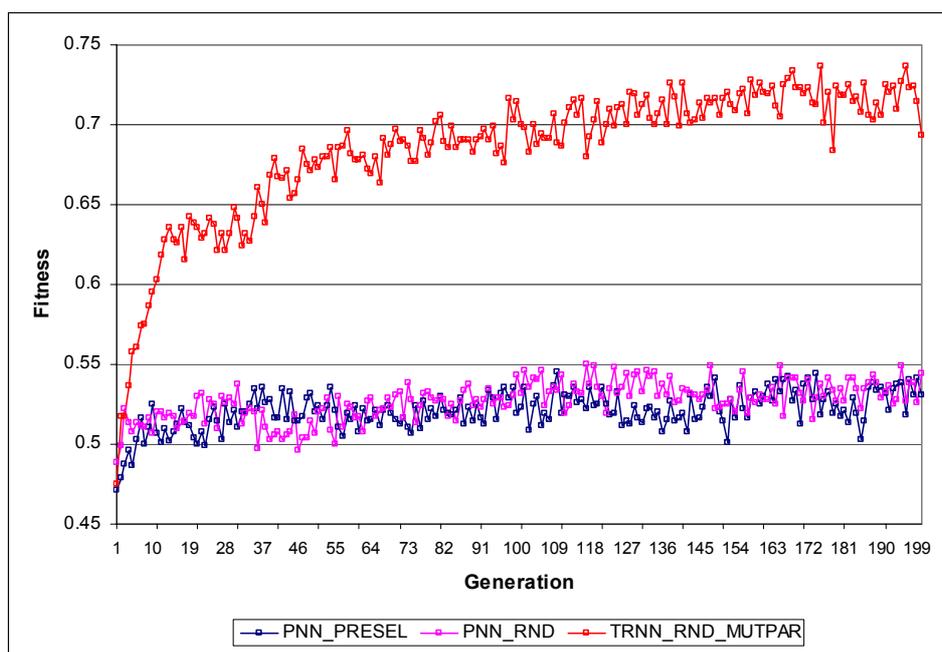


Fig. 4 – Comparación de fitness entre el mejor controlador de la figura 3 (TRNN_PRESEL_MUTPAR) y las dos alternativas de controladores PNN (con mutación total y controlada del signo de los pesos).

La figura 4 muestra el fitness promedio generado durante una corrida típica para las configuraciones de redes PNN (con mutación total y controlada de los signos de los pesos sinápticos) y una red FFNN recurrente con inicialización de signos de los pesos a priori y mutación controlada (mejor controlador de la figura 3).

Los resultados de los criterios de mutación seleccionados en este trabajo (sección 2.3) muestran que la mutación controlada (PNN_PRESEL en azul de la figura 4) no presenta grandes ventajas respecto de la mutación totalmente al azar (PNN_RND en magenta de la figura 4). Esto muestra que la diversidad es más potente que las mutaciones controladas a la hora de determinar los controladores más apropiados.

La simulación también muestra que el agente robótico es capaz de evitar obstáculos presentes dentro de su rango de sensores con la configuración de redes neuronales recurrentes FFNN o PNN. Además, se observó que el robot pierde la referencia de posición de la fuente de luz cuando pasa cerca de dicho punto, debido a que los sensores correspondientes no reciben su influencia por estar al frente del robot.

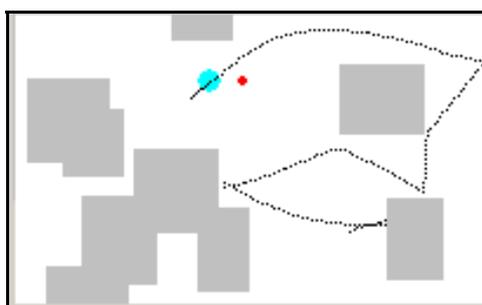


Fig. 5. a – Ejemplo de movimiento simulado de un robot con un neuro-controlador entrenado para evasión de obstáculos basado en una red no recurrente tipo FFNN con inicialización del signo de los pesos aleatorios (random_init). Las líneas muestran la trayectoria tomada por el robot durante el transcurso de la simulación.

