

APLICACIÓN DE UN ALGORITMO ESTADÍSTICO PARA EL ENTRENAMIENTO DE REDES NEURONALES PARA EL CONTROL DE UN SISTEMA CON RETARDO

Raúl Santiago Martínez, José Manuel Cruz Olguín, Virgilio López-Morales, Anilú Franco-Árcega, Manuel Alejandro Ojeda-Misses*

Universidad Autónoma del Estado de Hidalgo
Instituto de Ciencias Básicas e Ingeniería, Área Académica de Computación y Electrónica, Carretera Pachuca-Tulancingo Km. 4.5, C. P. 42090, Mineral de la Reforma, Hidalgo.
e-mail: sa381306@uaeh.edu.mx, cr233109@uaeh.edu.mx, virgilio@uaeh.edu.mx, afranco@uaeh.edu.mx, manuel_ojeda@uaeh.edu.mx

RESUMEN

Este trabajo propone un algoritmo de entrenamiento estadístico para redes neuronales aplicado al control de sistemas con retardo. A diferencia de técnicas clásicas, el algoritmo utiliza parámetros estadísticos como la media y desviación estándar para estimar la significancia del error, aplicando correcciones mediante la función sigmoide. Se evalúan dos arquitecturas: una red neuronal unicapa y otra multicapa. Los resultados muestran que ambas pueden compensar el retardo, aunque la red multicapa ofrece mayor precisión y estabilidad en seguimiento. Finalmente, se presenta un experimento con un servomecanismo con un retardo en el estado y tres redes neuronales para el control neuronal con realimentación de estado.

Palabras clave: retardo, red neuronal, estadístico, control, error.

ABSTRACT

This work proposes a statistical training algorithm for neural networks applied to the control of systems with delays. Unlike classical techniques, the algorithm uses statistical parameters such as the mean and standard deviation to estimate error significance, applying corrections using the sigmoid function. Two architectures are evaluated: a single-layer neural network and a multi-layer one. The results show that both can compensate for delay, although the multi-layer network offers greater tracking accuracy and stability. Finally, an experiment is presented with a state-delayed servomechanism and three neural networks for state-feedback neural control.

Keywords: delay, neural network, statistical, control, error.

1. INTRODUCCIÓN.

El control de sistemas con retardo representa un reto crítico en ingeniería, ya que los retardos en la señal de entrada o salida afectan la estabilidad y el desempeño. Estos retardos se encuentran en sistemas reales como redes de comunicación, procesos industriales, sistemas robóticos, entre otros, que complican el diseño de controladores tradicionales. Por ende, se requieren estrategias de control que ayuden a compensar los efectos de los retardos garantizando precisión y robustez en tiempo real. El uso de redes neuronales en el control de sistemas con retardos ha ganado atención por su capacidad de modelar dinámicas temporales complejas. Las redes neuronales con tiempo de retardo han sido empleadas para predecir señales con

retardo en sistemas no lineales [1]. Por su parte, las redes neuronales recurrentes (RNN) y sus versiones LSTM han demostrado eficacia en el control estocástico de sistemas con retardo, al permitir modelar dependencias de largo plazo y estabilizar el entrenamiento [2]. En escenarios donde el retardo es incierto o variable, los controladores adaptativos basados en redes neuronales han resultado eficaces. Por ejemplo, en sistemas de teleoperación con retardo constante se han implementado arquitecturas bilaterales con RNN que mejoran el seguimiento ante perturbaciones externas [3]. En el caso de exoesqueletos, se han desarrollado algoritmos que estiman el retardo en línea, mejorando la precisión y la seguridad del control [4].

Una línea de investigación destacada ha sido la integración de redes neuronales con modelo de control predictivo en la que se emplean modelos entrenados en línea junto con predictores tipo Smith. Estos esquemas han demostrado efectividad en el control de sistemas complejos como robots o drones, aunque requieren recursos computacionales considerables [5]. Otra estrategia relevante es el uso de controladores adaptativos óptimos que combinan aprendizaje numérico y optimización en línea, diseñados especialmente para sistemas lineales con retardo. Estas técnicas logran estabilidad y buen desempeño en el seguimiento, aunque su aplicación práctica puede estar limitada por la complejidad de la planta [6].

