



*Mejora de la precisión del seguimiento de trayectoria de un brazo robótico mediante un algoritmo de red neuronal inicializado con Lyapunov*

*Improving the trajectory tracking accuracy of a robotic arm using a neural network algorithm initialized with Lyapunov*

*Melhorando a precisão do rastreamento de trajetória de um braço robótico usando um algoritmo de rede neural inicializado com Lyapunov*

Jesús Rodríguez-Flores <sup>I</sup>

[jesus.rodriguez@ute.edu.ec](mailto:jesus.rodriguez@ute.edu.ec)

<https://orcid.org/0000-0001-6254-2348>

Jonnathan Chamba-Cruz <sup>III</sup>

[jonnathan.chamba@ute.edu.ec](mailto:jonnathan.chamba@ute.edu.ec)

<https://orcid.org/0000-0003-2792-7271>

Luis Sánchez-Muyulema <sup>II</sup>

[luis.sanchez@ute.edu.ec](mailto:luis.sanchez@ute.edu.ec)

<https://orcid.org/0000-0001-5078-3734>

Andrés Morocho-Caiza <sup>IV</sup>

[andres.morocho@epoch.edu.ec](mailto:andres.morocho@epoch.edu.ec)

<https://orcid.org/0000-0003-3146-8784>

**Correspondencia:** [jesus.rodriguez@ute.edu.ec](mailto:jesus.rodriguez@ute.edu.ec)

Ciencias Técnicas y Aplicadas  
Artículo de Investigación

\* **Recibido:** 13 de agosto de 2023 \* **Aceptado:** 10 de septiembre de 2023 \* **Publicado:** 10 de octubre de 2023

- I. Universidad UTE, Ecuador.
- II. Universidad UTE, Ecuador.
- III. Universidad UTE, Ecuador.
- IV. Escuela Superior Politécnica de Chimborazo, Ecuador.



## Resumen

Se plantea un método de corrección adaptativa de la desviación en la trayectoria de un brazo robótico mediante una red neuronal, abordando problemas de inestabilidad y desviación en seguimiento de movimientos cuando el manipulador es afectado por factores externos. Se emplea una red neuronal de flujo directo o no realimentada para analizar las características dinámicas del manipulador y se formula un sistema de control adaptativo basado en esta tecnología. La estabilidad y convergencia del sistema se demuestran utilizando la función de Lyapunov, siendo esta la condición de inicialización de pesos de la red neuronal. Se emplea MATLAB/Simulink para simular los parámetros dinámicos del manipulador y se comparan los resultados con un sistema de control PID. Las simulaciones muestran que, con ciertos parámetros, el brazo robótico sigue una trayectoria esperada con precisión. Cuando el entorno externo perturba el movimiento del brazo, la red neuronal adaptativa minimiza el error de seguimiento y reduce la fluctuación en el par de entrada. En resumen, el uso de este método mejora significativamente la precisión en el control de la trayectoria de movimiento y reduce la vibración del manipulador.

**Palabras Clave:** Manipulador; Control; Red neuronal adaptativo; Trayectoria; Robótica.

## Abstract

A method for adaptive correction of the deviation in the trajectory of a robotic arm is proposed using a neural network, addressing problems of instability and deviation in movement tracking when the manipulator is affected by external factors. A forward-flow or non-feedback neural network is used to analyze the dynamic characteristics of the manipulator and an adaptive control system is formulated based on this technology. The stability and convergence of the system are demonstrated using the Lyapunov function, this being the weight initialization condition of the neural network. MATLAB/Simulink is used to simulate the dynamic parameters of the manipulator and the results are compared with a PID control system. Simulations show that with certain parameters, the robotic arm follows an expected trajectory precisely. When the external environment disturbs the arm movement, the adaptive neural network minimizes the tracking error and reduces the fluctuation in the input torque. In summary, the use of this method significantly improves the accuracy of motion trajectory control and reduces manipulator vibration..

**Keywords:** Manipulator; Control; adaptive neural network; Trajectory; Robotics.

## Resumo

Um método para correção adaptativa do desvio na trajetória de um braço robótico é proposto utilizando uma rede neural, abordando problemas de instabilidade e desvio no rastreamento de movimento quando o manipulador é afetado por fatores externos. Uma rede neural de fluxo direto ou sem feedback é usada para analisar as características dinâmicas do manipulador e um sistema de controle adaptativo é formulado com base nesta tecnologia. A estabilidade e convergência do sistema são demonstradas através da função Lyapunov, sendo esta a condição de inicialização do peso da rede neural. MATLAB/Simulink é usado para simular os parâmetros dinâmicos do manipulador e os resultados são comparados com um sistema de controle PID. Simulações mostram que, com certos parâmetros, o braço robótico segue com precisão uma trajetória esperada. Quando o ambiente externo perturba o movimento do braço, a rede neural adaptativa minimiza o erro de rastreamento e reduz a flutuação no torque de entrada. Em resumo, o uso deste método melhora significativamente a precisão do controle da trajetória do movimento e reduz a vibração do manipulador.

**Palavras-chave:** Manipulador; Ao controle; rede neural adaptativa; Trajetória; Robótica.

## Introducción

En la contemporaneidad, la industria manufacturera se enfrenta a una coyuntura de cambio y transformación sin precedentes. La necesidad apremiante de innovación en este sector responde a la imperativa búsqueda de eficiencia, precisión y adaptabilidad en los procesos productivos. La introducción de tecnologías de vanguardia constituye un elemento central para afrontar los retos actuales y futuros. En este contexto, la mejora de la precisión del seguimiento de trayectoria de un brazo robótico mediante un algoritmo de red neuronal, emerge como un pilar esencial en la senda de la innovación manufacturera.

La convergencia entre la robótica y la inteligencia artificial se erige como un hito transcendental en la evolución de la industria manufacturera. La implementación de algoritmos basados en redes neuronales proporciona un marco teórico y técnico para abordar con precisión el desafío de mejorar la precisión del seguimiento de la trayectoria de un brazo robótico. Esta metodología amalgama la capacidad de aprendizaje y adaptación propia de las redes neuronales con la ejecución precisa y coordinada de un brazo robótico, representando un avance significativo en el horizonte de la manufactura de precisión.

En este contexto, la investigación y desarrollo de métodos de control adaptativos para mitigar la desviación en la trayectoria del movimiento de un brazo robótico, se erige como un imperativo científico y tecnológico de primera magnitud. La aplicación de algoritmos de red neuronal ofrece una respuesta prometedora, al permitir la corrección dinámica y en tiempo real de las desviaciones en la trayectoria de movimiento. Este enfoque no solo atenúa las imprecisiones causadas por perturbaciones externas, sino que también proyecta una nueva frontera en la capacidad de los sistemas robóticos para operar con precisión milimétrica en entornos manufactureros de alta exigencia.

