



Análisis gráfico de la neuro-dinámica en un robot capaz de realizar aprendizaje imitativo

Graphic analysis of the neural-dynamics in a robot able to do imitative learning

Dante Giovanni Sterpin Buitrago¹, Fernando Martínez Santa²

Fecha de recepción: Agosto 28 de 2015

Fecha de aceptación: Septiembre 25 de 2015

Como citar: Sterpin, D., & Martínez, F. (2015). Análisis gráfico de la neuro-dinámica en un robot capaz de realizar aprendizaje imitativo. *Revista Tecnura*, 19 (CITIE), 116-128. doi: <http://dx.doi.org/10.14483/udistrital.jour.tecnura.2015.ICE.a14>

Resumen

El Perceptrón multicapa es una red neuronal artificial capaz de aprender a funcionar de cierta forma deseable mediante ejemplos y, una vez entrenado, es capaz de responder correctamente en situaciones para las cuales no se le entrenó explícitamente, debido a su capacidad de generalización. Dichos ejemplos los especifica el diseñador o pueden asimilarse con el apoyo de un mapa auto-organizado. En el presente artículo se presentan la verificación del entrenamiento y el análisis del funcionamiento del Perceptrón multicapa empleado durante el desarrollo de un modelo de aprendizaje por imitación, enfocado al desplazamiento de un robot móvil en un espacio bidimensional. Esto se logró diseñando un método de comparación entre las gráficas de los atractores generados por el robot instructor y el robot aprendiz. Dicho método está soportado en procesamiento digital de imágenes.

Palabras clave: Red neuronal artificial, Perceptrón multicapa, Aprendizaje supervisado, Mapa auto-organizado de Kohonen, Perceptrón auto-supervisado, Aprendizaje imitativo, Atractor caótico.

Abstract

Multilayer Perceptron is an artificial neural network able to learn to work in a specific desired way through examples, and once it is trained, it is able to react correctly in situations for which it was not specifically trained, due to its generalization capability. These examples are specified by the designer or can be assimilated with the support of a self-organized map. This paper shows the training verification and the working analysis of the multilayer perceptron used during the development of an imitation learning model; which is focused to a mobile robot displacement over two-dimensional space. This was done designing a comparison method between the attractor's graphs generated by the instructor robot and the apprentice robot. This method is supported in digital image processing.

Key words: Artificial neural network, Multilayer Perceptron, Supervised learning, Kohonen Self-organized Map, Self-supervised Perceptron, Imitative Learning, Chaotic Attractor.

¹ Investigador grupo IDECUN, Corporación Unificada Nacional de Educación Superior CUN, Programa de Ingeniería Electrónica, Bogotá D.C., Colombia. Contacto: dante_sterpin@cun.edu.co.

² Investigador grupo AXON, Corporación Unificada Nacional de Educación Superior CUN, Programa de Ingeniería de Sistemas, Bogotá D.C., Colombia. Contacto: fernando_martinez@cun.edu.co.

INTRODUCCIÓN

En el programa de Ingeniería Electrónica de la Corporación Unificada Nacional de Educación Superior (CUN) se desarrolló un modelo de aprendizaje imitativo inspirado en la hipótesis del replicador neuro-memético (Fernando *et al.*, 2010). En dicho modelo se emplearon dos redes neuronales artificiales muy conocidas –Perceptrón multicapa (MLP) y Mapa Auto-organizado de Kohonen (SOM), con el propósito de entrenar al SOM para asimilar los patrones conductuales de un MLP instructor (MLP_T) con el fin de auto-entrenar un MLP aprendiz (MLP_L), de tal forma que el modelo híbrido (SOM + MLP_L) resulta imitando al MLP_T (Sterpin, 2011). A dicho híbrido se le denominó Perceptrón auto-supervisado (SSP).

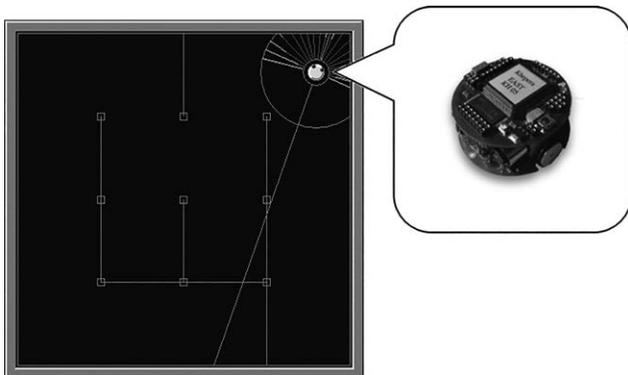


Figura 1. Robot circular simulado en un laberinto virtual.

Fuente: Elaboración propia.

En dicho desarrollo el MLP es capaz de controlar un robot virtual, y el MLP_T se pre-entrenó convencionalmente para que dicho robot pudiera recorrer el espacio accesible en un laberinto virtual, manteniéndose a cierta distancia de las paredes por el lado derecho. En la figura 1 se detalla una vista superior de dicho robot en un posible laberinto.

El aprendizaje imitativo se ha simulado ampliamente mediante diversos modelos, incluyendo

redes neuronales artificiales (Borenstein y Ruppín, 2004), (Curran y O’Riordan, 2004), (Castillo, 2010) y (Castillo, Escobar, Hermosillo y Lara, 2013). El SSP dista de otros modelos imitadores en cuanto a que, por ahora, se le ha considerado como un agente, el cual podría operar de manera individualista o como parte de una comunidad de varios replicadores neuro-meméticos en un único sistema cerebral artificial (Sterpin, 2015). En el presente artículo se detalla el diseño del MLP_T empleado, se presentan las visualizaciones mediante las cuales se verificó el respectivo entrenamiento de cada MLP, y además se analiza las visualizaciones mediante las cuales se calculó la semejanza conductual entre el instructor y el aprendiz.

METODOLOGÍA

Considerando el funcionamiento convencional de un Perceptrón multicapa (MLP) se definió un patrón de entrenamiento acorde con el objetivo de navegación del robot, el cual es el desplazamiento libre en un laberinto; pero considerando que en muchas de las situaciones posibles sería necesario emplear valores intermedios entre los extremos bipolares con los que suele operar un MLP.

Funcionamiento del MLP_T

El MLP es una posible forma de interconectar neuronas artificiales organizándolas en capas: una receptora, una o varias intermedias, y una efectora, de tal manera que las neuronas de la primera reciben información sensorial y las de la última entregan información motriz. Definiendo una sola capa intermedia, la propagación neuro-informática se calcula empleando los valores sensoriales para obtener la activación de las neuronas intermedias, y utilizando estos valores como entrada para calcular la activación de las neuronas efectoras.

