

IDENTIFICACIÓN NO PARAMÉTRICA DE UN BRAZO ROBÓTICO DE 2 GDL UTILIZANDO REDES NEURONALES RBF

NON-PARAMETRIC IDENTIFICATION OF A 2 DOF ROBOTIC ARM USING RBF NEURONAL NETWORKS

Miuler Junior Blas Marquina¹, Carlos Junior Flores Ramírez Jose², Adrian Humberto Mejías Sandoval³, Angel Omar Terrones Escobedo⁴, Josmell Henry Alva Alcántara⁵

¹²³⁴⁵Universidad Nacional de Trujillo, Facultad de Ingeniería, Escuela Académica Profesional de Ingeniería Mecatrónica, Av. Juan Pablo II s.n., Trujillo, La libertad, Perú, mblas@unitru.edu.pe jfloresra@unitru.edu.pe amejiass@unitru.edu.pe aterrones@unitru.edu.pe josalva@unitru.edu.pe

RESUMEN

En este trabajo se presenta un método para determinar un modelo del sistema dinámico de un manipulador planar de 2 GDL, utilizando redes neuronales Radial Basis Function. Obtener un modelo del robot manipulador analíticamente es complicado, debido a los efectos de fricción, inercia de los actuadores, desgastes, etc, siendo este, además, un sistema no lineal y variante en el tiempo. Lo que se propone es entrenar una red neuronal RBF para que esta represente el comportamiento dinámico de un manipulador robótico de 2 GDL. La elección de esta red es debido a que gozan de características en el modelado de sistemas dinámicos como la capacidad de aproximar funciones no lineales con precisión, y la posibilidad de permitir diseño de controladores utilizando diversas metodologías. La obtención del modelo se basa en la identificación, para lo cual será preciso implementar un prototipo del robot a identificar, realizar el modelo cinemático del robot que nos permita obtener datos de salida para el algoritmo de identificación no paramétrica por redes neuronales. Después, se deben tener los datos de entrada, voltaje de actuadores, y posición como salida para la red neuronal, luego realizar un preprocesamiento, normalizando los datos y poder dividir estos para entrenamiento, validación y prueba. Se tienen que definir los parámetros internos de la red neuronal, número de capas, neuronas por cada capa, tipo de aprendizaje, y criterios de parada del algoritmo. En los experimentos, se alcanzó un error cuadrático medio de 7.45 con 500 neuronas en la capa oculta.

Palabras clave: Brazo robótico, identificación no-paramétrica, redes neuronales, sistemas no lineales.

ABSTRACT

This paper presents a method to determine a model of the dynamic system of a 2 DOF planar manipulator, using Radial Basis Function neural networks. Obtaining a model of the robot manipulator analytically is complicated, due to the effects of friction, inertia of the actuators, wear, etc. This is also a non-linear and time-variant system. The aim is to train a RBF neural network to represent the dynamic behavior of a 2 DOF robotic manipulator. The choice of this network is due to their characteristics in the modeling of dynamic systems such as the ability to approximate non-linear functions with precision, and the ability to allow the design of controllers using various methodologies. Obtaining the model is based on identification, for which it will be necessary to implement a prototype of the robot to be identified, to perform the kinematic model of the robot that will allow us to obtain output data for the algorithm of nonparametric identification by neural networks. Afterwards, input data, actuator voltage, and position as output for the neural network must be obtained, then preprocessing, normalizing the data and dividing them for training, validation and testing. The internal parameters of the neural network, number of layers, neurons per layer, type of learning, and stop criteria of the algorithm must be defined. In the experiments, a mean quadratic error of 7.45 was reached with 500 neurons in the occult layer.

Keywords: robotic arm, non-parametric identification, neural networks, non-linear systems.

