

# Topología de una RNA para calcular la cinemática inversa del Puma 560 para aplicaciones en tiempo real

David Bedolla-Martínez<sup>1</sup>, Felipe de Jesús Trujillo-Romero<sup>2</sup>, Esther Lugo-González<sup>3</sup>

<sup>1</sup> Universidad de la Sierra Juárez, Instituto de Estudios Ambientales,  
Ixtlán de Juárez, Oaxaca, México

<sup>2</sup> Universidad de Guanajuato, Departamento de Ingeniería Electrónica,  
División de Ingenierías Campus Irapuato-Salamanca, Guanajuato, México

<sup>3</sup> Universidad Tecnológica de la Mixteca, Instituto de Electrónica y Mecatrónica,  
Huajuapán de León, Oaxaca, México

bedolla@unsij.edu.mx, elugog@mixteco.utm.mx,  
fdj.trujillo@ugto.mx

**Resumen.** Se presenta una red neuronal artificial que resuelve la cinemática inversa de un robot manipulador Puma 560 de 6 grados de libertad, utilizando el menor número de parámetros en su topología que aseguren un buen desempeño. La topología de la red neuronal artificial se obtuvo usando un algoritmo genético para determinar el número de neuronas en las capas ocultas. Como resultado se presenta una red neuronal para eficientar la respuesta de los resultados para su implementación en tiempo real. Se muestran gráficas del error cuadrático medio, así como la comparación del método analítico y la red neuronal para los 6 grados de libertad.

**Palabras clave:** cinemática inversa, Puma 560, red neuronal artificial, algoritmo genético.

## Topology of an RNA to Calculate the Inverse Kinematics of the Puma 560 for Real-Time Applications

**Abstract** We present an artificial neural network that solves the inverse kinematics of a Puma 560 manipulator robot with 6 degrees of freedom, using the least number of parameters in its topology that ensure good performance. The topology of the artificial neural network was obtained using a genetic algorithm to determine the number of neurons in the hidden layers. As a result, a neural network is presented to improve the response of the results for its implementation in real time. Graphs of the mean square error are shown, as well as the comparison of the analytical method and the neural network for the 6 degrees of freedom.

**Keywords:** inverse kinematics, Puma 560, artificial neural network, genetic algorithm.

## 1. Introducción

Los robots manipuladores son de gran importancia debido a las diferentes aplicaciones donde se usan, desde la industria manufacturera hasta servicios de robótica

de rehabilitación. Para desarrollar estas tareas se debe conocer la posición y orientación del efector final. La acción individual de cada junta debe ser controlada de tal forma que se ejecute la trayectoria de movimiento deseado. Específicamente, la localización espacial del efector final se determina con el desplazamiento traslacional y rotacional de cada junta con respecto a una referencia dada [1].

La cinemática inversa de un robot manipulador tradicionalmente se obtiene de forma analítica donde se usan funciones trigonométricas que necesitan varias operaciones para obtener un valor aproximado. En el área de sistemas embebidos que operan en tiempo real, los ciclos de reloj que toma cada operación son críticos, al igual que el espacio en memoria. Por esta razón se utilizarán como método alternativo las RNA's (redes neuronales artificiales).

Las RNA's se han utilizado para el cálculo de la cinemática inversa de robots manipuladores de 3 grados de libertad [2] y robots bípedos [3]. Una alternativa es encontrar la mejor topología de una RNA para minimizar el error en el cálculo de la cinemática inversa, variando los parámetros de aprendizaje y los pesos iniciales [4].

Durante la optimización, es necesario modificar los parámetros de la topología. Este proceso se realiza comúnmente a prueba y error [5] y [6], necesitando una inversión considerable de tiempo y experiencia por tanto se usa un Algoritmo Genético (AG).

Los AG's han sido usados para optimización o entrenamiento de RNA's [7]. En [8] se probó la técnica diseño robusto de RNA's para obtener los parámetros que consigan el mejor diseño de una RNA para resolver un problema específico. En [9] se usa un AG para entrenar una RNA de tipo Feedforward. En [10], [11] y [12] se realiza una búsqueda de la mejor topología de una RNA para conseguir un mejor desempeño usando entrenamiento estándar.