Las neuronas artificiales en un MLP tienen n entradas $[x_i]$ con un peso sináptico $[w_{ij}]$ asociado con el fin de reforzarlas o inhibirlas en el cálculo de la estimulación entrante. Con la diferencia entre dicha

estimulación y cierto umbral $[\theta_i]$ calculan un valor denominado potencial posináptico $[h_i]$, con el cual calculan una reacción frente a la situación percibida, empleando una función de tipo sigmoïdal

$$y_i = \left[\frac{2}{1 + e^{(-\lambda_i \times h_i)}} \right] - 1, \quad \text{donde } h_i = \sum_{j=1}^n (x_j \times w_{ij}) - \theta_i \quad (1)$$

Vale mencionar que los valores de las sinapsis $[w_{ij}]$ representan el conocimiento mediante el cual cada neurona *sabe* cómo reaccionar a la estimulación entrante. Además, una neurona puede reaccionar positiva o negativamente a la estimulación entrante, pero esto no corresponde a un juicio de valor, pues una reacción negativa podría ser tan necesaria como una reacción positiva. Por ejemplo, al controlar un motor la reacción negativa podría hacerlo girar en cierto sentido mientras la reacción positiva lo haría girar en sentido contrario.

En la figura 2 se muestra el MLP empleado para controlar el robot de la figura 1. La neurona A recibe la medición del sensor izquierdo mientras la neurona B recibe la del sensor derecho. La neurona C entrega la información para controlar la velocidad del motor izquierdo mientras la neurona D entrega la del motor derecho. Tanto las neuronas intermedias $\{M, N, O, P, Q\}$ como las efectoras $\{C, D\}$ se activan mediante sigmoïdal bipolar con rango $[-10 ; +10]$, pues dicho robot simula la motricidad de una plataforma móvil circular con dos motores centro-laterales cuya velocidad se asumió

bipolar, debido al carácter continuo de los sensores. En la ecuación 1 se detalla la respectiva expresión formal para dichas neuronas, empleando una sigmoïdal bipolar de tipo exponencial.

independiente y variante entre $[-10 ; +10]$, donde $(+10)$ les implicaría rotar con máxima velocidad en el sentido de las manecillas del reloj, mientras (-10) les implicaría rotar con máxima velocidad al contrario. Para este MLP se empleó la versión cuadrática de la sigmoïdal bipolar detallada en las ecuaciones 3 y 5, donde las respectivas expresiones de potenciales posinápticos se detallan en las ecuaciones 2 y 4.

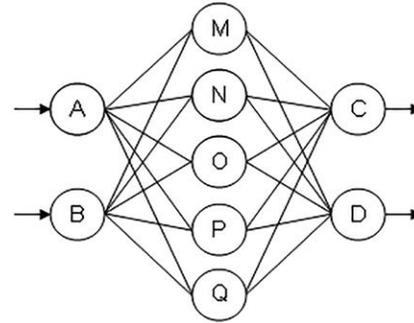


Figura 2. MLP con 2 neuronas sensoriales (k), 5 intermedias (j) y 2 efectoras (i).

Fuente: Elaboración propia.

$$h_j = \sum_{k=1}^n (x_k * w_{jk}) - \theta_j \quad (2)$$

$$y_j = 20 * \left\{ \begin{array}{l} 0,0; \text{para } (h_j + v_j) < (v_j - \lambda_j v_j) \\ \frac{1}{2} * \left(\frac{h_j + \lambda_j v_j}{\lambda_j v_j} \right)^4; \text{para } (h_j + v_j) < v_j \\ 1 - \left[\frac{1}{2} * \left(\frac{\lambda_j v_j - h_j}{\lambda_j v_j} \right)^4 \right]; \text{para } (h_j + v_j) \geq v_j \\ 1,0; \text{para } (h_j + v_j) \geq (v_j + \lambda_j v_j) \end{array} \right\} - 10 \quad (3)$$

$$h_i = \sum_{j=1}^s (y_j * w_{ij}) - \theta_i \quad (4)$$

$$y_i = 20 * \left\{ \begin{array}{l} 0,0; \text{para}(h_i + v_i) < (v_i - \lambda_i v_i) \\ \frac{1}{2} * \left(\frac{h_i + \lambda_i v_i}{\lambda_i v_i} \right)^4; \text{para}(h_i + v_i) < v_i \\ 1 - \left[\frac{1}{2} * \left(\frac{\lambda_i v_i - h_i}{\lambda_i v_i} \right)^4 \right]; \text{para}(h_i + v_i) \geq v_i \\ 1,0; \text{para}(h_i + v_i) \geq (v_i + \lambda_i v_i) \end{array} \right\} - 10 \quad (5)$$

Con el fin de controlar cíclicamente el robot se empleó la medida de los sensores $\{x_a; x_b\}$ en la ecuación 2 para obtener los potenciales $[h_j] : \{h_m; h_n; h_o; h_p; h_q\}$, los cuales se emplearon en la ecuación 3 para calcularlas salidas $[y_j] : \{y_m; y_n; y_o; y_p; y_q\}$, las cuales se emplearon en la ecuación 4 para obtener los potenciales $[h_i] : \{h_c; h_d\}$, los cuales se emplearon en la ecuación 5 para calcular las salidas $[y_i] : \{y_c; y_d\}$ y aplicarlas respectivamente a los motores.

Entrenamiento del MLP_T

Para un MLP con n entradas $[x_k]$ y m salidas $[y_i]$ el diseñador humano define un conjunto de p ejemplos sobre el cómo reaccionar correctamente frente a la estimulación entrante. En cada ejemplo $[p_\mu]$ se especifica las m salidas deseadas $[t_i]$ para alguna posible situación. Con la ecuación 6 se calcula el error cuadrático medio $[E_\mu]$ en cada iteración (τ) del entrenamiento, el cual pretende minimizar el

error modificando los pesos sinápticos y los umbrales de activación aplicando la ecuación 7 en cada nueva iteración ($\tau + 1$). La ecuación 8 muestra la expresión general para calcular las variaciones de pesos sinápticos y umbrales de activación en cada iteración (τ). Al desarrollar las expresiones de la ecuación 8 se obtienen las expresiones de la ecuación 9, donde son útiles las expresiones de la ecuación 10 cuando se emplea la versión cuadrática de la sigmoide bipolar como función de activación.

