



Ingeniare. Revista Chilena de Ingeniería

ISSN: 0718-3291

facing@uta.cl

Universidad de Tarapacá

Chile

Correa, Felipe; Gallardo, José; Muñoz, Nelson; Pérez, Ricardo
Estudio comparativo basado en métricas para diferentes arquitecturas de navegación
reactiva

Ingeniare. Revista Chilena de Ingeniería, vol. 24, núm. 1, enero, 2016, pp. 46-54

Universidad de Tarapacá

Arica, Chile

Disponible en: <http://www.redalyc.org/articulo.oa?id=77243535005>

- Cómo citar el artículo
- Número completo
- Más información del artículo
- Página de la revista en redalyc.org

redalyc.org

Sistema de Información Científica

Red de Revistas Científicas de América Latina, el Caribe, España y Portugal

Proyecto académico sin fines de lucro, desarrollado bajo la iniciativa de acceso abierto

Estudio comparativo basado en métricas para diferentes arquitecturas de navegación reactiva

Comparative study based on metrics for various reactive navigation architectures

Felipe Correa¹ José Gallardo¹ Nelson Muñoz² Ricardo Pérez¹

Recibido 24 de noviembre de 2014, Aceptado 8 de mayo de 2015

Received: November 24, 2014, Accepted: May 8, 2015

RESUMEN

El presente estudio compara cuatro arquitecturas de navegación autónoma para robots móviles, con el fin de determinar cuál presenta una mayor eficiencia en la realización de tareas. Las arquitecturas comparadas son: reactivo puro, redes neuronales, lógica difusa y una arquitectura híbrida neuro-difusa. La comparación se realizó recopilando datos de las métricas de desempeño basadas en seguridad, suavidad de curva (consumo energético), velocidad y distancia, en la plataforma robótica Pioneer 3-DX en un entorno simulado con MobileSim, el software oficial de MobileRobot®. Los datos obtenidos fueron analizados estadísticamente, generando una base formal de comparación. Los resultados muestran que la arquitectura neuro-difusa presenta mayores niveles de seguridad y ahorro energético que las otras arquitecturas, demostrando así, que la utilización de arquitecturas en forma sinérgica logra mayor eficiencia.

Palabras clave: Navegación autónoma, robótica móvil, navegación reactiva, redes neuronales, lógica difusa, neuro-difuso, métricas.

ABSTRACT

This study compares four autonomous navigation architectures for mobile robots, in order to determinate which has a greater efficiency in tasks solving. These architectures are: reactive, neuronal network, fuzzy logic and neuro-fuzzy hybrid architecture. The contrast within architectures was assessed collecting data of performance metrics based on safety, smoothness of curve (energy consumption), speed and distance. The comparison was done using the robotic platform, Pioneer 3-DX within a simulated environment of MobileSim, the official software of MobileRobot®. The collected data were statistically analyzed, generating, therefore a formal base of comparison. The outcomes of this procedure indicate that neuro-fuzzy architecture presents higher levels of security and energy efficiency in contrast with other architectures. Thus, using synergic architectures allows achieving a greater efficiency.

Keywords: Autonomous navigation, mobile robotic, reactive navigation, neuronal network, fuzzy logic, neuro-fuzzy, metrics.

NAVEGACIÓN REACTIVA

En robótica móvil es importante tomar en cuenta que los vehículos deberán operar bajo la consideración

de la incertidumbre de sus sensores, que los entornos por los que se desplazaran pueden ser desconocidos y por lo tanto que deben responder adecuadamente a los cambios dinámicos [1]. Los

¹ Departamento de Ingeniería de Sistemas y Computación. Universidad Católica del Norte. Av. Angamos 0610. Antofagasta, Chile. E-mail: felipe.correa.santis@gmail.com; jgallardo@ucn.cl; rperez@ucn.cl

² Facultad de Ingenierías. Politécnico Colombiano Jaime Isaza Cadavid. Carrera 48 #7-151. Medellín, Colombia. E-mail: ndmunoz@elpoli.edu.co

vehículos autoguiados, conocidos por sus siglas en inglés AGV (Automated Guided Vehicle) definen un campo ampliamente investigado de la robótica móvil, la aplicación de estos vehículos varía en función de las necesidades como por ejemplo: transporte de materiales, exploración de zonas desconocidas o peligrosas y apoyo a personas con capacidades especiales entre otras [2]. En la actualidad los AGV son muy utilizados en la industria [3].