*Autor para correspondencia: mblas@unitru.edu.pe





INTRODUCCIÓN

En años recientes, el desarrollo de las redes neuronales ha permitido una mejora en los métodos de identificación de sistemas no lineales. Particularmente, las redes del tipo Función de Base Radial (RBF, por sus siglas en inglés) se han utilizado para aproximar sistemas dinámicos no lineales, con buenos resultados y un grado de precisión sobresaliente (Barrientos *et al.* 2007). Así mismo, estas pueden asegurar la consecución de un sistema estable, y la garantía de una convergencia en el error durante el proceso de identificación (Yan *et al.* 2000).

Se ha planteado la utilización de redes RBF, puesto que estas han probado ser aproximadores universales, es decir, capaces de aproximar muchos comportamientos dinámicos, obteniéndose una respuesta con estas superior a las obtenidas con otro tipo de red, como las típicas Perceptrón de Capa Múltiple (MLP, en inglés) (Pairan *et al.* 2020).

Es posible, incluso, lograr el control de robots manipuladores a partir de las redes neuronales, señalándose en una investigación, que la RBF es

la red más idónea para estos casos, por su velocidad de entrenamiento y la precisión de sus predicciones (Isasi 2004). Contándose a la red RBF como un método probado para la resolución de problemas dinámicos y cinemáticos de robots manipuladores (Jin *et al.* 2018).

En la presente investigación se plantea el entrenamiento de una red neuronal RBF para la identificación de un Brazo Robótico Planar de 2 Grados de Libertad, teniendo como entradas al sistema los valores de voltaje efectivo suministrado a los actuadores (motores DC), y como salidas las coordenadas cartesianas del efector final.

MÉTODOS

Implementación del brazo robótico de 2GDL

La estructura robótica está compuesta por 2 motores DC, con sus respectivos eslabones metálicos como se puede apreciar en la figura 3. Los sensores MPU 6050, que nos permitirán saber los ángulos de rotación, se colocan encima de cada eslabón para saber el ángulo de rotación.

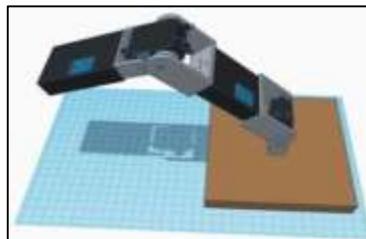


Figura 1. Diseño CAD brazo robótico 3GDL.



Figura 2. Brazo robótico implementado.

La configuración electrónica usa dos Arduinos. El primero para la lectura de los sensores MPU 6050 conectados mediante protocolo I2C al microcontrolador y en configuración tipo bus

entre ellos, el segundo Arduino se conecta a través. El sistema tiene un disparador mediante un pulsador, el cual activa el inicio de la toma de datos y del movimiento de los servomotores.

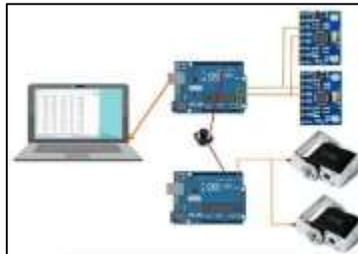


Figura 3. Configuración electrónica para toma de datos.

Obtención de datos de entrada

Para generar las entradas de voltaje al motor DC, se emplea un módulo puente H manejado por un controlador Arduino. Este módulo aplica una señal de voltaje sobre los dos motores, para esto se hizo un arreglo para que el voltaje sea de la misma magnitud sobre los dos motores, pero en cada uno una polaridad opuesta. Es decir, mientras que sobre un motor se aplica de 0 a 7V, sobre el otro -7 a 0V.

Para registrar la información del voltaje aplicado sobre el motor en cada experimento, se optó por emplear el sensor de voltaje INA226, el cual

puede registrar voltajes positivos aplicados sobre una carga. Se mide el voltaje que varía de 0 a 7V, sabiendo que el voltaje sobre el otro motor tiene el mismo valor, pero con polaridad opuesta.