Los algoritmos de aprendizaje juegan un papel fundamental en el control neuronal. El algoritmo Delta Rule (también conocido como regla de Widrow-Hoff) ha sido ampliamente utilizado para entrenar redes de una sola capa, mejorando en la capacidad de almacenamiento de patrones en redes tipo Hopfield [8]. Por otro lado, el algoritmo de retropropagación de errores (Backpropagation) [9], [10], junto con variantes de segunda orden, sigue siendo el estándar para entrenar redes multicapa y recurrentes mediante la técnica de retropropagación temporal, y ha sido objeto de numerosas mejoras recientes en cuanto a velocidad y precisión [11]. Asimismo, el algoritmo de Levenberg-Marquardt (LM) [9], [10] ha sido adoptado en contextos donde se requiere una convergencia más rápida y precisa en redes neuronales feedforward y recurrentes. Este

panorama muestra un avance constante en el control de sistemas con retardo mediante redes neuronales, donde las arquitecturas recurrentes y predictivas lideran la investigación.

2. CONTROL DEL SISTEMA CON RETARDO MEDIANTE UNA RED NEURONAL

2.1. Sistema con retardo

Se considera un sistema dinámico con retardo en el estado:

$$\dot{x}(t) = -ax(t - \tau) - cx(t) + bu(t) \quad (1)$$

donde $x(t)$ es la salida del sistema, $u(t)$ es la entrada de control, $\tau > 0$ es el retardo del sistema y $a, b, c > 0$ son parámetros del sistema. Es importante mencionar que, el término $x(t - \tau)$ introduce un retardo que afecta la estabilidad y el desempeño del sistema. La señal de control $u(t)$ en el dominio del tiempo, no afecta la salida hasta después de un intervalo τ . Esto implica que el controlador debe prever el efecto retardado de sus acciones para lograr un seguimiento adecuado de la referencia $r(t)$. La propuesta de solución parte de la formulación de una señal de control $u(t)$ dada por una red neuronal unicapa y una multicapa con parámetros adaptativos con error definido como:

$$e(t) = r(t) - x(t) \quad (2)$$

donde $r(t)$ es la señal de referencia. El controlador $u(t)$ es definido como una función de errores actuales y pasados que tiene como objetivo compensar y anticipar el efecto del retardo dado como:

$$u(t) = N(e(t), e(t - \Delta t), \dots, e(t - k\Delta t), W(t), b(t)) \quad (3)$$

donde $e(t)$ es el error de seguimiento, Δt es el intervalo de muestreo, k es la cantidad de pasos históricos usados para compensar el retardo τ , $N(\bullet)$ es la red neuronal adaptativa con parámetros $W(t), b(t)$, $W(t)$ son los pesos sinápticos y $b(t)$ los sesgos de la red neuronal, respectivamente.

El controlador es capaz de adaptar los pesos sinápticos y los sesgos en línea para minimizar el error anticipado $e(t + \tau)$ compensando el retardo en el sistema:

$$\lim_{t \rightarrow \infty} E(e(t + \tau)) \approx 0 \quad (4)$$

Es decir, hará que la salida $x(t)$ siga a la referencia $r(t)$ a pesar de que la acción del controlador sea retrasada por el retrado τ . En sistemas con retardo temporal τ , la acción de control $u(t)$ llega al proceso con un desfase, lo que puede provocar inestabilidades o un seguimiento deficiente de la señal de referencia $r(t)$. Para compensar este efecto, el controlador implementado mediante una red neuronal adaptativa considera no solo el error instantáneo, sino también una ventana temporal de errores pasados $e(t - \Delta t), \dots, e(t - k\Delta t)$, donde Δt es el

intervalo de muestreo y k un número entero que aproxima el retardo τ . De esta manera, la red neuronal genera una señal de control $u(t)$ que anticipa el comportamiento del sistema frente al retardo, ajustando sus parámetros $W(t), b(t)$ para minimizar el error futuro $e(t + \tau)$. Este enfoque permite lograr una compensación dinámica y adaptativa del retardo, asegurando que la salida del sistema $x(t)$ pueda seguir con precisión la referencia $r(t)$ pese al desfase inherente al proceso.

2.2. Algoritmo de entrenamiento basado en datos estadísticos

Los métodos de entrenamiento en redes neuronales se basan en definir una función de error que mide el desempeño según los pesos sinápticos y el sesgo. En este trabajo se emplea una red neuronal feedforward de una sola capa y una multicapa. La señal de salida de la neurona será insertada en la entrada de control $u(t)$ y la señal de entrada es dada mediante $e(t) = r(t) - x(t)$. La señal de error $e(t)$ ayuda compensar la señal de control, cuyo fin es aplicar los ajustes correspondientes de los pesos sinápticos y los sesgos $W(t), b(t)$ de la neurona. Los ajustes son realizados minimizando el índice de desempeño:

$$J(t) = 1/2e^2(t) \quad (5)$$

Las redes neuronales son entrenadas con un algoritmo de aprendizaje basado en datos estadísticos y distribuciones de probabilidad. Este se fundamenta en el Teorema del Límite Central (TLC), que establece que la media muestral tiende a una distribución normal [17]. El método propuesto toma muestras de la salida de la red y calcula la media y la desviación estándar para ajustar el aprendizaje.

