

SIMULACIÓN DEL CONTROL DE UN SISTEMA DE PRIMER ORDEN EMPLEANDO UN ALGORITMO NEURO INSPIRADO

*SIMULATION OF THE CONTROL OF A FIRST ORDER SYSTEM
EMPLOYING NEURO-INSPIRED ALGORITHM*

Valentín García Cervantes

Benemérita Universidad Autónoma de Puebla, México

valentin.garcia@alumno.buap.mx

Amparo Dora Palomino Merino

Benemérita Universidad Autónoma de Puebla, México

amparo.palomino@correo.buap.mx

Juan Escareno

Limoges University, XLIM Research Institute UMR CNRS 7252, Francia

juan.escareno-castro@unilim.fr

María Aurora Diozcora Vargas Treviño

Benemérita Universidad Autónoma de Puebla, México

aurora.vargas@correo.buap.mx

Recepción: 23/noviembre/2023

Aceptación: 26/diciembre/2023

Resumen

Actualmente el desarrollo de sistemas autónomos se ha convertido en un tópico bastante importante, su relevancia continúa creciendo en diversos ámbitos de la sociedad. Sin embargo, la autonomía de los nuevos sistemas se ve limitada por la capacidad de las baterías. En este artículo se propone un control neuronal que utiliza una red neuronal de tipo función base radial. Utilizando el software MATLAB – Simulink, el control neuro inspirado es implementado sobre un sistema dinámico de primer orden con presencia de perturbaciones. Se realiza un análisis comparativo del control propuesto con un controlador PI clásico para los casos de una perturbación constante, perturbación variante en el tiempo y perturbación dependiente del estado. En general, el algoritmo neuro inspirado muestra un comportamiento rápido, logra aproximar la posición deseada en tiempo reducido y mantiene un error mínimo. Esto permite reducir tiempos de simulación, lo que se traduce en menor costo computacional y energético.

Palabras Clave: Control neuronal, Función Base Radial, PI, Redes neuronales.

Abstract

Currently, the development of autonomous systems has become a highly significant topic, and its relevance continues to grow in various fields of society. However, the autonomy of these new systems is constrained by battery capacity. This article proposes a neuronal control approach that utilizes a radial basis function neural network. Using MATLAB – Simulink software, the neuro inspired control is implemented on a first-order dynamic system with the presence of disturbances. A comparative analysis is conducted between the proposed control and a classic PI controller in scenarios involving constant disturbances, time-variant disturbances, and state-dependent disturbances. In general, the neuro-inspired algorithm exhibits rapid response, achieves close approximation to the desired position in a short amount of time, and maintains minimal error. This leads to reduced simulation times, translating to lower computational and energy costs.

Keywords: *Neural networks, Neuronal control, PI, Radial Basis Function.*

1. Introducción

Las aplicaciones de sistemas autónomos se han multiplicado exponencialmente en diversos ámbitos sociales. Una categoría en pleno auge son los sistemas multiagente (SMA), los cuales permiten solucionar problemas complejos mediante la inteligencia distribuida y el trabajo colaborativo, sin embargo, su autonomía se ve limitada por la capacidad de sus baterías, por lo que actualmente se buscan nuevas formas de conseguir una eficiencia energética. En este sentido, los algoritmos de control han evolucionado a fin de realizar tareas más complejas mostrando una autonomía elevada, en donde los sistemas de control neuronal se distinguen por la plasticidad y desempeño frente a condiciones inesperadas.

Las redes neuronales artificiales (RNA) tratan de emular el comportamiento de la mente humana, caracterizada por el aprendizaje a través de la experiencia. Una forma de ver a una RNA es como un algoritmo que a su vez se compone de diferentes algoritmos, que realizan cálculos locales más pequeños a medida que los

datos se propagan a través de él [Galeone, 2019]. Existen diferentes modelos de RNA que permiten solucionar problemas difíciles de resolver para algoritmos de control convencionales. Las redes neuronales tienen la capacidad de aprender, por lo que, si en un sistema se conoce la entrada, el valor de salida deseado y el valor de salida actual, un controlador integrado por una red neuronal es capaz de modificar sus parámetros aprendiendo la dinámica de la planta hasta conseguir un sistema confiable.

