Scientific Paper / Artículo Científico

UNIVERSIDAD POLITÉCNICA SALESIANA ECUADOR
INGENIUS
Revista de Ciencia y Tecnología
https://doi.org/10.17163/ings.n36.2026.08
pISSN: 1390-650X / eISSN: 1390-860X

Neural inverse control of a rotary flexible link

Control neuronal por modelo inverso de un eslabón flexible rotatorio

Carlos Alberto Saldaña Enderica[1,2,*], José Ramón Llata[1,2], Carlos Torre-Ferrero[1,2]

Received: 11-11-2025, Received after review: 04-02-2026, Accepted: 28-04-2026, Published: 01-07-2026

Abstract

This paper presents a data-driven inverse-model control scheme for a rotary flexible-link (RFL) system, with θ denoting the base angular position and α 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 θ tracking and α 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.

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 θ y la deflexión relativa de la punta por α. 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 θ y la mitigación de las oscilaciones en α. 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

[1,*]Departamento de Tecnología Electrónica, Ingeniería de Sistemas y Automática, Universidad de Cantabria, España.
[2]Facultad de Sistemas y Telecomunicaciones, Universidad Estatal Península de Santa Elena, Ecuador.
Corresponding author ✉: cse386@alumnos.unican.es.

Suggested citation: C. A. Saldaña Enderica, J. R. Llata and C. Torre-Ferrero "Neural inverse control of a rotary flexible link," *Ingenius, Revista de Ciencia y Tecnología*, N.° 36, pp. 98-109, 2026, DOI: https://doi.org/10.17163/ings.n36.2026.08.

1. Introduction

Single-link flexible rotary manipulators constitute a representative case study in both real-world applications, such as lightweight robotics and deployable structures, and academic testbeds for advanced control. Their underactuated nature and inherent structural flexibility generate oscillations and dynamic couplings that hinder the synthesis of robust and precise controllers, particularly when information about the physical model of the system is incomplete or uncertain [1].

In this context, data-driven control is an attractive alternative to classical methods whose design relies on detailed parametric models. Rather than depending on high-fidelity analytical identifications, data-driven approaches design controllers directly from experimental or simulated data, which is particularly useful for complex and nonlinear systems [2]. Within this framework, inverse-model control schemes based on multilayer perceptron (MLP) artificial neural networks (ANNs) have demonstrated the ability to approximate the inverse dynamics of manipulators and other nonlinear systems with considerably less explicit modeling effort [3–5]. These networks can capture highly nonlinear input–output relationships from data and have demonstrated strong performance in trajectory-tracking tasks and in compensating for unmodeled dynamic effects [6].

Recent literature on data-driven control and model inversion can be organized into three main research lines. The first focuses on the design of optimal controllers without relying on an explicit physical model, using finite impulse response (FIR) structures combined with LASSO-type regularization and convex formulations for inverse control. Overall, these methods reduce the structural complexity of the controller without compromising performance, while providing formal guarantees of data-based optimality [7–9]. In this context, hyperparameter selection, the effective number of FIR coefficients, and the quality of the experimental data are critical factors for achieving good generalization.

The second line addresses model inversion through machine learning, particularly by using MLPs to approximate the direct or inverse dynamics of manipulators. MLPs can outperform classical methods, such as SVMs, in dynamic prediction when their hyperparameters are properly tuned, although their performance remains sensitive to the representativeness of the training data [10]. In addition, several approaches incorporate optimization and regularization through metaheuristic techniques to improve computational efficiency and reduce overfitting in neural networks used for control [11].

The third line focuses on the robust synthesis of fixed-order controllers and the integration of noisy, non-ideal data. Data-driven approaches enable the model-matching problem and robustness requirements to be addressed directly in the data space, but they also introduce new trade-offs among robustness to uncertainty, computational complexity, and ease of implementation [9]. For rotary systems with flexible links, recent experimental studies combine state-space identification with guided reinforcement learning techniques [12], using metrics related to tracking performance, vibration reduction, and disturbance rejection. However, comparability among studies remains limited by the lack of standardized testing protocols and benchmark scenarios.

Despite these advances, several specific gaps can still be identified in the state of the art:

- The lack of studies that systematically integrate the data-based identification of an RFL, the construction of a neural inverse model, and its quantitative validation under multiple reference scenarios.
- The limited integration of neural inversion with formal model order selection criteria, supported by metrics such as FPE, MDL, and percentage fit [9], which allow the selection of moderate-dimensional models to be justified over lower-order alternatives.
- The absence of detailed analyses of closed-loop BIBO stability when neural controllers are employed, beyond qualitative observations based on simulations.

In this context, this work aims to analyze and demonstrate the feasibility of a data-driven inverse-model control scheme for a rotary flexible-link system, using an MLP-type ANN to approximate the inverse dynamics. Specifically, the proposed workflow integrates: i) continuous state-space identification of the plant from experimental data, following the framework of [12]; ii) construction of a regressor set based on delayed states and inputs for inverse-model training; iii) synthesis and selection of the neural architecture according to objective metrics; and iv) evaluation of the inverse-controller performance in terms of tracking error, vibration suppression, and discrete BIBO stability.