Obtención de datos de salida reales

Para los datos de salida que serían la posición X e Y del efector final respecto la base, nos basamos en la cinemática directa de un brazo robótico de 2GDL mediante los parámetros de Denavit-Hartenberg. [4]

Las longitudes de cada eslabón las tenemos desde la elaboración del brazo robótico y los ángulos de

rotación los obtenemos mediante módulos MPU6050 integrados a cada eslabón.

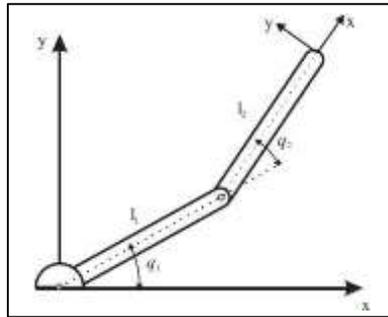


Figura 5. Robot planar de 2GDL.

Esta será nuestra referencia de la cual hallaremos los parámetros de Denavit- Hartenberg.

Tabla 1. Parámetros de Denavit- Hartenberg.

Articulación	θ_{ii}	d_i	a_{ii}	a_{ii}
1	q_1	0	13.6	0
2	q_2	0	12	0

A partir de estos datos y del cálculo de la cinemática directa obtenemos que las coordenadas reales del efector final son:

$$P_x = 12\cos(q_1+q_2)+13.6\cos(q_1)$$

$$P_y = 12\sin(q_1 + q_2) + 13.6\sin(q_1)$$

$$178.16^\circ \leq q_1 \leq 16.81^\circ$$

$$49.86^\circ \leq q_2 \leq -317.44^\circ$$

Construcción de la Red de Base Radial, Entrenamiento y Pruebas

Construcción de la Red RBF

En el presente paper, se clasificaron los Voltajes RMS de ambos motores DC mencionados

anteriormente, por lo cual se obtuvieron 535 datos. Los cuales, los datos del 1-427 son utilizados como datos de entrenamiento y los datos de 428-535 se utilizaron como datos de validación.

En las redes RBF, el número de entrada de neuronas, depende de los factores que pueden afectar la posición del efector final, las cuales, como se mencionó anteriormente son 2, V_{rms1} y V_{rms2} . El valor de salida son las posiciones en el eje X y eje Y, por lo que la salida son 2. En el software MATLAB, se utiliza la función “newrbf”, proveniente del Neural Network Toolbox, esta función se utiliza para crear una función precisa de Redes Neuronales Función de Base Radial, que pueden elegir automáticamente

el número de capa oculta, lo que hace que los pronósticos sean más precisos. Para usar los datos de entrada en la red neuronal RBF, primero se realizó un preprocesamiento de estos, lo cual consta en normalizar los datos antes de poder entrenarlo, ya que, las funciones de activación, en nuestro caso función sigmoidea, estarán en un rango de [0,1], y si no se normaliza probablemente no podrán aproximarse a la salida deseada.

Posteriormente se seleccionan los parámetros aleatoriamente para la función “newrbf”, los

cuales son: SPREAD, este nos da la difusión de funciones de base radial, MN, nos da el número máximo de neuronas ocultas y goal, la cual se refiere a la meta a la que tiene que llegar el error cuadrático medio. Se eligió los valores de 7.45 y 0.2 respectivamente. En la figura 6 se puede observar el número de neuronas en la capa de entrada, 427, refiriéndose al número de entradas y de la capa oculta lineal se tiene 2 capas en la cual tiene 500 neuronas internas, y finalmente la capa de salida, la cual es 2.

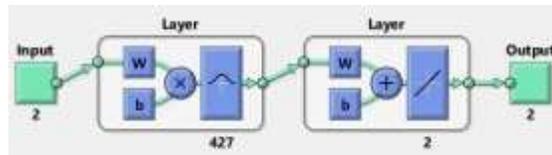


Figura 6. Estructura de la red neuronal RBF.

Entrenamiento y pruebas

Utilizando las funciones anteriormente propuestas, el entrenamiento y pruebas se realiza de forma directa en el entorno de programación MATLAB. Para el ámbito de las pruebas se hace uso de la función “sim”, del mismo Toolbox anteriormente expuesto, esta función simula nuestra red entrenada frente a unos datos de pruebas, ya sea datos validados o de entrenamiento.