Uno de los desafíos de la robótica móvil, es lograr dotar a los AGV de la mayor flexibilidad y adaptación en entornos cambiantes o desconocidos. Debido a la carencia de información del robot sobre su entorno, los algoritmos reactivos deberán trabajar con sistemas sensoriales que les permiten obtener información fiable del entorno mientras el robot se desplaza [4]. Los sensores permiten al robot, mediante un algoritmo, planificar tareas de forma dinámica, mejorando la percepción y por ende su adaptabilidad [2].

En esta área existen dos conceptos altamente diferenciados, la planificación local y la global. La planificación global, es el componente deliberativo y se enfoca en realizar la planificación de acciones a largo plazo [5], es decir, en función de la completa información del entorno, planifica las trayectorias o las rutas óptimas a seguir por el robot móvil. Mientras que la planificación local es el componente reactivo, y apunta a la evasión de obstáculos en tiempo real en entornos con información incompleta o desconocida. De esta última forman parte varias de las técnicas más llamativas de la robótica móvil [4].

Los controladores modernos operan en una estructura de capas que integra estos dos comportamientos. El comportamiento reactivo opera en la capa inferior y tiene la responsabilidad de evadir los obstáculos [6]. La capa intermedia se denomina ejecutiva y opera con la información proporcionada por la capa reactiva, su función es realizar la planificación a corto plazo, también puede encargarse de realizar la localización y el mapeo [7]. En la capa superior se ejecuta el comportamiento deliberativo, en esta se ejecutan las operaciones de planificación y toma de decisiones a mediano y largo plazo [7].

Sin embargo la operación de robots móviles presenta limitaciones que generan pérdida de su efectividad,

estas limitaciones puede ser producto del desgaste normal de sus componentes, variaciones repentinas en la carga, la combinación de la información de diferentes sistemas de sensores, o la coordinación de diferentes módulos de control que gobiernan el robot, etc. Es por esto que se ha optado por incorporar técnicas inteligentes al control, con el fin de satisfacer los requerimientos demandados por la operación. Destacan los algoritmos genéticos, la lógica difusa y las redes neuronales entre las técnicas utilizadas con este propósito [8-1].

Cada una de las técnicas antes mencionadas posee características especiales, las cuales son propicias para resolver diferentes tipos de problemas. Pero la complejidad presente en los problemas de navegación en la robótica móvil ha motivado la implementación conjunta de estas técnicas, para así lograr mejores resultados de lo que se lograría de forma independiente [1].

La lógica difusa es una técnica de inteligencia artificial que permite transformar sentencias complejas del lenguaje natural en formalismos matemáticos [9], permitiendo así, expresar el conocimiento del experto en términos del lenguaje natural [10].

Su estructura opera en función de reglas en lenguaje natural y funciones de pertenencia, lo cual permite realizar operaciones similares a las que realizaría un humano [5]. Esto le entrega gran robustez frente a los cambios de estado y le permite operar información con un alto nivel de incertidumbre [9].

Por su parte las redes neuronales artificiales, pretenden emular el funcionamiento del cerebro humano a través de una arquitectura de alta conectividad entre neuronas [5], propiedad que les permite aprender y adaptarse situaciones desconocidas [10]. Estas características permiten a las redes neuronales artificiales (RNA), resolver problemas de reconocimiento, identificación y clasificación [5].

El entrenamiento más comúnmente utilizado para las RNA es el Backpropagation, el cual consiste en un tipo de entrenamiento supervisado. Este entrenamiento emplea un ciclo de propagación – adaptación de dos etapas y dos conjuntos de datos, uno de entrenamiento y otro de prueba.

En la primera etapa, denominada propagación hacia delante, la red calcula el valor de salida en función del patrón presentado. El valor resultante se compara con el valor esperado y se obtiene la señal del error, con lo que se da inicio a la segunda etapa, la cual se denomina propagación hacia atrás o retropropagación. En esta etapa se actualizan los pesos sinápticos de las neuronas en función de la contribución de la neurona con el valor resultante y el error obtenido. Este proceso se realiza en ciclos, hasta que la red converge, es decir, alcanza un valor determinado de error [1].

Al revisar las técnicas antes mencionadas se puede notar claramente sus diferencias y también sus potencialidades, y son estas las que han motivado a muchos investigadores a trabajar en soluciones que combinen sinérgicamente la lógica difusa y las redes neuronales [10].

MÉTRICAS

En la robótica móvil muchas investigaciones se centran en el diseño y optimización de las arquitecturas de control, con el objetivo de mejorar el desempeño del robot. Sin embargo las métricas son un tema al cual han dado poca importancia, limitándose a medir indicadores simples como distancia recorrida o tiempo empleado. Esta situación dificulta la comparación objetiva de los controladores y además impide evaluar correctamente los avances en este campo, como también medir sus diferentes capacidades [11].