En [13] una RNA se usa en combinación de un algoritmo genético para el entrenamiento. Aunque hay algoritmos para entrenar RNA's, es necesario presentar una topología predeterminada, comúnmente se han usado AG's para obtener dicha topología.

En este trabajo se presenta la topología de una RNA que minimiza el error de entrenamiento usando el algoritmo de Levenberg-Marquardt, dicha topología se encontró usando un algoritmo genético.

### 1.1. Cinemática inversa del Puma 560

Las siguientes ecuaciones se toman de [14], El Unimation Puma 560 es un robot con seis grados de libertad con juntas rotativas.

De la tabla 1 se tiene  $\alpha_i$ ,  $a_i$ ,  $d_i$  y  $\theta_i$  que son los parámetros Denavit-Hartenberg del robot de 6 grados de libertad.

**Tabla 1.** Parámetros Denavit-Hartenberg.

$i$	$\alpha_{i-1}$	$a_{i-1}$	$d_i$	$\theta_i$
1	0	0	0	$\theta_1$
2	$-90^\circ$	0	0	$\theta_2$
3	0	$a_2$	$d_3$	$\theta_3$
4	$-90^\circ$	$a_3$	$d_4$	$\theta_4$
5	$90^\circ$	0	0	$\theta_5$
6	$-90^\circ$	0	0	$\theta_6$

La matriz de transformación homogénea se muestra en la ecuación (1), donde  $p_{x,y,z}$  representa la posición del efector final. Los valores de  $r_{ij}$  representan la matriz de orientación.

$${}^0_6T = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1)$$

La solución de la cinemática inversa está dada por las ecuaciones (2-8). Los valores de  $s_i$  y  $c_i$ , corresponden a los valores de senos y cosenos de  $\theta_i$  respectivamente y los valores de  $a_i$  y  $d_i$ , corresponden a los parámetros Denavit-Hartenberg.

Para  $\theta_1$  se escribe como:

$$\theta_1 = \text{atan2}(p_x, p_y) - \text{atan2}\left(d_3, \pm\sqrt{p_x^2 + p_y^2 - d_3^2}\right), \quad (2)$$

donde

$$d_3 = -s_1 p_x + c_1. \quad (3)$$

Para  $\theta_3$

$$\theta_3 = \text{atan2}(a_3, d_4) - \text{atan2}\left(K, \pm\sqrt{a_3^2 + d_4^2 - K^2}\right), \quad (4)$$

donde

$$K = \frac{p_x^2 + p_y^2 + p_z^2 - a_2^2 - a_3^2 - d_3^2 - d_4^2}{2a_2}. \quad (5)$$

Para  $\theta_2$

$$\theta_2 = \text{atan2}\left((-a_3 - a_2 c_3)p_z - (c_1 p_z + s_1 p_y)(d_4 - a_2 s_3), (a_2 s_3 - d_4)p_z - (a_3 + a_2 c_3)(c_1 p_x + s_1 p_y)\right). \quad (6)$$

Para  $\theta_4$

$$\theta_4 = \text{atan2}(-r_{13}s_1 + r_{23}c_1, -r_{13}c_1c_{23} - r_{23}s_1c_{23} + r_{33}s_{23}). \quad (7)$$

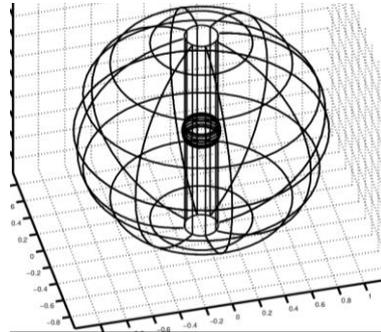
Para  $\theta_5$

$$\theta_5 = \text{atan2}\left(-r_{11}(c_1c_{23}s_4 - s_1c_4) - r_{21}(s_1c_{23}s_4 + c_1c_4) + r_{31}(s_{23}s_4), r_{11}[(c_1c_{23}c_4 + s_1s_4)c_5 - c_1s_{23}s_5] + r_{21}[(s_1c_{23}c_4 - c_1s_4)c_5 - s_1s_{23} - r_{31}(s_{23}c_4c_5 + c_{23}s_5)]\right). \quad (8)$$

El ángulo  $\theta_6$  modifica la orientación a lo largo del eje z del efector final.