RESULTADOS Y DISCUSIÓN

Los datos de entrada

Se tuvieron que obtener datos de entrada con los voltajes RMS de los motores DC, en términos del tiempo de operación. Como se planteó previamente. Además, observamos en las figuras 7 y 8 los voltajes de entradas a los que son sometidos nuestros motores DC.

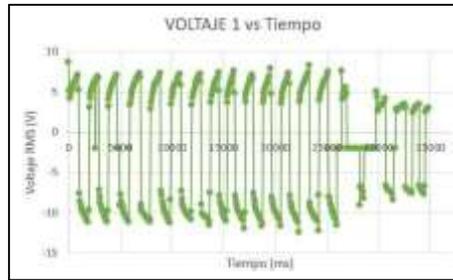


Figura 7. Voltaje RMS de entrada al eslabón 1.

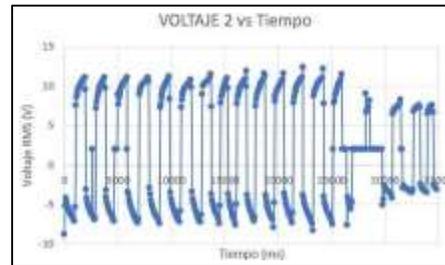


Figura 8. Voltaje RMS de entrada al eslabón 2.

En años recientes, el desarrollo de las redes neuronales ha permitido una mejora en los métodos de identificación de sistemas no lineales. Particularmente, las redes del tipo Función de Base Radial (RBF, por sus siglas en inglés) se han utilizado para aproximar sistemas dinámicos no lineales, con buenos resultados y un grado de precisión sobresaliente (Barrientos *et al.* 2007). Así mismo, estas pueden asegurar la consecución de un sistema estable, y la garantía de una convergencia en el error durante el proceso de identificación (Yan *et al.* 2000).

Datos de salida

Mediante la cinemática directa obtuvimos las posiciones X e Y del efector final respecto a la base a partir de los voltajes RMS ingresados a los motores DC.

Como se mencionó anteriormente, se seleccionó una meta de error cuadrático medio de 0.2, esto es factible siempre y cuando las neuronas lo permitan y puedan llegar a ese error mínimo. En la figura 8, se observa que la Red RBF, alcanzó un error cuadrático medio de 7.35 con las 500 neuronas de la capa oculta, pudiendo ser menos si se implementa más neuronas en la capa oculta.

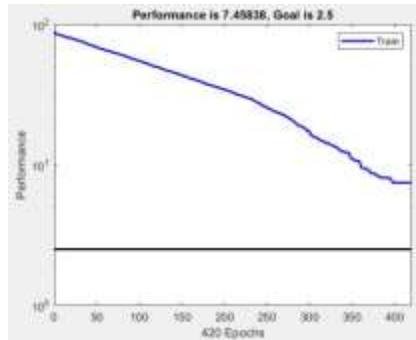


Figura 9. Error cuadrático medio de la red neuronal RBF.

Se ha planteado la utilización de redes RBF, puesto que estas han probado ser aproximadores universales, es decir, capaces de aproximar muchos comportamientos dinámicos, obteniéndose una respuesta con estas superior a las obtenidas con otro tipo de red, como las típicas Perceptrón de Capa Múltiple (MLP, en inglés) (Pairan *et al.* 2020).

En la figura 10 se observan los datos de la posición real que se pretenden predecir.

Por último, en la figura 11 se puede observar las predicciones de las posiciones con los datos de validación.