La sinergia entre robótica y automatización ha emergido como un catalizador crucial para la optimización de la producción en la industria contemporánea. La incorporación de sistemas robóticos altamente especializados y automatizados en los procesos manufactureros permite una ejecución precisa y constante de tareas complejas. Esto no solo reduce los tiempos de ciclo y aumenta la capacidad de producción, sino que también minimiza errores humanos, elevando así la calidad de los productos finales. La robótica colaborativa, en particular, ha abierto nuevas perspectivas al permitir una interacción segura y eficiente entre humanos y robots en el espacio de trabajo, lo que se traduce en una mayor flexibilidad y versatilidad en la línea de producción.

Por otro lado, la mejora de la precisión del seguimiento en trayectoria de un brazo robótico mediante un algoritmo de red neuronal representa un avance significativo en el campo de la robótica de precisión. Esta metodología aprovecha la capacidad de aprendizaje y adaptación de las redes neuronales para corregir de forma dinámica las desviaciones en la trayectoria del brazo robótico. Al reducir la incidencia de errores en la ejecución de movimientos, se garantiza una mayor precisión en la manipulación de objetos y componentes en el proceso productivo. Este enfoque no solo promete una producción más eficiente, sino que también abre la puerta a aplicaciones de alta precisión en áreas como la industria aeroespacial o la fabricación de dispositivos médicos.

La inestabilidad en el movimiento y el considerable error de seguimiento en la trayectoria de un manipulador robótico cuando se ve influenciado por factores externos representa una problemática de relevancia crítica en la ingeniería robótica. Estos desafíos surgen de la compleja interacción entre el manipulador y su entorno, donde perturbaciones imprevistas pueden desencadenar respuestas no lineales que comprometen la precisión y seguridad de las operaciones. Fenómenos como vibraciones, cambios súbitos en la carga y resistencias inesperadas pueden tener un impacto

sustancial en la ejecución de movimientos, lo que requiere de estrategias avanzadas para mitigar estos efectos adversos y garantizar un desempeño óptimo del manipulador en condiciones cambiantes.

La resolución de este problema de inestabilidad y error de seguimiento de trayectoria demanda un enfoque multidisciplinario que integre la mecánica, la electrónica y la informática. Estrategias de control adaptativo, basadas en algoritmos de aprendizaje y ajuste continuo, se presentan como soluciones prometedoras para contrarrestar la influencia de factores externos. Asimismo, la incorporación de tecnologías de sensorica avanzada y sistemas de retroalimentación en tiempo real juegan un papel determinante en la detección y corrección de desviaciones. En definitiva, abordar esta problemática no solo potencia la eficiencia de los manipuladores robóticos, sino que también fortalece su capacidad para operar en entornos dinámicos y cambiantes, promoviendo así su aplicabilidad en un amplio espectro de sectores industriales.

### **Trabajos relacionados**

Los trabajos relacionados a la temática tratada se agrupan con bases en sus enfoques y métodos similares:

- Control de trayectorias de singularidad en manipuladores robóticos

El artículo "Control of Singularity Trajectory Tracking for Robotic Manipulator by Genetic Algorithms"(Rebouças Filho et al., 2019) propone un enfoque para controlar el seguimiento de trayectorias de singularidad en manipuladores robóticos utilizando algoritmos genéticos. El estudio se centra en la aplicación de la inteligencia artificial para mejorar la precisión y la eficiencia en la manipulación de objetos.

- Control adaptativo y robusto para manipuladores de robots industriales

El artículo "A novel Robust Adaptive Control Using RFWNNs and Backstepping for Industrial Robot Manipulators with Dead-Zone" (Quynh et al., 2020) propone un nuevo controlador para mejorar el seguimiento de la posición de los manipuladores de robots industriales en presencia de dinámicas y perturbaciones desconocidas. El controlador utiliza redes neuronales recurrentes de ondaletas difusas para aproximar las dinámicas desconocidas y un compensador de zona muerta para tratar los problemas de zona muerta.

El artículo "An Adaptive Backstepping Nonsingular Fast Terminal Sliding Mode Control for Robust Fault Tolerant Control of Robot Manipulators" (Van et al., 2019) propone una metodología

de control basada en un algoritmo de deslizamiento de modo terminal rápido no singular de retroceso adaptable para mejorar el control robusto y tolerante a fallos de manipuladores de robots. La metodología se centra en reducir el error de seguimiento de una trayectoria deseada, incluso en presencia de incertidumbres paramétricas.

El artículo "Adaptive model-free control for robotic manipulators" (Safaei et al., 2017) presenta un enfoque de control adaptativo sin modelo para manipuladores robóticos. El método propuesto utiliza un controlador de redes neuronales adaptativo de tiempo fijo diseñado dentro del marco TSMC y el marco PPCA para lograr la convergencia global del controlador.

- Control de sistemas no lineales utilizando redes neuronales

El artículo "Backstepping terminal sliding mode control of robot manipulator using radial basis functional neural networks"(Vijay & Jena, 2017) presenta un enfoque de control novedoso para manipuladores robóticos utilizando redes neuronales de funciones de base radial y el modo deslizante terminal de retroceso. El método propuesto se centra en mejorar la precisión y la estabilidad del control de los manipuladores, lo que lo hace aplicable a diferentes sistemas no lineales.

- Control de manipuladores de brazos robóticos

El artículo "Trajectory Tracking Control of Robotic Manipulator via Variable Gain Iterative Learning Algorithm"(Zheng et al., 2018) presenta un enfoque de control de seguimiento de trayectorias para manipuladores robóticos utilizando un algoritmo de aprendizaje iterativo de ganancia variable. El método propuesto mejora la precisión y la estabilidad del control de los manipuladores, lo que es importante en aplicaciones industriales y de robótica.

El artículo "Trajectory Tracking of Robotic Arm Based on Power Regulation of Actuator Using Neural Averaged Subgradient Control"(Hernandez-Sanchez et al., 2022) presenta un enfoque de control para el seguimiento de trayectorias de un brazo robótico. El método propuesto utiliza un control basado en subgradientes promediados y redes neuronales para regular la potencia del actuador, lo que mejora la precisión y la estabilidad del control del brazo robótico.

El artículo "The design of a neural network-based adaptive control method for robotic arm trajectory tracking"(Xu & Wang, 2022) presenta un enfoque de control basado en redes neuronales para el seguimiento de trayectorias de un brazo robótico. El método propuesto utiliza un controlador adaptativo que mejora la precisión y la estabilidad del seguimiento de la trayectoria, lo que es importante en aplicaciones industriales y de robótica.



El artículo "Robotic Arm Trajectory Tracking Control Based on An RBF Neural Network Adaptive Control Algorithm"(Qin et al., 2022) presenta un enfoque de control para el seguimiento de trayectorias de un brazo robótico utilizando un algoritmo de control adaptativo basado en redes neuronales de funciones de base radial (RBF). El método propuesto mejora la precisión y la estabilidad del control del brazo robótico, lo que es importante en aplicaciones industriales y de robótica.