Este método se conoce como aprendizaje supervisado debido a la necesidad de ejemplos capaces de guiar el proceso de ajuste sináptico. Al aplicarlo se espera que, para cada ejemplo $[p_\mu]$ del patrón de entrenamiento, las m salidas obtenidas $[y_i]$ se acerquen respectivamente a las m salidas deseadas $[t_i]$. Finalmente cabe recordar que, para dotar de mayor autonomía al MLP con respecto al diseñador humano, puede emplearse un SOM con la finalidad de aprehender un conjunto

$$E_\mu(\tau) = \frac{1}{2} \sum_{i=1}^m [t_i^\mu - y_i^\mu(\tau)]^2 \quad (6)$$

$$w_{ij}(\tau + 1) = w_{ij}(\tau) + \Delta w_{ij}(\tau); \theta_i(\tau + 1) = \theta_i(\tau) + \Delta \theta_i(\tau) \quad (7)$$

$$\Delta w_{ij} = -\varepsilon \sum_{\mu=1}^p \frac{\partial E_\mu}{\partial w_{ij}}; \Delta \theta_i = -\varepsilon \sum_{\mu=1}^p \frac{\partial E_\mu}{\partial \theta_i} \quad (8)$$

$$\begin{aligned} \Delta w_{ij} &= \varepsilon \sum_{\mu=1}^p \frac{(t_i^\mu - y_i^\mu) * \partial y_i}{\partial h_i} * y_j^\mu; \Delta w_{jk} = \varepsilon \sum_{\mu=1}^p \frac{\sum_{i=1}^m \left[\frac{(t_i^\mu - y_i^\mu) * \partial y_i}{\partial h_i} * w_{ij} \right] * \partial y_j}{\partial h_j} * x_k^\mu \\ \Delta \theta_i &= -\varepsilon \sum_{\mu=1}^p \frac{(t_i^\mu - y_i^\mu) * \partial y_i}{\partial h_i}; \Delta \theta_j = -\varepsilon \sum_{\mu=1}^p \frac{\sum_{i=1}^m \left[\frac{(t_i^\mu - y_i^\mu) * \partial y_i}{\partial h_i} * w_{ij} \right] * \partial y_j}{\partial h_j} \end{aligned} \quad (9)$$

$$\frac{\partial y_i}{\partial h_i} = 20 * \left\{ \begin{array}{l} 0,0; (h_i + v_i) < (v_i - \lambda_i v_i) \\ \frac{2}{\lambda_i v_i} * \left(\frac{h_i + \lambda_i v_i}{\lambda_i v_i} \right)^3; (h_i + v_i) < v_i \\ \frac{2}{\lambda_i v_i} * \left(\frac{\lambda_i v_i - h_i}{\lambda_i v_i} \right)^3; (h_i + v_i) \geq v_i \\ 0,0; (h_i + v_i) \geq (v_i + \lambda_i v_i) \end{array} \right\}$$

$$\frac{\partial y_j}{\partial h_j} = 20 * \left\{ \begin{array}{l} 0,0; (h_j + v_j) < (v_j - \lambda_j v_j) \\ \frac{2}{\lambda_j v_j} * \left(\frac{h_j + \lambda_j v_j}{\lambda_j v_j} \right)^3; (h_j + v_j) < v_j \\ \frac{2}{\lambda_j v_j} * \left(\frac{\lambda_j v_j - h_j}{\lambda_j v_j} \right)^3; (h_j + v_j) \geq v_j \\ 0,0; (h_j + v_j) \geq (v_j + \lambda_j v_j) \end{array} \right\} \quad (10)$$

de ejemplos capaz de auto-supervisar el entrenamiento del MLP (Sterpin, 2011). El algoritmo para entrenar el SOM dista bastante del anteriormente descrito para el MLP, y en este artículo se obviará detallarlo para evitar confusión.

Para el MLP_T se definieron 64 ejemplos de entrenamiento considerando el rango operativo de

los motores en el robot virtual, y adicionalmente su rango sensorial. Dicho robot simula dos sensores fronto-laterales con la capacidad de medir su distancia con respecto a las paredes del laberinto con valores en el rango [-10 ; +10], donde (-10) simboliza que no hay pared, mientras (+10) corresponde a la cercanía máxima de alguna pared.

Tabla 1. Patrón completo de entrenamiento para el MLPT.

| μ | Situación posible | | Reacción deseada | |
|-------|-----------------------|-----------------------|----------------------|----------------------|
| | Sensor X _a | Sensor X _b | Motor T _c | Motor T _d |
| 1 | -10,0 | -10,0 | 9,0608 | -2,6177 |
| 2 | -10,0 | -7,1428 | 7,9463 | 2,7814 |
| 3 | -10,0 | -4,2856 | 5,9172 | 5,5592 |
| 4 | -10,0 | -1,4284 | 2,4701 | 6,5151 |
| 5 | -10,0 | 1,4284 | -2,4882 | 6,0991 |
| 6 | -10,0 | 4,2856 | -6,1272 | 3,5602 |
| 7 | -10,0 | 7,1428 | -8,2324 | -3,5472 |
| 8 | -10,0 | 10,0 | -9,264 | -7,9344 |
| 9 | -7,1428 | -10,0 | 7,7758 | 0,1557 |
| 10 | -7,1428 | -7,1428 | 5,6948 | 4,0612 |
| 11 | -7,1428 | -4,2856 | 2,2141 | 5,6147 |
| 12 | -7,1428 | -1,4284 | -2,6495 | 5,5648 |
| 13 | -7,1428 | 1,4284 | -6,172 | 3,4956 |
| 14 | -7,1428 | 4,2856 | -8,2858 | -2,6506 |

| μ | Situación posible | | Reacción deseada | |
|-------|-----------------------|-----------------------|----------------------|----------------------|
| | Sensor X _a | Sensor X _b | Motor T _c | Motor T _d |
| 15 | -7,1428 | 7,1428 | -9,3241 | -7,7764 |
| 16 | -7,1428 | 10,0 | -9,7876 | -9,1326 |
| 17 | -4,2856 | -10,0 | 5,4956 | 2,2049 |
| 18 | -4,2856 | -7,1428 | 2,0003 | 4,5222 |
| 19 | -4,2856 | -4,2856 | -2,7632 | 4,9287 |
| 20 | -4,2856 | -1,4284 | -6,1897 | 3,3403 |
| 21 | -4,2856 | 1,4284 | -8,2681 | -1,9492 |
| 22 | -4,2856 | 4,2856 | -9,3513 | -7,4817 |
| 23 | -4,2856 | 7,1428 | -9,8122 | -9,1806 |
| 24 | -4,2856 | 10,0 | -9,9663 | -9,5662 |
| 25 | -1,4284 | -10,0 | 1,829 | 3,2083 |
| 26 | -1,4284 | -7,1428 | -2,8338 | 4,1762 |
| 27 | -1,4284 | -4,2856 | -6,1813 | 3,0958 |
| 28 | -1,4284 | -1,4284 | -8,2354 | -1,3448 |