$$\bar{x}(t) = \frac{x(t_1) + x(t_2) + \dots + x(t_n)}{n} \quad (6)$$

Para las primeras n muestras aplicando la ley de los grandes números se considera que los promedios muestrales convergen y, por lo tanto, convergen en probabilidad al valor esperado $\bar{x}(t)$ cuando $n \rightarrow \infty$, además se obtiene la desviación estándar:

$$\sigma(t) = \sqrt{\frac{\sum_{i=1}^n (x_i(t) - \bar{x}(t))^2}{n}}$$

A continuación, se utiliza el TLC [17] para determinar el valor de $Z(t)$ de los datos obtenidos:

$$Z(t) = \frac{r(t) - \bar{x}(t)}{\sigma(t)} \quad (7)$$

donde $x(t)$ es la salida del sistema y $\bar{x}(t)$ es la media estimada, la diferencia $r(t) - \bar{x}(t)$ es definido como el error $e(t)$ y $\sigma(t)$ es la desviación estándar. En el contexto del TLC, es un valor

estandarizado que indica cuántas desviaciones estándar se encuentra una media muestral respecto a la media poblacional [18]. Esto permite usar la distribución normal estándar para calcular probabilidades, incluso con datos de distribuciones no normales [17]. El TLC establece que, con muestras grandes, la distribución de las medias muestrales tiende a ser normal. Esto permite estandarizar la media muestral, calcular $Z(t)$ respecto a la media poblacional en errores estándar y estimar probabilidades. Una vez estimada $Z(t)$, la probabilidad es calculada como:

$$P(t) = P(Z \leq z) = \Phi(z) \quad (8)$$

donde $\Phi(z)$ es la función de distribución acumulada (FDA) de la distribución normal estándar. En términos de cálculo, la probabilidad acumulada se expresa como:

$$\Phi(z) = \int_{-\infty}^z \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt \quad (9):$$

La función probabilística indica la probabilidad de que una variable tome un valor específico, mientras que la función de distribución representa la probabilidad acumulada hasta un cierto punto. Para el ajuste de pesos en la red neuronal, se utilizan funciones de activación como la sigmoide o la tangente hiperbólica:

$$W(t+1) = \Psi(P(t)) = W(t) + \frac{L}{1 + e^{-\alpha P(t)}} \quad (10)$$

$$W(t+1) = \Psi(P(t)) = W(t) + \frac{e^{P(t)} - e^{-P(t)}}{e^{P(t)} + e^{-P(t)}} \quad (11)$$

donde ambas están en función de la probabilidad $P(t)$. Mientras que, para el sesgo es definido como:

$$b(t+1) = b(t) + \frac{L}{1 + e^{-\alpha P(t)}} \quad (12)$$

$$b(t+1) = b(t) + \frac{e^{P(t)} - e^{-P(t)}}{e^{P(t)} + e^{-P(t)}} \quad (13)$$

Estas funciones permiten llevar a cabo el entrenamiento a la red neuronal mediante el algoritmo estadístico propuesto.

3. RESULTADOS EXPERIMENTALES

El sistema con retardo empleado para los resultados numéricos es definido como:

$$\dot{x}(t) = -1.2x(t - \tau) - 0.5x(t) + 2.2u(t) \quad (14)$$

El anterior sistema es puesto a prueba bajo el control basado en la red neuronal unicapa y multicapa empleando el algoritmo de entrenamiento estadístico propuesto en este trabajo. Con señal de referencia $r(t) = 0.5\sin(2t) + 0.1\cos(1.5t)$ cabe mencionarse que el controlador fue simulado para diversos valores de retardo $\tau = 0.1, 4.8, 5$ y $\Delta t = 0.01$ segundos.

3.1. Algoritmo con red neuronal unicapa

La red neuronal unicapa utilizada en esta implementación cuenta con una única neurona y función de activación tangente hiperbólica (ver Figura 1), cuyo objetivo es generar una señal de control $u(t)$ a partir del error de seguimiento $e(t)$ entre la referencia $r(t)$ y la salida de la planta $x(t)$ definido por la ecuación (2).