## 2. Metodología

Los datos de entrenamiento, validación y prueba, se generaron de forma analítica usando las ecuaciones para el robot Puma 560 (ecuaciones (2)-(8)). El toolbox de P. Corke, "Robotic Toolbox" [15] se usó para generar los ejemplos de entrenamiento, los cuales se encuentran dentro de su espacio de trabajo (ver figura 1). Cada ejemplo consiste en 6 valores para definir la posición y orientación del efector final y 6 valores para definir la solución de la cinemática inversa.



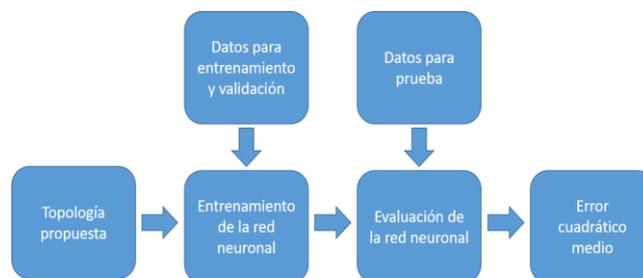
**Fig. 1.** Espacio de trabajo del Robot Puma 560 (Tomado de [16]).

Cada red es entrenada con diferentes pesos iniciales aleatorios para evitar caer en mínimos locales, posteriormente es evaluada usando el error cuadrático medio. El objetivo es encontrar una RNA que presente el mejor desempeño mientras se varía el número de neuronas de dos capas ocultas, con el fin de obtener el mínimo número de parámetros.

Los parámetros para la implementación del algoritmo genético son:

- Los alelos del gen son de tipo entero.
- Los genes de un individuo tienen la forma de  $[n_1 \ n_2]$ .
- El tamaño de la población es de 10 individuos.
- La selección se hizo por torneos.
- El máximo número de generaciones es de 1000.
- El rango de búsqueda en el número de neuronas es de 6 a 30.
- La población inicial se genera aleatoriamente.
- La probabilidad de cruce y mutación es de 0.8 y 0.1 respectivamente.

La figura 2 muestra el diagrama de flujo del AG, que procesa cada RNA. Como primer paso una RNA es generada con  $n_1$  y  $n_2$  para la primera y segunda capa oculta posteriormente la RNA se entrena, valida y evalúa obteniendo el error cuadrático medio.



**Fig. 2.** Diagrama de flujo de la función fitness.

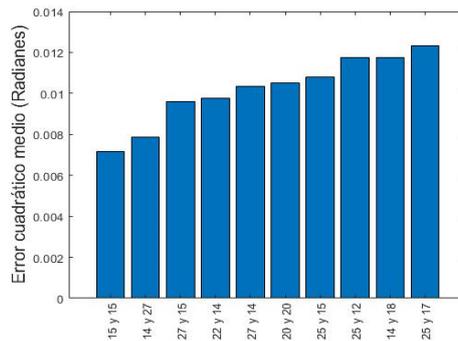
En la tabla 2 se muestran las consideraciones para el entrenamiento de las RNA's:

**Tabla 2.** Parámetros de entrenamiento de la RNA.

Parámetro	Valor
Método de entrenamiento	Levenberg-Marquardt
Número máximo de épocas	20,000
Función de activación en las capas ocultas	Sigmoidal
Número de neuronas en cada capa	6 - 30
Pesos iniciales	Aleatorio
Mínimo gradiente	$1 \times 10^{-7}$
Función de desempeño	Error cuadrático medio
División de datos	70% Entrenamiento
	20% Validación
	10% Evaluación final

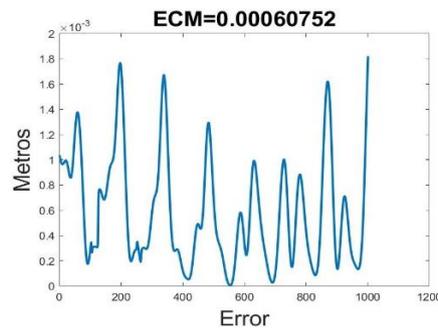
### 3. Resultados

Al término de la ejecución del AG se tienen 10 RNA's con el mejor desempeño (ver Fig. 3). La red con el error cuadrático medio menor es elegida.



