



# CONTROL NEURONAL POR MODELO INVERSO DE UN ESLABÓN FLEXIBLE ROTATORIO

## NEURAL INVERSE CONTROL OF A ROTARY FLEXIBLE LINK

Carlos Alberto Saldaña Enderica<sup>1,2,\*</sup> , José Ramón Llata<sup>1,2</sup> ,  
Carlos Torre-Ferrero<sup>1,2</sup> 

Recibido: 11-11-2025, Recibido tras revisión: 04-02-2026, Aceptado: 28-04-2026, Publicado: 01-07-2026

### Resumen


Este trabajo presenta un esquema de control por modelo inverso basado en datos para un sistema rotatorio de eslabón flexible, en el que la posición angular de la base se denota por  $\theta$ , y la deflexión relativa de la punta, por  $\alpha$ . La planta, correspondiente a un sistema rotatorio de eslabón flexible (*rotary flexible link*, RFL), se identifica a partir de datos experimentales y se modela en espacio de estados continuo de cuarto orden. A partir de este modelo, se diseña un controlador por modelo inverso implementado mediante una red neuronal artificial (RNA) de tipo perceptrón multicapa (MLP), entrenada con regresores formados por estados y entradas retardadas. La validación incluye métricas cuantitativas de error, análisis de respuesta transitoria y una certificación indirecta de estabilidad BIBO en tiempo discreto, obtenida mediante la identificación de un modelo lineal equivalente en lazo cerrado. Se comparan seis arquitecturas MLP en tres escenarios de referencia. La configuración seleccionada muestra el mejor compromiso entre el seguimiento de  $\theta$  y la mitigación de las oscilaciones en  $\alpha$ . Además, el trabajo integra en un flujo único la identificación basada en datos, la selección sistemática de la arquitectura y el análisis de estabilidad.


**Palabras clave:** control por modelo inverso; eslabón flexible rotatorio; estabilidad BIBO; identificación de sistemas; perceptrón multicapa; redes neuronales artificiales; supresión de vibraciones

### Abstract

This paper presents a data-driven inverse-model control scheme for a rotary flexible-link (RFL) system, with ( $\theta$ ) denoting the base angular position and ( $\alpha$ ) the relative tip deflection. The plant is identified from experimental data and represented as a continuous-time fourth-order state-space model. On this basis, an inverse-model controller is designed and implemented using an artificial neural network (ANN) of the multilayer perceptron (MLP) type, trained on regressors composed of delayed states and inputs. Validation relies on quantitative error metrics, transient-response analysis, and an indirect discrete-time BIBO (Bounded Input–Bounded Output) stability certification obtained by identifying an equivalent closed-loop linear model. Six MLP architectures are compared under three reference scenarios. The selected configuration achieves the best trade-off between ( $\theta$ ) tracking and ( $\alpha$ ) vibration attenuation, with bounded closed-loop signals and competitive settling times. The work integrates, into a single workflow, data-driven identification of the RFL, systematic MLP architecture selection, and discrete-time BIBO stability analysis, providing a reproducible framework for designing and objectively comparing inverse-model neural controllers in subactuated flexible systems.

**Keywords:** rotary flexible-link system; data-driven inverse-model control; artificial neural networks; multilayer perceptron; system identification; vibration suppression; BIBO stability.

<sup>1,\*</sup>Departamento de Tecnología Electrónica, Ingeniería de Sistemas y Automática, Universidad de Cantabria, España. 

<sup>2</sup>Facultad de Sistemas y Telecomunicaciones, Universidad Estatal Península de Santa Elena, Ecuador.   
Autor para correspondencia ✉: cse386@alumnos.unican.es.

Forma sugerida de citación: C. A. Saldaña Enderica, J. R. Llata y C. Torre-Ferrero, “Control neuronal por modelo inverso de un eslabón flexible rotatorio,” *Ingenius, Revista de Ciencia y Tecnología*, N.º 36, pp. 98-109, 2026. DOI: <https://doi.org/10.17163/ings.n36.2026.08>.

## 1. Introducción

Los manipuladores flexibles de un solo eslabón rotatorio constituyen un caso de estudio representativo tanto en aplicaciones reales – como la robótica ligera y las estructuras desplegables – como en bancos de pruebas académicos de control avanzado. Su naturaleza subactuada y la presencia de flexibilidad estructural generan oscilaciones y acoplamientos dinámicos que dificultan la síntesis de controladores robustos y precisos, especialmente cuando se dispone de información incompleta o incierta sobre el modelo físico del sistema [1].

En este contexto, el enfoque de control basado en datos (*data-driven*) constituye una alternativa atractiva frente a los métodos clásicos, cuyo diseño descansa en modelos paramétricos detallados. En lugar de depender de identificaciones analíticas de alta fidelidad, los enfoques *data-driven* permiten diseñar controladores directamente a partir de datos experimentales o simulados, lo que resulta especialmente útil en sistemas complejos y no lineales [2]. Dentro de esta familia, los esquemas de control por modelo inverso basados en redes neuronales artificiales (RNA) de tipo perceptrón multicapa (*multilayer perceptron*, *MLP*) han demostrado capacidad para aproximar la dinámica inversa de manipuladores y otros sistemas no lineales con un esfuerzo de modelado explícito considerablemente menor [3–5]. Estas redes permiten capturar relaciones entrada–salida altamente no lineales a partir de datos y han mostrado buen desempeño en tareas de seguimiento de trayectorias y compensación de efectos dinámicos no modelados [6].

La literatura reciente sobre control *data-driven* e inversión de modelos puede organizarse en tres líneas principales. La primera línea aborda el diseño de controladores óptimos sin utilizar un modelo físico explícito, mediante estructura FIR (*Finite Impulse Response*), regularización tipo LASSO y formulaciones convexas para control inverso. En general, estos métodos contribuyen a reducir la complejidad estructural del controlador sin comprometer el rendimiento y, además, proporcionan garantías formales de optimalidad basadas en datos [7–9]. En este contexto, la selección de hiperparámetros, la cantidad efectiva de coeficientes FIR y la calidad de los datos experimentales son factores determinantes para lograr una adecuada generalización.

Un segundo eje se centra en la inversión de modelos mediante aprendizaje automático, especialmente con el uso de redes MLP para aproximar la dinámica directa o inversa de manipuladores. Se ha observado que las MLP pueden superar a métodos clásicos, como las máquinas de vectores de soporte (support vector machines, SVM), en tareas de predicción dinámica, siempre que la sintonización de hiperparámetros sea adecuada. No obstante, su desempeño depende de la representatividad de los datos de entrenamiento [10].

Asimismo, existen propuestas que incorporan optimización y regularización mediante metaheurísticas, con el fin de mejorar la eficiencia computacional y reducir el sobreajuste en redes empleadas en control [11].

El tercer eje está relacionado con la síntesis robusta de controladores de orden fijo y la integración de datos ruidosos y no ideales. Los enfoques basados en *data-driven* permiten tratar el problema de ajuste de modelos (*model matching*) y la robustez directamente en el espacio de datos, pero introducen nuevos compromisos entre la robustez frente a la incertidumbre, la complejidad computacional y la facilidad de implementación [9]. En el caso de sistemas rotatorios con eslabones flexibles, los estudios experimentales recientes combinan identificación en espacio de estados con técnicas de aprendizaje por refuerzo guiado [12], empleando métricas de seguimiento, reducción de vibraciones y robustez ante perturbaciones. Sin embargo, la comparabilidad entre trabajos se ve limitada por la ausencia de protocolos y escenarios de prueba estandarizados.