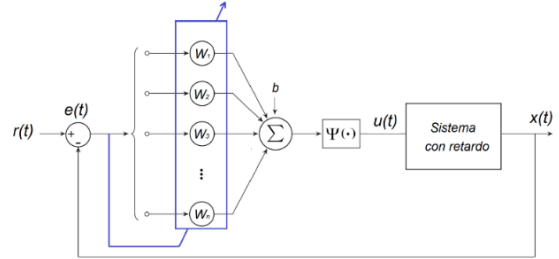


Figura 1. Red neuronal unicapa para el control del sistema con retardo.

La salida de la red neuronal actúa como la acción de control calculada como:

$$u(t) = \tanh(W(t)e(t) + b(t)) \quad (15)$$

donde $W(t) \in \mathbb{R}$, $b(t) \in \mathbb{R}$ son los pesos sinápticos y los sesgos, y $\tanh(\bullet)$ es la función de activación no lineal. A continuación, es empleada la normalización estadística del error para estimar $Z(t)$ del TLC. El Teorema Central del Límite (TCL) establece que el promedio de variables aleatorias independientes con media y varianza finitas tiende a una distribución normal sin importar la distribución original, entonces $Z(t) \rightarrow N(0,1)$ cuando $n \rightarrow \infty$. Esta convergencia ocurre porque al sumar muchas variables independientes, sus irregularidades se cancelan, generando una distribución simétrica. Luego se estima la probabilidad acumulada $P(t)$ dada por $\Phi(z)$ y la función sigmoide para ajustar el peso sináptico y el sesgo. En el experimento, durante 10 s con pasos de 0.01 s, se aplicó una señal de referencia senoidal y cosenoidal, mostrando resultados para los retardos $\tau = 0.1$ y $\tau = 5$ (ver Figura 2).

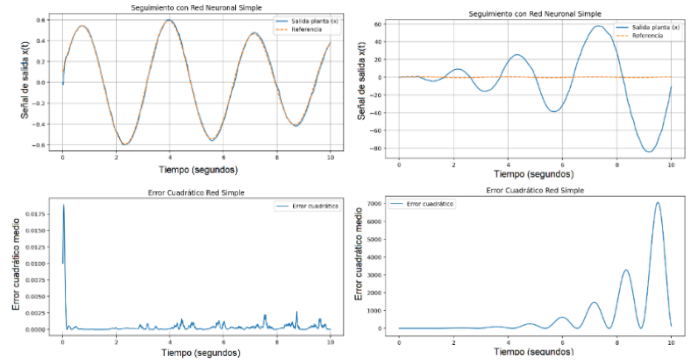


Figura 2. Señal de salida y referencia para el seguimiento de trayectoria del sistema con retardo de $\tau = 0.1$, $\tau = 5$ y sus errores cuadráticos medios.

La red neuronal simple logró generar una señal de control que permitió un seguimiento aceptable de la referencia, con ajuste progresivo del peso sináptico y del sesgo. El error cuadrático promedio disminuyó durante la simulación, aunque se observaron oscilaciones moderadas. Para retardos en el rango de 0.1 a 4.4 s, la salida fue estable, aunque presentó oscilaciones, y para retardos superiores a 5 s, la salida del sistema fue inestable.

3.2. Algoritmo con red neuronal multicapa

De manera similar, se aplicó el algoritmo a la red neuronal multicapa de la Figura 3 (quince capas ocultas de diez neuronas cada una para sistemas lineales). Esta red logró un seguimiento más preciso de la referencia, con ajustes dinámicos de pesos y sesgos en las capas internas.

Para cada capa $\ell = 1, \dots, C$, donde C es el número de capas (15 para este caso), se definen $W^{(\ell)} \in \mathbb{R}^{N \times N}$, $b^{(\ell)} \in \mathbb{R}^N$, $u^{(\ell)} \in \mathbb{R}^N$, donde N es el número de neuronas y ℓ la capa de salida.

La retroalimentación estadística permitió una mejor compensación del retardo inherente al sistema, donde la salida de capa es estimada como:

$$u^{(\ell)} = \tanh(W^{(\ell)} u^{(\ell-1)} + b^{(\ell)}) \quad (16)$$

donde los pesos y los sesgos son actualizados usando la regla local en cada capa:

$$W_{ij}^{(\ell)}(t+1) = W_{ij}^{(\ell)}(t) + \frac{2}{1 + e^{-P_i^{(\ell)}(t)}} \quad (17)$$

$$b_{ij}^{(\ell)}(t+1) = b_{ij}^{(\ell)}(t) + \frac{2}{1 + e^{-P_i^{(\ell)}(t)}} \quad (18)$$

Para la parte experimental, durante un intervalo de 10 segundos se observó una menor variabilidad del error cuadrático medio a lo largo del tiempo, así como una mayor estabilidad en los parámetros sinápticos una vez alcanzada la convergencia.