A diferencia de los métodos de control tradicionales, los cuales requieren modelos matemáticos precisos y un conocimiento previo del sistema a controlar, un controlador basado en redes neuronales utiliza aproximaciones basadas en datos. Estas redes pueden aprender de manera autónoma a partir de la experiencia, lo que le permite mejorar continuamente su comportamiento y rendimiento en función de las condiciones cambiantes del entorno o del sistema bajo control, esto da lugar a un controlador más eficiente en presencia de perturbaciones.

El control basado en redes neuronales es un enfoque innovador y poderoso, en donde, una de sus principales ventajas es su capacidad para manejar sistemas complejos y no lineales. Estos sistemas a menudo presentan comportamientos difíciles de modelar mediante enfoques tradicionales, sin embargo, el aprendizaje y la adaptabilidad de las RNA permiten que los sistemas de control neuronal se ajusten automáticamente a cambios en el entorno o a perturbaciones en el sistema, lo que mejora la robustez y la estabilidad del control.

Las RNA se han aplicado ampliamente al control adaptativo de sistemas no lineales, su capacidad de aprendizaje junto con su capacidad de capturar relaciones no lineales les permite ajustar y optimizar el comportamiento del sistema de control para lograr una operación más eficiente desde el punto de vista energético [Shuai, 2018], [Revanesh, 2023]. En algunos casos, el entrenamiento inicial de la red neuronal puede requerir una cantidad significativa de energía y recursos computacionales. Sin embargo, una vez entrenada, la capacidad de las redes neuronales para adaptarse, aprender y optimizar su comportamiento puede contribuir significativamente a mejorar la eficiencia energética en diversas aplicaciones de control [Shengnan, 2023].

En este trabajo se realiza la implementación mediante simulación de un control neuronal, el cual utiliza una red neuronal del tipo función base radial (RBF) sobre un sistema dinámico de primer orden en presencia de diferentes tipos de perturbaciones; perturbación constante, perturbación variante en el tiempo y perturbación dependiente del estado.

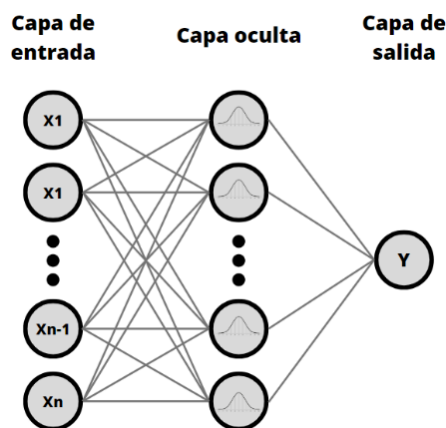
Redes Neuronales de tipo función base radial (RBF)

Las redes neuronales de función base radial (RBF por sus siglas en inglés) han demostrado tener una buena capacidad de aproximación no lineal de manera rápida [Zhang, 2023]. Las funciones de activación en una RBF son Gaussianas, estas permiten modelar relaciones no lineales complejas en los datos [Wang, 2022]. Las RBF son potentes herramientas computacionales que se han utilizado ampliamente en las áreas de reconocimiento de patrones, modelado e identificación de sistemas. Las redes neuronales RBF tienen varias ventajas que las hacen atractivas y efectivas en diversas aplicaciones, algunas de sus ventajas se enlistan a continuación [García, 2017]:

- **Eficiencia computacional:** Las RBF requieren menos neuronas en la capa oculta en comparación con otras redes neuronales, esto se debe a que las funciones radiales permiten capturar relaciones no lineales de manera más eficiente, lo que reduce la complejidad computacional y el tiempo de entrenamiento.
- **Capacidad para abordar problemas no lineales:** Las redes neuronales RBF son especialmente adecuadas para problemas de aproximación y clasificación que involucran relaciones no lineales complejas en los datos. Su arquitectura basada en funciones radiales les permite modelar y representar estas relaciones de manera efectiva.
- **Escalabilidad:** Las RBF pueden adaptarse y escalar para manejar diferentes tamaños de conjuntos de datos y aplicaciones. Pueden ser utilizadas tanto para pequeños como grandes problemas, lo que las hace adecuadas para una gran variedad de entornos.