## Metodología

### Características dinámicas del brazo robótico

El robot a tratar tiene un mecanismo de cadena abierta que consta de  $N + 1$  eslabones rígidos y  $N$  articulaciones (Gautier, 1986). La base del robot se numera como eslabón 0 y el eslabón terminal como eslabón  $N$ . La articulación  $i$  se encuentra entre el eslabón  $i$  y el eslabón  $i - 1$ , y tiene un grado de libertad (rotacional o translacional). Se fija un marco de coordenadas  $R_i$  con respecto al eslabón  $i$ . El eje de la articulación  $i$  se encuentra a lo largo de  $Z_i$ . Se utilizarán los parámetros modificados de Denavit y Hartenberg (D&H) para definir las matrices de transformación. La variable de articulación  $q_i$  es  $\theta_i$  si la articulación  $i$  es rotacional y  $r_i$  si la articulación  $i$  es translacional.

La orientación de las coordenadas  $R_i$  con respecto a las coordenadas  $R_{i-1}$  se define como se muestra en la ecuación (1).

$${}^{i-1}A_i = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 \\ \cos(\alpha_i)\sin(\theta_i) & \cos(\alpha_i)\cos(\theta_i) & -\sin(\alpha_i) \\ \sin(\alpha_i)\sin(\theta_i) & \sin(\alpha_i)\cos(\theta_i) & \cos(\alpha_i) \end{bmatrix} \quad (1)$$

Un vector  $V$  expresado en el marco de coordenadas  $R_i$  se denotará como  ${}^iV$ . La posición del origen de  $R_i$  con respecto al origen de  $R_{i-1}$ , denotado como  ${}^{i-1}P_i$ , presentada por la ecuación (2).

$${}^{i-1}P_i = [d_i \quad -r_i \cdot \sin(\alpha_i) \quad r_i \cdot \cos(\alpha_i)]^T \quad (2)$$

La ecuación de movimiento de un mecanismo articulado puede ser descrita por la ecuación (3)

$$\tau = M\ddot{q} + V\dot{q}\dot{q} + F\dot{q}^2 + G \quad (3)$$

Donde



$$\dot{q} = [\dot{q}_1 \quad \dot{q}_2 \quad \cdots \quad \dot{q}_N]^T \quad (4)$$

$$\dot{q}\dot{q} = [\dot{q}_1\dot{q}_2 \cdots \dot{q}_1\dot{q}_N \quad \dot{q}_2\dot{q}_3 \cdots \dot{q}_2\dot{q}_N \cdots \dot{q}_{N-1}\dot{q}_N]^T$$

$$\dot{q}^2 = [\dot{q}_1^2 \quad \dot{q}_2^2 \quad \cdots \quad \dot{q}_N^2]^T$$

El vector  $\tau = N \times 1$ , representa la fuerza de los actuadores o torques. La matriz  $M$  es la matriz de inercia de un tamaño de  $N \times N$ , esta matriz es simétrica y no singular. La matriz  $G = N \times 1$  es el vector que representa la carga gravitatoria.  $V$  y  $F$  representan las matrices de Coriolis y centrífuga, respectivamente, de dimensiones  $N \times (N - 1)/2$  y  $N \times N$ . Los elementos  $M$ ,  $V$ ,  $F$ ,  $G$  son funciones de la variable  $q$  de las uniones y de la suposición de los parámetros de Denavit-Hartenberg y de los parámetros inerciales. Para obtener la ecuación (3) se emplea el formalismo del método de Lagrange. Si se asume la siguiente notación, donde  $m_i$  es la masa del enlace  $i$ -ésimo,  ${}^i J_i$  es la matriz de inercia del enlace  $i$ -ésimo referido al origen  $O_i$ , respecto a  $R_i$ , siendo  $m_i$  planteado como  ${}^i S_i = [mX_i \quad mY_i \quad mZ_i]^T$  siendo estos los momentos del enlace  $i$ -ésimo.

Considerando lo expuesto, se puede reescribir la ecuación (3) como se muestra en la ecuación (5)

$$\tau = \sum_{m=1}^M x_m (M^m \ddot{q} + V^m \dot{q}\dot{q} + F^m \dot{q}^2 + G^m) \quad (5)$$

$$\tau = \sum x_m D^m = D \cdot X \quad (6)$$

Donde:

$$M^m = \frac{\partial}{\partial x_m} M \quad (7)$$

$$V^m = \frac{\partial}{\partial x_m} V \quad (8)$$

$$F^m = \frac{\partial}{\partial x_m} F \quad (9)$$

$$G^m = \frac{\partial}{\partial x_m} G \tag{10}$$

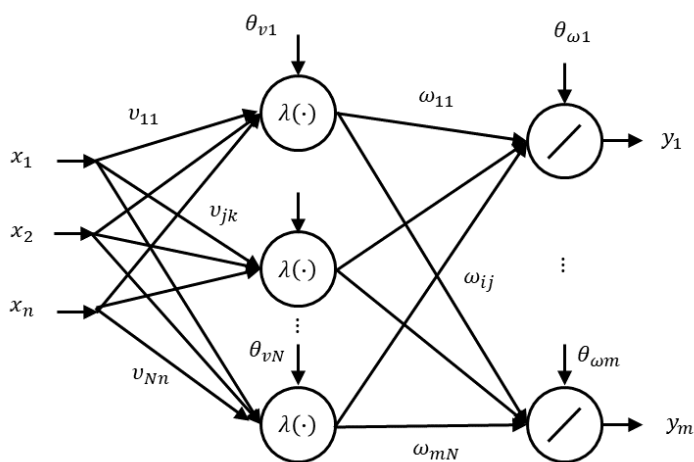
Considerando que  $x = f(q)$ , siendo  $f$  la solución positiva de la ecuación de la cinemática, al obtener la segunda derivada se tiene la ecuación (11), siendo  $J(q)$  la matriz Jacobiana del manipulador y su pseudo inversa obtenida con la ecuación (12).

$$x'' = J'(q)q' + J(q)q'' \tag{11}$$

$$J^\xi = [J(q)]^T (J(q)[J(q)]^T)^{-1} \tag{12}$$

### Redes neuronales múlticapa de flujo directo

Considere un sistema de tres capas, siendo estas una capa de entrada, una capa oculta y finalmente, una capa de salida, como se muestra en la Figura 1. Cada neurona de la capa oculta presenta una salida definida por la ecuación (13), indexada  $j$ , desde  $j = 1$  hasta  $N$ , siendo  $N$  la cantidad de neuronas ocultas, siendo el indexador  $k$  el encargado de indexar las entradas, de tal manera que inicia en  $k = 1$  y termina en  $k = n$ , siendo  $n$  la cantidad de entradas o neuronas de entrada del sistema. Esta capa presenta una función de activación que pudiere ser de cualquier naturaleza, es decir lineal o no lineal, no obstante, se ha implementado una función de activación no lineal como la presentada en la ecuación (14).