**Fig. 3.** Desempeño de las redes neuronales finales.

Como resultado se obtuvo una RNA con un error cuadrático medio de 0.00060752 metros con 15 neuronas en la primer y segunda capa oculta. En la figura 4 se muestra el error entre la trayectoria deseada y la obtenida con la RNA.



**Fig. 4.** Error entre la trayectoria deseada y la salida de la RNA.

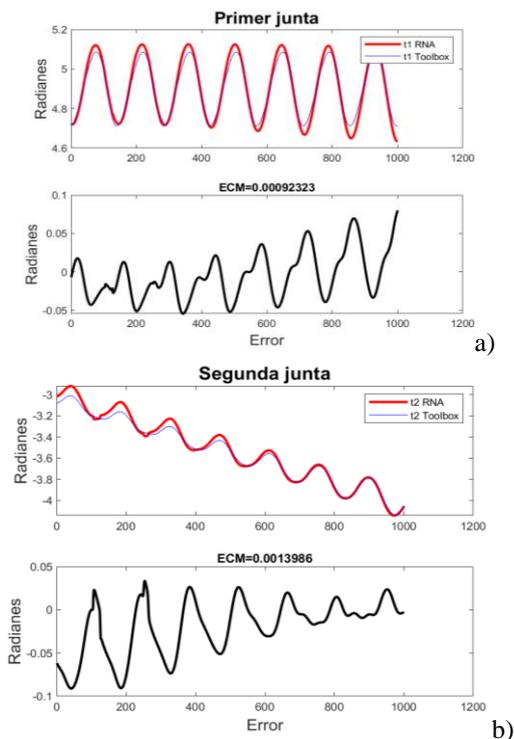


Fig. 5. Resultado para la primer y segunda junta.

En la tabla 3 se muestra el error cuadrático medio en radianes para cada junta.

Tabla 3. Máximo error para cada junta.

Junta	Error cuadrático medio (rad)
1°	0.00092323
2°	0.0013986
3°	0.0030617
4°	0.037452
5°	0.00089664
6°	0.11495

En la figura 5, 6 y 7 se muestra la comparación entre los datos de evaluación (en azul) y la salida de la RNA (en rojo) y su gráfica de error (en negro) para cada junta. Los datos de evaluación se tomaron del 10% de un total de 286746 ejemplos analíticos.

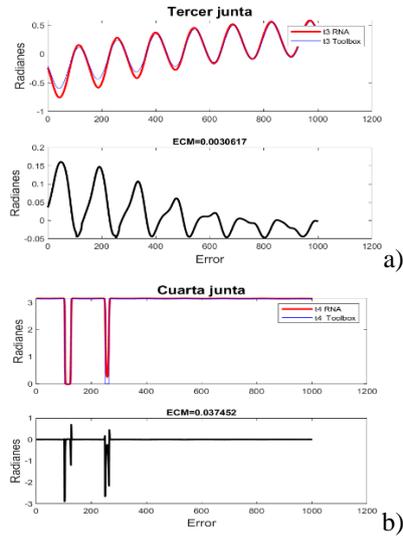


Fig. 6. Resultado para la tercer y cuarta junta.

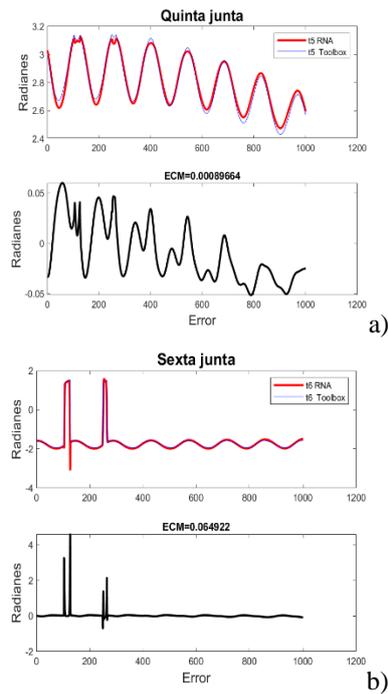


Fig. 7. Resultados para la quinta y sexta junta.