En una red neuronal RBF, las neuronas en la capa oculta utilizan funciones radiales para medir la distancia existente entre los datos de entrada y ciertos puntos de referencia o centros. Estos centros representan puntos significativos y se utilizan para ponderar la influencia de cada neurona en la capa oculta en la predicción o clasificación final. En la figura 1 se presenta un esquema de las redes neuronales RBF, en general, siguen la siguiente arquitectura [Wang, 2022], [Doug, 2020]:

- Capa de entrada: En esta primera capa, los datos de entrada son proporcionados a la red.
- Capa oculta: Se encuentra interconectada entre todos sus nodos con la capa de entrada y es activada a través de la función radial (gaussiana).
- Capa de salida: Una vez que las neuronas de la capa oculta se activan, sus salidas se combinan ponderadamente para formar la salida de la red. Esta capa es activada a través de una función lineal continua.



Fuente: elaboración propia.

Figura 1 Esquema de una red neuronal de función base radial.

Estado del arte

Existen múltiples sistemas dinámicos cuyos modelos matemáticos se caracterizan por ser de primer orden. En algunos casos las herramientas clásicas no logran un sistema de control que sea estable, posea un buen desempeño y rechace adecuadamente perturbaciones [Regalón, 2012]. Una solución prometedora son los controladores por modos deslizantes (SMC, por sus siglas en inglés), es uno de los métodos de control robusto y no lineal, tiene varias ventajas

tales como la robustez contra perturbaciones externas, sin embargo, su principal desventaja es que agrega un término discontinuo en el controlador y la discontinuidad da como resultado una frecuencia de conmutación infinita, lo que se ve reflejado en el consumo energético de este tipo de controladores [Panhale, 2020], [Utkin, 2017], [Pérez, 2018]. Por lo que los SMC son inviables si lo que se busca es reducir el costo computacional y energético.

Las redes neuronales RBF fueron propuestas a finales de los años 1980, pero gracias a la creciente popularidad de la inteligencia artificial han tomado mayor relevancia en la actualidad. Existen diferentes publicaciones que incorporan las RBF, sin embargo, la mayoría de ellas solo son empleadas de manera complementaria. [Zhang, 2023] y [Tian, 2020] realizan una implementación híbrida con un control por modo deslizante adaptativo mediante redes neuronales RBF, lo cual mejora el tiempo de convergencia. Por otro lado, [Gómez, 2023] combina un controlador PI con una red neuronal base radial y la incorpora para compensar de manera “off-line” la salida del modelo. Finalmente, [Jing, 2014] y [Wang, 2018] proponen una variante de PID adaptativo, el cual hace uso de redes neuronales RBF para realizar la sintonía de las ganancias. Todas estas propuestas mantienen una implementación complementaria, por lo que se requiere un módulo de control y un módulo que contiene la red neuronal RBF. En este artículo se propone una red neuronal RBF, la cual sustituye por completo el módulo de control, simplificando el sistema y aprovechando los beneficios que poseen las RBF.

2. Métodos

En la presente sección se describe el desarrollo del control neuronal. Se realiza un análisis teórico, diseño del control, análisis de estabilidad por Lyapunov y una breve descripción sobre la implementación del sistema en el software de simulación Matlab – Simulink.

Planteamiento del problema

Considerando un sistema dinámico de primer orden, representado por la ecuación 1.

$$\dot{x}(t) = f(x(t)) + u(t) + \zeta(t) \quad (1)$$

Donde $f(\cdot) \in \mathbb{R}$ representa la dinámica no lineal del sistema, que en este trabajo se denota como una incertidumbre dependiente del estado, $\zeta(t) \in \mathbb{R}$ es una perturbación exógena variante en el tiempo, sabiendo que $\zeta(t) \neq 0$, la cual es desconocida, pero se puede delimitar a $\|\zeta_i(t)\| \leq \zeta_{max}$, y finalmente, $u(t) \in \mathbb{R}$ es la entrada de control.

Dado que las dinámicas no lineales $f(x)$ y $g(x)$ son desconocidas, por lo tanto, una combinación lineal finita de funciones de activación ponderadas para aproximar dichos términos no lineales dentro de un conjunto prescrito Ω .

Teorema de aproximación universal: Sea $f(x(t))$ una función suave en el conjunto compacto Ω . Por lo tanto, existe una arquitectura de red neuronal de tres capas basada en una única capa intermedia de η neuronas que presenta una función de activación adecuada $\phi(x(t)) \in \mathbb{R}^\rho$ y un vector de pesos ideales $\omega \in \mathbb{R}^\rho$, tal que puede ser descrita por la ecuación 2.

$$f(x) = \omega^T \phi(x(t)) + \varepsilon \quad (2)$$

Donde ε representa el error de aproximación de la RNA. En cuanto a la función de activación, se puede seleccionar entre una tangente hiperbólica, sigmoide o gaussiana (RBF). El peso ideal $\omega(\cdot)$, la función de activación $\phi(x)$ y el error de aproximación ε tienen su límite superior respectivamente en $\|\omega(\cdot)\| \leq \omega_{max}$, $\|\phi(\cdot)\| \leq \phi_{max}$ y $\|\varepsilon\| \leq \varepsilon_{max}$, con $\omega_{max}, \phi_{max}, \varepsilon_{max} > 0 \in \mathbb{R}$.