**Figura 1.** Red neuronal de flujo directo o feedforward

$$u_j = \lambda \left( \sum_{k=1}^n v_{jk} \cdot x_k + \theta_{vj} \right) \quad (13)$$

$$\lambda(p) = \frac{1}{1 + e^{-p}} \quad (14)$$

Las neuronas de salida se indexan desde  $i = 1$  hasta  $i = m$ , por lo que la expresión resultante es como se muestra en la ecuación (15). En esta ecuación el indexador  $j$  asigna la salida de la capa oculta y su respectivo peso.

$$y_i = \sum_{j=1}^N \omega_{ij} \cdot u_j + \theta_{\omega i} \quad (15)$$

En definitiva, el sistema neuronal de propagación directa con  $n$  entradas y  $m$  salidas, con funciones de activación no lineal en las capas ocultas y lineal en la capa de salida, queda como se muestra en la ecuación (16).

$$y_i = \sum_{j=1}^N \omega_{ij} \cdot \lambda \left( \sum_{k=1}^n v_{jk} \cdot x_k + \theta_{vj} \right) + \theta_{\omega i} \quad (16)$$

Al sustituir los valores de los pesos de la red neuronal  $v_{jk}$  y  $\omega_{ij}$  en matrices de pesos  $V^T$  y  $W^T$ , respectivamente, se puede representar la ecuación (16) como se muestra en (17)

$$y = W^T \lambda(V^T x) \quad (17)$$

## Controlador neuronal adaptativo

### Análisis de error

Al estudiar el parámetro de valor de entrada  $\tau(t)$  del par de control, la trayectoria de operación real del manipulador final del robot puede ser lo más consistente posible con la trayectoria de operación esperada. Por lo tanto, el valor de entrada del controlador debe utilizar de manera efectiva la redundancia de los grados de libertad del brazo del robot para lograr un seguimiento efectivo de las sub tareas del programa. El error de seguimiento del espacio de operación  $e(t) \in R^m$  se puede

expresar mediante la ecuación (18) en la cual  $x_d$  es la trayectoria esperada en el espacio de operación.

$$e(t) = x_d - x \quad (18)$$

El error de seguimiento de la sub tarea definida  $e_N(t) \in R^m$  se expresa mediante la ecuación (19).

En la ecuación (19),  $g(\cdot) \in R^m$  se establece de acuerdo con el objetivo de control de la sub tarea.

$$e_N(t) = (I_n - J^\xi D)(g - q') \quad (19)$$

Para proporcionar incentivos para la determinación de los objetivos de control de las sub tareas del programa, se toma la derivada temporal de la ecuación (18), después de la simplificación, el sistema de seguimiento de trayectoria de espacio de operación en bucle cerrado puede ser finalmente escrito como se presenta en la ecuación (20), donde  $\alpha \in R^n$ , siendo  $\alpha$  una matriz diagonal de ganancias y  $r(t)$  perteneciente a  $R^n$  es la señal filtrada del error de seguimiento, dicha señal satisface la ecuación (21).

$$e' = -\alpha e + Jr \quad (20)$$

$$r = J^\xi(x'_d + \alpha e) + (I_n - J^\xi J)g - q' \quad (21)$$

Usando la ecuación (19), se establece el valor del parámetro de entrada de control para asegurar que el error del espacio de operación y el error de seguimiento después del filtrado se ajusten dentro del rango especificado. Luego, se pueden utilizar las propiedades de la matriz inversa para lograr los requisitos anteriores, obteniéndose la ecuación (22). Finalmente, se puede observar a partir de la ecuación (22) que  $e_N(t)$  puede ser ajustado incidiendo sobre la variable  $r(t)$ , de modo que el control de la sub-tarea del programa pueda ser realizado.

$$e_N = (I_n - J^\xi J)r \quad (22)$$

### Control por redes neuronales

El diseño del controlador basado en redes neuronales de flujo directo o feedforward, se basa en los siguientes considerandos:

Consideración 1: Tanto  $x_d$ ,  $g(t)$  así como  $g'(t)$ , son funciones acotadas en el tiempo.

Consideración 2: Todos los parámetros de las funciones cinética y cinemática están limitadas bajo todas las condiciones de  $D$ .

Para mayor comodidad en la notación, la matriz de pesos de la red neuronal se especifica con base en la ecuación (23), al mismo tiempo, los errores de estimación de los pesos se presentan en la ecuación (24). El valor de peso ideal de la red neuronal está acotado y satisface las relaciones planteadas en la ecuación (25), siempre y cuando se conozca  $Z_M$ , siendo  $tr(\cdot)$  la función traza que devuelve la suma de los elementos de la diagonal en una matriz cuadrada. Al mismo tiempo, bajo la condición de conocida  $y$ , se puede estimar el error en la estimación de valores presentes en la capa oculta, siendo esta desviación presentada en la ecuación (26) tomando en consideración los valores de entrada.

$$Z \equiv diag\{W, V\} \quad (23)$$

$$\tilde{W} = W - \hat{W} \quad (24)$$

$$\tilde{V} = V - \hat{V}$$

$$\tilde{Z} = Z - \hat{Z}$$

$$\|Z\|_F \leq Z_M \quad (25)$$

$$Z_F^2 = tr(Z^T Z)$$

$$\tilde{\lambda} = \lambda(V^T x) - \lambda(\hat{V}^T x) \quad (26)$$

En cuanto al error de seguimiento del brazo robótico después de la proyección, sus características dinámicas se pueden escribir haciendo uso de la ecuación (27).

$$M \cdot r' = -V^m r - Kr + \hat{W}^T \hat{\lambda} \tilde{V}^T x + \tilde{W}^T \hat{\lambda} + \omega - J^T e \quad (27)$$

En la ecuación (28) la matriz de ganancia se define a partir de un valor positivo para  $K = \alpha$ .

$$\omega = \tilde{W}^T \lambda(\tilde{V}^T x) + 2W^T \lambda(\tilde{V}^T x) + \varepsilon \quad (28)$$

En consecuencia, el valor límite  $\omega(t)$  del término de interferencia puede ser determinado por las constantes  $c_0$ ,  $c_1$  y  $c_2$ . Las estimaciones de peso  $\hat{W}$  y  $\hat{V}$  están acotadas, y los errores de seguimiento del espacio de operación y de la subtarea pueden ser ajustados para ser arbitrariamente pequeños. Comprobación para la función candidata de Lyapunov de la siguiente ecuación:

$$\varphi = \frac{1}{2}e^T + \frac{1}{2}r^T Mr + \frac{1}{2}tr(\tilde{W}^T F_\omega^{-1} \tilde{W}) + \frac{1}{2}tr(\tilde{V}^T G_v^{-1} \tilde{V}) \quad (29)$$

Tomando la derivada temporal de la función de Lyapunov  $\varphi$ , combinando las ecuaciones  $\tilde{W}' = -\hat{W}'$ ,  $\tilde{V}' = -\hat{V}'$  y la regla de aprendizaje adaptativo, se puede obtener la (30).

$$\varphi' = -e^T \alpha e - r^T K r \quad (30)$$

La estabilidad de la función de Lyapunov está determinada por  $\varphi' \leq 0$ , lo que garantiza que  $r(t)$ ,  $\hat{W}$  y  $\hat{V}$  estén acotados.