Sin embargo, al aplicar un conjunto de métricas enfocadas a comparar las características de la navegación misma, como por ejemplo, riesgo de colisión y suavidad de trayectoria; de forma ordenada y sistemática, se obtiene una base de comparación para algoritmos de control, ayudando así a poder determinar cuál de ellos es mejor para su implantación en un robot móvil [11].

Para determinar y comparar bajo una base formal el riesgo de colisión de diversas arquitecturas de control reactivo, las métricas seleccionadas para cumplir con este fin son las métricas de seguridad. Estas representan el nivel de seguridad presente en la trayectoria del robot móvil, considerando la distancia medida entre el vehículo y los objetos durante su misión [11].

- **Medida de Seguridad 1 (SM1):** Representa la distancia promedio de todas las mediciones durante el transcurso de su misión [11].
- **Medida de Seguridad 2 (SM2):** se calcula como el promedio de las menores mediciones realizadas por los sensores, por ciclo de control, durante el transcurso de su misión [11].

Las medidas de seguridad permiten comparar cuál de las distintas arquitecturas posee un mejor desempeño para evadir obstáculos, pero no permite determinar cuál de ellas presenta un comportamiento más eficiente a la hora de realizar esta tarea.

Un robot que realiza trayectorias suaves demuestra consistencia en la relación decisión – acción, la capacidad de anticiparse y responder a eventos de forma oportuna. Además, las trayectorias están estrechamente relacionadas con la energía y tiempo, por lo cual, la suavidad de ellas se refleja en ahorros de energía y tiempo [11].

- **Energía de Curvatura (Bending Energy BE):** Esta métrica depende de la función de curvatura k (ver ecuación (1)). Esta métrica se calcula como la sumatoria de los cuadrados de la curvatura (k) en cada punto de la trayectoria [11] como se muestra en ecuación (2).

$$k(x_i) = \frac{f''(x_i)}{(1 + (f'(x_i))^2)^{\frac{3}{2}}} \quad (1)$$

$$B_E = \frac{1}{n} \sum_{i=1}^n k^2(x_i, f(x_i)) \quad (2)$$

- **TB_E:** Esta métrica considera tanto la suavidad como la longitud de la trayectoria, a diferencia de B_E, esta métrica no es un promedio de los valores de curvatura en los puntos, por el contrario es el valor acumulado, por lo cual es una fuerte representación de la suavidad total de la trayectoria descrita por el robot [11]. En la ecuación (3) se presenta la fórmula de esta métrica.

$$TB_E = \sum_{i=1}^n k^2(x_i, f(x_i)) \quad (3)$$

Mientras los valores logrados en las métricas de suavidad sean más bajos, se estará en presencia de un mejor algoritmo de control, el cual es eficiente al controlar la trayectoria, realizar acciones para anticiparse a los eventos del entorno y poseer un menor costo energético.

Adicionalmente a las métricas ya descritas, resulta útil considerar valores simples de obtener como velocidad promedio y distancia recorrida, para así lograr una mayor comprensión del comportamiento general del robot. Para el caso de la velocidad promedio basta con calcular el promedio de las velocidades instantáneas o bien dividir la distancia total recorrida por el tiempo utilizado siendo esta última, una medida menos certera. La distancia recorrida, puede calcularse como la sumatoria de la distancia recorrida cada periodo de control n entre dos puntos, desde un punto inicial i hasta uno final f , como se muestra en la ecuación (4).

$$d = \sum_{i=1}^n \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \quad (4)$$

EXPERIMENTO

El experimento consistió en la implementación de cuatro tipos de arquitecturas reactivas (reactivo puro, reactivo neuronal, difuso y neuro – difuso), las cuales fueron ejecutadas sobre la plataforma robótica Pioneer 3-DX, en el simulador de robot móviles MobileSim. Se utilizó la interfaz de ARIA y luego se compararon los resultados obtenidos.

MobileSim, forma parte del SDK de MobileRobot®, este software proporciona una conexión de control a través del puerto TCP que simula la conexión por el puerto serial en el robot real. También permite manejar los archivos de configuración que representan a los robots y sensores ultrasónicos, laser, odométricos con sus respectivos errores relativos. Además incluye todo el código fuente para entender, implementar, personalizar y mejorar las simulaciones [12].