The approach is not intended to compete with high-level techniques in terms of absolute performance, but rather to demonstrate the value of a clear and reproducible methodology for designing, training, and validating a neural inverse controller in rotary systems with flexible links. This methodology can be extrapolated to other complex dynamic systems characterized by difficult analytical modeling, structural couplings, and sensitivity to disturbances, and it is presented through a representative case study that facilitates replication in control laboratories.

2. Materials and Methods

This section describes the state-space model of the RFL, the acquisition of experimental data, and the design of the ANN-MLP inverse neural controller used in the simulations.

2.1. General Description of Inverse Model Control

Inverse model control is based on training an ANN to approximate the inverse dynamics of the plant from input-output data pairs

$$\big(\boldsymbol{\varphi}(k),\, u(k)\big), \tag{1}$$

where $\boldsymbol{\varphi}(k)$ is a regressor vector containing delayed outputs and inputs, and $u(k)$ is the associated control signal [13,14]. During the training phase, the plant is excited using step signals of different magnitudes and random application times to cover a wide operating range and obtain a truly representative dataset [15]. The resulting records are then divided into training, validation, and test sets to adjust the ANN parameters while simultaneously mitigating overtraining, thereby preserving adequate generalization capability [4], [16].

The Levenberg-Marquardt algorithm is employed for training because of its fast convergence and ability to achieve low errors in moderate-size networks [16,17]. Robustness is reinforced through data normalization, partitioning into training, validation, and test sets, and early stopping based on the validation error.

2.2. General Workflow of the Methodology

The development of the neural controller is structured into three main stages:

1. Continuous state-space identification of the RFL from experimental data, yielding a fourth-order (A, B, C, D) model.
2. Generation of synthetic data from this model to train the neural inverse model.
3. Design, evaluation, and selection of the most suitable MLP architecture based on tracking metrics and link oscillation amplitude.

The platform used is the Quanser QLabs Virtual Rotary Flexible Link environment, implemented in MATLAB/Simulink® R2022b and QUARC 2.15, according to the workflow described in [12].

2.3. Identification of the Rotary Flexible-Link System

A single rotary flexible-link system actuated at the base is considered. The following variables are used:

$$\begin{aligned} &\theta(t):\ \text{angular position of the base,}\\ &\alpha(t):\ \text{relative deflection at the tip,}\\ &u(t) \in [-10,\, 10]\ \text{V.} \end{aligned} \tag{2}$$

For the identification of the continuous model, experimental data obtained from the Quanser virtual environment were used. The angular position θ and the relative deflection α were recorded with a sampling time

$$T_s = 2\ \text{ms}, \tag{3}$$

sufficient to capture the low-frequency bending modes without aliasing, consistent with previous studies on the same prototype [12], [18]. The tests were performed while maintaining $\alpha(t)$ within the interval $[-5°,\, 5°]$ to operate in a quasi-linear regime [1].

To ensure persistence of excitation and cover the operating range of interest, a pseudo-random square voltage signal was applied to $u(t)$, with amplitude and frequency varying within the safe interval $[-10,\, 10]$ V. Throughout the entire process, the operating constraints defined for the virtual model were respected [19]. Subsequently, the data were divided into:

- an identification set, used to estimate the model parameters; and
- a validation set, reserved exclusively to evaluate the predictive capability of the model.

2.3.1. Identification Procedure and Order Selection

A fourth-order continuous state-space model with one input and two outputs is adopted:

$$\begin{aligned} \dot{x}(t) &= A\,x(t) + B\,u(t)\\ y(t) &= C\,x(t) + D\,u(t) \end{aligned} \tag{4}$$

Where

$$x(t) = \begin{bmatrix}\theta(t) & \alpha(t) & \dot{\theta}(t) & \dot{\alpha}(t)\end{bmatrix}^{\top}, y(t) = \begin{bmatrix}\theta(t) & \alpha(t)\end{bmatrix}^{\top}. \tag{5}$$

Prediction Error Methods (PEM) were applied to the previously described data for model identification. As a result, a fourth-order model was obtained with $D = 0$, since the inertia of the motor–link assembly and the structural flexibility prevent instantaneous changes in the outputs in response to variations in $u(t)$ [12], [19].

The selection of a fourth-order model is supported by the comparative analyses reported in [20]. In particular, lower-order models do not adequately represent the bending dynamics, and increasing the order does not lead to significant improvements in FPE, MSE, or percentage fit. However, it increases computational cost and the risk of overfitting.