La acotación de la función  $r(t)$  asegura la acotación de las funciones  $e(t)$  y  $e'(t)$ , como una consecuencia de haber acotado a  $x(t)$ ,  $x'(t)$  e  $y(t)$ , esta acción o medida ocasiona que la trayectoria se encuentre también acotada en el intervalo esperado. La acotación de la señal representada en el lado derecho de la ecuación (27) verifica la acotación de  $r(t)$  y  $\varphi$ , y la continuidad consistente de  $\varphi$ . Dado que el cambio de  $\varphi$  con el tiempo tiende a cero para  $t$ ,  $e(t)$  y  $r(t)$  también tienden a cero, por lo que se puede concluir a partir de la ecuación (22) que  $e_N(t)$  también tiende a cero.

### Criterios y parámetros de valoración

Durante la simulación de las pruebas, se obtienen resultados cuya valoración permiten determinar la calidad comparativa y bondad de desempeño de los controladores

### Desviación Relativo Porcentual de Posicionamiento

Compara la posición del manipulador reportada por los sensores respecto a la consigna. El criterio se determina aplicando la ecuación (31).

$\sigma_{\%Y} = 100 \times \frac{\sqrt{\sum_{i=1}^N (Y_i - Y_{sp_i})^2}}{\sqrt{\sum_{i=1}^N (Y_{sp_i})^2}}$	(31)
-------------------------------------------------------------------------------------------------------------	------

**Determinación de una derivada de alta precisión para una señal  $\hat{\delta}_{(i)}$  arbitraria cualquiera.**

Una derivada de alta precisión se consigue al considerar una mayor cantidad de términos en un desarrollo polinomial de Taylor, por lo que con 4 términos se consigue una derivada en correspondencia con el valor teórico ante una valoración con una señal sinusoidal. La ecuación (32) muestra la expresión resultante para el cálculo.

$$\frac{d\hat{\delta}_{(i)}}{dt} = \begin{pmatrix} \hat{\delta}_{(i-1)} - \hat{\delta}_{(i)} & \frac{T_s^2}{2} & \frac{-T_s^3}{6} & \frac{T_s^4}{24} \\ \hat{\delta}_{(i-2)} - \hat{\delta}_{(i)} & \frac{4T_s^2}{2} & \frac{-8T_s^3}{6} & \frac{16T_s^4}{24} \\ \hat{\delta}_{(i+1)} - \hat{\delta}_{(i)} & \frac{T_s^2}{2} & \frac{T_s^3}{6} & \frac{T_s^4}{24} \\ \hat{\delta}_{(i+2)} - \hat{\delta}_{(i)} & \frac{4T_s^2}{2} & \frac{8T_s^3}{6} & \frac{16T_s^4}{24} \end{pmatrix} \cdot \begin{pmatrix} -T_s & \frac{T_s^2}{2} & \frac{-T_s^3}{6} & \frac{T_s^4}{24} \\ -2T_s & \frac{4T_s^2}{2} & \frac{-8T_s^3}{6} & \frac{16T_s^4}{24} \\ T_s & \frac{T_s^2}{2} & \frac{T_s^3}{6} & \frac{T_s^4}{24} \\ 2T_s & \frac{4T_s^2}{2} & \frac{8T_s^3}{6} & \frac{16T_s^4}{24} \end{pmatrix}^{-1} \quad (32)$$

**Velocidad Eficaz**

La velocidad eficaz es un parámetro global que evalúa el cambio en el tiempo de la posición del manipulador, en este caso, se plantean tanto para la consigna, como para las posiciones alcanzadas por los controladores durante el desempeño de posicionamiento. Para determinar este parámetro se hace uso de las ecuaciones (32) y (33).

$$\dot{Y}_{rms} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\dot{Y}_i)^2} \quad (33)$$

**Valor eficaz del Torque**

El torque eficaz es un parámetro global que presenta el valor cuadrático medio del Torque, durante el desempeño de posicionamiento del manipulador, su forma de calcularlo se presenta en la ecuación (34).

$$T_{rms} = \sqrt{\frac{1}{N} \sum_{i=1}^N (T_i)^2} \quad (34)$$



### Potencia Cuantitativa

La potencia cuantitativa, es un valor aproximado cuando se considera de manera no vectorial tanto el torque y la velocidad, es decir, se considera tan sólo la magnitud del vector. Determinando valores eficaces, tanto para el torque como para la velocidad, se establece un criterio de desempeño bajo igualdad de condiciones para ambos controladores al momento de cumplir la consigna en el tiempo y posicionar el manipulador del brazo robótico. Es importante destacar, que el indicador recibe el nombre de potencia cuantitativa porque expresa un número que permite efectuar comparación, pero no se corresponde con un valor determinístico de la potencia total empleada por el brazo robótico durante el cumplimiento de la trayectoria.

$$P_c = \frac{1}{N} \sqrt{\left(\sum_{i=1}^N (T_i)^2\right) \left(\sum_{i=1}^N (\dot{Y}_i)^2\right)} \quad (35)$$

### Análisis de Resultados

Simulación y análisis. Con el fin de evaluar el efecto de seguimiento del controlador de movimiento controlado por la red neuronal adaptativa, se llevó a cabo una ronda de simulaciones sobre el error de seguimiento del controlador de movimiento en MATLAB/Simulink.

Se simula y analiza el error de seguimiento del brazo robótico diseñado para tal prueba. Los resultados de la simulación del brazo robótico se efectúan en dos escenarios, que, por ser simulado, permiten simultaneidad para los casos. Un escenario consiste en evaluar el brazo robótico ante un controlador clásico PID, y el otro escenario, consiste en evaluar el brazo robótico ante un controlador neuronal multicapa de flujo directo o feedforward, pero ante ambos escenarios la condición de consigna es la misma con el propósito de valorar la capacidad de desempeño de ambos controladores. La Tabla 1 presenta las especificaciones de prueba de simulación del mismo brazo robótico ante un control PID Clásico y un control Neuronal multicapa de flujo directo o feedforward, adicionalmente, se presentan los datos estructurales, mínimos, empleados para la especificación de la simulación.