A pesar de estos avances, se identifican varios vacíos específicos en el estado del arte:

- La falta de estudios que integren de forma sistemática la identificación basada en datos de un RFL, la construcción de un modelo inverso neuronal y su validación cuantitativa bajo múltiples escenarios de referencia.
- La escasa articulación entre la inversión neuronal y los criterios formales de selección del orden del modelo, apoyados en métricas como FPE, MDL o ajuste porcentual [9], que permitan justificar la elección de modelos de dimensión moderada frente a alternativas de menor orden.
- La ausencia de análisis detallados sobre la estabilidad BIBO del lazo cerrado cuando se emplean controladores neuronales, más allá de observaciones cualitativas basadas en simulaciones.

En este contexto, el objetivo de este trabajo es analizar y demostrar la viabilidad de un esquema de control por modelo inverso basado en datos para un sistema rotatorio de eslabón flexible, empleando una RNA de tipo MLP como aproximador de la dinámica inversa. Específicamente, se propone un flujo de trabajo que integra: i) identificación en espacio de estados continuo de la planta a partir de datos experimentales siguiendo el marco de [12]; ii) construcción de un conjunto de regresores con estados y entradas retardadas para entrenar el modelo inverso; iii) síntesis y selección de la arquitectura neuronal en función de métricas objetivas; y iv) evaluación del desempeño del controlador inverso en términos de error de seguimiento, supresión de vibraciones y estabilidad BIBO discreta.

El enfoque no pretende competir en términos de rendimiento absoluto con técnicas de alto nivel, sino evidenciar el valor de una metodología clara y reproducible para diseñar, entrenar y validar un controlador neuronal inverso en sistemas rotatorios con eslabón flexible. La metodología propuesta es extrapolable a otros sistemas dinámicos complejos caracterizados por modelado analítico de alta dificultad, acoplamientos estructurales y sensibilidad a perturbaciones. Además, se aplica a un estudio de caso representativo que facilita su replicación en laboratorios de control.

## 2. Materiales y métodos

Esta sección describe el modelo en espacio de estados del RFL, el proceso de obtención de datos experimentales y el diseño del controlador neuronal por modelo inverso, implementado mediante una arquitectura RNA-MLP y utilizado en las simulaciones.

### 2.1. Descripción general del control por modelo inverso

El control por modelo inverso se basa en entrenar una RNA para que aproxime la dinámica inversa de la planta a partir de pares

$$(\varphi(k), u(k)), \quad (1)$$

donde  $\varphi(k)$  es un vector de regresores que contiene salidas y entradas retardadas, y  $u(k)$  es la señal de control asociada [13, 14]. Durante la fase de entrenamiento, la planta se excita con señales de tipo escalón, de distintas magnitudes y con tiempos de aplicación aleatorios para cubrir un rango de operación amplio y obtener un conjunto de datos representativos [15]. A continuación, los registros se separan en conjuntos de entrenamiento, validación y prueba, con el fin de ajustar los parámetros de la RNA y, al mismo tiempo, mitigar el sobreentrenamiento; de este modo, se preserva una adecuada capacidad de generalización [4], [16].

Como algoritmo de entrenamiento, se emplea Levenberg-Marquardt, conocido por su rapidez de convergencia y su capacidad para alcanzar errores pequeños en redes de tamaño moderado [16, 17]. La robustez se refuerza mediante la normalización de datos, la partición en conjuntos de entrenamiento, validación y prueba, y la parada temprana basada en el error de validación.

### 2.2. Flujo general de la metodología

El desarrollo del controlador neuronal se articula en tres etapas principales:

1. Identificación en espacio de estados continuo del RFL a partir de datos experimentales, con el

fin de obtener un modelo  $(A, B, C, D)$  de cuarto orden.

2. Generación de datos sintéticos a partir de dicho modelo para entrenar el modelo inverso neuronal.
3. Diseño, evaluación y selección de la arquitectura MLP más adecuada, en función de métricas de seguimiento y de la amplitud de oscilación del eslabón.

La plataforma utilizada corresponde al entorno Quanser *QLabs Virtual Rotary Flexible Link*, instrumentado en MATLAB/Simulink® R2022b y QUARC 2.15, de acuerdo con el flujo de trabajo descrito en [12].

### 2.3. Identificación del sistema rotatorio de eslabón flexible

Se trabaja con un RFL de un solo eslabón flexible accionado en la base. Se emplean las siguientes variables:

$$\begin{aligned} \theta(t) &: \text{posición angular de la base} \\ \alpha(t) &: \text{deflexión relativa en la punta} \\ u(t) &\in [-10, 10] \text{ V, que corresponde a la señal} \end{aligned} \quad (2)$$

de entrada.

Para la identificación del modelo continuo, se emplearon datos experimentales tomados en el entorno virtual de Quanser. La posición angular  $\theta$  y la deflexión relativa  $\alpha$  se registraron con un tiempo de muestreo

$$T_s = 2 \text{ ms}, \quad (3)$$

suficiente para capturar los modos de flexión de baja frecuencia sin *aliasing* y en concordancia con estudios previos sobre el mismo prototipo [12], [18]. Las pruebas se realizaron manteniendo  $\alpha(t)$  en el intervalo  $[-5^\circ, 5^\circ]$  a fin de operar en un régimen cuasilíneo [1].

Para asegurar la persistencia de la excitación y cubrir el rango operativo de interés, se aplicó al sistema una señal de tensión cuadrada pseudoaleatoria en  $u(t)$ , con amplitud y frecuencia variables dentro del intervalo seguro  $[-10, 10]$  V. Durante todo el proceso, se respetaron las restricciones de operación definidas para el modelo virtual [19]. Luego, los datos se separaron en:

- un conjunto de identificación, utilizado para estimar los parámetros del modelo; y
- un conjunto de validación, reservado exclusivamente para evaluar la capacidad predictiva del modelo.

### 2.3.1. Procedimiento de identificación y selección de orden

Se adopta un modelo continuo en espacio de estados de cuarto orden, con una entrada y dos salidas:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \quad (4)$$

donde

$$x(t) = [\theta(t) \quad \alpha(t) \quad \dot{\theta}(t) \quad \dot{\alpha}(t)]^T, y(t) = [\theta(t) \quad \alpha(t)]^T. \quad (5)$$

Se emplearon métodos de error de predicción (*Prediction Error Methods*, PEM) sobre los datos descritos anteriormente para la identificación. Como resultado, se obtuvo un modelo de cuarto orden, con matriz  $D = 0$ , debido a que la inercia del conjunto motor-eslabón y la flexibilidad estructural impiden cambios instantáneos en las salidas ante variaciones de  $u(t)$  [12], [19].

La selección de un modelo de cuarto orden se apoya en los análisis comparativos de [20]. En particular, los modelos de orden menor no logran representar adecuadamente la dinámica de flexión. Por otra parte, al aumentar el orden del modelo, no se observa una mejora importante en FPE, MSE ni en el ajuste porcentual; sin embargo, se incrementan el costo computacional y el riesgo de sobreajuste.

### 2.3.2. Modelo continuo identificado

Aplicando el procedimiento descrito anteriormente se obtuvo el siguiente modelo continuo en espacio de estados de cuarto orden, con una entrada y dos salidas.