| μ | Situación posible | | Reacción deseada | | μ | Situación posible | | Reacción deseada | |
|-------|-------------------|--------------|------------------|-------------|-------|-------------------|--------------|------------------|-------------|
| | Sensor X_a | Sensor X_b | Motor T_c | Motor T_d | | Sensor X_a | Sensor X_b | Motor T_c | Motor T_d |
| 29 | -1,4284 | 1,4284 | -9,3391 | -6,9884 | 47 | 4,2856 | 7,1428 | -10,0 | -10,0 |
| 30 | -1,4284 | 4,2856 | -9,8202 | -9,1689 | 48 | 4,2856 | 10,0 | -10,0 | -10,0 |
| 31 | -1,4284 | 7,1428 | -9,9718 | -9,6519 | 49 | 7,1428 | -10,0 | -8,0509 | -0,2146 |
| 32 | -1,4284 | 10,0 | -9,999 | -9,7138 | 50 | 7,1428 | -7,1428 | -9,2062 | -4,7172 |
| 33 | 1,4284 | -10,0 | -2,8641 | 3,2915 | 51 | 7,1428 | -4,2856 | -9,7565 | -8,6608 |
| 34 | 1,4284 | -7,1428 | -6,148 | 2,7597 | 52 | 7,1428 | -1,4284 | -9,9648 | -9,7125 |
| 35 | 1,4284 | -4,2856 | -8,188 | -0,8475 | 53 | 7,1428 | 1,4284 | -9,9992 | -9,898 |
| 36 | 1,4284 | -1,4284 | -9,3028 | -6,2607 | 54 | 7,1428 | 4,2856 | -10,0 | -10,0 |
| 37 | 1,4284 | 1,4284 | -9,8132 | -9,0939 | 55 | 7,1428 | 7,1428 | -10,0 | -10,0 |
| 38 | 1,4284 | 4,2856 | -9,9732 | -9,6997 | 56 | 7,1428 | 10,0 | -10,0 | -10,0 |
| 39 | 1,4284 | 7,1428 | -9,9993 | -9,8082 | 57 | 10,0 | -10,0 | -9,1444 | -3,9692 |
| 40 | 1,4284 | 10,0 | -10,0 | -10,0 | 58 | 10,0 | -7,1428 | -9,7227 | -8,1454 |
| 41 | 4,2856 | -10,0 | -6,0915 | 2,3277 | 59 | 10,0 | -4,2856 | -9,9524 | -9,6789 |
| 42 | 4,2856 | -7,1428 | -8,1267 | -0,4693 | 60 | 10,0 | -1,4284 | -9,9987 | -9,9159 |
| 43 | 4,2856 | -4,2856 | -9,2588 | -5,4891 | 61 | 10,0 | 1,4284 | -10,0 | -10,0 |
| 44 | 4,2856 | -1,4284 | -9,7902 | -8,9423 | 62 | 10,0 | 4,2856 | -10,0 | -10,0 |
| 45 | 4,2856 | 1,4284 | -9,9711 | -9,7188 | 63 | 10,0 | 7,1428 | -10,0 | -10,0 |
| 46 | 4,2856 | 4,2856 | -9,9994 | -9,8649 | 64 | 10,0 | 10,0 | -10,0 | -10,0 |

Fuente: Elaboración propia.

Visualización del entrenamiento

Para detallar gráficamente el proceso de entrenamiento se emplearon planos cartesianos con el propósito de simbolizar las tres combinaciones relevantes de los datos implicados: Situación sensorial del robot, Actuación deseada para el robot y Actuación aprendida por el MLP. La notación de dichos planos es: $[X_a \times X_b]$, $[T_c \times T_d]$ y $[Y_c \times Y_d]$, respectivamente.

En la figura 3a se especifica una situación posible para el robot en el laberinto –donde el sensor izquierdo $[x_a]$ mide -4,2856 y el sensor derecho $[x_b]$ mide -1,4284, mientras en la figura 3b se detalla la ubicación de la dupla {-4,2856 ; -1,4284} en el plano $[X_a \times X_b]$. Dichos valores no requieren unidades de medida pues se asume que el rango sensorial [-10 ; +10] corresponde al [0 ; 100]% del rango de medición de un sensor real.

En la figura 4a se detalla el sentido con el cual se asume necesario que el robot reaccione en dicha situación, mientras en la figura 4b se detalla la ubicación de la respectiva combinación de

velocidades{-6,1897 ; 3,3403} en el plano $[T_c \times T_d]$. Dichos valores están en el rango motriz [-10; +10] se compone de dos partes y así, el rango [0; +10] simboliza el [0; 100]% del rango de velocidades de un motor real girando en sentido horario, mientras el rango [0; -10] simboliza el [0; 100]% del rango de velocidades girando en sentido contrario.

Para visualizar la dinámica del entrenamiento se empleó el plano $[Y_c \times Y_d]$. En cada iteración (τ) se graficaron allí las duplas $\{y_c ; y_d\}$ obtenidas en el 80% de las posibles situaciones definidas en el patrón de entrenamiento y empleando los pesos sinápticos vigentes en (τ). Considerando que el entrenamiento del MLP consiste en lograr valores útiles para una gran cantidad de variables: $\{(j * k) \text{ pesos} + (j) \text{ umbrales}\} + \{(i * j) \text{ pesos} + (i) \text{ umbrales}\}$, con el propósito de cumplir requerimientos, y considerando que obtener dichos valores no es calculable mediante métodos convencionales, dicho proceso es complejo (Maldonado y Gómez, 2010) y caótico, es decir auto-organizativo, pues a partir de un comportamiento inservible el sistema es atraído hacia el comportamiento deseable –donde

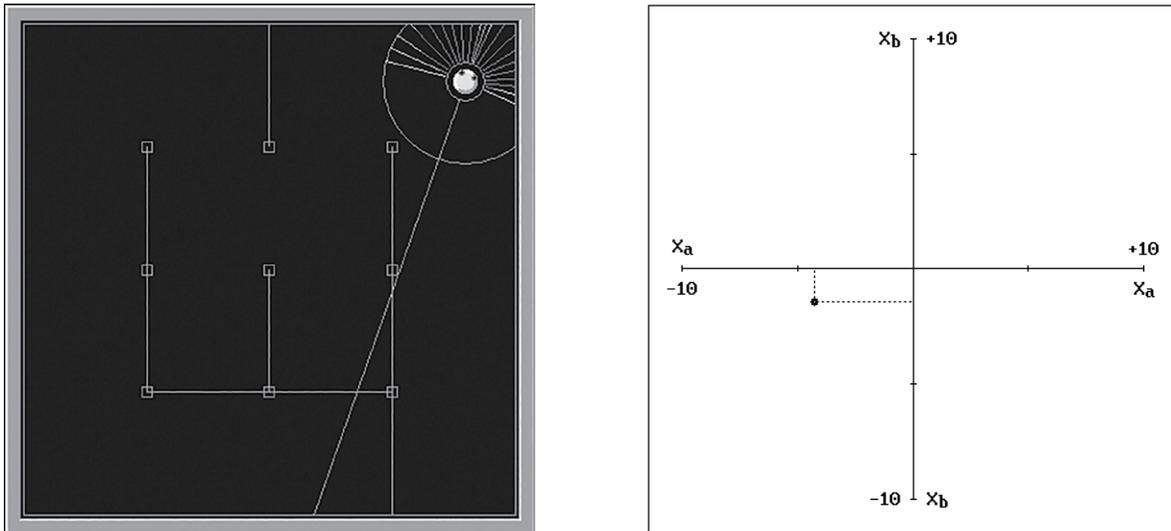


Figura 3. a) Ubicación real, y b) Representación cartesiana, de una situación posible.

Fuente: Elaboración propia.