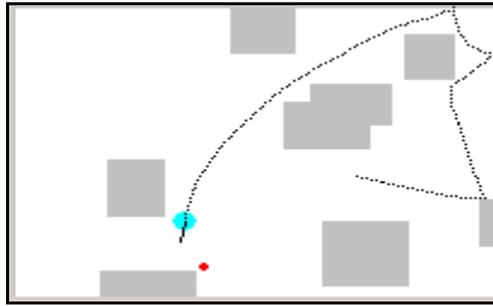


Fig. 5. b - Ejemplo de movimiento simulado de un robot con un neuro-controlador entrenado para navegación basado en una red recurrente tipo PNN con inicialización del signo de los pesos aleatorio (random_init). Las líneas muestran la trayectoria tomada por el robot durante el transcurso de la simulación y el punto rojo representa la fuente de luz o punto de meta del robot.

4. Conclusiones y trabajos futuros

El trabajo presentado describe, en términos generales, controladores robóticos instanciados y entrenados en un ambiente simulado para evasión de obstáculos. Diferentes neuro-controladores fueron entrenados en simulación y el mejor de ellos fue seleccionado en cada generación a fin de obtener el controlador resultante más apropiado para la tarea. El modelo del robot fue desarrollado basado en criterios evolutivos y adaptativos.

Algunos neuro-controladores mostraron comportamientos particulares tales como la tendencia a realizar trayectorias circulares luego de evitar un obstáculo. Este comportamiento será estudiado en el futuro. También se ampliará el estudio de la disposición de los sensores en el robot a fin de minimizar comportamientos erráticos del mismo cuando alcanza la meta.

Este trabajo demuestra una vez más la viabilidad del uso de controladores basados en redes neuronales en robótica evolutiva, mostrando sus potenciales en lo que concierne a adaptabilidad y capacidad de aprendizaje.

Trabajos futuros estarán orientados al desarrollo de neuro-controladores con arquitecturas similares a las presentadas y orientadas al control de robots reales en ambientes físicos estáticos y móviles.

Referencias

- [1] Hebb, D. O. A Textbook of Psychology. Philadelphia, PA., W. B. Saunders, 1958, pp 44-45.
- [2] Nelson, A. L.; Grant, E.; Galeotti, J. M.; Rhody, S. Maze exploration behaviors using an integrated evolutionary robotic environment. *Robotic and Autonomous Systems* 46. 2004, pp. 159-173.
- [3] Nolfi S. and Floreano, D. *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*. MA: MIT Press/Bradford Books. 2000.
- [4] Tuci, E.; Harvey, I.; Quinn, M. Evolving integrated controllers for autonomous learning robots using dynamic neural networks. *Proceedings of The Seventh International Conference on the Simulation of Adaptive Behavior (SAP'02)*, 4-9 August 2002, Edimburgh, UK.
- [5] Lipson, H. *Uncontrolled Engineering: A review of Nolfi and Floreano's Evolutionary Robotics*. 2000.

Robotics. 2000.