Las matrices identificadas son:

$$A = \begin{bmatrix} 0.2629 & -0.6923 & 2.0550 & 1.0130 \\ -11.5200 & 23.9900 & -54.7900 & -30.0000 \\ -4.2910 & -10.4200 & -63.9600 & -20.6300 \\ -3.4710 & 77.0200 & 24.4800 & -10.1900 \end{bmatrix} \quad (6)$$

$$B = \begin{bmatrix} -0.05998 \\ 1.6080 \\ 7.1510 \\ -6.6660 \end{bmatrix} \quad (7)$$

$$C = \begin{bmatrix} 29.6200 & 0.7060 & 0.1893 & 0.1340 \\ -0.1510 & 0.8635 & -0.7005 & -0.5094 \end{bmatrix} \quad (8)$$

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (9)$$

Este modelo reproduce la dinámica acoplada entre  $\theta$  y  $\alpha$  en el rango operativo considerado. Estudios previos realizados con el mismo prototipo RFL han mostrado que modelos continuos de este orden capturan adecuadamente el comportamiento del sistema en tareas de seguimiento y supresión de vibraciones [12], [20].

### 2.4. Diseño del controlador neuronal por modelo inverso

A partir del modelo continuo identificado (A,B,C,D), se construyó una simulación en espacio de estados del RFL mediante una discretización de Euler con paso  $dt=0.002$  s. La entrada utilizada fue una señal escalonada aleatoria, con nuevos niveles cada 2 s en el rango

$$u(k) \in [-1.5, 1.5] \text{ V}, \quad (10)$$

aplicada durante 1000 s. De esta forma, se obtuvieron datos persistentes de

$$\theta, \alpha, \dot{\theta}, \dot{\alpha}, y, u, \quad (11)$$

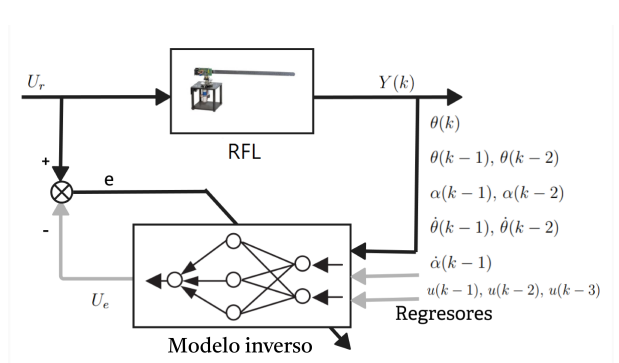
que constituyen la base para la generación de regresores y la estimación del modelo inverso neuronal.

La Tabla 1 muestra la configuración empleada para el entrenamiento del modelo inverso. La selección se realizó con el objetivo de promover una convergencia estable, evitar el sobreajuste y mantener un equilibrio adecuado entre precisión y capacidad de generalización.

**Tabla 1.** Parámetros de entrenamiento de la red neuronal

Parámetro	Valor
Algoritmo de optimización	Levenberg-Marquardt ( <code>trainlm</code> )
Número máximo de épocas	600
Criterio de error (objetivo)	$10^{-4}$
Proporción de entrenamiento	85 %
Proporción de validación	10 %
Normalización de datos	<code>mapminmax</code> en entradas y salidas

El diagrama de la Figura 1 resume el proceso de identificación inversa. Este esquema permite visualizar claramente cómo se organiza el flujo de información durante el entrenamiento de la RNA, destacando la relación entre los regresores empleados y la señal de control que la red debe aprender a predecir.



**Figura 1.** Esquema del proceso de identificación de la dinámica inversa del RFL.

### 2.4.1. Arquitecturas evaluadas

Se evalúan diferentes configuraciones de MLP, las cuales difieren en el número de neuronas y de capas ocultas, mientras se mantiene fija la función de activación lineal en la salida. Las configuraciones se detallan en la Tabla 2.

**Tabla 2.** Configuraciones evaluadas para la red neuronal MLP

Conf.	Capas ocultas	Funciones de activación
C1	[15]	<i>logsig-purelin</i>
C2	[20]	<i>logsig-purelin</i>
C3	[20, 10]	<i>logsig-logsig-purelin</i>
C4	[25, 15]	<i>logsig-logsig-purelin</i>
C5	[12, 2]	<i>logsig-logsig-purelin</i>
C6	[30, 20]	<i>logsig-logsig-purelin</i>

### 2.4.2. Modelo inverso y entrenamiento

Para cada configuración de la Tabla 2, se entrenó una red neuronal MLP *feedforward*, utilizando el algoritmo de Levenberg–Marquardt, para aproximar la señal de control  $u(k)$  a partir de un vector de regresores que contiene valores actuales y retardados de las variables medidas. En notación tipo NARX/NARMAX, dicho vector se define como:

$$\varphi(k) = \begin{bmatrix} (I, q^{-1}, q^{-2}) \theta(k), (q^{-1}, q^{-2}) \alpha(k), \\ (q^{-1}, q^{-2}) \dot{\theta}(k), q^{-1} \dot{\alpha}(k), \\ (q^{-1}, q^{-2}, q^{-3}) u(k) \end{bmatrix}^T. \quad (12)$$

donde  $q^{-1}$  denota el operador de retardo en una muestra ( $q^{-1}x(k) = x(k-1)$ ,  $q^{-2}x(k) = x(k-2)$ , etc.), e  $I$  es el operador identidad.

De forma explícita, el vector de regresores queda como

$$\varphi(k) = \begin{bmatrix} \theta(k), \theta(k-1), \theta(k-2), \alpha(k-1), \alpha(k-2), \\ \dot{\theta}(k-1), \dot{\theta}(k-2), \dot{\alpha}(k-1), \\ u(k-1), u(k-2), u(k-3) \end{bmatrix}^T. \quad (13)$$

Es decir, la RNA recibe como entrada una ventana temporal formada por las muestras actuales y pasadas de la posición de la base  $\theta$ , la deflexión de la punta  $\alpha$ , sus velocidades correspondientes y la señal de control  $u$  en los tres últimos instantes de muestreo.

En las pruebas preliminares, se consideraron también vectores de regresores con más retardos, por ejemplo, incluyendo  $\theta(k-3)$ ,  $\alpha(k-3)$  o  $u(k-4)$ .

Sin embargo, al aumentar el número de retardos, la dimensión de  $\varphi(k)$  crece sin aportar información dinámica relevante adicional, lo que empeora la condición del problema de entrenamiento. En la práctica, la red tendía a no converger dado que el error de validación permanecía alto, o se estabilizaba en mínimos

locales con desempeño insuficiente en el seguimiento de referencia. Por este motivo, se adoptó el vector de regresores de la ecuación (13), que proporciona un compromiso adecuado entre memoria temporal y capacidad de aprendizaje.

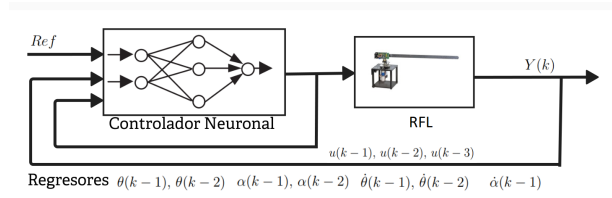
La salida de la red implementa entonces una aproximación del modelo inverso de la planta:

$$u(k) = \mathcal{N}_\psi(\varphi(k)), \quad (14)$$

donde  $\mathcal{N}_\psi(\cdot)$  representa la MLP parametrizada por el vector de pesos  $\psi$ .

### 2.4.3. Implementación en lazo cerrado

Para implementar el controlador neuronal basado en el modelo inverso entrenado, la RNA-MLP genera la señal de control  $u(k)$  necesaria para que el sistema RFL siga la referencia  $\theta_{\text{ref}}(k)$ . El esquema conceptual del controlador en lazo cerrado se muestra en la Figura 2.