**Tabla 1.** Especificaciones de prueba de simulación del mismo brazo robótico ante un control PID Clásico y un control Neuronal multicapa de flujo directo o feedforward

Consigna	$\theta = 0.4 \cos\left(\frac{2\pi}{1s}t\right)$	Parámetros del manipulador
----------	--------------------------------------------------	----------------------------

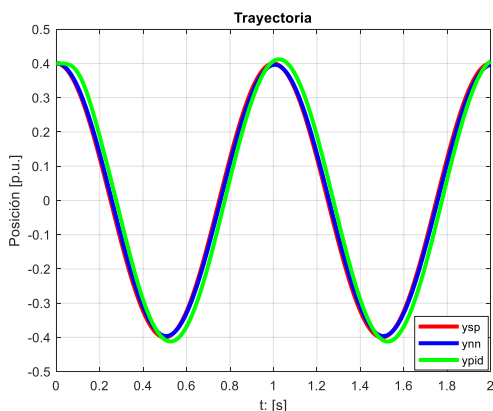
Condición Inicial	$\theta(0) = [0 \ 0 \ 0]^T$		$l_1$	$l_2$	$l_3$
Parámetro de control K	$K = diag(40,40,40)$	Dimensión Eslabón	0.62m	0.41m	0.34m
Parámetro de Interferencia	$r' = 0.08sin\left(\frac{2\pi}{2s}t\right)$				
Condición gravitatoria	Tiempo de Simulación	Masa Eslabón	$m_1$	$m_2$	$m_3$
$g = 9.82 \frac{m}{s^2}$	T inicial    T final				
	0 seg.        2.8 seg.	3.5 kg        2.5 kg        2.0 kg			

La capacidad de seguimiento es un criterio de valoración en control que se encuentra reñido con el gasto energético o estrés inherente ya sea mecánico, eléctrico o de cualquier otra naturaleza de la variable de interés del sistema estudiado, es por ello que las funciones de costo asociadas a la optimización de controladores suelen penalizar el proceso de corrección con base en un aporte del gradiente de cambio o velocidad de corrección, como se muestra en la ecuación (36), no obstante ello, el precio debe pagarse para cuando se desea fidelidad en el seguimiento y se incorporan otros criterio, tal como se muestra en la ecuación (35), en esta ecuación se aprecia una estimación de la potencia que se constituye, en su naturaleza, en un criterio de valoración o función de costo del desempeño de los controladores.

$J = \int_0^{t_f} \left( \varepsilon^2 - \alpha \left( \frac{d\varepsilon}{dt} \right)^2 \right) dt$	<b>(36)</b>
------------------------------------------------------------------------------------------------------	-------------

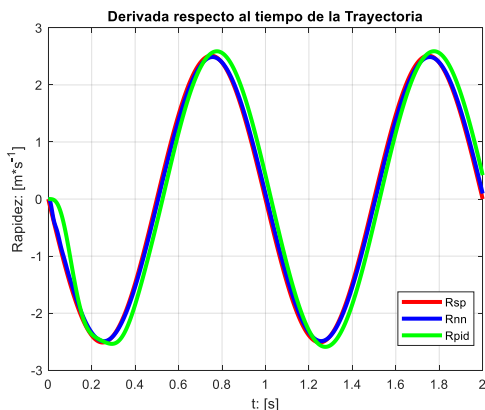
La **Figura 2** muestra como la consigna de posición, en trazo rojo, varía en el tiempo, y simultáneamente las acciones de control del brazo robótico, reaccionan para lograr la posición con el menor error posible y el menor tiempo, por ende, con el menor desfase. Lograr la posición con el menor desfase es una condición de seguimiento en el tiempo, que por simple observación se concluye que el trazo azul, correspondiente al control neuronal es de desempeño superior al trazo verde, logrado por el control pid clásico, y en efecto, su valoración con base en la desviación

porcentual de posición respecto  $Y_{sp}$  o consigna de posición en el tiempo, muestra que  $\sigma_{\%Y_{nn}} \leq \sigma_{\%Y_{pid}}$  Tal y como se aprecia en **Tabla 2**.

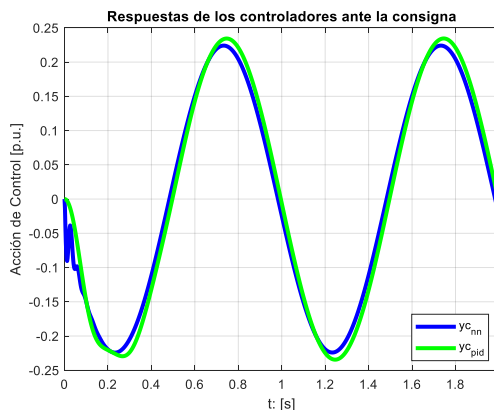


**Figura 2.** Trayectoria del brazo robótico ante consigna, trazo rojo, siendo el resultado de seguimiento de la consigna en trazo azul para el control neuronal y entrazo verde para el control pid clásico.

Adicionalmente, se tiene que, procesando la posición en el tiempo, haciendo uso de la ecuación (32) se obtiene la razón de cambio o velocidad tanto asociada a la consigna como a las posiciones producto de la acción de control neuronal como pid clásica, dicho comportamiento se observa en la **Figura 3 a)**, siendo la **Figura 3 b)** el comportamiento de la acción de control actuando sobre los brazos robóticos. El procesamiento de las velocidades permite obtener su valor eficaz, observándose que  $\dot{Y}_{pid} \geq \dot{Y}_{nn}$ , tal y como se observa en la **Tabla 2**, esta condición es de esperarse, puesto que la calidad del seguimiento implica menor velocidad, ya que en presencia de desviación se debe recorrer más distancia en el mismo tiempo, situación presente en la acción de control pid clásica.



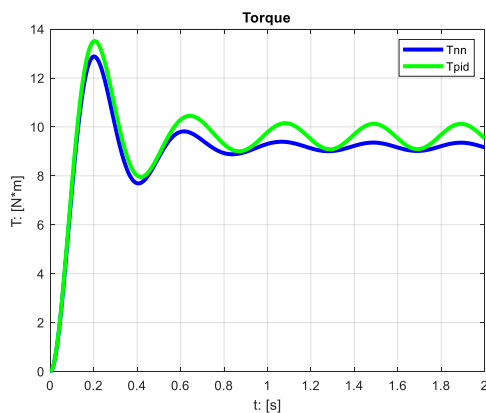
a)



b)

**Figura 3.** a) Razón de cambio en el tiempo: trazo rojo, consigna, trazo azul, posición con el control neuronal, trazo verde, posición con el control pid clásico. b) Acción de control: trazo azul, salida del controlador neuronal, trazo verde, salida del controlador pid clásico.

Considerándose que el torque es medido a partir de la simulación, su comportamiento en el tiempo puede apreciarse en **Figura 4**, siendo el comportamiento del torque en el tiempo como consecuencia de la acción de control neuronal mostrado en trazo azul y en verde el torque logrado con la acción de control pid clásica. Los valores eficaces, para los torques obtenidos, se determinan haciendo uso de la ecuación (34), y sus valores se observan en la **Tabla 2**, resultando que  $T_{nn} \leq T_{pid}$ .