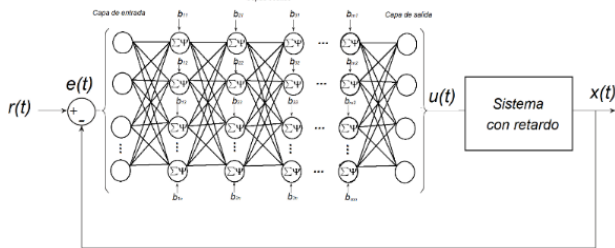


Figura 3. Red neuronal multicapa para el control del sistema con retardo.

Los resultados obtenidos presentan una comparativa entre el algoritmo estadístico propuesto, el algoritmo de Backpropagation y el algoritmo de Levenberg-Marquard para el control del sistema con retardo. Obteniendo los siguientes resultados para $\tau = 0.1, 4.7, 5$ (como se muestran en las Figuras 3, 4 y 5). Donde los resultados son similares, a diferencia de BP y LM, el algoritmo estadístico no depende del gradiente del error para actualizar los pesos. Mientras BP sigue la derivada del error y LM combina gradiente y aproximación de segundo orden, el

enfoque estadístico utiliza la normalización del error y su transformación probabilística como factor de corrección adaptativo. Esto permite compensar retardos y perturbaciones de manera flexible, aunque con una convergencia menos predecible en comparación con la precisión y velocidad de BP y LM.

1) Para el retardo $\tau = 0.1$

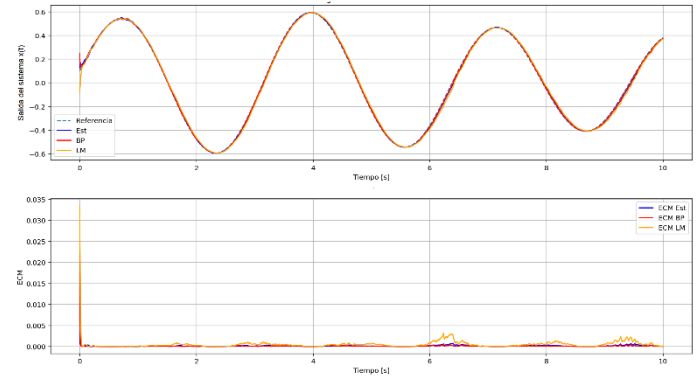


Figura 3. Señal de salida y referencia para el seguimiento de trayectoria del sistema con retardo de $\tau = 0.1$ y su error cuadrático medio.

3) Para el retardo $\tau = 4.8$

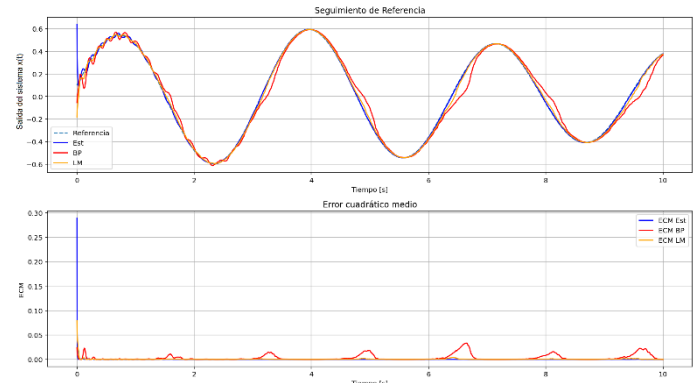


Figura 4. Señal de salida y referencia para el seguimiento de trayectoria del sistema con retardo de $\tau = 4.8$ y su error cuadrático medio.