La Librería ARIA es una biblioteca de C++ que proporciona una interfaz y marco para el control y la recepción de datos de las plataformas. Con ARIA, a través de las interfaces y servicios que proporciona, se puede obtener acceso y controlar

información vital del robot, por ejemplo controlar la velocidad del robot y monitorear su posición, el nivel de batería y leer los sensores ultrasónicos entre otros [12].

La suite de MobileRobot (MobileSim y ARIA) es una eficaz herramienta para la simulación y operación de robots móviles debido a que:

- La Librería ARIA se comporta de la misma forma en el entorno simulado como en el real, por lo tanto, el programa siempre opera como si estuviera ejecutándose en el robot real [12].
- El Simulador permite confeccionar nuevas configuraciones e incorpora errores de sensor y odométricos, lo cual le da al robot un comportamiento similar al real [12].

El robot Pioneer 3-DX (Figura 1), es un robot diferencial de tres ruedas, dos de ellas motrices y la tercera de apoyo. Consta de 3 baterías que le permiten una autonomía de 8 a 10 horas y un anillo ultrasónico de 16 sensores, (8 en la parte frontal y 8 en la parte trasera) Figura 2, y puede alcanzar una velocidad de hasta 1.2 metros por segundos [12].



Figura 1. Robot Pioneer 3-DX.

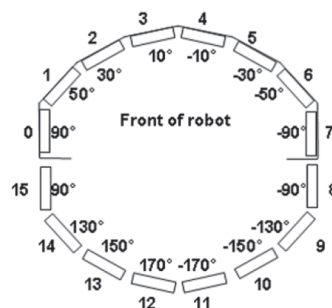


Figura 2. Distribución del anillo ultrasónico.

HIPÓTESIS PLANTEADA

A través de técnicas de redes neuronales y lógica difusa es posible lograr, un controlador inteligente que permita al robot móvil realizar acciones de evasión tempranas frente a obstáculos presentes y emergentes, variando su velocidad de desplazamiento, en función de la composición del entorno inmediato y distante. Con lo cual se conseguirá una mayor eficiencia en razón del tiempo de ejecución de una tarea.

ARQUITECTURAS UTILIZADAS

Para la realización de la comparación se implementaron cuatro arquitecturas de control, a saber: Reactivo puro, Neuronal, Difuso y Neuro-Difuso.

Arquitectura reactiva

En la arquitectura reactiva, se elaboró un algoritmo en base a condición-acción. Las condiciones establecidas para navegación, como se muestran en la Tabla 1, se establecieron en función de las lecturas de los sensores ultrasónicos, modificando así, las velocidades de las ruedas (Derecha e Izquierda) permitiendo al robot desplazarse en línea recta, virar o rotar sobre su propio eje.

Tabla 1. Arquitectura reactiva.

Condición	Acción
Si $df < 1$	$V_r = -240$ mm/s
	$V_l = 240$ mm/s
Si $dl > 1$ y $dr > 1$ y $df > 2.5$	$V_r = 480$ mm/s
	$V_l = 480$ mm/s
Si $dl > 1$ y $dr > 1$ y $df \leq 2.5$	$V_r = 240$ mm/s
	$V_l = 240$ mm/s
$dl > 1$ y $dr < 1$	$V_r = 240$ mm/s
	$V_l = 0$ mm/s
$dl < 1$ y $dr > 1$	$V_r = 0$ mm/s
	$V_l = 240$ mm/s

La Tabla 1 resume los 5 comportamientos programados al robot móvil Pioneer 3-DX, donde d_i representa la distancia medida por el sensor i (r: right, f: front, l: left); mientras que V_j representa la velocidad del motor j (r: right, l: left). De esta manera la arquitectura procesa las mediciones de distancia mínimas obtenidas por los sensores ultrasónicos y reacciona modificando las velocidades en el motor correspondiente, como se puede ver en la Figura 3.

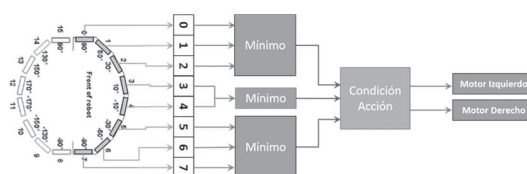


Figura 3. Configuración de la arquitectura Reactiva.

Arquitectura Neuronal

En la arquitectura Neuronal, se entrenó una red feed-forward del tipo Backpropagation, con la finalidad de identificar hitos (paredes, puertas y obstáculos) en el entorno.