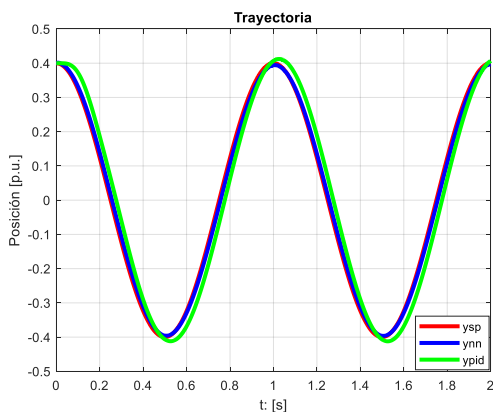
**Figura 4.** Torque medido a partir de la simulación, para lograr el posicionamiento del brazo robótico ante la consigna. El trazo azul es la medición del torque ante una acción de control neuronal, y el trazo verde, ante una acción de control pid clásico.

Como la velocidad no puede ser usada como criterio excluyente o definitivo, para valorar la bondad en el seguimiento de trayectorias o consignas de posicionamiento, se plantea usar como criterio la potencia cuantitativa. Es decir, la desviación en el posicionamiento, debe considerarse y este debe ponderarse con la potencia cuantitativa, bajo este criterio se tiene: Desviación porcentual de Posición respecto a  $Y_{sp}$ , 4.02% para el control neuronal, contra 16.73% para el control pid, Potencia cuantitativa, 16.192186 W para el control neuronal, contra 17.649309 W para el control pid clásico, tal y como se observa en la **Tabla 2**.

<b>Tabla 2.</b> Resultados del desempeño Ante una misma consigna del Controlador del brazo robótico ante un control PID Clásico y un control Neuronal multicapa de flujo directo o feedforward
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Desviación porcentual de Posición respecto a $Y_{sp}$		Velocidad Eficaz		
$\sigma_{\%Y_{nn}}$	$\sigma_{\%Y_{pid}}$	$\dot{Y}_{sp}$ [m/s]	$\dot{Y}_{nn}$ [m/s]	$\dot{Y}_{pid}$ [m/s]
4.02	16.73	1.776264	1.759991	1.830270
Torque eficaz		Potencia cuantitativa		
$T_{nn}$ [Nm]	$T_{pid}$ [Nm]	$P_{nn}$ [W]	$P_{pid}$ [W]	
9.200149	9.643006	16.192186	17.649309	

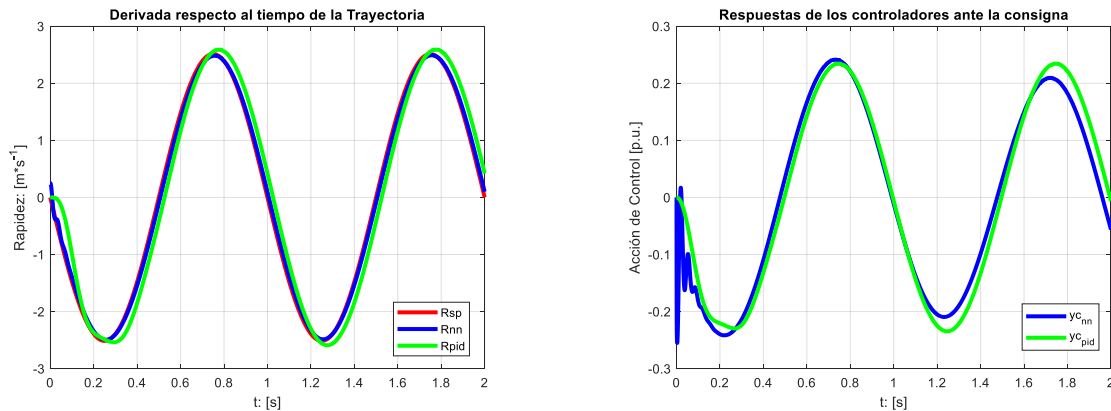
En la **Figura 5** se repite el estudio presentado en la **Figura 2**, en este caso en presencia de perturbación en el manipulador del brazo robótico, los resultados encontrados son semejantes, es decir, se tiene que  $\sigma_{\%Y_{nn}} \leq \sigma_{\%Y_{pid}}$  Tal y como se aprecia en la **Tabla 3**.



**Figura 5.** Trayectoria del brazo robótico ante consigna y en presencia de perturbación aditiva en el manipulador del brazo robótico, trazo rojo, siendo el resultado de seguimiento de la consigna en trazo azul para el control neuronal y entraso verde para el control PID clásico.

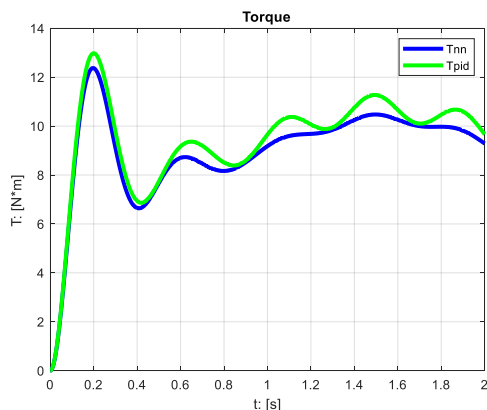
La **Figura 6** muestra resultados semejantes a los presentados en la **Figura 3**, dado que el seguimiento se cumple, gracias a la acción de control, la **Figura 6 a)** presenta un comportamiento semejante mostrado en la **Figura 3 a)**, no obstante en presencia de perturbación se muestra en la

**Tabla 3** un ligero incremento en la velocidad eficaz presente en el desempeño de seguimiento de la consigna por parte de la regulación con control neuronal, esto se traduce como en un esfuerzo de control para garantizar la bondad de seguimiento, y en consecuencia se muestra en la figura **Figura 6 b)** mayor actuación por parte del control neuronal.



**Figura 6.** a) Razón de cambio en el tiempo en presencia de perturbación aditiva en el manipulador del brazo robótico: trazo rojo, razón de cambio de la consigna, trazo azul, razón de cambio de la posición con el control neuronal, trazo verde, razón de cambio de la posición con el control PID clásico. b) Acción de control: trazo azul, salida del controlador neuronal, trazo verde, salida del controlador PID clásico.

La **Figura 7** muestra una acción antagónica, respecto a la perturbación, presentes en las mediciones de ambas acciones de control, tanto neuronal como PID clásico. No obstante, al igual que en la **Figura 4**, el trazo verde, que se corresponde con la acción de control PID, muestra un mayor esfuerzo para intentar garantizar la bondad en el seguimiento.



**Figura 7.** Torque medido a partir de la simulación, para lograr el posicionamiento del brazo robótico ante la consigna y en presencia de perturbación aditiva en el manipulador del brazo robótico. El trazo azul es la medición del torque ante una acción de control neuronal, y el trazo verde, ante una acción de control PID clásico.

La **Tabla 3** muestra en resumen los parámetros de valoración para poder comparar la bondad de desempeño de ambos controladores, tanto el neuronal como el PID clásico. Respecto a esta prueba en particular, se ratifica un mejor desempeño por parte del regulador neuronal respecto al controlador PID clásico.