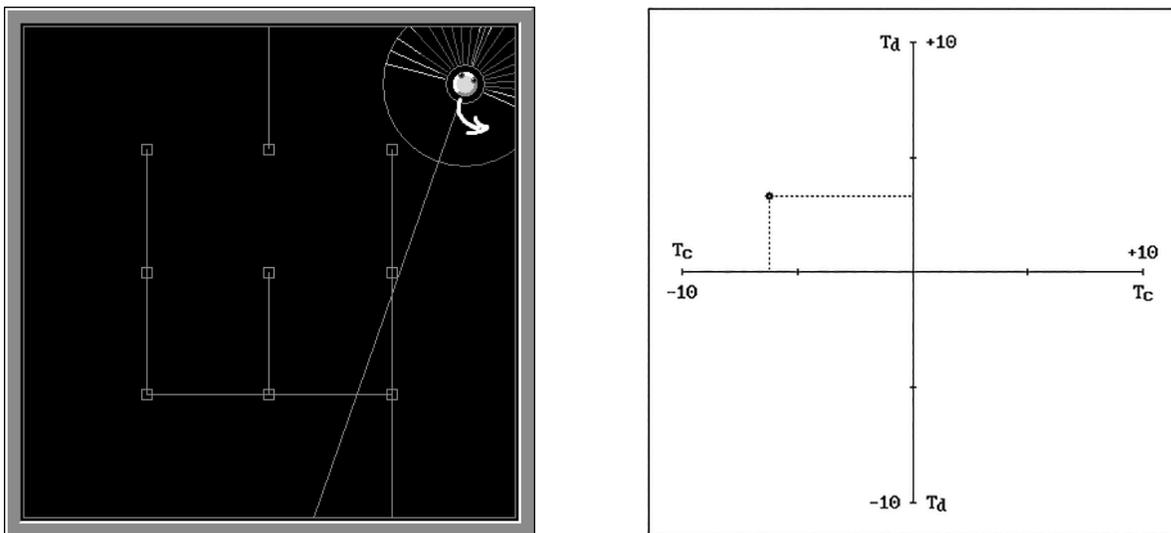


Figura 4. a) Sentido real, y b) Representación cartesiana, de una reacción necesaria.

Fuente: Elaboración propia.

el patrón auto-organizativo es en este caso el patrón de entrenamiento. La visualización del orden implícito en procesos caóticos se denomina *atractor extraño* o fractal, y los hay de dos tipos: punto o ciclo (Restrepo, 2007).

Al instruir un MLP los atractores son de tipo punto y se definen en el patrón de entrenamiento con

el conjunto de salidas deseadas en cada situación posible. En la figura 5a el punto *omega* (ω) marca el 20-avo atractor $\{-6,1897 ; 3,3403\}$ de la tabla 1, pues es justo ahí a donde será atraído el sistema tras iniciar en el punto *alpha* (α). Este último punto es la reacción $\{y_c ; y_d\}$ del MLP al estimularlo con la 20-ava situación sensorial $\{-4,2856 ; -1,4284\}$ de la

tabla 1, utilizando sinapsis y umbrales con valores aleatorios entre $[-0,1 ; +0,1]$. Empleando $\lambda_i=2,2$ y $v_i=10$ en las ecuaciones 5 y 10, además de $\lambda_j=6,8$ y $v_j=10$ en las ecuaciones 3 y 10, se entrenó el MLP durante 300.000 iteraciones. En cada iteración se ajustaron los valores de sinapsis y umbrales, con

los cuales se re-calculaba y se graficaba la respectiva reacción $\{y_c ; y_d\}$. En la figura 5b se visualiza la trayectoria resultante desde *alpha* hacia *omega*.

En la figura 6 están graficados los 64 ejemplos del patrón de entrenamiento presentado en la tabla 1. La figura 6a muestra la distribución uniforme

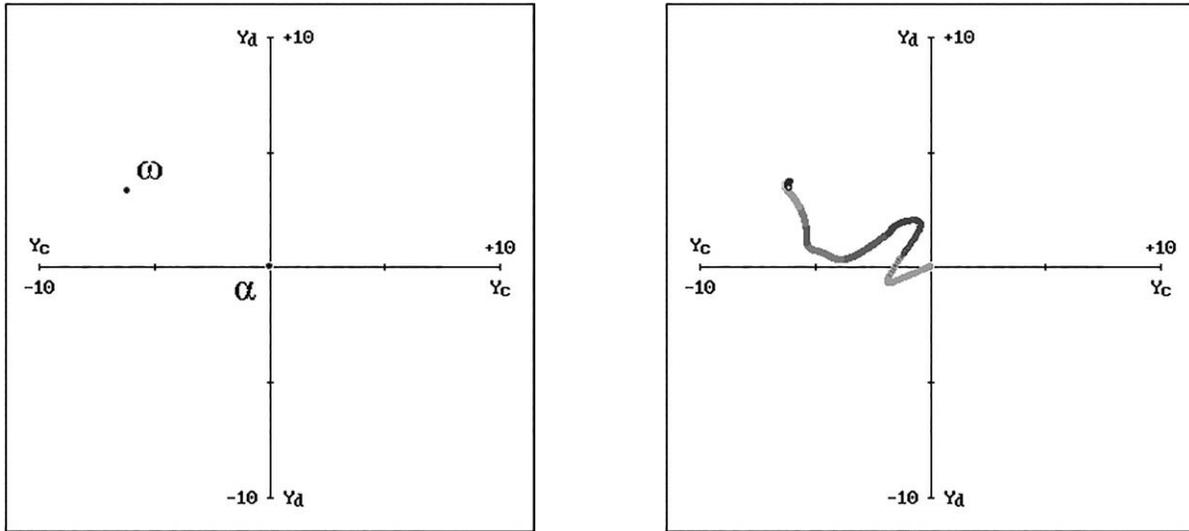


Figura 5. a) *alpha*&*omega*, y b) Trayectoria auto-organizativa, del aprendizaje.

Fuente: Elaboración propia.

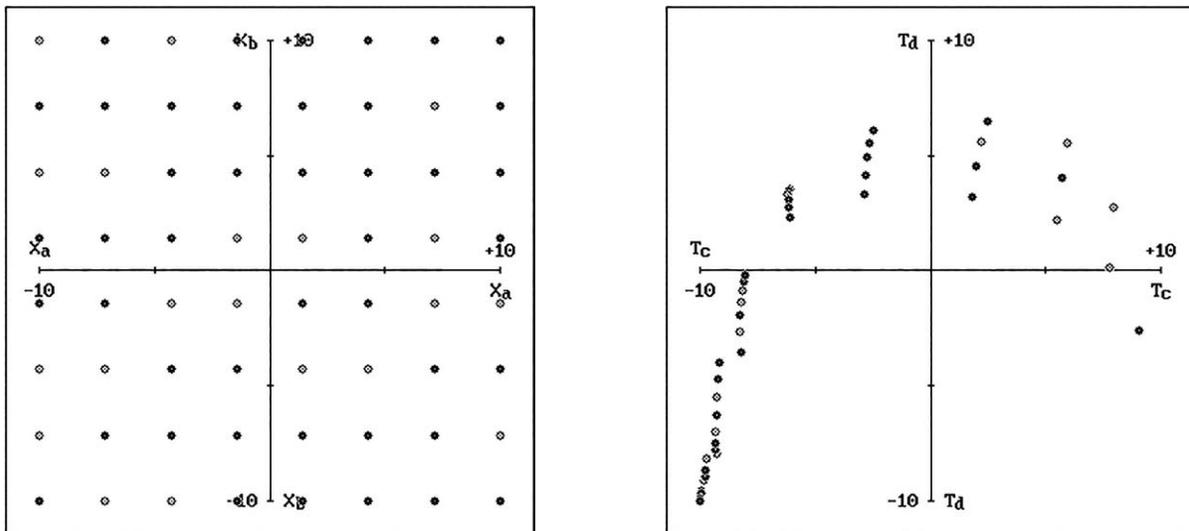


Figura 6. a) Situaciones posibles y b) Actuaciones deseadas, para entrenar el MLPT.