**Figura 2.** Diagrama del controlador neuronal por modelo inverso del RFL.

## 3. Resultados y discusión

### 3.1. Resultados de identificación del modelo RFL

Para el conjunto de validación, el modelo continuo de cuarto orden proporciona:

- Un ajuste en el dominio temporal para  $\theta$  en el rango 99.85–96.95 % – predicción a un paso y simulación libre, respectivamente –, con índices  $FPE = 2.89 \times 10^{-13}$  y  $MSE = 1.326 \times 10^{-6}$ ;
- Un ajuste en simulación libre del 96.25 % para  $\theta$  y del 81.41 % para  $\alpha$ , lo que indica que la dinámica acoplada entre la rotación de la base y la flexión del eslabón queda razonablemente capturada en el rango operativo considerado.

En trabajos preliminares con la misma plataforma, se ha comprobado que modelos de orden inferior ( $n = 2, 3$ ) degradan notablemente la capacidad de predicción sobre  $\alpha$ , mientras que los órdenes superiores no aportan mejoras significativas en FPE o MSE, pero sí incrementan la complejidad y la sensibilidad al ruido [1], [20]. Por tanto, el orden  $n = 4$  se adopta como

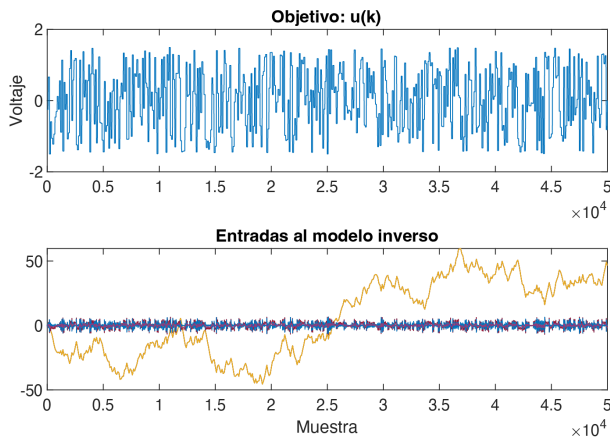
compromiso entre la fidelidad del modelo y una complejidad razonable, y sirve como base para la síntesis del controlador por modelo inverso neuronal desarrollada en este artículo.

### 3.2. Caracterización del conjunto de entrenamiento

La Figura 3 resume el conjunto de datos en lazo abierto utilizado como base para el entrenamiento del modelo inverso neuronal. En la subgráfica superior, se representa la señal de control  $u(k)$ , mientras que en la inferior se muestran las entradas al modelo inverso, es decir, los regresores construidos a partir de salidas y entradas retardadas.

Se aprecia que  $u(k)$  recorre de manera reiterada y persistente su rango operativo, sin intervalos prolongados en régimen estacionario ni señales claras de saturación. En coherencia con ello, los regresores exhiben variaciones suaves y acotadas a lo largo de todo el horizonte de muestras, lo que permite cubrir de forma densa la región de operación en torno al punto de trabajo.

Desde el punto de vista experimental, esta distribución temporal asegura una excitación suficiente del sistema en el dominio de interés y evita zonas del espacio de estados con muestreo insuficiente. En consecuencia, el conjunto ilustrado en la Figura 3 resulta adecuado para el entrenamiento supervisado del modelo inverso neuronal, pues proporciona ejemplos representativos de la dinámica relevante para el posterior control en lazo cerrado.



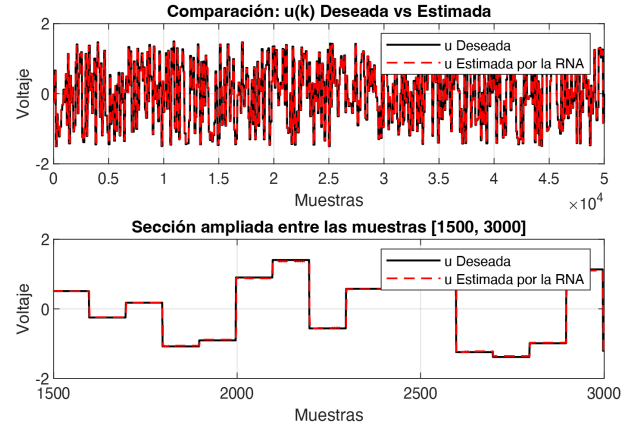
**Figura 3.** Conjunto de entrenamiento para el modelo inverso neuronal. Arriba: señal de control  $u(k)$ ; abajo: regresores en tiempo discreto (salidas y entradas retardadas), que actúan como entradas del modelo inverso.

### 3.3. Entrenamiento y evaluación del modelo inverso neuronal

Tras entrenar con las configuraciones de la Tabla 2, se obtuvieron las curvas de error cuadrático medio

(MSE) correspondientes a las arquitecturas con comportamiento más representativo.

La comparación entre la señal de control real  $u(k)$  y la estimada por la RNA entrenada se presenta en la Figura 4. Las dos trayectorias prácticamente se solapan en la mayor parte del horizonte de validación, y las discrepancias visibles se concentran en los transitorios de mayor pendiente. Este comportamiento es consistente con un ajuste adecuado del modelo inverso en el rango de operación considerado.



**Figura 4.** Comparación entre la señal real  $u(k)$  y la señal estimada por la RNA en el conjunto de validación.

El entrenamiento se detuvo en 15 épocas al cumplirse el criterio de validación temprana. El mejor resultado en validación apareció en la época 9, con un error cuadrático medio (MSE) de 0.012968, en el conjunto de entrenamiento se logró MSE de 0.0163 (RMSE  $\approx$  0.128), lo cual indica error pequeño y una generalización razonable de la RNA.

La configuración C5 se seleccionó para las pruebas en lazo cerrado debido a los resultados anteriores: MSE bajo, aprendizaje estable (entrenamiento y validación sin separarse) y una  $u(k)$  estimada que acompaña de cerca la referencia, sin discrepancias marcadas.

### 3.4. Criterios de aceptación del diseño para controlar el sistema

Se establecieron los siguientes umbrales de desempeño para evaluar las diferentes arquitecturas neuronales:

$$|e_{ss,\theta}| < 2^\circ, \quad (15)$$

$$t_{est,\theta} < 6 \text{ s } (\pm 2\%), \quad (16)$$

$$|\alpha|_{pico} < 0.15 \text{ rad } (\approx 8.6^\circ). \quad (17)$$

En la práctica, no todas las configuraciones neuronales logran satisfacer simultáneamente los tres umbrales en todos los escenarios de prueba. En particular, estos umbrales se emplean para comparar de forma objetiva las arquitecturas C1-C6, a partir de las métricas cuantitativas presentadas en las tablas de esta sección.

### 3.5. Evaluación en lazo cerrado: influencia de la arquitectura MLP

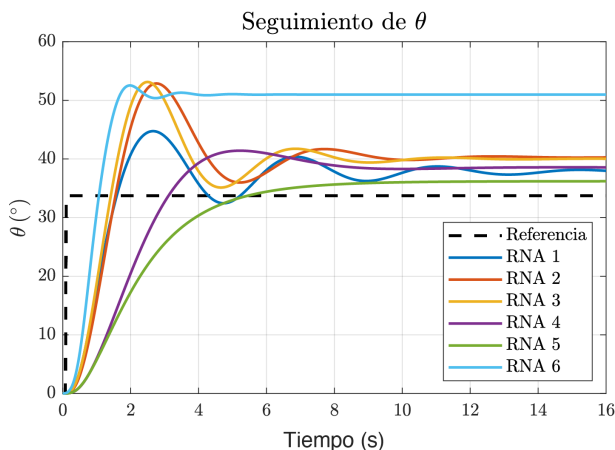
Para cada configuración de MLP de la Tabla 2, se entrenó un modelo inverso neuronal con el mismo conjunto de datos y se evaluó su desempeño en lazo cerrado en tres escenarios de referencia en  $\theta$ : i) escalón simple, ii) escalón periódico de amplitud variable; y iii) escalón periódico constante. Como métricas, se emplearon:

- el error cuadrático medio (RMSE);
- los índices integrales IAE e ITAE;
- el sobreimpulso y el tiempo de establecimiento en  $\theta$ ;
- el valor pico de  $|\alpha|$  como indicador de vibración en la punta.