<b>Tabla 3.</b> Resultados del desempeño Ante una misma consigna del Controlador del brazo robótico ante un control PID Clásico y un control Neuronal multicapa de flujo directo o feedforward en presencia de perturbación				
Desviación porcentual de Posición respecto a $Y_{sp}$		Velocidad Eficaz		
$\sigma_{%Y_{nn}}$	$\sigma_{%Y_{pid}}$	$\dot{Y}_{sp}$ [m/s]	$\dot{Y}_{nn}$ [m/s]	$\dot{Y}_{pid}$ [m/s]
4.04	16.75	1.776264	1.760661	1.831142
Torque eficaz		Potencia cuantitativa		
$T_{nn}$ [Nm]	$T_{pid}$ [Nm]	$P_{nn}$ [W]	$P_{pid}$ [W]	
9.235465	9.678435	16.260527	17.722598	



Los resultados presentados exhiben una superioridad del control neuronal respecto al control PID clásico, la razón de este resultado es explicado tomando en consideración la concepción de su forma matemática de operación. La red neuronal se encuentra operando como una función de caracterización de una ganancia no lineal, que mejor ajusta a la respuesta de seguimiento del brazo robótico. La razón de la implementación de la red neuronal y su factibilidad, obedece al tipo de sistema, es decir, el brazo robótico por la naturaleza de sus servomecanismos se puede clasificar como un sistema tipo 1 (Ogata, 2010, pp. 225). La razón de la clasificación como sistema tipo 1, hace que, por naturaleza, el controlador clásico óptimo sea proporcional, o proporcional derivativo. La presencia de un efecto integral puede ser justificada para corregir desgastes en los mecanismos, y este efecto siempre es mínimo. Finalmente, aunque el  $K_v$  del sistema (Ogata, 2010, pp 227), bajo el concepto de regulación clásica, se mejora incrementando la ganancia dinámica, este efecto se logra incorporando en serie al regulador una compensación de atraso de menos de 5 grados respecto al lugar de raíces del polo del sistema (Ogata, 2010, pp 321), y en consecuencia, aunque esta valoración está dada para una rampa, no es la mejor opción ante una perturbación sinusoidal, lo que puede ocasionar que el controlador se aleje de su propósito original, que es garantizar el seguimiento.

## Conclusión

En el presente artículo se analiza la bondad de las redes neuronales feedforward o redes neuronales múlticapa de flujo directo, para efectuar el control de posicionamiento de un brazo robótico tomando en consideración su efector o manipulador. Adicionalmente, se muestra como con ayuda del estudio de estabilidad de Lyapunov es posible especificar pesos a las diferentes capas de la red neuronal, que desde el punto de vista de estabilidad garantizan una operación con una mínima desviación, tanto como se desee o especifique, por lo tanto se plantea como una opción de partida para la implementación de un ajuste de pesos implementando el gradiente decreciente o cualquier técnica de optimización, o simplemente, para cuando se desee operar con inmediatez implementando esta técnica. Finalmente, haciendo uso de MATLAB/Simulink, se contrastaron resultados de la operación del brazo robótico ante el seguimiento de una misma consigna pero empleando diferentes controladores, el propuesto con redes neuronales, inicializados sus pesos con el método de Lyapunov, y el implementado con un PID clásico sintonizado con el auto-tuning de MATLAB. Los resultados, aunque satisfactorios, numéricamente hablando, mostraron un mejor

desempeño por parte del regulador implementado con redes neuronales. Es importante decir que el proceso de inicialización de pesos, de la red neuronal, empleando Lyapunov, debe considerarse un punto de partida que permitirá, empleando entrenamiento o conceptos adaptativos, mejorar aún más el desempeño del regulador con redes neuronales feedforward o redes neuronales múlticapa de flujo directo.

## Referencias

- Gautier, M. (1986). Identification of Robots Dynamics. IFAC Proceedings Volumes.  
[https://doi.org/10.1016/S1474-6670\(17\)59465-3](https://doi.org/10.1016/S1474-6670(17)59465-3)
- Hernandez-Sanchez, A., Mireles-Perez, C., Poznyak, A., Andrianova, O., Chertopolokhov, V., & Chairez, I. (2022). Trajectory Tracking of Robotic Arm Based on Power Regulation of Actuator Using Neural Averaged Subgradient Control.  
<https://doi.org/10.1016/J.IFACOL.2022.07.018>
- Ogata, K. (2010). Ingeniería de control moderna. Pearson Educación.
- Qin, B., Zhang, W., Dong, S., Wang, S., & Jiang, Y. (2022). Robotic Arm Trajectory Tracking Control Based on An RBF Neural Network Adaptive Control Algorithm.  
<https://doi.org/10.1109/ICIST55546.2022.9926773>
- Quynh, N. X., Nan, W. Y., & Yen, V. T. (2020). A novel Robust Adaptive Control Using RFWNNs and Backstepping for Industrial Robot Manipulators with Dead-Zone. Journal of Intelligent and Robotic Systems. <https://doi.org/10.1007/S10846-019-01089-9>
- Rebouças Filho, P. P., Suane, S. P., Praxedes, V. N., Hemanth, J., & de Albuquerque, V. H. C. (2019). Control of singularity trajectory tracking for robotic manipulator by genetic algorithms. Journal of Computational Science, 30, 55–64.  
<https://doi.org/10.1016/j.jocs.2018.11.006>
- Safaei, A., Koo, Y., & Mahyuddin, M. (2017). Adaptive model-free control for robotic manipulators. IRIS. <https://doi.org/10.1109/IRIS.2017.8250090>
- Van, M., Mavrovouniotis, M., & Ge, S. (2019). An Adaptive Backstepping Nonsingular Fast Terminal Sliding Mode Control for Robust Fault Tolerant Control of Robot Manipulators. IEEE Transactions on Systems, Man, and Cybernetics: Systems.  
<https://doi.org/10.1109/TSMC.2017.2782246>

- Vijay, M., & Jena, D. (2017). Backstepping terminal sliding mode control of robot manipulator using radial basis functional neural networks. *Computers & Electrical Engineering*.  
<https://doi.org/10.1016/J.COMPELECENG.2017.11.007>
- Xu, K., & Wang, Z. (2022). The design of a neural network-based adaptive control method for robotic arm trajectory tracking. <https://doi.org/10.1007/S00521-022-07646-Y>
- Zheng, H., Liu, Y., Li, C., & Wang, G. (2018). Trajectory Tracking Control of Robotic Manipulator via Variable Gain Iterative Learning Algorithm.  
<https://doi.org/10.23919/CHICC.2018.8482870>

© 2023 por los autores. Este artículo es de acceso abierto y distribuido según los términos y condiciones de la licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional (CC BY-NC-SA 4.0) (<https://creativecommons.org/licenses/by-nc-sa/4.0/>).