La topología implementada en la red neuronal (RNA) consta de una capa de entrada, dos capas ocultas, debido a que el problema que se debe solucionar no es linealmente separable, requiere toma de decisiones precisas y mapeo fluido [13]. Por último una capa de salida. La Figura 4 ilustra una representación gráfica de las topologías y los problemas a resolver.

Estructura de la red	Tipo de región de decisión	Solución OR Exclusivo	clases con regiones geométricas	Formas de superficie de decisión mas generales
	Hiperplano simple			
	Regiones convexas abiertas o cerradas			
	Arbitrario (complejidad limitada por el número de nodos)			

Figura 4. Regiones de decisión para distintas topologías [14].

Una vez seleccionada la cantidad de capas ocultas en la RNA, se debió determinar la cantidad de nodos o neuronas presentes en cada capa. Este es un factor crítico en la implementación de la red, el cual afectara directamente la convergencia de esta [13].

Si bien, no existe un método exacto para determinar la cantidad de neuronas en las capas ocultas, existen tres heurísticas ampliamente aceptadas, las que proporcionan un punto de partida para la determinación de la topología de la RNA [15].

El modelo final de la RNA fue producto de un proceso iterativo de prueba y error [13]. La topología

resultante pudiese contar con más neuronas que las establecidas por las heurísticas, obteniendo así una RNA sobre dimensionada, la cual, de todos modos obtendría resultados concordantes con la generalización [16]. De manera que la RNA implementada consta de 3 neuronas en la capa de entrada, Dos capas ocultas, la primera con 9 neuronas y la segunda con 6. Finalmente una capa de salida con una neurona. En la Figura 5 se presenta gráficamente la topología de la RNA.

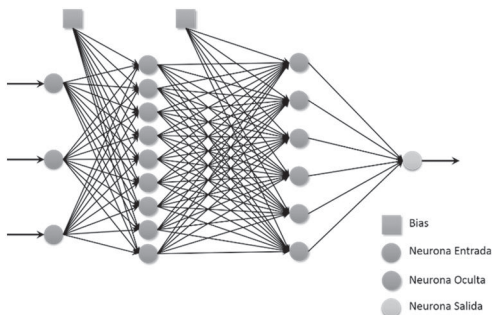


Figura 5. Topología de la red neuronal.

El propósito de la red neuronal, y el patrón para el cual fue entrenada, es recibir la información proveniente del entorno a través de los sensores ultrasónicos, determinando patrones en función de las discontinuidades del entorno y entregar valores que propicien una navegación más segura.

La configuración de la arquitectura neuronal, requiere dos RNA iguales, las cuales reciben la información proveniente de los sensores ultrasónicos ubicados en los flancos del robot. De esta manera la RNA puede detectar los hitos de forma correcta sin la influencia de lo que ocurre en el lado opuesto del robot. Finalmente los valores de salida de la RNA alimentan el mismo bloque de condición/acción que en el modelo reactivo puro, como se muestra en la Figura 6.

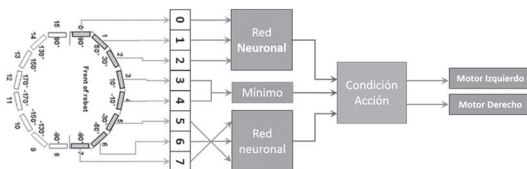


Figura 6. Configuración de la arquitectura Neuronal.

Arquitectura Difusa

La arquitectura difusa utiliza tres variables de entrada (left, front y right) cada una tipificada en cinco

valores lingüísticos (“Muy cerca”, “Cerca”, “Medio”, “Lejos”, “Muy Lejos”) y dos variables de salida, las velocidades del motor derecho e izquierdo, tipificadas en seis valores lingüísticos (“Extremo”, “Muy Lento”, “Lento”, “Normal”, “Rápido”, “Muy Rápido”) y 125 reglas para controlar su comportamiento, en la Figura 7 se muestran las funciones de pertenencia utilizadas en el motor de inferencia difusa.

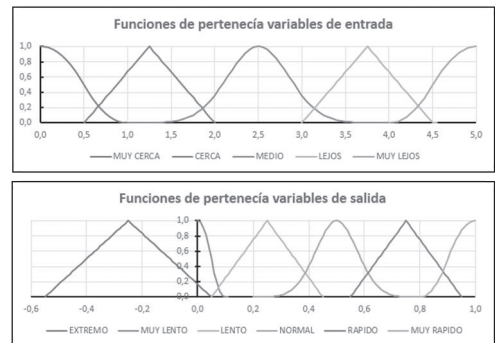


Figura 7. Funciones de Pertenencia.