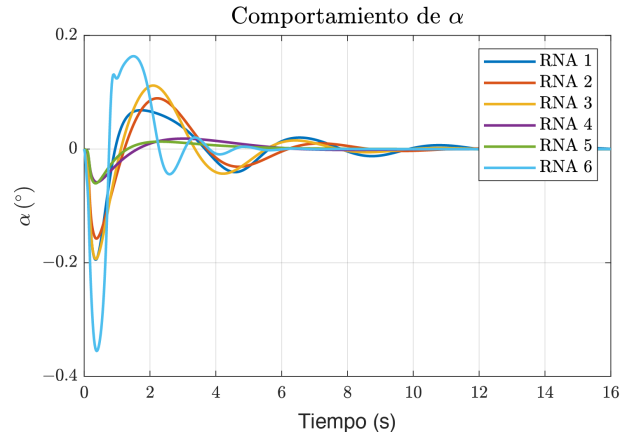
A continuación, se presentan los resultados para cada tipo de referencia, incluyendo las tablas con las métricas numéricas que respaldan las conclusiones.

#### 3.5.1. Escalón simple

La Figura 5 muestra la respuesta de  $\theta(t)$  ante un escalón simple para las seis configuraciones de la RNA, mientras que la Figura 6 ilustra el comportamiento del eslabón flexible  $\alpha(t)$ .



**Figura 5.** Comparación de la respuesta de  $\theta(t)$  para las seis configuraciones de RNA ante un escalón simple.



**Figura 6.**  $\alpha(t)$ : escalón simple para las seis configuraciones de RNA.

Los parámetros dinámicos extraídos de estas respuestas se resumen en la Tabla 3, mientras que las métricas de error correspondientes se presentan en la Tabla 4.

**Tabla 3.** Parámetros dinámicos ante señal escalón simple

Conf.	Sobreelong. [%]	$t_{\text{est}}$ [s]	Rango $\alpha$ [°]
C1	32.6	11.28	[-11.1, 3.9]
C2	56.7	8.52	[-9.0, 5.1]
C3	57.4	7.68	[-11.1, 6.4]
C4	22.7	7.48	[-3.3, 1.1]
C5	7.3	7.56	[-3.5, 0.8]
C6	55.7	2.20	[-20.4, 9.4]

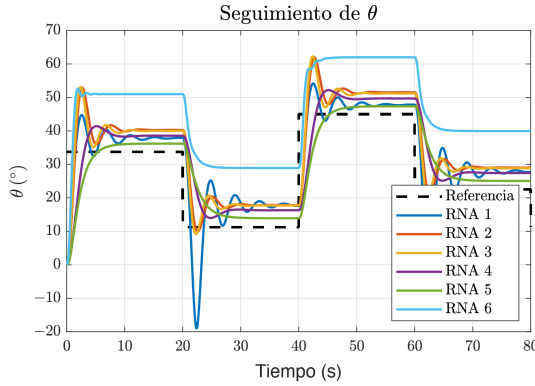
**Tabla 4.** Métricas de error para cada configuración (escalón simple)

Conf.	RMSE	IAE	ITAE
C1	7.6990	109.4850	840.0724
C2	9.9954	164.9611	1349.4845
C3	9.5792	156.5936	1287.0993
C4	9.7163	142.9398	1030.5302
C5	9.1722	102.2241	529.4320
C6	17.4475	343.1674	3442.3005

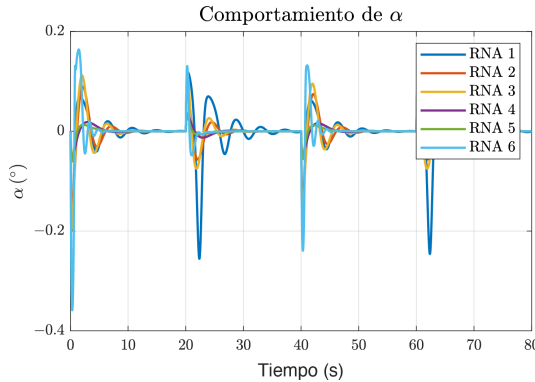
A partir de estas cifras, la configuración C5 presenta el compromiso más favorable: registra la menor sobreelongación (7.3 %) y uno de los menores tiempos de establecimiento (7.56 s), con un rango de oscilación de la punta muy acotado,  $\alpha \in [-3.5^\circ, 0.8^\circ]$ . Además, sus errores agregados son los más bajos del conjunto o cercanos a los mínimos, con RMSE = 9.17, IAE = 102.22 e ITAE = 529.43. En contraste, C6 presenta una sobreelongación marcada (55.7 %), un valor final de  $50.99^\circ$  claramente superior a la referencia y un rango de  $\alpha$  de  $[-20.4^\circ, 9.4^\circ]$ . Además, registra RMSE = 17.45, IAE = 343.17 e ITAE = 3442.30, valores varias veces mayores que los obtenidos con C5.

### 3.5.2. Escalón periódico variable

En la Figura 7 se muestra la respuesta de  $\theta$  ante una referencia de tipo escalón periódico con amplitud variable para las seis configuraciones, mientras que la Figura 8 ilustra el comportamiento de  $\alpha$ .



**Figura 7.**  $\theta(t)$ : escalón periódico variable para las seis configuraciones RNA.



**Figura 8.**  $\alpha(t)$ : escalón periódico variable.

A partir de los resultados de las Tablas 5 y 6, la configuración C5 vuelve a mostrar el compromiso más favorable entre la precisión en el seguimiento de  $\theta$  y el amortiguamiento de las vibraciones en la punta. Por ello, el análisis se centra en el rango de flexión  $\alpha$  y en los errores acumulados.

**Tabla 5.** Rango de flexión de la punta  $\alpha$  (entrada escalón periódico variable)

Conf.	Rango $\alpha$ [°]
C1	$[-14.6^\circ, 6.7^\circ]$
C2	$[-9.1^\circ, 5.3^\circ]$
C3	$[-11.2^\circ, 6.7^\circ]$
C4	$[-3.4^\circ, 2.0^\circ]$
C5	$[-3.5^\circ, 2.1^\circ]$
C6	$[-20.5^\circ, 9.4^\circ]$

En este escenario, C5 es la configuración que mantiene la flexión de la punta más acotada, con

un rango  $\alpha \in [-3.5^\circ, 2.1^\circ]$ , sensiblemente menor que el observado en el resto de configuraciones (C1–C4), cuyos intervalos se sitúan entre aproximadamente  $[-14.6^\circ, 6.7^\circ]$  y  $[-3.4^\circ, 2.0^\circ]$ . Esto implica que, para la misma señal de referencia, C5 reduce de forma consistente la amplitud de las oscilaciones en la punta sin degradar el seguimiento del perfil angular  $\theta(k)$ . Solo a modo de caso extremo, C6 concentra el comportamiento más vibratorio, alcanzando  $\alpha \in [-20.5^\circ, 9.4^\circ]$  y desviaciones más acusadas en  $\theta(k)$  respecto a la trayectoria deseada.

En términos de error acumulado, C5 presenta el mejor registro: RMSE = 8.39 y, de forma consistente, los menores índices integrales (IAE = 419.35, ITAE = 14944.07). Las configuraciones C1–C4 quedan relativamente cerca, pero dentro del mismo orden de magnitud.