2.3.2. Identified Continuous Model

By applying the procedure described above, the following fourth-order continuous state-space model with one input and two outputs was obtained. The identified matrices are:

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

$$B = \begin{bmatrix} -0.05998 \\ 1.6080 \\ 7.1510 \\ -6.6660 \end{bmatrix} \tag{7}$$

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

$$D = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{9}$$

This model reproduces the coupled dynamics between θ and α within the considered operating range. Previous studies using the same RFL prototype have shown that continuous models of this order adequately capture the system behavior in tracking and vibration-suppression tasks [12], [20].

2.4. Design of the Neural Inverse Model Controller

From the identified continuous model (A,B,C,D), a state-space simulation of the RFL was constructed using Euler discretization with a step size of $dt = 0.002s$. The input was a random step signal, with new levels assigned every 2 within the range

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

applied for 1000 s. In this way, persistent data for

$$\theta,\ \alpha,\ \dot{\theta},\ \dot{\alpha},\ y\ and\ u, \tag{11}$$

which constituted the basis for regressor construction and estimation of the neural inverse model.

Table 1 shows the configuration used for training the inverse model. The selection was made to promote stable convergence, avoid overfitting, and maintain an adequate balance between accuracy and generalization capability.

Table 1. Training Parameters of the Neural Network

Parameter	Value
Optimization algorithm	Levenberg-Marquardt (`trainlm`)
Maximum number of iterations	600
Error criterion (objective)	10^{-4}
Training set	85%
Validation set	10%
Data normalization	`mapminmax` on inputs and outputs

The diagram in Figure 1 summarizes the inverse identification process. This scheme provides a clear visualization of how the information flow is organized during ANN training, highlighting the relationship between the employed regressors and the control signal that the network must learn to predict.

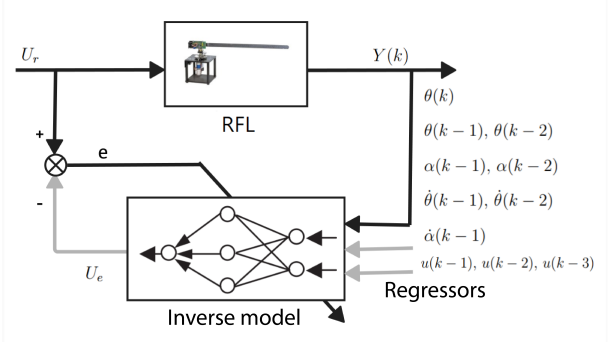

Figure 1. Scheme of the RFL inverse dynamics identification process.

2.4.1. Evaluated Architectures

Different MLP configurations are evaluated, varying in the number of neurons and hidden layers while maintaining a fixed linear activation function at the output. The configurations are detailed in Table 2.

Table 2. Evaluated Configurations for the MLP Neural Network

Conf.	Hidden layers	Activation functions
C1	[15]	*logsig–purelin*
C2	[20]	*logsig–purelin*
C3	[20, 10]	*logsig–logsig–purelin*
C4	[25, 15]	*logsig–logsig–purelin*
C5	[12, 2]	*logsig–logsig–purelin*
C6	[30, 20]	*logsig–logsig–purelin*

2.4.2. Inverse Model and Training

For each configuration in Table 2, a feedforward MLP neural network was trained using the Levenberg–Marquardt algorithm to approximate the control signal $u(k)$ from a regressor vector composed of current and delayed values of the measured variables. In NARX/NARMAX-type notation, this vector is defined as:

$$\boldsymbol{\varphi}(k) = \big[(I, q^{-1}, q^{-2})\,\theta(k),\ (q^{-1}, q^{-2})\,\alpha(k),\ (q^{-1}, q^{-2})\,\dot{\theta}(k),\ q^{-1}\dot{\alpha}(k),\ (q^{-1}, q^{-2}, q^{-3})\,u(k)\big]^{\top}. \tag{12}$$

where q^{-1} denotes the one-sample delay operator ($q^{-1}x(k) = x(k-1)$, $q^{-2}x(k) = x(k-2)$, etc.), and I is the identity operator.

Explicitly, the regressor vector can be expressed as

$$\boldsymbol{\varphi}(k) = \big[\theta(k),\ \theta(k-1),\ \theta(k-2),\ \alpha(k-1),\ \alpha(k-2),\ \dot{\theta}(k-1),\ \dot{\theta}(k-2),\ \dot{\alpha}(k-1),\ u(k-1),\ u(k-2),\ u(k-3)\big]^{\top}. \tag{13}$$

That is, the ANN receives as input a "time window" formed by the current and past samples of the base position θ, the tip deflection α, their corresponding velocities, and the control signal u over the three most recent sampling instants.

In the preliminary tests, regressor vectors with more delays were also considered, for example, including $\theta(k-3)$, $\alpha(k-3)$ or $u(k-4)$.

However, increasing the number of delays causes the dimension of $\boldsymbol{\varphi}(k)$ to grow without providing further relevant dynamic information, which worsens the conditioning of the training problem. In practice, the network tended not to converge, with the validation error remaining high, or stabilized at local minima with insufficient performance in reference tracking. For this reason, the regressor vector in (13) was adopted, as it provides an adequate compromise between temporal memory and learning capability.