El error de aproximación desconocido ε_i puede ser reducido incrementando el número de neuronas. El teorema de aproximación universal permite aproximar uniformemente $f(\xi)$ usando la ecuación 3.

$$\hat{f}(x(t)) = \hat{\omega}^T \phi(x(t)) \quad (3)$$

Donde $\omega = [\omega_1, \dots, \omega_\rho]^T$ y $\phi = [\phi_1, \dots, \phi_\rho]^T$, donde ρ representa el número de neuronas. En este trabajo, consideramos funciones de base radial (RBF) como función de activación. En la ecuación 4 se presenta la función de activación utilizada.

$$\phi = \exp\left(-\frac{(x - \mu)^2}{\sigma^2}\right) \quad (4)$$

Donde μ y σ representan la media y la varianza de las entradas (x).

Diseño del controlador

La política de control propuesta pretende alcanzar el objetivo de regulación mientras compensa la dinámica desconocida dependiente del estado, $f(x)$ y las perturbaciones externas $\zeta(t)$. En otras palabras, el esquema de control de primer orden pretende minimizar el error de posición, como se representa en ecuación 5.

$$\lim_{t \rightarrow \infty} \|x(t) - x_d(t)\| \rightarrow 0 \quad (5)$$

En ese sentido, se reescribe el modelo en una dinámica de error de primer orden, ecuación 6. Donde $e = x(t) - x_d(t)$. En cuanto al diseño del control se considera el término global representado por la ecuación 7.

$$\dot{e}(t) = f(x(t)) + u(t) + \zeta(t) - \dot{x}_d(t) \quad (6)$$

$$\bar{u}(t) = f(x(t)) + u(t) + \zeta(t) \quad (7)$$

Lo que permite reescribir la dinámica de error anterior mediante ecuación 8.

$$\dot{e}(t) = \bar{u}(t) - \dot{x}_d(t) \quad (8)$$

El controlador propuesto para estabilizar la ecuación 6 se encuentra dado por la ecuación 9, y cuya regla de adaptación/aprendizaje se presenta en la ecuación 10. Donde $\alpha \in \mathbb{R}$ es la tasa de aprendizaje.

$$\bar{u}(t) = -\hat{\omega}^T \varphi(e(t)) + \varepsilon + \dot{x}_d(t) \quad (9)$$

$$\dot{\hat{\omega}}(t) = -\alpha e(t) \varphi(e(t)) \quad (10)$$

El peso estimado se calcula a partir de la minimización del error de estimación del peso, expresado por la ecuación 11, y su derivada temporal por ecuación 12.

$$\tilde{\omega} = \omega - \hat{\omega} \quad (11)$$

$$\dot{\tilde{\omega}} = -\dot{\hat{\omega}} \quad (12)$$

Análisis de estabilidad

Sea una función candidata de Lyapunov dada por la ecuación 13.

$$V = \frac{1}{2} e(t)^2 + \frac{1}{2\alpha} \tilde{\omega}^T(t) \tilde{\omega}(t) \quad (13)$$

Diferenciando ecuación 13 con respecto al tiempo y considerando ecuación 12, se obtiene ecuación 14.

$$\dot{V} = e(t)\dot{e}(t) - \frac{1}{\alpha}\tilde{\omega}^T(t)\dot{\hat{\omega}}(t) \quad (14)$$

Considerando las ecuaciones 8 y 9, se puede reescribir la ecuación 14 de la siguiente forma, presentada en la ecuación 15.

$$\dot{V} = e(t)(-\omega^T\varphi(e(t)) + \varepsilon) - \frac{1}{\alpha}\tilde{\omega}^T(t)\dot{\hat{\omega}}(t) \quad (15)$$

Usando la ecuación 11, se obtiene la ecuación 16.