4) Para el retardo $\tau = 5$

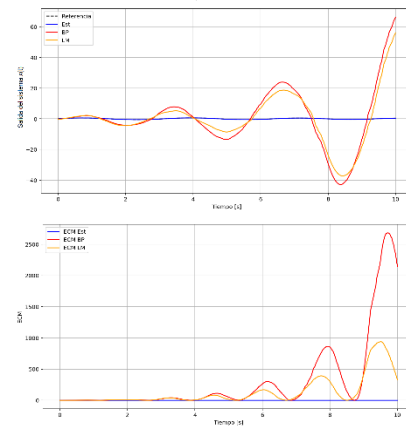


Figura 5. Señal de salida y referencia para el seguimiento de trayectoria del sistema con retardo de $\tau = 5$ y su error cuadrático medio

3.3. Aplicación de la red neuronal multicapa para el control de un sistema con retardo

En esta sección se implementa un resultado experimental mediante algoritmo estadístico usando la arquitectura multicapa para la red Perceptrón para el control en tiempo real mediante un servomecanismo. Este servomecanismo opera con programas como Matlab-Simulink mediante una computadora [19] cuya interfaz de hardware es usada para la transcripción de datos como se muestra en la Figura 6. El servomecanismo ha sido modelado en [19] como $\dot{x}(t) = -ax(t) + bu(t)$, donde se agrega en el estado un retardo:

$$\dot{x}(t) = -1.16x(t - \tau) - 1.16x(t) + 86.5u(t) \quad (19)$$

donde el retardo es $\tau = 0.1$ y los parámetros [19] son $a = 1.16$ y $b = 86.5$

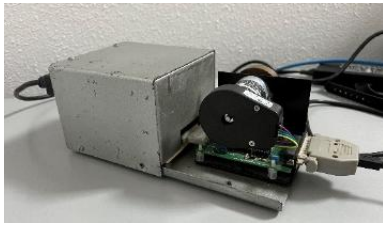


Figura 7. Prototipo del servomecanismo.

El diagrama de bloques [9], [10] de la Figura 9 emplea una red neuronal para la identificación del sistema, la señal de control $u(t)$ está conectada al servomecanismo como su entrada y genera una salida de identificación Y_{NN} que se compara con la salida del sistema para producir la señal de error de identificación utilizada para el entrenamiento dada por $E_1 = Y_s(t) - Y_{NN}(t)$.

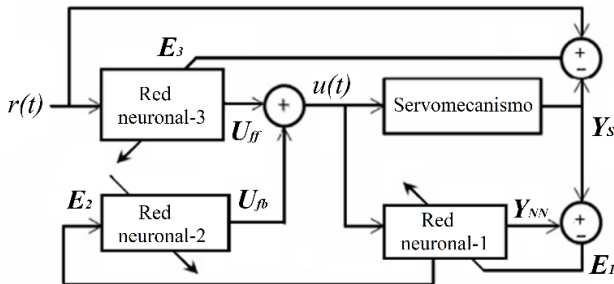


Figura 8. Diagrama de bloques del control neuronal con retroalimentación de estado.

La segunda red neuronal se alimenta con el estado interno mediante la identificación de la red neuronal-1 para generar una señal de control de retroalimentación de estado U_{fb} . Además, esta red neuronal-2 se entrena utilizando el error de control dado por E_1 . En consecuencia, la red neuronal-3 se entrena con la señal de error de control y converge al modelo inverso del sistema de lazo cerrado. Esta red neuronal-3 se alimenta con la señal de referencia dada como $r(t)$ para generar la señal de control de avance dada por U_{ff} . Finalmente, las tres redes

neuronales empleadas para validar el algoritmo de entrenamiento se presentan con las dimensiones de entrada, interna y de salida correspondientes. La señal de control es la suma de las señales de avance y retroalimentación; esto se describe como:

$$u(t) = U_{fb}(t) + U_{ff}(t) \quad (20)$$

El servomecanismo empleado para la identificación y el control del sistema en este trabajo es el servomecanismo presentado, ilustrado en la Figura 7. En este experimento, se considera el par motor y la posición angular del enlace como señales de entrada y salida, respectivamente. Donde la ecuación 27 se inserta en la ecuación 28 y al sistema se inserta una referencia dada por:

$$r(t) = 3 + 5 \sin(0.1t) + 2.5 \sin(0.2t) \quad (21)$$

Obtención la siguiente respuesta para el control mediante el servomecanismo en tiempo real. La Figura 9 muestra el seguimiento de la trayectoria mediante el diagrama de bloques del control neuronal de avance con retroalimentación de estado.

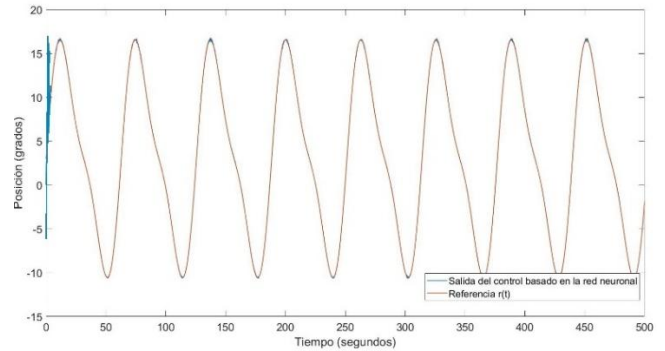


Figure 9. Seguimiento de trayectoria utilizando el diagrama de bloques del control neuronal con retroalimentación de estado.