The network output therefore approximates the inverse model of the plant:

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

where $\mathcal{N}_{\boldsymbol{\psi}}(\cdot)$ represents the MLP parameterized by the weight vector $\boldsymbol{\psi}$.

2.4.3. Closed-Loop Implementation

To implement the neural controller based on the trained inverse model, the MLP ANN generates the control signal $u(k)$ required for the RFL system to follow the reference $\theta_{\text{ref}}(k)$. The conceptual scheme of the closed-loop controller is shown in Figure 2.

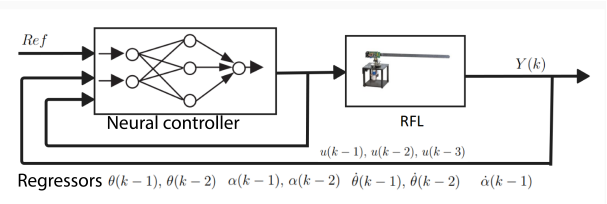

Figure 2. Diagram of the RFL neural inverse-model controller.

3. Results and Discussion

3.1. Results of the RFL Model Identification

For the validation set, the fourth-order continuous model provides:

- A time-domain fit for θ in the range of 99.85–96.95 % for one-step-ahead prediction and free-run simulation, with FPE $= 2.89 \times 10^{-13}$ and MSE $= 1.326 \times 10^{-6}$;
- A free-run simulation fit of 96.25 % for θ and 81.41 % for α, indicating that the coupled dynamics between the base rotation and the link flexure are reasonably captured within the considered operating range

In preliminary studies using the same platform, lower-order models ($n = 2, 3$) have been found to significantly degrade the prediction capability for α, whereas higher-order models do not provide significant improvements in FPE or MSE, but do increase complexity and sensitivity to noise [1], [20]. Therefore, the order $n = 4$ is adopted as a compromise between model fidelity and reasonable complexity, serving as the basis for the synthesis of the neural inverse-model controller developed in this article.

3.2. Characterization of the Training Set

Figure 3 summarizes the open-loop dataset used to train the neural inverse model. The upper subplot shows the control signal $u(k)$, whereas the lower subplot presents the inverse-model inputs, that is, the regressors constructed from delayed outputs and inputs.

The control signal $u(k)$ repeatedly and persistently spans its operating range, without prolonged steady-state intervals or clear signs of saturation. Consistently, the regressors exhibit smooth and bounded variations throughout the sample horizon, allowing dense coverage of the operating region around the operating point.

From an experimental perspective, this temporal distribution ensures sufficient excitation of the system within the domain of interest and avoids poorly

sampled regions of the state space. Consequently, the dataset illustrated in Figure 3 is suitable for the supervised training of the neural inverse model, as it provides representative examples of the dynamics relevant to subsequent closed-loop control.

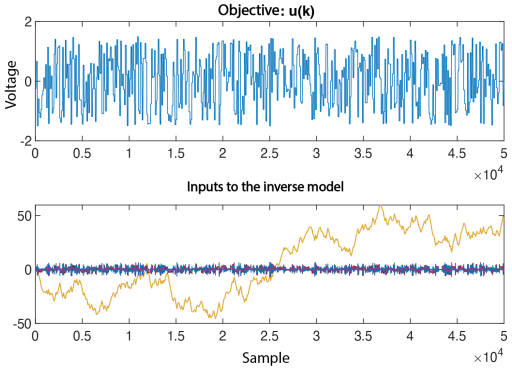

Figure 3. Training set used for the neural inverse model. Top: control signal $u(k)$; bottom: discrete-time regressors (delayed outputs and inputs) acting as inputs to the inverse model.

3.3. Training and Evaluation of the Neural Inverse Model

After training the neural inverse model using the configurations listed in Table 2, the mean squared error (MSE) curves corresponding to the architectures with the most representative behavior were obtained.

The comparison between the actual control signal $u(k)$ and that estimated by the trained ANN is presented in Figure 4. The two trajectories practically overlap over most of the validation horizon, with the visible discrepancies concentrated in the transients with the steepest slopes. This behavior is consistent with an adequate fit of the inverse model within the considered operating range.

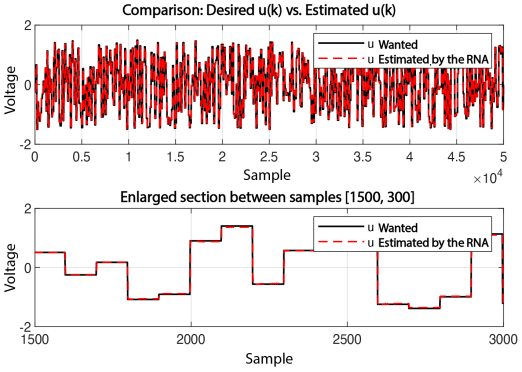

Figure 4. Comparison between the actual $u(k)$ and the $u(k)$ estimated by the ANN for the validation set.