$$\dot{V} = -e(t)\hat{\omega}^T\varphi(e(t)) - e(t)\tilde{\omega}^T\varphi(e(t)) + e(t)\varepsilon - \frac{1}{\alpha}\tilde{\omega}^T(t)\dot{\hat{\omega}}(t) \quad (16)$$

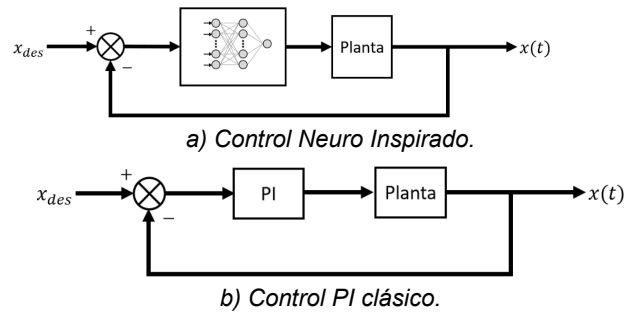
Donde se ha sustituido $\omega = \tilde{\omega} + \hat{\omega}$. Mediante ecuación 12 se obtiene ecuación 17. Considerando que se puede despreciar a ε ya que se reduce a medida que el número de neuronas aumenta y $\|\hat{\omega}^T\|, \|\varphi(e(t))\| > 0$, permiten concluir que el vector de pesos de error $\tilde{\omega}$ y el error $e(t)$ están y permanecen acotados dentro de una vecindad de origen para $t > 0$.

$$\dot{V} = -e(t)\|\hat{\omega}^T\|\|\varphi(e(t))\| + e(t)\varepsilon \quad (17)$$

Simulación del Control Neuronal Adaptativo

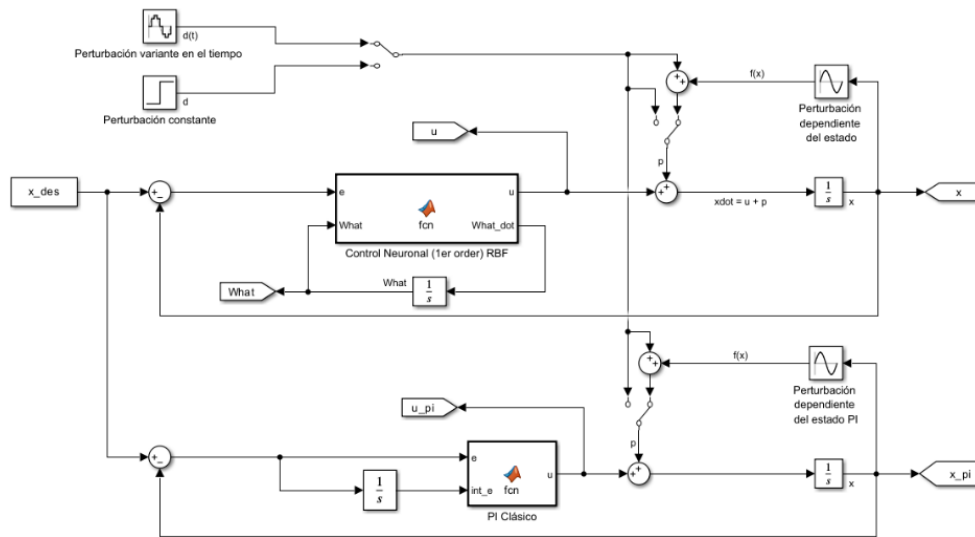
Partiendo de un sistema dinámico de primer orden, la red neuronal de base radial es diseñada para sustituir a un controlador clásico, por lo que utiliza como variable principal el error de posición, es decir la diferencia entre la posición deseada y la posición actual ($e = x_d - x(t)$). El control que se utiliza con fines comparativos es un PI clásico. En la figura 2 se presenta un diagrama a bloques del lazo cerrado de control tanto del controlador PI clásico como del control neuronal adaptativo.

Para la simulación, se utilizó el software MATLAB – Simulink para obtener la respuesta de los controladores en presencia de diferentes tipos de perturbaciones. Los casos analizados son; perturbación contante ($\dot{x}(t) = u(t) + \delta$), perturbación variante en el tiempo ($\dot{x}(t) = u(t) + \zeta(t)$) y perturbación dependiente del estado ($\dot{x}(t) = u(t) + \zeta(t) + f(x(t))$). A continuación, en la figura 3 se presenta el diagrama a bloques implementado en Simulink para la simulación de los 3 casos.