Fuente: Elaboración propia.

de las posibles situaciones sensoriales, mientras la figura 6b muestra las respectivas actuaciones deseables.

En dichas gráficas se emplean dos tipos de puntos –negros y blancos, con el fin de representar la necesaria diferenciación entre los ejemplos $[p_i]$ utilizados para entrenar directamente al MLP y los que se emplean para monitorear el error de generalización. Los puntos negros referencian a los primeros mientras los blancos referencian a los segundos. Dicha diferenciación se realiza aleatoriamente, y con dicho monitoreo se procura garantizar que, tras el entrenamiento, el MLP *sepa* reaccionar –lo más coherente posible, en situaciones para las cuales no se le entrenó explícitamente. En el SSP dicha diferenciación es probabilística (Sterpin, 2011).

En la figura 7b puede detallarse el conjunto de trayectorias correspondiente al entrenamiento de los 44 ejemplos escogidos al azar con la finalidad de instruir al MLP. Ideográficamente se le denominó *Culebrilla cognitiva* a la trayectoria en figura 5b, y por ello al conjunto de *Culebrillas cognitivas* –como el de la figura 7b, se le denominó *Cabellera de Medusa*.

RESULTADOS

Una vez entrenado el MLP_T se le hizo controlar al robot y en la figura 8a pueden detallarse, en orden descendente, cuatro series de tiempo mediante las cuales se visualizó la dinámica particular de cada percepción $[x_a]$ $[x_b]$ y, de cada una de las respectivas activaciones neuronales $[y_c]$ $[y_d]$ obtenidas del MLP y aplicadas al robot. Dichas señales simbolizan la cabal ejecución de la tarea en el laberinto. En la figura 8b se detallan, en orden descendente, los dos atractores de tipo ciclo que resultan al combinarse bidimensionalmente $\{x_a ; x_b\}$ y $\{y_c ; y_d\}$.

Empleando este MLP como instructor y un SSP como aprendiz (Sterpin, 2011) se observó que este último evidenció un comportamiento cualitativamente semejante al instruido. Para detallar el patrón de entrenamiento autónomamente adquirido por el SOM del SSP aprendiz, la figura 9a presenta la distribución de las posibles situaciones, mientras la figura 9b presenta las respectivas salidas deseadas. En las dos gráficas de la figura 9 los puntos-cruz referencian valores sinápticos del SOM que no influyen en el auto-entrenamiento del SSP.

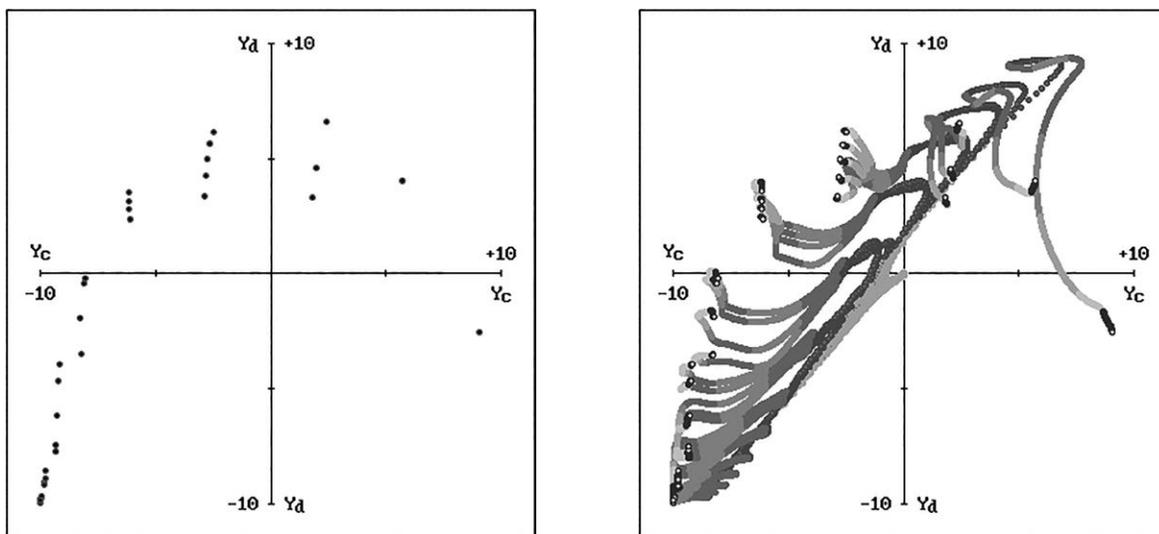


Figura 7. a) Atractores, y b) Trayectorias auto-organizativas, del entrenamiento.

Fuente: Elaboración propia.

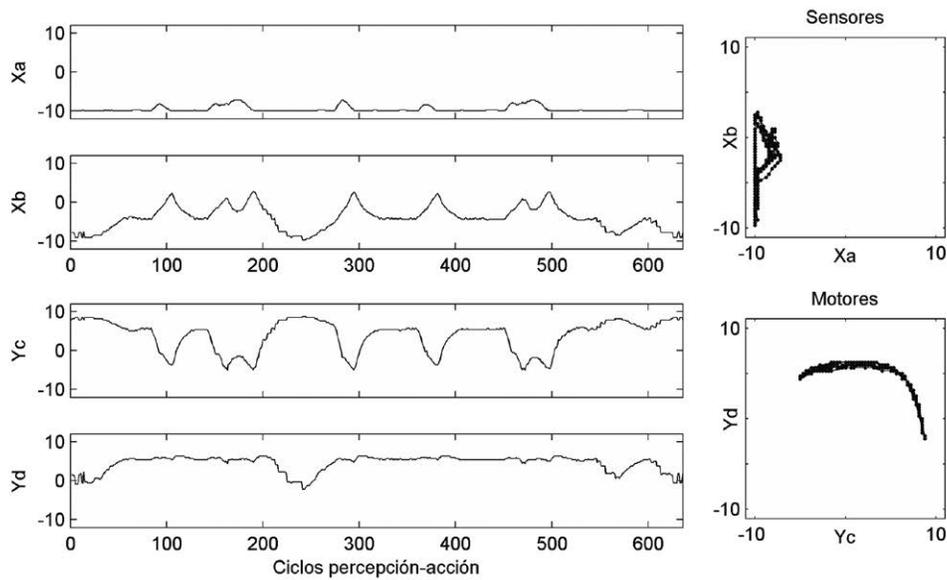


Figura 8. a). Señales senso-motrices y b). Atractores conductuales, del instructor funcionando.