Training stopped after 15 epochs after satisfying the early-stopping criterion based on validation performance. The best validation result was obtained at epoch 9, with a mean squared error (MSE) of 0.012968. For the training set, an MSE of 0.0163 (RMSE $\approx$ 0.128) was achieved, indicating a small error and reasonable ANN generalization capability.

Configuration C5 was selected for the closed-loop tests based on the previous results: low MSE, stable learning with no divergence in training or validation, and an estimated $u(k)$ that closely follows the reference without marked discrepancies.

3.4. Design Acceptance Criteria for Controlling the System

The following performance thresholds were established to evaluate the different neural architectures:

$$|e_{ss,\theta}| < 2^\circ, \tag{15}$$

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

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

In practice, not all neural configurations can simultaneously satisfy the three thresholds under all test scenarios. Therefore, these thresholds are used to objectively compare architectures C1–C6 based on the quantitative metrics reported in the tables of this section.

3.5. Closed-Loop Evaluation: Influence of the MLP Architecture

For each MLP configuration in Table 2, a neural inverse model was trained using the same dataset, and its closed-loop performance was evaluated under three reference scenarios in θ: i) single step, ii) periodic step with variable amplitude, and iii) constant periodic step. The following metrics were used:

- Root mean squared error (RMSE);
- Integral indices IAE and ITAE;
- Overshoot and settling time for θ;
- Peak value of $|\alpha|$ as an indicator of tip vibration.

The results for each reference type are presented below, including the tables with the numerical metrics supporting the conclusions.

3.5.1. Single Step

Figure 5 shows the response of $\theta(t)$ to a single step for the six ANN configurations, while Figure 6 illustrates the behavior of the flexible link $\alpha(t)$.

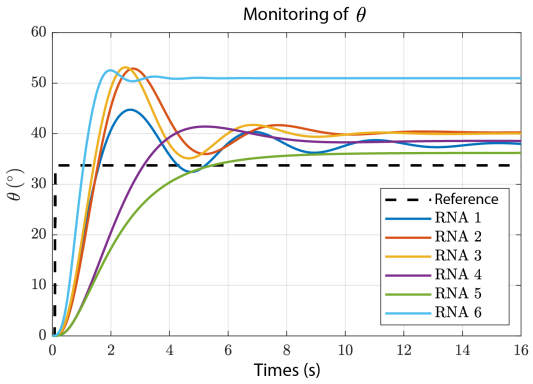

Figure 5. Comparison of the $\theta(t)$ response under a single-step reference for the six ANN configurations.

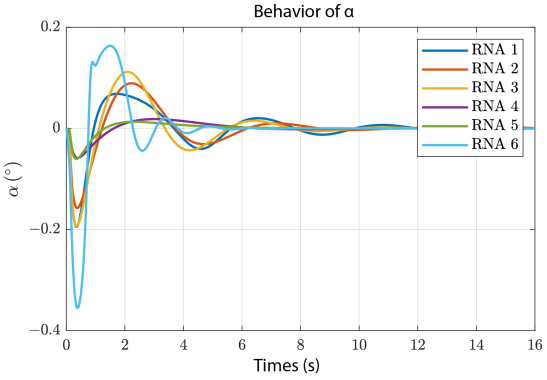

Figure 6. Behavior of $\alpha(t)$ for a single-step reference under the six ANN configurations.

The dynamic parameters extracted from these responses are summarized in Table 3, while the error metrics are presented in Table 4.

Table 3. Dynamic Parameters under a Single-Step Signal

Conf.	Overshoot [%]	t_{est} [s]	Range α [°]
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]

Table 4. Error Metrics for Each Configuration (Single Step)

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

From these figures, configuration C5 provides the most favorable compromise: it exhibits the lowest overshoot (7.3%) and one of the shortest settling times (7.56 s), with a highly contained tip-oscillation range of $\alpha \in [-3.5°, 0.8°]$. In addition, its aggregated errors are the lowest in the set, or close to the minimum values, with RMSE = 9.17, IAE = 102.22, and ITAE = 529.43. In contrast, C6 exhibits a pronounced overshoot (55.7%), a final value of 50.99°, clearly above the reference, and an α range of $[-20.4°, 9.4°]$. Furthermore, it records RMSE = 17.45, IAE = 343.17 and ITAE = 3442.30, values several times higher than those obtained with C5.

3.5.2. Variable Periodic Step

Figure 7 shows the θ response to a variable-amplitude periodic step reference for the six configurations, while Figure 8 illustrates the behavior of α.