Para la entrada de escalón periódico variable, C5 vuelve a ser la alternativa más sólida: combina el menor RMSE (8.39), los errores integrales más bajos (IAE = 419.35, ITAE = 14944.07) y la menor flexión de la punta, con  $\alpha \in [-3.5^\circ, 2.1^\circ]$ . El resto de configuraciones presenta un intercambio menos favorable entre precisión y vibración; C6, en particular, acumula tanto los mayores errores como el mayor rango de vibración,  $\alpha \in [-20.5^\circ, 9.4^\circ]$ .

**Tabla 6.** Métricas de error para entrada escalón periódico variable

Conf.	RMSE	IAE	ITAE
C1	8.8691	522.1624	20022.8260
C2	9.1393	615.1249	23273.9122
C3	8.7681	589.7227	22370.7007
C4	8.9462	543.6551	20005.5536
C5	8.3926	419.3452	14944.0715
C6	18.3018	1432.3069	57311.4578

### 3.5.3. Escalón periódico constante

Finalmente, se analizó el comportamiento del sistema para una referencia de tipo escalón periódico constante. La Figura 9 muestra la respuesta de  $\theta(t)$ , mientras que la Figura 10 presenta la respuesta de  $\alpha(t)$ .

En términos de flexión, las configuraciones C4 y C5 son las que mejor contienen la vibración de la punta, con rangos muy reducidos en torno a los  $-6^\circ$ :  $\alpha \in [-6.1^\circ, -5.7^\circ]$  y  $\alpha \in [-6.5^\circ, -6.1^\circ]$ . El resto de configuraciones muestra deflexiones significativamente mayores: C2 y C3 se mantienen en bandas estrechas, pero más alejadas del eje neutro, aproximadamente entre  $-16.5^\circ$  y  $-22.1^\circ$ , mientras que C1 y C6 concentran los peores casos, con rangos extremos de  $\alpha \in [-61.9^\circ, -30.9^\circ]$  y  $\alpha \in [-25.5^\circ, -19.2^\circ]$ , respectivamente. Esto indica que, para la misma entrada periódica, C4 y C5 son claramente las configuraciones que inducen las menores oscilaciones en la punta.

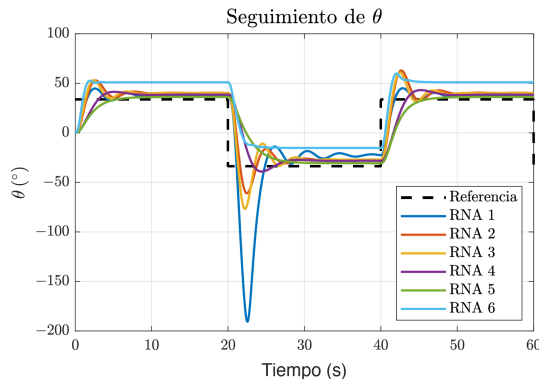


Figura 9.  $\theta(t)$ : escalón periódico constante.

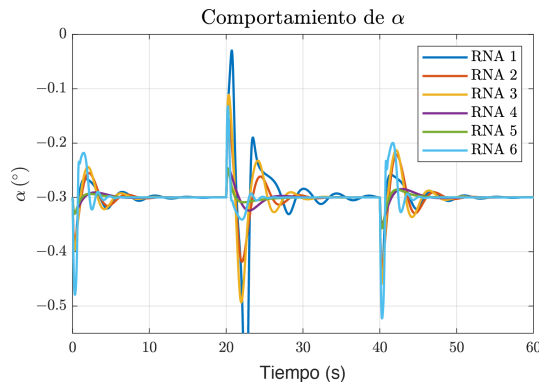


Figura 10.  $\alpha(t)$ : escalón periódico constante.

Tabla 7. Rango de flexión de la punta  $\alpha$  para la referencia de escalón periódico constante

Conf.	Rango $\alpha$ [°]
C1	[-61.9, -30.9]
C2	[-16.5, -15.0]
C3	[-22.1, -21.6]
C4	[-6.1, -5.7]
C5	[-6.5, -6.1]
C6	[-25.5, -19.2]

Para esta segunda evaluación, C5 encabeza los índices integrales con  $IAE = 752.84$  e  $ITAE = 29147.57$ , seguida muy de cerca por C4 ( $IAE = 774.76$ ,  $ITAE = 29764.48$ ). Aunque C2 y C3 reportan valores de RMSE ligeramente menores (15.47 y 15.82), esa ventaja puntual no se traduce en un mejor comportamiento global, ya que viene acompañada de errores integrales mayores y de una flexión de la punta más alejada del eje neutro. En el extremo opuesto, C1 y C6 continúan mostrando comportamientos alejados del deseado.

Tabla 8. Métricas de error para la referencia de escalón periódico constante

Conf.	RMSE	IAE	ITAE
C1	31.7209	1282.9760	54425.4854
C2	15.4694	834.2291	32438.9853
C3	15.8182	847.0670	33297.9880
C4	16.8459	774.7553	29764.4846
C5	17.7625	752.8363	29147.5748
C6	22.7026	1628.4212	65823.4133

### 3.6. Comparativa global de desempeño

A lo largo de los tres escenarios de prueba, la configuración C5 se perfila como la opción más robusta: combina rangos de flexión de la punta significativamente más acotados que los de la mayoría de configuraciones alternativas y, al mismo tiempo, mantiene RMSE, IAE e ITAE entre los valores más bajos en los distintos experimentos. Las configuraciones C1 y C6, por el contrario, concentran las peores combinaciones de error y vibración, mientras que las configuraciones C2-C4 logran un desempeño intermedio. Esto respalda la elección de C5 como arquitectura de referencia para el control por modelo inverso en el eslabón flexible ensayado.

### 3.7. Comparación con líneas base

Siguiendo el criterio de [18], se consideran como referencia tres enfoques de control probados previamente en sistemas RFL de características dinámicas similares: una "red neuronal clásica" (entrenada directamente en lazo cerrado), un controlador difuso y la arquitectura propuesta RNA-C5 basada en modelo inverso. Aunque las configuraciones físicas no son estrictamente idénticas (longitud del eslabón, inercia, etc.), las tres soluciones abordan la misma tarea de seguimiento en  $\theta$  con limitación de vibraciones en la punta  $\alpha$ , por lo que las cifras se utilizan como *comparación orientativa* del orden de magnitud del desempeño.

Bajo este marco, RNA-C5 resulta ser la estrategia más rápida, con  $t_{est} \approx 7.56$  s frente a la red neuronal clásica ( $\approx 9$  s) y un controlador difuso sensiblemente más lento ( $> 60$  s). En cuanto al pico de  $|\alpha|$ , expresado en grados, los valores de *fuzzy* y de la red neuronal clásica se sitúan en torno a  $3.3^\circ$ , mientras que RNA-C5 alcanza aproximadamente  $3.5^\circ$ , correspondiente al máximo absoluto del rango  $[-3.5^\circ, 0.8^\circ]$ .

En conjunto, RNA-C5 ofrece un tiempo de establecimiento claramente competitivo, a costa de un pico ligeramente mayor en la vibración de la punta. Esto es coherente con el énfasis de diseño puesto en el seguimiento inverso de  $\theta$  antes que en la minimización explícita de las oscilaciones en  $\alpha$ . Estas comparaciones no pretenden constituir un estudio exhaustivo entre controladores, sino situar el rendimiento de RNA-C5

dentro del rango de soluciones ya exploradas para sistemas RFL similares [18].