Fuente: Elaboración propia.

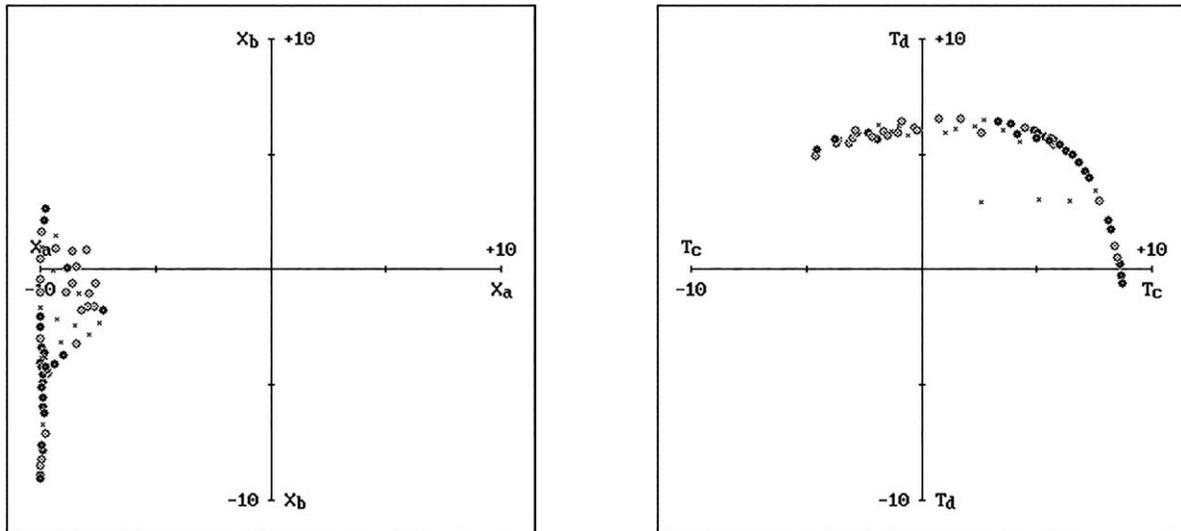


Figura 9. a) Situaciones posibles y b) Actuaciones deseadas, para auto-entrenar el SSP.

Fuente: Elaboración propia.

En la figura 10b puede detallarse la *Cabella de Medusa* correspondiente al entrenamiento de los 28 ejemplos definidos probabilísticamente para instruir al SSP. Finalmente, la figura 11b presenta los atractores del comportamiento evidenciado por el robot aprendiz en el laberinto y así,

al comparar esta última visualización con la figura 8b se presenta a continuación un método cuantitativo para verificar la semejanza conductual entre el robot instructor y el aprendiz, soportado en procesamiento digital de imágenes.

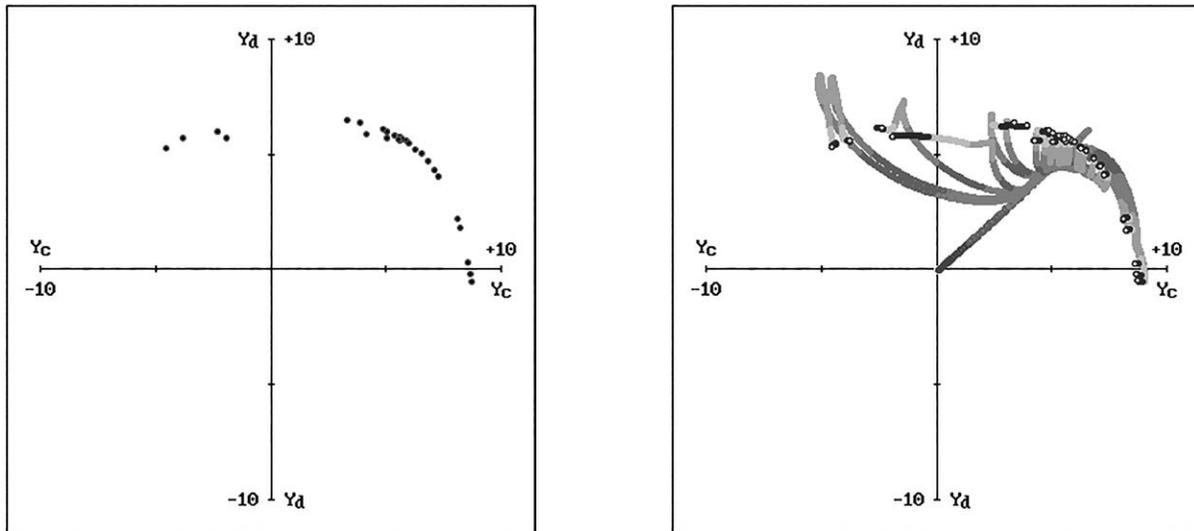


Figura 10. a) Atractores, y b) Trayectorias auto-organizativas, del auto-entrenamiento.

Fuente: Elaboración propia.

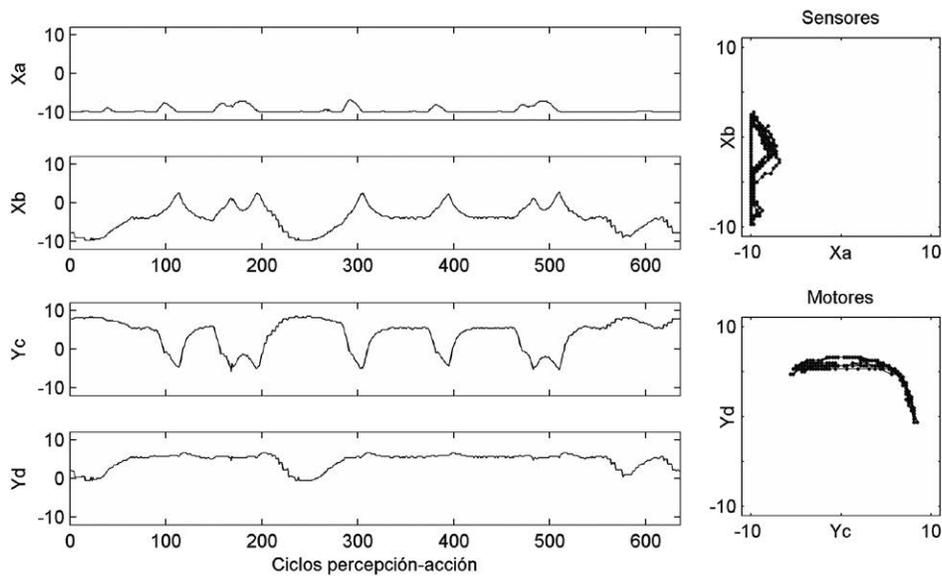


Figura 11. a). Señales senso-motrices y b). Atractores conductuales, del robot aprendiz.