Esta arquitectura alimenta al motor inferencia difusa con los mínimos valores sensados (Flanco derecho, Frontal y Flanco Izquierdo). El motor difuso opera fuzzyficando los valores de entrada, evaluando las reglas y calculando el predicado. Para así entregar como salida, los valores porcentuales con que deben operar los motores derecho e izquierdo del robot, como se puede ver en la Figura 8.

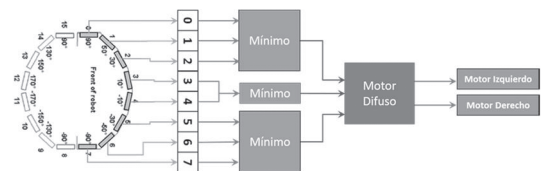


Figura 8. Configuración de la arquitectura Difusa.

Arquitectura Neuro-Difusa

La implementación de la arquitectura Neuro - Difusa fue diseñada con la red neuronal y el motor difuso ya mencionados, para evitar la ambigüedad entre las arquitecturas. Estas fueron configuradas de forma que, las salidas de la red neuronal alimentan directamente al motor de inferencia difusa. Es decir, la arquitectura Neuro-Difusa es una combinación de las ambas arquitecturas.

El anillo ultrasónico del robot sensa el entorno, alimentando con los valores de los flancos a las redes neuronales. Luego las RNA procesan la

información detectando los hitos del entorno, entregando un valor de distancia que propicie una navegación más segura.

El motor de inferencia difusa, es alimentado con las salidas de las RNAs y el valor mínimo de los dos sensores frontales. Este procesa la información de entrada y determina los valores porcentuales de las velocidades con que deben operar los motores del robot. La configuración de esta arquitectura se ilustra en la Figura 9.

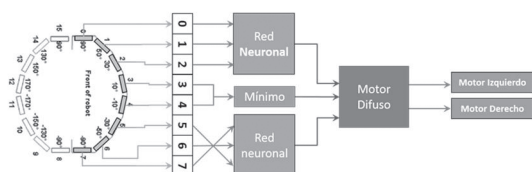


Figura 9. Configuración de la arquitectura Neuro-Difusa.

DISEÑO DEL EXPERIMENTO

Cada arquitectura fue probada 15 veces sobre el mismo entorno, con un intervalo de tiempo de 5 minutos, almacenando la información necesaria para calcular las métricas de comparación. El entorno seleccionado para estas pruebas es el plano de un departamento (ver Figura 10), el cual fue modificado, aumentando su nivel de complejidad para presentar un verdadero desafío a los controladores diseñados. El escenario diseñado para la experimentación considera 6 áreas trampa –zonas rayadas– en las cuales el controlador debe presentar un comportamiento inteligente y entendimiento del entorno para salir de estas.

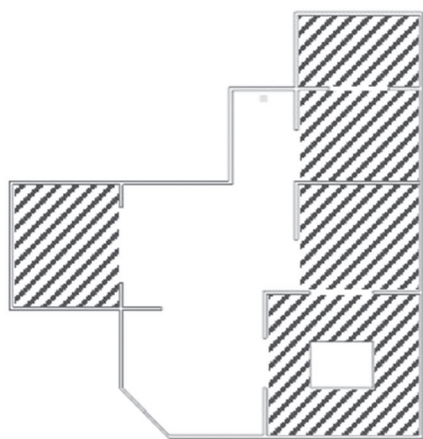


Figura 10. Ambiente para la simulación.

Es relevante destacar que en el caso del motor de inferencia difuso experimentó una tasa de colisión de un 45%, por lo cual no pudo completar los 5 minutos las 15 veces que fue probado. Para poder mantener la base de comparación, esta arquitectura tuvo que ser simulada un mayor número de veces.

RESULTADOS

El análisis comparativo de las cuatro arquitecturas fue realizado utilizando las métricas descritas. La Tabla 2 muestra los valores promedio de las métricas definidas para comparar el rendimiento y performance de los cuatro tipos de arquitecturas.

Tabla 2. Resultados de las simulaciones.

	Tbe	Be	SM1	SM2	d	vProm
Neuro difuso	908,88	0,72	3.761,20	2.151,24	196,81	0,54
Difuso	12.976,58	4,49	3.777,29	1.973,65	404,02	0,79
Neuronal	12.727,73	4,63	3.870,36	2.326,51	181,91	0,37
Reactivo	7.279,70	2,50	3.278,47	1.488,42	180,68	0,34

Para la validación de los resultados obtenidos se utilizó el análisis de varianza –ANOVA– de un solo factor, suponiendo la normalidad de los datos de acuerdo al teorema del límite central [17].