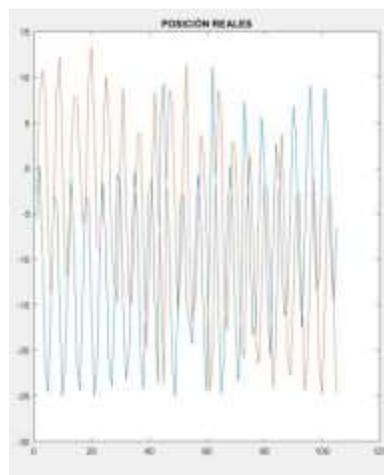


Figura 10. Posiciones Reales.

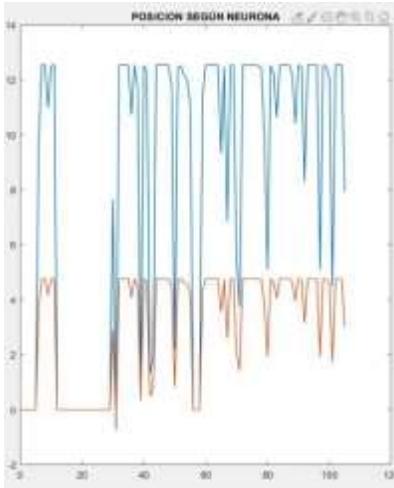


Figura 11. Posiciones predichas por la Red RBF.

Es posible, incluso, lograr el control de robots manipuladores a partir de las redes neuronales, señalándose en una investigación, que la RBF es la red más idónea para estos casos, por su velocidad de entrenamiento y la precisión de sus predicciones (Isasi 2004). Contándose a la red RBF como un método probado para la resolución de problemas dinámicos y cinemáticos de robots manipuladores (Jin *et al.* 2018).

CONCLUSIONES

La arquitectura de la red RBF planteada, fue incapaz de predecir correctamente el comportamiento del sistema robótico. Así mismo modificar la naturaleza del actuador, pasando de un servomotor a otro motor DC, implicó un incremento en la dificultad para medir datos. Finalmente, la carencia de algunos equipos y

dispositivos para implementar sistemas de medición de voltaje bipolar más precisos, representó una restricción que afectó significativamente la calidad de los datos de entrada para el entrenamiento de la red.

RECOMENDACIONES

La toma de datos válidos con el motor DC representó una tarea significativamente dificultosa, y a pesar de obtener valores adecuados, la arquitectura presentada no se ajustó de mejor manera, por lo que en futuros trabajos se deben explorar otras arquitecturas.

CONFLICTO DE INTERÉS

El autor no tiene conflicto de interés de ninguna índole.

REFERENCIAS

- Barrientos, A., Peñín, L., Balaguer, C., & Aracil, R. (2007). Fundamentos de Robótica. McGraw-Hill.
<https://eltrasteroloco.files.wordpress.com/2017/03/267380685-fundamentos-de-robotica.pdf>
- Jin, L., Li, S., Yu, J., & He, J. (2018). Robot manipulator control using neural networks: A survey. *Neurocomputing*, 285, pp. 23–34.
<https://www.sciencedirect.com/science/article/abs/pii/S0925231218300158>
- Pairan, M. F., Shamsudin, S. S., & Zulkafli, M. F. (2020). Neural Network-Based System Identification for Quadcopter Dynamic Modeling: A Review. *Journal of Advanced Mechanical Engineering Applications*, 1(2), pp. 20–33.
https://www.researchgate.net/publication/345605104_Neural_Network-Based_System_Identification_for_Quadcopter_Dynamic_Modeling_A_Review
- Pedro Isasi, I. M. G. (2004). *Redes neuronales artificiales - Un enfoque práctico*. Pearson Prentice Hall, pp.63. <http://librodigital.sangregorio.edu.ec/librosusgp/08867.pdf>
- Yan, L., Sundararajan, N., & Saratchandran, P. (2000). Nonlinear System Identification Using Lyapunov Based Fully Tuned Dynamic RBF Networks. *Neural Processing Letters* 2000 12:3, 12(3), pp. 291–303.
https://www.researchgate.net/publication/225845823_Nonlinear_System_Identification_Using_Lyapunov_Based_Fully_Tuned_Dynamic_RBF_Networks