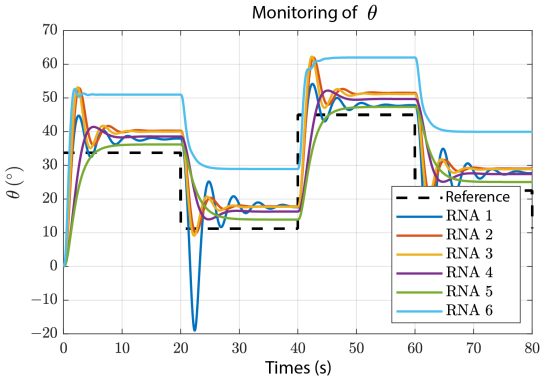

Figure 7. $\theta(t)$: response under a variable-amplitude periodic step reference for the six ANN configurations.

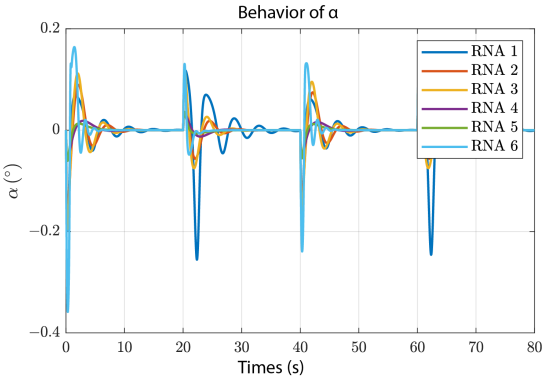

Figure 8. $\alpha(t)$: under a variable-amplitude periodic step reference.

From the results in Tables 5 and 6, configuration C5 again shows the most favorable compromise between θ tracking accuracy and damping of tip vibrations. The analysis focuses on the α flexion range and the accumulated errors.

In this scenario, C5 maintains the most contained tip flexure, with a range of $\alpha \in [-3.5°, 2.1°]$, noticeably smaller than those observed in the remaining configurations (C1–C4), whose intervals range approximately from $[-14.6°, 6.7°]$ to $[-3.4°, 2.0°]$ This indicates that, for the same reference signal, C5 consistently reduces the amplitude of tip oscillations without degrading the tracking of the angular profile $\theta(k)$. In contrast, C6 represents the most extreme case, concentrating the strongest vibratory behavior, reaching $\alpha \in [-20.5°, 9.4°]$ and exhibiting more pronounced deviations in $\theta(k)$ with respect to the desired trajectory.

In terms of accumulated error, C5 achieves the best performance, with RMSE = 8.39 and the lowest integral indices, IAE = 419.35 and ITAE = 14944.07. C1–C4 remain relatively close, with moderate increases but within the same order of magnitude.

For the variable periodic step input, C5 again proves to be the most robust alternative: it combines the lowest RMSE (8.39), the lowest integral errors (IAE = 419.35, ITAE = 14944.07), and the smallest tip flexure, with $\alpha \in [-3.5°, 2.1°]$. The remaining configurations exhibit a less favorable trade-off between accuracy and vibration, whereas C6 accumulates both the largest errors and the widest vibration range, with $\alpha \in [-20.5°, 9.4°]$.

Table 5. Tip-Flexure Range α Under a Variable-Amplitude Periodic Step Input

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

Table 6. Error Metrics Under a Variable-Amplitude Periodic Step Input

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. Constant Periodic Step

Finally, the system behavior was analyzed under a constant periodic step reference. Figure 9 shows the response of $\theta(t)$, while Figure 10 presents the response of $\alpha(t)$.

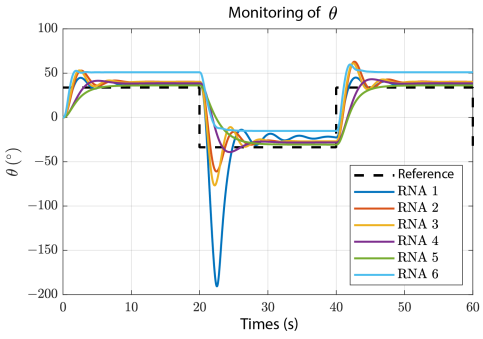

Figure 9. $\theta(t)$: response under a constant periodic step reference.

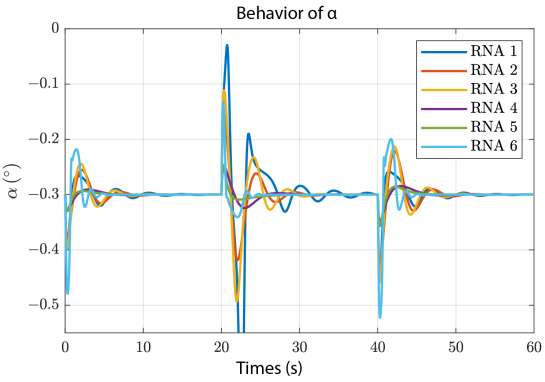

Figure 10. $\alpha(t)$: response under a constant periodic step reference.