Finalmente, en la Figura 10 se muestra el MSE total correspondiente a cada número de iteraciones para la identificación y el control mediante el servomecanismo, donde se observa que el MSE disminuye y se encuentra alrededor de 0.05.

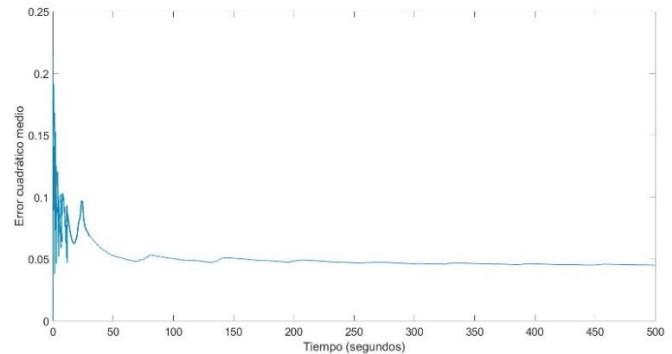


Figure 10. Error cuadrático medio del seguimiento de trayectoria utilizando el diagrama de bloques del control neuronal de avance con retroalimentación de estado.

El algoritmo estadístico presenta varias limitaciones. Su desempeño depende fuertemente de la arquitectura de la red neuronal, incluyendo el número de capas y neuronas, así como

de los pesos y tasas de aprendizaje iniciales, lo que puede provocar subajuste, sobreajuste o convergencia lenta. Está diseñado para plantas con retardos constantes y dinámicas relativamente estables; retardos variables o cambios abruptos en la planta pueden degradar su desempeño. Además, aunque la normalización estadística reduce la sensibilidad al ruido, perturbaciones significativas pueden afectar la estimación del error y la estabilidad del aprendizaje. Finalmente, el método es más efectivo con señales de referencia continuas y suaves, y su costo computacional aumenta con redes más grandes, lo que puede limitar su aplicabilidad en sistemas de tiempo real.

4. CONCLUSIONES

Se presentó un algoritmo de entrenamiento basado en estadística inferencial para redes neuronales, diseñado para el control de sistemas con retardo sin necesidad de modelado explícito. A diferencia de los métodos tradicionales, el algoritmo emplea medidas estadísticas como la media y la desviación estándar para evaluar la significancia del error, aplicando una corrección dinámica mediante funciones de activación no lineales, como la sigmoide. Esto permite un aprendizaje adaptativo del sistema, útil cuando los retardos son inciertos o variables.

Los resultados simulados evidencian que el algoritmo mejora la robustez del control frente a variaciones en el retardo y permite un ajuste dinámico del comportamiento del sistema. La comparación entre redes de arquitectura unicapa y multicapa muestra que, aunque ambas compensan los efectos del retardo, las multicapa ofrecen mayor generalización, precisión y estabilidad en la trayectoria de salida. Además, el enfoque es computacionalmente eficiente al evitar la estimación directa de parámetros del modelo o del retardo.

No obstante, el algoritmo presenta limitaciones: su desempeño depende de la arquitectura de la red, los pesos iniciales y las tasas de aprendizaje, pudiendo generar subajuste, sobreajuste o convergencia lenta. Está pensado para plantas con retardos constantes y dinámicas estables; retardos variables o cambios abruptos pueden afectar su desempeño. Aunque la normalización estadística reduce la sensibilidad al ruido, perturbaciones significativas pueden comprometer la estabilidad del aprendizaje. Su efectividad disminuye con señales discontinuas y su costo aumenta con redes más grandes, limitando su aplicabilidad en tiempo real.

En conjunto, el trabajo sugiere que los métodos estadísticos en redes neuronales representan una alternativa flexible y eficaz para el control de sistemas dinámicos con retardo, especialmente cuando el conocimiento del modelo es parcial o incierto.