Los resultados obtenidos demuestran que la RNA obtenida es competitiva con el método analítico (Ec. 2-8) para su aplicación en tiempo real.

## 4. Conclusiones

La principal ventaja de la RNA es la velocidad con la que resuelve la cinemática inversa a comparación de los métodos analíticos que requieren la aproximación de funciones trigonométricas. El error de salida es relativamente pequeño en la mayoría de los casos. La desventaja es que el proceso de optimización toma tiempo considerable debido a que cada individuo (RNA) debe ser entrenado y evaluado. Como trabajo a futuro se busca reducir el error cuadrático medio usando diferentes algoritmos de entrenamiento y la implementación de la RNA obtenida en un sistema embebido para aplicaciones de tiempo real.

## Referencias

1. Ramos, G.: Cinemática inversa de Robots industriales. *Tecnura 10*, Con Ciencias, Colombia pp. 54–58 (2002)
2. Hossian, A., Sierra, E., Fernández, E., Britos, P., García, R.: El problema cinemático en manipuladores robóticos industriales, un abordaje de solución mediante redes neuronales artificiales. En: VI Jornadas Iberoamericanas de Ingeniería de Software e Ingeniería del conocimiento, Lima, Perú, pp. 427–434 (2017)
3. Vele, O.: Cinemática inversa de un robot bípedo. Investigación personal, Researchgate.net, pp. 1–20 (2006)
4. Castillo, P.A., García, F., Merelo, J., Prieto, M.: Diseño de redes neuronales artificiales mediante algoritmos evolutivos. *Revista Iberoamericana de Inteligencia Artificial* 5(14), 2–32 (2001)
5. Giraldo, L., Delgado, E., Castellanos, G.: Cinemática inversa de un brazo robot utilizando algoritmos genéticos. *Avances en sistemas e informática*, Medellín, 3(1), 29–34 (2006)
6. Kardan, M.R., Setayeshi, S., Koohi-Fayegh, R., Ghiassi-Nejad, M.: Neutron spectra unfold in bonner spheres spectrometry using neural networks. *Radiation protection dosimetry* 104(1), 27–30 (2003)
7. Vaca-González, J.J., Peña-Caro, C.A., Vaca- González, H.: Cinemática inversa de robot serial utilizando algoritmo genético basado en MCDS. *Tecnura* 19(44), 33–46 (2015)
8. Reyes-Alfaro, A., Ortiz-Rodríguez, J.M., Reyes-Haro, A., Castañeda-Miranda, R., Solís-Sánchez, L.O., Vega-Carrillo, H.R.: Optimización de redes neuronales artificiales para la reconstrucción del espectro de neutrones y sus dosis equivalentes. *ISSSD*, Cusco, Perú, pp. 4–5 (2014)
9. Montana, D.J., Davis, L.: Training feedforward neural network using genetic algorithms. In: *IJCAI'89 Proceedings of the 11th international joint conference on Artificial intelligence*, pp. 762–767 (1989)
10. Miller, G., Todd, P., Hedge, S.: Designing neural networks using genetic algorithms. In: *Proceedings of the 3rd International Conference on Genetic Algorithms*, pp. 2–3 (1989)
11. Kitano, H.: Designing Neural Networks using genetic algorithms with graph Generation System. *Complex System*, vol. 4, pp. 461–476 (1990)
12. Delgado, J. A.: Redes neuronales y algoritmos genéticos. En: *Jornadas de Ingeniería Eléctrica y Electrónica (FIEE)*, vol. 13, pp. 65–68 (1992)
13. Tellez, P.: Training feedforward neural networks using neural networks and genetic algorithms. In: *International conference on computing, communications and control technologies*, pp. 3–4 (2006)

14. Craig, J.J.: Introduction to robotics: mechanics and control. Pearson Education, México, (2005)
15. Corke, P.: Robotic Toolbox. [www.petercorke.com](http://www.petercorke.com). Último acceso: 03/05/2019
16. Ghosal, A.: Manipulator Kinematics. Handbook of Manufacturing Engineering and Technology, pp. 1777–1808 (2015)