### 3.8. Análisis de estabilidad BIBO en tiempo discreto

El análisis de estabilidad de un sistema dinámico controlado por una red neuronal artificial (RNA) suele abordarse mediante procedimientos indirectos, dado que, en general, las redes neuronales no permiten derivar una expresión cerrada del modelo en lazo cerrado. En este trabajo, se emplean herramientas de identificación de sistemas para obtener una función de transferencia en tiempo discreto que aproxime la dinámica efectiva del canal  $\theta_{\text{ref}} \rightarrow \theta$  cuando el sistema RFL es controlado por la RNA C5.

Una de las formas más habituales de evaluar la estabilidad en sistemas discretos es mediante el criterio BIBO (*Bounded Input–Bounded Output*): un sistema es BIBO estable si, ante cualquier entrada acotada, su salida permanece acotada. En el caso de sistemas lineales invariantes en el tiempo (LTI, por sus siglas en inglés) en tiempo discreto, esta propiedad se comprueba analizando la ubicación de los polos de su función de transferencia en el plano  $z$ . Sea un sistema descrito por:

$$G(z) = \frac{b_0 + b_1 z^{-1} + \dots + b_m z^{-m}}{1 + a_1 z^{-1} + \dots + a_n z^{-n}}, \quad (18)$$

donde  $b_i$  y  $a_i$  son coeficientes reales. En consecuencia, el sistema es BIBO estable si, y solo si, todos los polos, es decir, las raíces del denominador, se ubican estrictamente dentro del círculo unitario:

$$|p_i| < 1, \quad \forall i. \quad (19)$$

#### 3.8.1. Identificación del modelo en lazo cerrado

Para realizar este análisis, se registraron las señales de referencia ( $\theta_{\text{ref}}$ ) y de salida ( $\theta$ ) ante un escalón, con el sistema operando bajo el control de la red C5. Con base en estos datos, se empleó la herramienta *System Identification Toolbox* de MATLAB para estimar una función de transferencia discreta de segundo orden, con un retardo de una muestra, que representa el comportamiento en lazo cerrado:

$$G(z) = \frac{0.0002971 z^{-1}}{1 - 1.966 z^{-1} + 0.9667 z^{-2}}. \quad (20)$$

El sistema tiene un tiempo de muestreo de  $T_s = 0.002$  s. A partir de esta función, se obtuvieron los polos y sus módulos:

- Polos:  $p_{1,2} = 0.9830 \pm 0.0203i$ ,
- Módulos:  $|p_1| = |p_2| = 0.9832$ .

Finalmente, ambos polos identificados cumplen  $|p_{1,2}| < 1$ ; por tanto, el lazo cerrado con la RNA C5 se clasifica como BIBO estable de manera local, al quedar todas las raíces del denominador dentro del círculo unitario. El hecho de que  $|p_{1,2}|$  sea cercano a 1 es consistente con una dinámica de establecimiento más lenta, pero con amortiguamiento marcado, en concordancia con los tiempos observados en las respuestas al escalón.

Este método indirecto de evaluación matemática de la estabilidad permite validar el desempeño del controlador neuronal sin necesidad de conocer de forma explícita el modelo interno del sistema. Es importante enfatizar que la función de transferencia  $G(z)$  identificada representa una aproximación lineal del comportamiento en lazo cerrado alrededor del punto de operación y para el conjunto de datos considerado. Por tanto, la conclusión de estabilidad BIBO obtenida a partir de los polos  $|p_{1,2}| = 0.9832 < 1$  es válida en un entorno local de las condiciones de operación utilizadas, asociadas a pequeñas variaciones en amplitud y frecuencia de referencia, pero no implica estabilidad global frente a saturaciones severas, grandes no linealidades o cambios drásticos en la dinámica del RFL. Esta interpretación es consistente con el uso de modelos lineales identificados para sistemas inherentemente no lineales como los manipuladores flexibles [1], y complementa los resultados temporales presentados en las subsecciones anteriores.

### 3.9. Discusión

#### 3.9.1. Relación con los criterios de diseño

Los criterios de aceptación definidos en las ecuaciones (15)–(17) se establecieron como metas particularmente exigentes. Para la entrada de escalón simple, ninguna configuración los satisface de manera simultánea; sin embargo, C5 es la que más se aproxima: mantiene una sobreelongación moderada, tiempos de respuesta del orden de 7.6 s y un pico de flexión en la punta de  $|\alpha|_{\text{pico}} \approx 3.5^\circ$ , claramente por debajo del límite de diseño ( $8.6^\circ$ ). En las entradas de escalón periódico, constante y variable, donde la sobreelongación y el tiempo de establecimiento dejan de ser medidas representativas, C5 conserva un rango reducido de  $\alpha$  (del orden de  $\pm 3.5^\circ$ ) y los menores errores integrales. Esto la posiciona como el compromiso más favorable entre seguimiento de  $\theta$  y amortiguamiento de vibraciones.

#### 3.9.2. Influencia de la arquitectura MLP

La comparación entre C1 y C6 evidencia que las redes demasiado pequeñas o excesivamente grandes no resultan adecuadas para este desarrollo. Las configuraciones con una sola capa y pocas neuronas (C1–C2) no representan adecuadamente la dinámica inversa del RFL; por ello se observan vibraciones mayores en la

punta. En el otro extremo, C6 puede acelerar la respuesta, pero empeora de manera sistemática RMSE, IAE e ITAE, y además, aumenta el rango de  $\alpha$ . En cambio, la arquitectura intermedia C5, con dos capas ocultas [12, 2], ofrece un mejor equilibrio. Se obtienen los mejores —o segundos mejores— valores de error en todos los escenarios, mantiene  $\alpha$  acotada y no pide esfuerzos de control extremos, evitando tanto falta de capacidad como sobreajuste.

### 3.9.3. Limitaciones y alcance

Los resultados se basan en simulaciones realizadas sobre un modelo identificado con ajustes superiores al 96 % en  $\theta$  y al 80 % en  $\alpha$ ; esto permite comparar arquitecturas, pero no sustituye la validación experimental, en la que pueden aparecer fricciones no lineales, holguras, ruido y saturaciones. El análisis BIBO discreto confirma estabilidad local ( $|p_{1,2}| = 0.9832 < 1$ ), pero no garantiza un comportamiento global ante referencias o perturbaciones extremas. Finalmente, la comparación con controladores difusos y redes neuronales convencionales, tomada de [18], es solo orientativa, al provenir de prototipos similares, pero no idénticos; una evaluación concluyente requerirá implementar todas las estrategias en el mismo banco de pruebas, lo cual se plantea como trabajo futuro.

### 3.9.4. Trabajos futuros

Como siguiente etapa, se propone implementar la arquitectura RNA-C5 en un prototipo real de RFL, evaluar su desempeño bajo ruido, saturaciones y retardos de cómputo, y comparar sus métricas RMSE, IAE, ITAE y rango de  $\alpha$  con los resultados obtenidos en simulación. Asimismo, se plantea analizar la sensibilidad del lazo cerrado frente a variaciones moderadas de masa, rigidez, fricción, amplitud y frecuencia de referencia. Finalmente, se recomienda comparar la propuesta con controladores PID, LQR, difusos y neuronales convencionales implementados en el mismo banco de pruebas.

## 4. Conclusiones

Este trabajo demostró que es posible diseñar y evaluar un controlador neuronal por modelo inverso para un sistema rotatorio con eslabón flexible, utilizando únicamente datos experimentales provenientes de un modelo identificado, sin recurrir a un modelado analítico detallado de alta complejidad.

A partir de un modelo en espacio de estados, con buen ajuste en  $\theta$  y aceptable en  $\alpha$ , se construyeron escenarios de prueba en lazo cerrado y se entrenaron seis arquitecturas MLP. Entre ellas, C5 [12, 2] muestra el mejor compromiso: mantiene  $|\alpha|_{\text{pico}} < 8.59^\circ$  y, en los tres tipos de referencia considerados, alcanza los

mejores valores de RMSE, IAE e ITAE, en comparación con configuraciones con poca capacidad (C1–C2) o con exceso de tamaño (C6).