Fuente: elaboración propia.

Figura 2 Diagramas a bloques de los controladores simulados.



Fuente: elaboración propia.

Figura 3 Diagrama a bloques en MATLAB – Simulink.

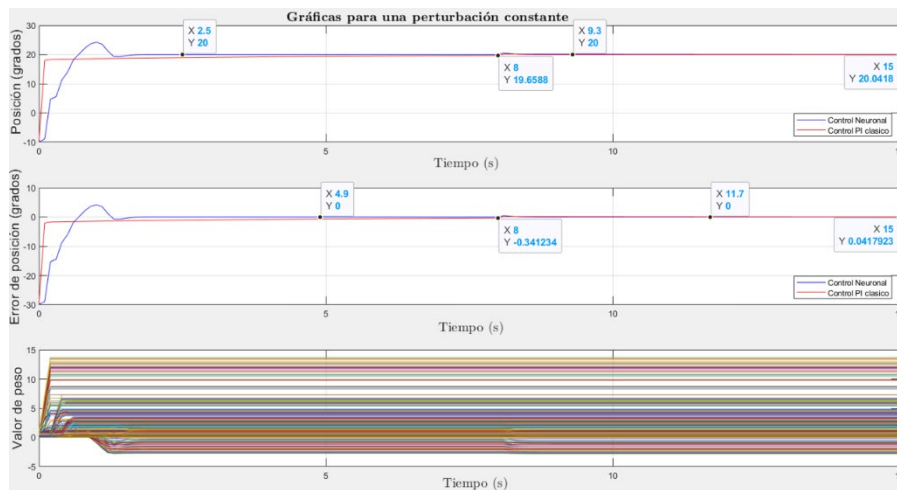
3. Resultados

A continuación, se presentan los resultados obtenidos correspondientes a cada uno de los casos mencionados anteriormente, considerando una posición deseada $x_d = 20$ y una condición inicial $x_0 = -10$.

Perturbación constante

En la actual subsección se presentan y comparan los resultados obtenidos del control neuronal y el PI clásico para un sistema de la forma $\dot{x}(t) = u(t) + \delta$. En la figura 4 se puede visualizar la respuesta para el sistema con una perturbación constante $\delta = 5$ a partir del tiempo $t = 8$ segundos. La primera gráfica representa la posición a través del tiempo, se puede observar que el control neuronal reacciona

mucho más rápido que el PI clásico, disminuyendo el error a prácticamente 0 a partir de los 2.5 segundos, mientras que el PI nunca converge completamente. La segunda gráfica corresponde al error de posición, en donde se puede observar que el control neuro inspirado logra un error exactamente igual a 0 en el segundo 11.7. Por otro lado, el PI clásico mantiene un error de centésimas. Finalmente, en la última gráfica se puede observar el comportamiento de los pesos de la red RBF.



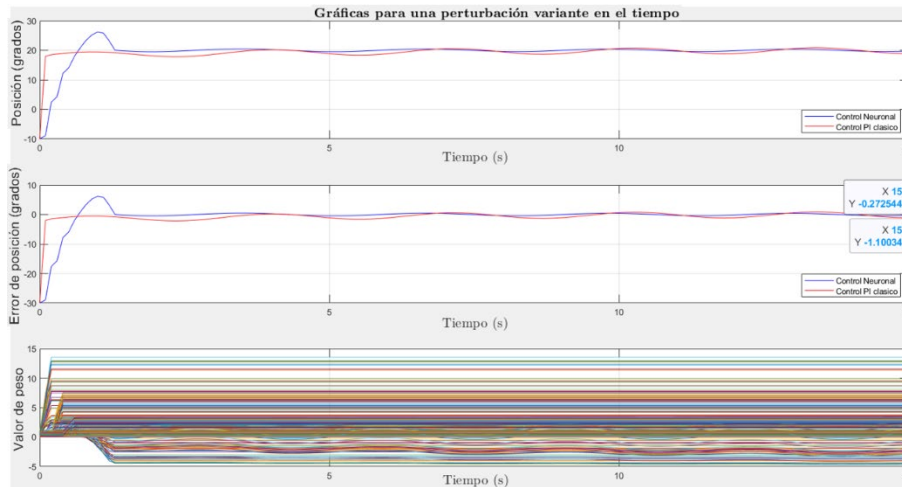
Fuente: elaboración propia.