5. REFERENCIAS

- [1] M. Sugiyama, H. Sawai y A. Waibel, "Review of TDNN architectures for speech recognition," en *Proc. IEEE International Symposium on Circuits and Systems*, 1991. <https://doi.org/10.1109/ISCAS.1991.176402>
- [2] J. Han y R. Hu, "Recurrent neural networks for stochastic control problems with delay," *arXiv*, 2021. <https://doi.org/10.48550/arXiv.2101.01385>
- [3] Q. Mu, F. Long y B. Li, "Adaptive neural network prescribed performance control for dual switching nonlinear time-delay system," *Scientific Reports*, vol. 13, p. 8132, 2023. <https://doi.org/10.1038/s41598-023-35307-0>
- [4] Z. Wang et al., "Model-free neural network control with time-delay estimation for exoskeletons," *Journal of Systems & Control Engineering*, 2017.
- [5] J. Q. Huang y F. L. Lewis, "Neural-network predictive control for nonlinear dynamic systems with time-delay," *IEEE Transactions on Neural Networks*, vol. 14, no. 2, pp. 377–389, 2003. <https://doi.org/10.1109/TNN.2003.809424>
- [6] X. A. Ji y G. Orosz, "Learning time delay systems with neural ordinary differential equations," *arXiv*, 2022. <https://doi.org/10.48550/arXiv.2206.14288>
- [7] B. Guś y J. Możaryn, "Comparison of artificial neural network adaptive control techniques for a nonlinear system with delay," *arXiv*, 2023. <https://doi.org/10.48550/arXiv.2304.13468>
- [8] D. L. Prados, "The capacity of artificial neural networks using the delta rule," Doctoral dissertation, Louisiana State University, 1989. https://doi.org/10.31390/gradschool_disstheses.4869
- [9] Baruch, I. S., Arellano-Quintana, V., & Pérez-Reynaud, E. Complex-valued neural network topology and learning applied for identification and control of nonlinear systems. *Neurocomputing*, 2017, 233, 104–115. <https://doi.org/10.1016/j.neucom.2016.09.109>
- [10] Baruch, I. S., & Gaspar, M. C. R. (2009). A Levenberg-Marquardt learning applied for recurrent neural identification and control of a wastewater treatment bioprocess. *International Journal of Intelligent Systems*, 24, 1094–1114. <https://doi.org/10.1002/int.20377>
- [11] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2014. <https://doi.org/10.1016/j.neunet.2014.09.003>
- [12] X. Fu, S. Li, M. Fairbank, D. C. Wunsch y E. Alonso, "Training recurrent neural networks with the Levenberg–Marquardt algorithm for optimal control of a grid connected converter," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 9, pp. 1900–1912, 2015. <https://doi.org/10.1109/TNNLS.2014.2361267>
- [13] X. A. Ji y G. Orosz, "Learning time delay systems with neural ordinary differential equations," *arXiv*, 2022. <https://doi.org/10.48550/arXiv.2206.14288>
- [14] R. Zhou, D. Wu, L. Fang, A. Xu y X. Lou, "A Levenberg–Marquardt backpropagation neural network for predicting forest growing stock based on least-squares parameters," *Forests*, vol. 9, no. 12, p. 757, 2018. <https://doi.org/10.3390/f9120757>
- [15] Ojeda, M. A., Baruch, I. S., & Soria, A. L. A real-time identification for hand-based movements using Recurrent Complex-Valued Neural Networks. 2019 *IEEE 4th Colombian Conference on Automatic Control (CCAC)*, 2019, 1–6. <https://doi.org/10.1109/CCAC.2019.8920864>
- [16] Ojeda-Misses, M. A., Martínez-Arango, H., López-Morales, V., Franco-Árcega, A., & Márquez-Grajales, M. (2024). Self-tuned closed-loop controller based on statistical data using a servomechanism. 2024 *XXVI Robotics Mexican Congress (COMRob)*, 27–32. <https://doi.org/10.1109/COMRob64055.2024.10777440>
- [17] Anderson, D. R., Sweeney, D. J., Williams, T. A., Camn, J. D., & Martin, K. (2008). *Estadística para administración y economía* (10ª ed.). Cengage Learning.
- [18] Ojeda-Misses, M., Martínez-Arango, H., Sampedro-Mendoza, A., Franco-Árcega, A., & López-Morales, V. (2025). Diseño de un controlador mediante datos estadísticos en lazo cerrado para un servomecanismo mediante una técnica de autosintonización. *RIIT Revista Internacional De Investigación E Innovación Tecnológica*, 12(72), 24–43. <https://revistas.uadec.mx/index.php/RIIT/article/view/104>
- [19] González-Vargas, A. M., Serna-Ramírez, J. M., Fory-Aguirre, C., et al. (2019). A low cost, free-software platform with hard real-time performance for control engineering education. *Computer Applications in Engineering Education*, 1–13. <https://doi.org/10.1002/cae.22084>