El análisis de estabilidad BIBO en tiempo discreto confirmó de forma local la estabilidad del lazo cerrado con RNA-C5 (polos con módulo  $|p_{1,2}| < 1$ ), en coherencia con las respuestas temporales obtenidas. Además, la comparación orientativa con soluciones difusas y neuronales de la literatura sugiere que el desempeño de la propuesta es competitivo en términos de tiempo de establecimiento y nivel de vibraciones.

En síntesis, los resultados son coherentes con tres ideas principales. En primer lugar, contar con una identificación previa de la planta con niveles aceptables de fidelidad es un elemento clave para habilitar el control inverso neuronal. En segundo lugar, una MLP de tamaño moderado, junto con un diseño adecuado del conjunto de regresores, permite equilibrar el seguimiento de la referencia y la supresión de vibraciones. Por último, la combinación de métricas temporales con la verificación de estabilidad local proporciona una base cuantitativa sólida para avanzar hacia la validación experimental, el análisis de robustez y la comparación sistemática con controladores de referencia.

## Rol de los autores

- **Carlos Alberto Saldaña Enderica:** conceptualización, metodología, software, análisis formal, investigación, gestión de datos y redacción – borrador original.
- **José Ramón Llata:** metodología, validación, supervisión y redacción – revisión y edición.
- **Carlos Torre-Ferrero:** metodología, validación, supervisión y redacción – revisión y edición.

## Referencias

- [1] D. Subedi, I. Tyapin, and G. Hovland, “Review on modeling and control of flexible link manipulators,” *Modeling, Identification and Control: A Norwegian Research Bulletin*, vol. 41, no. 3, pp. 141–163, 2020. [Online]. Available: <https://doi.org/10.4173/mic.2020.3.2>
- [2] W. Tang and P. Daoutidis, “Data-driven control: Overview and perspectives,” in *2022 American Control Conference (ACC)*. IEEE, 2022, pp. 1048–1064. [Online]. Available: <https://doi.org/10.23919/ACC53348.2022.9867266>
- [3] K. Narendra and K. Parthasarathy, “Identification and control of dynamical systems using neural networks,” *IEEE Transactions on Neural*

- Networks*, vol. 1, no. 1, pp. 4–27, Mar. 1990. [Online]. Available: <https://doi.org/10.1109/72.80202>
- [4] S. Haykin, *Neural Networks and Learning Machines*, 3rd ed. Upper Saddle River, NJ, USA: Pearson Education, 2009, accessed: 2026-05-19. [Online]. Available: <https://upsalesiana.ec/ing36ar8r4>
- [5] S. Shin, M. Kang, and J. Baek, “Dynamic model learning and control of robot manipulator based on multi-layer perceptron neural network,” *Transactions of the Korean Society of Mechanical Engineers - A*, vol. 47, no. 12, pp. 945–957, Dec. 2023. [Online]. Available: <https://doi.org/10.3795/KSME-A.2023.47.12.945>
- [6] M. Deja and A. P. Markopoulos, “Advances and trends in non-conventional, abrasive and precision machining,” *Machines*, vol. 9, no. 2, p. 37, Feb. 2021. [Online]. Available: <https://doi.org/10.3390/machines9020037>
- [7] M. Suzuki and O. Kaneko, “Data-driven control by using data-driven prediction and LASSO for FIR typed inverse controller,” *Electronics and Communications in Japan*, vol. 106, no. 3, Aug. 2023. [Online]. Available: <https://doi.org/10.1002/ecj.12405>
- [8] S. Yahagi and M. Suzuki, “Direct data-driven design for a sparse feedback controller based on VRFT and LASSO regression,” *IFAC-PapersOnLine*, vol. 55, no. 25, pp. 229–234, 2022. [Online]. Available: <https://doi.org/10.1016/j.ifacol.2022.09.351>
- [9] E. Garrabe, H. Jesawada, C. D. Vecchio, and G. Russo, “On convex data-driven inverse optimal control for nonlinear, non-stationary and stochastic systems,” *Automatica*, vol. 173, p. 112015, Mar. 2025. [Online]. Available: <https://doi.org/10.1016/j.automatica.2024.112015>
- [10] Marji, A. M. Widodo, Marjono, W. Firdaus Mahmudy, and A. Maulana Muhamad, “Comparison of multi-layer perceptron and support vector machine methods on rainfall data with optimal parameter tuning,” *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 7, 2023. [Online]. Available: <https://dx.doi.org/10.14569/IJACSA.2023.0140745>
- [11] N. V. Thieu, S. Mirjalili, H. Garg, and N. T. Hoang, “Metaperceptron: A standardized framework for metaheuristic-driven multi-layer perceptron optimization,” *Computer Standards & Interfaces*, vol. 93, p. 103977, Apr. 2025. [Online]. Available: <https://doi.org/10.1016/j.csi.2025.103977>
- [12] C. Saldaña Enderica, J. R. Llata, and C. Torreferrero, “Guided reinforcement learning with twin delayed deep deterministic policy gradient for a rotary flexible-link system,” *Robotics*, vol. 14, no. 6, p. 76, May 2025. [Online]. Available: <https://doi.org/10.3390/robotics14060076>
- [13] J. G. Guarnizo Marin, N. Díaz Aldana, and C. Trujillo Rodríguez, “Design and implementation of an inverse neural network controller applied to VSC converter for active and reactive power flow, based on regions of work,” *Revista Facultad de Ingeniería Universidad de Antioquia*, no. 72, pp. 20–34, Aug. 2014. [Online]. Available: <https://doi.org/10.17533/udea.redin.15045>
- [14] V. A. Rodríguez-Toro, J. E. Garzón, and J. A. López, “Control neuronal por modelo inverso de un servosistema usando algoritmos de aprendizaje levenberg-marquardt y bayesiano,” *arXiv*, 2011. [Online]. Available: <https://doi.org/10.48550/arXiv.1111.4267>
- [15] M. Sasaki, M. Takeda, J. Muguro, and W. Njeri, “Trajectory control of flexible manipulators using forward and inverse models with neural networks,” *Vibration*, vol. 8, no. 3, p. 48, Aug. 2025. [Online]. Available: <https://doi.org/10.3390/vibration8030048>
- [16] M. T. Hagan, H. B. Demuth, M. H. Beale, and O. D. Jesús, *Neural Network Design*, 2nd ed. Stillwater, OK, USA: Martin Hagan, 2024, free eBook available online, Accessed: 2026-05-19. [Online]. Available: <https://upsalesiana.ec/ing36ar8r16>
- [17] M. Hagan and M. Menhaj, “Training feedforward networks with the marquardt algorithm,” *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989–993, 1994. [Online]. Available: <https://doi.org/10.1109/72.329697>
- [18] J. Capa López, *Control de un manipulador flexible de un único segmento*. Universidad de Cantabria, 2022. [Online]. Available: <https://upsalesiana.ec/ing36ar8r18>
- [19] C. A. Saldaña Enderica, J. R. Llata, and C. Torreferrero, “Optimization of Q and R matrices with genetic algorithms to reduce oscillations in a rotary flexible link system,” *Robotics*, vol. 13, no. 6, p. 84, May 2024. [Online]. Available: <https://doi.org/10.3390/robotics13060084>
- [20] Quanser. (2021) Rotary flexible link system identification and LQR design. MATLAB Central File Exchange. [Online]. Available: <https://upsalesiana.ec/ing36ar8r20>