In terms of flexure, configurations C4 and C5 provide the best containment of tip vibration, with very narrow ranges around −6°: $\alpha \in [-6.1°, -5.7°]$ and $\alpha \in [-6.5°, -6.1°]$ (C5). The remaining configurations exhibit significantly larger deflections: C2 and C3 remain within narrow bands but farther from the neutral axis (approximately between −16.5° and −22.1°), whereas C1 and C6 represent the worst cases, with extreme α ranges of $\in [-61.9°, -30.9°]$ and $\alpha \in [-25.5°, -19.2°]$, respectively. This indicates that, for the same periodic input, C4 and C5 are clearly the configurations that induce the smallest tip oscillations.

For this second evaluation, C5 achieves the best integral indices, with IAE = 752.84 and ITAE = 29147.57, followed closely by C4 (IAE = 774.76, ITAE = 29764.48). Although C2 and C3 report slightly lower RMSE values, 15.47 and 15.82, respectively, this specific advantage does not translate into better overall performance, as it is accompanied by larger integral errors and tip flexure farther from the neutral axis. At the opposite extreme, C1 and C6 continue to exhibit behaviors far from the desired performance.

Table 7. Tip-Flexure Range α Under a Constant Periodic Step Reference

Conf.	Range α [°]
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]

Table 8. Error Metrics Under a Constant Periodic Step Reference

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. Global Performance Comparison

Across the three test scenarios, configuration C5 emerges as the most robust option. It achieves substantially more contained tip-flexure ranges than most alternative configurations while consistently maintaining RMSE, IAE, and ITAE among the lowest values across the experiments. In contrast, configurations C1 and C6 concentrate the least favorable combinations of error and vibration, whereas C2–C4 achieve intermediate performance. These results support the selection of C5 as the reference architecture for inverse-model control in the tested flexible-link system.

3.7. Comparison with Baseline Approaches

Following the criterion in [18], three control approaches previously tested in RFL systems with similar dynamic characteristics are considered as references: a "classical" neural network trained directly in closed loop, a fuzzy controller, and the proposed ANN–C5 inverse-model architecture. Although the physical configurations are not strictly identical in terms of link length, inertia, and related parameters, the three solutions address the same θ-tracking task with tip-vibration limitation in α. Therefore, the reported figures are used as an indicative comparison of the order of magnitude of the performance.

Within this framework, ANN–C5 proves to be the fastest strategy, with $t_{\text{est}} \approx 7.56$ s, compared with the classical neural network (≈ 9 s)) and the significantly slower fuzzy controller (> 60 s). Regarding the peak value of $|\alpha|$, expressed in degrees, the fuzzy and NN approaches yield values of approximately 3.3°, whereas ANN–C5 reaches approximately 3.5°, corresponding to the maximum absolute value within the range [−3.5°, 0.8°].

Overall, ANN–C5 offers a clearly competitive settling time at the expense of a slightly larger peak in tip vibration, which is consistent with the design emphasis placed on inverse θ-tracking rather than on the explicit minimization of oscillations in α. These comparisons are not intended to provide an exhaustive study of controller performance, but rather to position ANN–C5 within the range of solutions already explored for similar RFL systems [18].

3.8. Discrete-Time BIBO Stability Analysis

The stability analysis of a dynamic system controlled by an artificial neural network (ANN) is commonly conducted through indirect procedures because neural networks generally do not allow a closed-form expression of the closed-loop model to be derived. In this work, system identification tools are used to obtain a discrete-time transfer function that approximates the effective dynamics of the $\theta_{\text{ref}} \to \theta$ channel when the RFL system is controlled by ANN–C5.

One of the most widely used approaches for evaluating the stability of discrete-time systems is the BIBO (bounded-input bounded-output) criterion, according to which a system is stable if every bounded input produces a bounded output. For discrete-time linear time-invariant (LTI) systems, this property is verified by analyzing the location of the poles of the transfer function in the z-plane. The system is described by:

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

where b_i and a_ii are real coefficients. Accordingly, the system is BIBO stable if and only if all poles, namely the roots of the denominator, lie strictly inside the unit circle:

$$|p_i| < 1, \quad \forall i. \tag{19}$$

3.8.1. Identification of the Closed-Loop Model

To perform this analysis, the reference (θ_{ref}) and output (θ) signals were recorded in response to a step input while the system operated under ANN–C5 control. Based on these data, the MATLAB System Identification Toolbox was employed to estimate a second-order discrete-time transfer function with a one-sample delay, representing the closed-loop behavior:

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

The system has a sampling time of $T_s = 0.002$ s. From this function, the poles and their corresponding magnitudes were obtained:

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

Finally, since both identified poles satisfy $|p_{1,2}| < 1$, the closed loop with ANN–C5 is classified as locally BIBO stable, as all the roots of the denominator lie inside the unit circle. The fact that $|p_{1,2}|$ is close to 1 is consistent with slower settling dynamics but marked damping, in agreement with the times observed in the step responses.