Fuente: Elaboración propia.

El método se pretende hallar la diferencia de píxeles entre cada uno de los atractores, empleando imágenes digitales con cierta resolución. Primero se hizo un análisis de la resolución mínima requerida para el método propuesto, para lo cual se realizó un experimento donde se graficaron

imágenes binarias de cada atractor a diferentes resoluciones de imagen. Considerando el número de píxeles activos en cada caso se realizó una gráfica de tendencia, empleando la lluvia de puntos obtenidos con el muestreo, tal como se muestra en la figura 12. El dominio de la gráfica representa la

resolución empleada en las imágenes binarias de prueba, desde 50x50 hasta 700x700 píxeles.

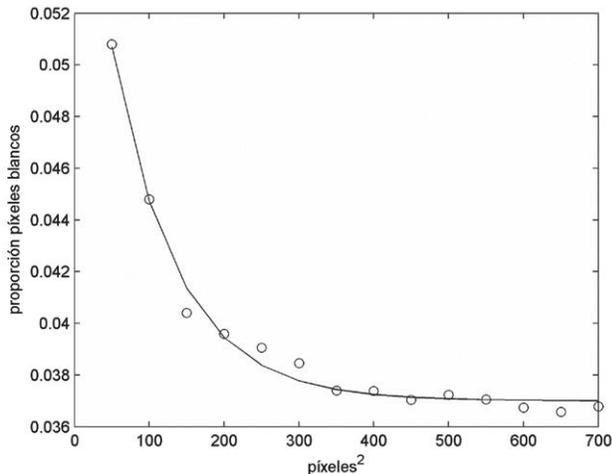


Figura 12. Muestreo del número de píxeles activos (señal punteada) y función ajustada al muestreo (señal continua), con respecto a la resolución de la imagen de los atractores.

Fuente: Elaboración propia.

El muestreo obtenido evidenció que a valores bajos de resolución se tiene una proporción de píxeles blancos β alta y decrece de forma exponencial en función de la resolución ρ , tendiendo a estabilizarse. Se ajustó dicho muestreo a una ecuación de tipo

exponencial, con la forma de la ecuación 11, con el fin de hallar el valor de resolución mínima.

$$\beta = a * e^{-b*\rho} + c \quad (11)$$

Con la curva ajustada se obtuvo los valores: $a=24,29*10^{-3}$, $b=11,47*10^{-3}$, $c=37*10^{-3}$, y al definir una tolerancia del 1%, con respecto al valor de estabilización c , se despejó de la ecuación 11 el valor de la resolución ρ , tal como se muestra en la ecuación 12, con lo cual se definió que la resolución mínima para analizar los atractores, la cual es de 365x365 píxeles.

$$\rho = \frac{1}{b} \ln\left(\frac{\beta-c}{a}\right) \quad (12)$$

El método propuesto consiste entonces en graficar cada uno de los atractores considerados, mediante una imagen binaria cuadrada, utilizando la resolución previamente obtenida y aplicando una operación morfológica de procesamiento de imágenes para rellenar los espacios vacíos, con el fin de obtener una figura sólida para comparar dichos atractores. La comparación se realizó calculando tanto la conjunción como la disyunción de cada uno de los píxeles de las imágenes de los atractores del instructor y del aprendiz, usando la ecuación 13. Donde x y y son las dimensiones de las imágenes, P_i cada píxel de la imagen del instructor



Figura 13. Imagen del atractor de los motores del robot instructor (a), del robot aprendiz (b), la conjunción (c) y la disyunción (d).

Fuente: Elaboración propia.

I_1 , P_2 del aprendiz I_2 y P_3 y P_4 de las imágenes resultantes I_3 y I_4 .

$$\begin{aligned} \forall P_1(x,y) \in I_1: P_3(x,y) &= P_1(x,y) \wedge P_2(x,y) \\ \forall P_2(x,y) \in I_2: P_4(x,y) &= P_1(x,y) \vee P_2(x,y) \end{aligned} \quad (13)$$

El resultado obtenido de la comparación de los atractores de los motores se muestra en la figura 13, donde se observan por separado los atractores del instructor, el aprendiz, la conjunción y la disyunción calculadas con la ecuación 13.

Finalmente se calcula el porcentaje de similitud entre los atractores como el cociente entre la sumatoria de píxeles activos de la imagen resultado de la conjunción y la sumatoria de los píxeles activos de la imagen de la disyunción, tal como se muestra en la ecuación 14. En donde al igual que en las ecuaciones 11 y 12 se denota la resolución como p .

$$S = \frac{\sum_{x=1}^p \sum_{y=1}^p P_3(x,y)}{\sum_{x=1}^p \sum_{y=1}^p P_4(x,y)} \quad (14)$$

Obteniendo como resultados unos porcentajes de similitud del 80,5% para los atractores de los sensores y del 53,6% para los de los motores. Lo anterior valida el método de aprendizaje imitativo explorado en este artículo, demostrando que se obtiene en promedio un 67% de similitud de comportamiento del robot aprendiz o imitador.

FINANCIAMIENTO:

Corporación Unificada Nacional de Educación Superior CUN

REFERENCIAS

- Borenstein, E. y Ruppín E. (2004). Envolving imitating agents and the emergence of a neural mirror system, En: *Artificial life IX*, pp. 146-151, Cambridge, MIT Press.
- Castillo, D., Escobar, E., Hermosillo, J., y Lara, B. (2013). Modelado de un sistema de neuronas espejo en un agente autónomo artificial. *Nova Scientia*, 5 (2) (10), pp. 51-72.
- Castillo, M. (2010). Control de motores de C.D. con aprendizaje por imitación basado en redes neuronales. México D.F.
- Curran, D. y O'Riordan, C. (2004). A comparison of population learning and cultural learning in artificial life societies, En: *Artificial life IX*, pp. 152-157, Cambridge, MIT Press.
- Fernando, C., Goldstein, R. y Szathmáry, E. (2010). The neuronal replicator hypothesis. *Neural computation*, Vol. 22, No. 11, pp. 2809-2857, MIT Press Journals.
- Maldonado, C., & Gómez, N. (2010). Modelamiento y simulación de sistemas complejos. *Documentos de investigación*, No. 66, Bogotá.
- Restrepo, C. (2007). *Teoría del caos*. Universidad del Cauca.
- Sterpin, D. (2011). Perceptrón auto-supervisado: Una red neuronal artificial capaz de replicación memética. *Revista Educación en Ingeniería*, No. 12, pp. 90-101, Bogotá, ACOFI.
- Sterpin, D. (2015). Cognición imitativa para un robot mediante una comunidad de replicadores neuro-meméticos. *Revista Tecnura*, Vol. 19, No. 44, pp. 15-31, Bogotá.