La hipótesis nula es, que las medias poblacionales de cada arquitectura, para una métrica determinada son iguales. Mientras que la hipótesis alternativa representa la no igualdad entre las medias poblacionales de cada arquitectura para cada métrica. La prueba ANOVA se le realizó a cada métrica por separado.

La primera métrica a la cual se le realizó la prueba ANOVA fue a TB_E , métrica que es el indicador más fuerte de un buen comportamiento en la evasión de obstáculos.

El análisis ANOVA para la métrica TB_E , muestra que la hipótesis nula es rechazada con un nivel de significancia del 95%. Por lo cual las medias poblacionales de las distintas arquitecturas no son similares [18]. El resultado del análisis ANOVA se muestra gráficamente en la Figura 11.

Tabla 3. Resultado prueba ANOVA.

RESUMEN					
Grupos	Cuenta	Suma	Promedio	Varianza	
Neuro-difuso	15	13633,2	908,9	249.993	
Difuso	15	194.648,8	12.976,6	47.899,573	
Neuronal	15	190.915,9	12.727,7	313.705.841	
Reactivo	15	109.195,4	7.279,7	85.294.503	

ANÁLISIS DE VARIANZA						
Origen de las variaciones	Suma de cuadrados	Grados de libertad	Promedio de los cuadrados	F	Probabilidad	Valor crítico para F
Entre grupos	1,46E+09	3,00	485.124.280	4,34	0,01	2,77
Dentro de los grupos	6,26E+09	56,00	111.787.478			
Total	7,72E+09	59,00				

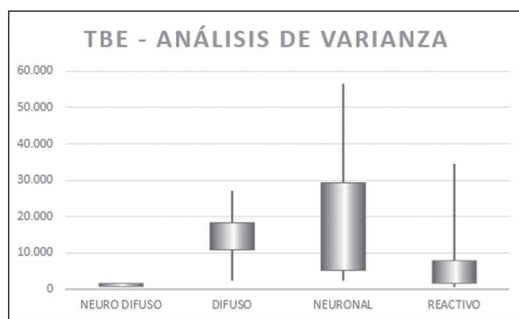


Figura 11. Gráfica de Análisis de Varianza para TBE.

Para poder determinar cuáles de estas medias son diferentes se aplicó el método de Tukey, con un nivel de significancia del 95%, dos grados de libertad 4 y 56. En consecuencia, la evidencia estadística muestra que el valor medio del TBE para un modelo Neuro-Difuso, será diferente y menor a las medias de las otras arquitecturas, como se puede ver en la Tabla 4 [17] y en la Figura 11.

Tabla 4. TBE, método de Tukey.

Neuro Difuso	Reactivo	Neuronal	Difuso
908.9	7,279.9	12,727.7	12,976.6

Este procedimiento se realizó con las otras métricas, primero se les aplicó el análisis ANOVA, y en todos los casos la hipótesis nula fue rechazada. Luego se aplicó la prueba de Tukey, para así determinar cuáles experimentos poseían medias poblacionales que pudieran ser similares.

CONCLUSIONES

El comportamiento del controlador Neuro – Difuso, si bien no es el que alcanza la mayor velocidad, es el segundo valor más alto. Junto con esto, esta arquitectura mostró mejor estabilidad tanto en la seguridad como en la suavidad de su trayectoria, situación que implica ahorro energético por parte del control.

La red neuronal implementada permitió que la configuración Neuronal y la Neuro-Difusa, realizaran acciones preventivas que lo posicionaran en zonas donde pudiera prever la aparición de obstáculos móviles. Solamente el control reactivo puro, no logro sortear las trampas del entorno, es decir que en algunas situaciones este controlador quedo realizando bucles indefinidos dentro de una estancia.

En conclusión, el controlador Neuro-Difuso, proporciona una arquitectura más eficiente, en base a las métricas utilizadas, navegando en forma segura, con un alto poder de evasión de obstáculos, siendo en cuanto al consumo de energía el más eficiente.

TRABAJOS FUTUROS

Para tener una mayor significación estadísticas de los resultados y poder comprar el rendimiento de cada arquitectura frente diferentes tipos de entornos, se debería realizar nuevamente el experimento, aumentando el número de simulaciones y la cantidad de entornos en los que se ejecuta.

AGRADECIMIENTOS

Se agradece a Sergio Correa Santis, Master of International Bussines, University of Queensland, Australia, por la traducción al inglés y el apoyo estadístico en el estudio.