Figura 4 Resultados obtenidos para una perturbación constante.

Perturbación variante en el tiempo

A continuación, se presentan y comparan los resultados obtenidos del control neuronal y el PI clásico para un sistema de la forma $\dot{x}(t) = u(t) + \zeta(t)$.

En la figura 5 se puede visualizar la respuesta para el sistema con una perturbación variante en el tiempo $\zeta(t) = 10 \sin(2t)$. La primera gráfica representa la posición a través del tiempo, se puede observar que el control neuronal nuevamente reacciona mucho más rápido que el PI clásico, disminuyendo el error rápidamente. La segunda gráfica corresponde al error de posición, en donde se puede observar con mayor precisión que el controlador neuronal logra un error cercano a 0, lo que se puede observar en la amplitud de las oscilaciones de ambos controles. Finalmente, en la última gráfica se puede observar el comportamiento de los pesos del control neuro inspirado.

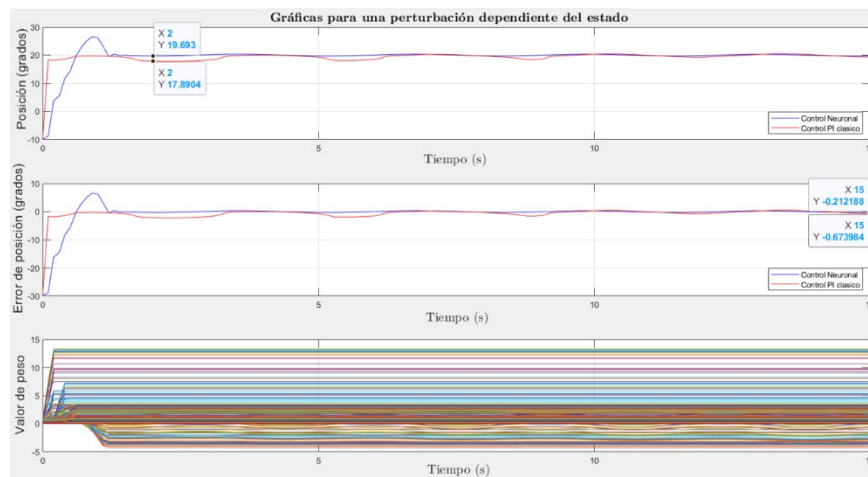


Fuente: elaboración propia.

Figura 5 Resultados obtenidos para una perturbación variante en el tiempo.

Perturbación dependiente del estado

Finalmente, se presentan y comparan los resultados obtenidos del control neuronal y el PI clásico para un sistema de la forma $\dot{x}(t) = u(t) + \zeta(t) + f(x(t))$. En la figura 6 se puede visualizar la respuesta para el sistema con una perturbación variante en el tiempo $\zeta(t) = 10 \sin(2t)$ y otra dependiente del estado $f(x(t)) = 4 \sin(3x(t))$.



Fuente: elaboración propia.

Figura 6 Resultados obtenidos para una perturbación dependiente del estado.

La primera gráfica representa la posición a través del tiempo, se puede observar que el control neuronal se estabiliza y disminuye el error más rápido que el PI

clásico. La segunda gráfica corresponde al error de posición, en donde se puede observar una menor amplitud de las oscilaciones del control neuronal. Al final de la simulación, de acuerdo con los “Data Tip”, el error de posición del control neuro inspirado es más de 3 veces menor que el control PI clásico. Finalmente, en la última gráfica se puede observar el comportamiento de los pesos del control neuronal.

4. Discusión

Utilizando el software de simulación Matlab – Simulink, se obtuvo el comportamiento de un sistema dinámico de primer orden al aplicar un control neuronal adaptativo y se comparó con un control PI clásico en presencia de diferentes tipos de incertidumbres, abarcando los casos; perturbación constante, perturbación variante en el tiempo y perturbación dependiente del estado.

El controlador PI clásico es ampliamente utilizado debido a su buen desempeño, ya que elimina el error en el estado estacionario y generalmente es un referente para comparar cualquier otra estrategia de control; sin embargo, se puede observar que el control neuronal adaptativo logra alcanzar errores aún menores.