- [6] Urzelai, J.; Floreano, D. Evolution of adaptive synapses: Robots with fast adaptive behavior in new environments. *Evolutionary Computation*, 9:495-524. 2001.
- [7] Tuci, E.; Quinn, M. Behavioural plasticity in autonomous agents: a comparison between two types of controller. *Proceedings of The Second European Workshop on Evolutionary Robotics EvoROB2003*, 14-16 April 2003, Essex, UK, pp. 661-672.
- [8] Russell, S.; Norvig, P. *Artificial Intelligence: A Modern Approach*, Prentice-Hall, Inc. 1995.
- [9] Cliff, D.; Harvey, I.; Husbands, P. Explorations in Evolutionary Robotics. *Adaptive Behavior*, 2:73-110. 1993.
- [10] Nolfi, S. Adaptation as a more powerful than decomposition and integration: Experimental evidences from evolutionary robotics. In P. K. Simpson (Ed.), *Proceedings of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'98)*, New York: IEEE Press, 141-146. 1998.
- [11] Floreano, D. *Ago Ergo Sum*. In Mulhauser, G. Editor, *Evolving Consciousness*. Benjamins Press, New York. 1997.
- [12] Steels, L. Building agents out of autonomous behavior systems. In L. Steels and R. Brooks, editors. *The Artificial Life route to Artificial Intelligence building situated embodied agents*, pp. 102-137. Lawrence Erlbaum, New Haven, 1993.
- [13] Fernando, C. *Accumulation of Adaptations in Plastic Neural Networks*. MSc. Dissertation. COGS, University of Sussex. 2002.
- [14] Floreano, D.; Mondada, F. Evolutionary Neurocontrollers for autonomous Mobile Robots. *Neural Networks*, 11:1461-1478, 1998.
- [15] Driscoll, J. A.; Peters, R. A. II. A development environment for evolutionary robotics. *Proceedings of the 2000 IEEE International Conference on Systems, Man, and Cybernetics*, vol., pp. 3841-3845. 2000.
- [16] Floreano, D.; Urzelai, J. Neural morphogenesis, synaptic plasticity, and evolution. *Theory in Biosciences*, 120 (3-4), 223-238. 2001.
- [17] Nolfi, S.; Floreano, D. Learning and evolution. *Autonomous Robots*, 7(1): 89-113, 1999.
- [18] Floreano, D.; Urzelai, J. (1999) Evolution of Adaptive-Synapse Controllers. In D. Floreano et al. (Eds.), *Advances in Artificial Life*. *Proceedings of the 5th European Conference on Artificial Life*, Berlin: Springer Verlag. (ECAL'1999). 1999.
- [19] Togelius, J. Evolution of the Layers in a Subsumption Architecture Robot Controller. Master of Science in Evolutionary and Adaptive Systems. University of Sussex, UK. 2003.
- [18] Floreano, D.; Urzelai, J. (1999) Evolution of Adaptive-Synapse Controllers. In D. Floreano et al. (Eds.), *Advances in Artificial Life*. *Proceedings of the 5th European Conference on Artificial Life*, Berlin: Springer Verlag. (ECAL'1999). 1999.
- [19] Togelius, J. Evolution of the Layers in a Subsumption Architecture Robot Controller. Master of Science in Evolutionary and Adaptive Systems. University of Sussex, UK. 2003.

of Science in Evolutionary and Adaptive Systems. University of Sussex, UK. 2003.

- [20] Goldberg, D. E. Genetic Algorithms in search, optimization, and machine learning. Addison-Wesley. 1989.
- [21] Braitenberg, V. Vehicles: Experiments in synthetic psychology. Cambridge, MA: MIT Press. 1984.
- [22] Karlson, J. YAKS Yet Another Khepera Simulator. URL: <http://r2d2.ida.his.se/> . 2002.
- [23] Téllez, R. A.; Angulo, C. Towards a general architecture for scaling up in evolutionary robotics. Technical Report. Spain: University of Catalonia. 2004.
- [24] Fernández León, J. A.; Tosini, M.; Acosta, G. G. Evolutionary Reactive Behavior for Mobile Robots Navigation. IEEE Conference on Cybernetics and Intelligent Systems (CIS). Proceedings of the 2004 IEEE CIS, Singapore, pp. 532-537. 2004.
- [25] Fernández León, J. A.; Acosta, G. G.; Mayosky, Miguel A. Estudio de Neuro-Controladores Evolutivos para Navegación de Robots Autónomos. Maestría en Ingeniería de Sistemas. UNCPBA, Argentina. 2005.
- [26] Khepera, mini robot. K-Team. <http://www.k-team.com/robots/khepera/index.html>. 2004.