REFERENCIAS

- [1] L. Rios, M. Bueno y S. Sanchez. “Generación de trayectorias para un robot móvil empleando redes neuronales”. Scientia et Technica. Año XIV N° 39, pp. 94-99. 2008. ISSN 0122-1701.
- [2] C. Uhrhan, A. Jaramillo, S. Plata, A. Buss y F. Mojica. “Escarabot: Vehículo Autoguiado para Transporte de Carga en Entornos

- Industriales". I Congreso Internacional de Ingenierías Eléctrica y Electrónica UIS CIIIE. Colombia. 2000. ISBN: 958-9318-69-X.
- [3] A. Pedroza, C. Sanchez y H. Rodriguez. "Control de un vehiculo guiado automaticamente (AGV)". Conciencia Tecnologica. Nº 034, pp. 10-15. 2007. ISSN: 1405-5597.
- [4] R. González, F. Rodríguez, J. Sánchez-Hermosilla y J. García. "Algoritmo de navegación reactiva de robots móviles para tareas bajo invernadero". XXVII Jornadas de Automática, pp. 888-865. Almeria, España. 2006.
- [5] O. Vivas y J. Mosso. "Sistema software neurodifuso para el control de un robot autonomo". I Congreso Internacional de Ingenierías Eléctrica y Electrónica UIS CIIIE. Colombia. 2000. ISBN: 958-9318-69-X.
- [6] R. Silva, J. García, V. Barrientos, M. Molina, V. Hernández y G. Silva. "Una panorámica de los robots móviles". Revista Electrónica de Estudios Telemáticos. Vol. 6 Nº 3, pp. 1-14. 2007. ISSN: 1856-4194.
- [7] M. Marinelli, H. Kuna, F. Puente, M. Mounier, C. Kornuta, E. De Silvestre, C. Kornuta y M. Kolb. "Aplicación de técnicas de inteligencia artificial para la navegación de robots móviles utilizando el simulador player/stage". XIII Workshop de Investigadores en Ciencias de la Computación, pp. 94-97. 2011. ISBN: 978-950-673-892-1
- [8] R. Núñez. "Modelamiento y optimización de procesos industriales utilizando sistemas neurodifusos". I Congreso Internacional de Ingenierías Eléctrica y Electrónica UIS CIIIE. Colombia. 2000. ISBN 958-9318-69-X.
- [9] H. Gonzalez y C. Mella. "Estudio comparativo de tres técnicas de navegación para robots móviles". Revista UIS Ingenierias. Vol. 6 Nº 1, pp. 77-84. 2007. ISSN: 1657-4583.
- [10] I. Baturone y A. Gersnoiez. "Diseño de sistemas neuro-difusos para control de robots mediante xfuzzy 3". Congreso Tecnologia, Aprendizaje y Enseñanza de la Electrónica TAAE 2010. Madrid, España. 2010.
- [11] N. Muñoz, L. Ortiz y N. Londoño. "Método de comparación de algoritmos de control de robots móviles". VII Conferencia Iberoamericana en Sistemas, Cibernética e Informática CИСCI 2008. Orlando, Estados Unidos. 2008.
- [12] Adept MobileRobots LLC. "Pioneer 3-DX Datasheet". URL: <http://mobilerobots.com/>. Fecha de Consulta: 01 de febrero de 2014.
- [13] J. Heaton. "Introduction to Neuronal Networks with Java". 2da Edición. K. Smith Ed. Heaton Research. Chesterfield, MO, Estados Unidos. 2008.
- [14] J. Peak. "Neural Network Methodologies and their Potential Application to Cloud Pattern Recognition". Computer Sciences Corp. Monterey, CA, Estados Unidos. 1991.
- [15] S. Geman, E. Bienenstock and R. Doursat. "Neural Networks and the Bias/Variance Dilemma". Neural Computation. Vol. 4 Nº1, pp. 1-58. 1992. ISSN: 0899-7667
- [16] S. Lawrence, C.L. Giles and A.C. Tsoi. "What Size Neural Network Gives Optimal Generalization? Convergence Properties of Backpropagation". Technical Report - Institute For Advanced Computer Studies. University of Maryland. Estados Unidos. 1996.
- [17] J. Devore. "Probabilidad y estadística para ingeniería y ciencias". 5ta Edición. Thomson Learning Ed. Mexico. 2001.
- [18] D. Montgomery y G. Runger. "Probabilidad y estadística aplicadas a la Ingeniería". McGraw-Hill Ed. Mexico. 1996.