El control neuronal adaptativo resultó ser más eficiente, pues le lleva un menor tiempo alcanzar la posición deseada x_d , además, el error es significativamente menor en el caso de la perturbación constante. Por otro lado, en el caso de perturbaciones dependiente del tiempo y dependiente del estado, se pueden observar oscilaciones menores, las cuales disminuyen su amplitud conforme transcurre el tiempo. Algunas de las deficiencias del control neuronal adaptativo es que en todos los casos se presenta un sobre impulso bastante grande, lo que podría hacerlo poco viable para determinadas aplicaciones, además, al realizar un entrenamiento en línea, los resultados no serán siempre los mismos, las variaciones son mínimas, sin embargo, se pueden obtener mejores o peores resultados. A pesar de ello, el controlador propuesto funciona de manera adecuada, con una mayor velocidad y precisión con respecto al PI clásico. La simulación se realizó sustituyendo directamente el bloque de control por la red neuronal RBF, lo que simplifica su implementación, siendo este otro de los principales beneficios de la propuesta frente a los sistemas actuales.

5. Conclusiones

Gracias a los resultados obtenidos se pueden concluir los siguientes puntos:

- El control neuronal es capaz de alcanzar el valor deseado de una manera más rápida, lo que permite un menor tiempo de simulación y por ende un menor costo computacional, algo que sin duda impacta en el consumo energético.
- El control propuesto realiza una adaptación (aprendizaje) en línea, por lo que no necesita de una etapa de entrenamiento previa, esto permite que aún con el cambio de las perturbaciones el sistema siga adaptándose y consiguiendo errores muy cercanos a cero.
- Este trabajo tiene la capacidad de ser escalable, es decir, posteriormente se puede realizar la simulación para sistemas de segundo orden y así poder visualizar el comportamiento de esta nueva familia de controladores neuronales adaptativos ante sistemas más complejos.

6. Bibliografía y Referencias

- [1] Doug, A. Neural networks: history and applications. Nova Science Publishers, Incorporated, 2020.
- [2] Galeone, P. Hands-on neural networks with TensorFlow 2.0: understand TensorFlow, from static graph to eager execution, and design neural networks. Packt Publishing Ltd, 2019.
- [3] Gómez, J. et al., Control PI neuro-adaptable en tiempo real de la humedad en el suelo usando un modelo híbrido. *Revista Iberoamericana de Automática e Informática industrial*. vol. 20, no 1, p. 93-103, 2023.
- [4] García, C. Redes neuronales de funciones base radiales. ULL. Facultad de Ciencias, 2017.
- [5] Jing, Y. et al., Inverted pendulum RBF neural network PID controller design. *International Symposium on Computer, Consumer and Control*. IEEE, 2014.
- [6] Panhale, A. et al., Robust motion control using novel first order sliding modes. *20th International Conference on Control, Automation and Systems (ICCAS)*. IEEE, 2020.

- [7] Pérez, C. et al., Control por modos deslizantes de un sistema de intercambio de calor: validación experimental. *Enfoque UTE* 9.4: 110-119, 2018.
- [8] Regalón, O. et al., Aplicación de algoritmos de control clásico, adaptable y robusto a sistemas dinámicos de parámetros variables. *Ingeniería Energética* 33.3: 184-195, 2012.
- [9] Revanesh, M. et al., Artificial neural networks-based improved Levenberg–Marquardt neural network for energy efficiency and anomaly detection in WSN. *Wireless Networks*: 1-16, 2023.
- [10] Shengnan, L. et al., Energy efficiency and coding of neural network. *Frontiers in Neuroscience* 16, 2023.
- [11] Shuai, L. et al., Neural Networks for Robot Arm Cooperation with a Full Distributed Control Topology. *Neural Networks for Cooperative Control of Multiple Robot Arms*: 49-74, 2018.
- [12] Tian, H. et al., Research on Adaptive Sliding Mode Robust Control Algorithm of Manipulator Based on RBF Neural Network. *Chinese Automation Congress (CAC)*. IEEE, 2020.
- [13] Utkin, V. et al., *Sliding mode control in electro-mechanical systems*. CRC press, 2017.
- [14] Wang, H. & Meng, F., Control Method of Nanomaterial Numerical Control Electronic Processing Based on RBF Neural Network. *Advances in Materials Science and Engineering*, 2022.
- [15] Wang, R. et al., Fuzzy neural network PID control based on RBF neural network for variable configuration spacecraft. *IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*. IEEE, 2018.
- [16] Zhang, X. et al., Research on fixed time sliding mode control strategy based on RBF neural network. *Journal of Measurement Science & Instrumentation* 14.2, 2023.