This indirect mathematical stability assessment enables the neural controller performance to be evaluated without requiring explicit knowledge of the internal system model. It should be emphasized that the identified transfer function $G(z)$ represents a linear approximation of the closed-loop behavior around the operating point and for the dataset considered. Therefore, the BIBO stability conclusion obtained from the poles $|p_{1,2}| = 0.9832 < 1$ is valid within a local neighborhood of the operating conditions used, with small variations in the amplitude and frequency of the reference, but it does not imply global stability under severe saturations, large nonlinearities, or drastic changes in the dynamics of the RFL. This interpretation is consistent with the use of identified linear models for inherently nonlinear systems such as flexible manipulators [1] and complements the time-domain results presented in the previous subsections.

3.9. Discussion

3.10. Relationship with the Design Criteria

The acceptance criteria defined in (15) to (17) were established as particularly demanding targets. For the single-step input, no configuration satisfies all of them simultaneously; however, C5 comes closest. It maintains moderate overshoot, response times on the order of 7.6 s and a peak tip flexure of $|\alpha|_{\text{peak}} \approx 3.5°$, which is clearly below the design limit of (8.6°). For the periodic step inputs, both constant and variable, overshoot and settling time are no longer representative measures. In these cases, C5 preserves a reduced α range, on the order of $\pm 3.5°$, and yields the lowest integral errors. These results position C5 as the most favorable compromise between θ-tracking and vibration damping.

3.11. Influence of the MLP Architecture

The comparison among C1–C6 shows that neither small networks nor excessively deep architectures are suitable for this application. Configurations with a single layer and few neurons (C1–C2) do not adequately represent the inverse dynamics of the RFL, resulting in larger tip vibrations. At the other extreme, C6 may accelerate the response, but it systematically worsens RMSE, IAE, and ITAE, while increasing the α range. By contrast, the intermediate architecture C5, with two hidden layers [12, 2], provides a better balance: it achieves the best or second-best error values in all scenarios, keeps α bounded, and avoids extreme control efforts, thereby mitigating both insufficient capacity and overfitting.

3.12. Limitations and Scope

The results are based on simulations using an identified model with fits above 96% for θ and 80% for α. This enables the comparison of architectures but does not replace experimental validation, where nonlinear friction, backlash, noise, and saturation effects may arise. The discrete-time BIBO analysis confirms local stability ($|p_{1,2}| = 0.9832 < 1$),but it does not guarantee global behavior under extreme references or disturbances. Finally, the comparison with fuzzy controllers and conventional neural networks reported in [18] is only indicative, since it is based on similar but not identical prototypes. A conclusive evaluation would require implementing all strategies on the same test bench, which is therefore proposed as a direction for future work.

3.13. Future Work

Future work will focus on bridging the gap between simulation and experimental implementation while strengthening the robustness of the proposed scheme:

- Implement the ANN–C5 architecture on the real RFL prototype and measure RMSE, IAE, ITAE, and the α range under noise, saturation, and computation delays, directly comparing the results with those obtained in simulation.
- Analyze the sensitivity of the closed loop to moderate variations in system parameters, including mass, stiffness, and friction, and in the reference signal, including amplitude and frequency, to verify the margins within which the observed stability is maintained.
- Test, on the same test bench, at least one classical controller, such as PID or LQR, and one fuzzy or conventional neural controller to provide a direct quantitative comparison with ANN–C5.

4. Conclusions

This work demonstrated the feasibility of designing and evaluating a neural inverse-model controller for a rotary flexible-link system using experimental data from an identified model, without resorting to complex analytical modeling.

Starting from a state-space model with a good fit for θ and acceptable fit for α, closed-loop test scenarios were constructed and six MLP architectures were

trained. Among them, C5 [12, 2] provided the best compromise, maintaining $|\alpha|_{\text{peak}} < 8.59°$ and, for the three reference types considered, achieving the best RMSE, IAE, and ITAE values when compared with configurations with insufficient capacity (C1–C2) or excessive size (C6).

The discrete-time BIBO stability analysis locally confirmed the stability of the closed loop with ANN–C5, with pole magnitudes satisfying $|p_{1,2}| < 1$, in agreement with the time-domain responses obtained. In addition, the indicative comparison with fuzzy and neural solutions reported in the literature suggests that the proposed approach is competitive in terms of settling time and vibration level.

Overall, the results support three main conclusions. First, prior plant identification with acceptable fidelity is a key element for enabling neural inverse control. Second, a moderately sized MLP, together with an adequate regressor design, allows reference tracking and vibration suppression to be balanced. Finally, combining time-domain metrics with local stability verification provides a solid quantitative basis for advancing toward experimental validation, robustness analysis, and systematic comparison with reference controllers.

Contributor Roles

- **Carlos Alberto Saldaña Enderica:** conceptualization, methodology, software, formal analysis, investigation, data curation, writing – original draft.
- **José Ramón Llata:** methodology, validation, supervision, writing – review and editing.
- **Carlos Torre-Ferrero:** methodology, validation, supervision, writing – review and editing.

References

[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. Torre-Ferrero, "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. Torre-Ferrero, "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