Received: 11 March 2021 | Revised: 3 August 2021 | Accepted: 1 September 2021

DOI: 10.1049/cth2.12202

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH PAPER

Combined adaptive neural network and regressor-based trajectory tracking control of flexible joint robots

Jorge Montoya-Cháirez[1] | Javier Moreno-Valenzuela[1] | Víctor Santibáñez[2] | Ricardo Carelli[3] | Fracisco G. Rossomando[3] | Ricardo Pérez-Alcocer[4]

[1] Instituto Politécnico Nacional-CITEDI, Av. Instituto Politécnico Nacional 1310, Tijuana, Baja California, México

[2] Tecnológico Nacional de México/Instituto Tecnológico de La Laguna, Torreón, México

[3] Instituto de Automática-Conicet-Universidad Nacional de San Juan, San Juan, Argentina

[4] CONACYT-Instituto Politécnico Nacional-CITEDI, Av. Instituto Politécnico Nacional 1310, Tijuana, Baja California, México

Correspondence
Javier Moreno-Valenzuela, Instituto Politécnico Nacional-CITEDI, Av. Instituto Politécnico Nacional 1310, Col. Nueva Tijuana, Tijuana, Baja California, México, 22435.
Email: moreno@citedi.mx

Funding information
TecNM projects; Secretaría de Investigación y Posgrado-Instituto Politécnico Nacional; CONACYT-Fondo Sectorial de Investigación para la Educación, Grant/Award Number: A1-S-24762; Consejo Nacional de Ciencia y Tecnología, Grant/Award Numbers: 134534, Cátedras 1537

Abstract

By relying on the input–output feedback linearization approach, a novel adaptive controller for flexible joint robots is proposed in this work. First, a model-based controller is developed to get a structure that is useful in the development of the adaptive controller. The adaptive version is developed by using two techniques. To stabilize the output function, an adaptive neural network controller is used, which approximates the non-linear function that contains the uncertainties. The desired rotor position required by the input–output feedback linearization controller is defined with the structure of a link dynamics adaptive regressor-based controller. The main reason to adopt the mentioned structure in the definition of the desired rotor link position is to guarantee its differentiability. Real-time experiment comparisons among the model-based controller, a model-based controller with desired compensation, an adaptive controller based on joint torque feedback, and an adaptive neural network-based controller are carried out. Experimental results support the theory reported in this document and the accuracy of the proposed approach.

1 | INTRODUCTION

Many of the controllers developed for robot manipulators are designed for rigid robots. The inclusion of the flexibility in the dynamic model when developing a motion controller would increase the precision of the task [1]. The flexibility in robot manipulators is included non-intentionally by harmonic drives, transmission belts, or long shafts, or is included intentionally with the purpose of increasing the safety with human interaction [2].

The study of controllers on flexible joint robots (FJRs) is not a new topic. Over the years, a lot of research has been done. Let us mention some of the past works. The feedback linearization technique was applied in [3], where the mathematical model used is globally feedback linearizable. In [4], a hybrid adaptive control for trajectory tracking of FJRs was proposed. The technique consists of one adaptive indirect controller to compensate uncertainties in the rigid-link equations, while an adaptive robust controller is devoted for compensating the dynamics on the actuators. A robust controller was designed in [5] for trajectory tracking of FJRs with two actuators. Three different controllers were developed in [6] by using the integrator backstepping, a model-based controller, an adaptive controller, and a robust controller. In [7], a partial state feedback controller was proposed by using an integrator backstepping approach for FJRs. An adaptive controller without velocity measurement for FJRs was proposed in [8]. Finally, in [9], three model-based controllers for FJRs were designed and tested. Nevertheless, the study on the development of controllers for FJRs is still an interesting topic, and much work has been done. Let us mention some recent works. An adaptive fuzzy controller for trajectory tracking of FJRs was developed in [10], where the controller is based on the backstepping theory, and fuzzy logic systems are used to approximate the unknown non-linearities of the dynamics. In [11], a voltage-based adaptive sliding mode controller for an n rigid-link flexible-joint robot manipulator was developed. An approach with a fractional-order observer-based adaptive controller for FJRs was presented in [12]. A time-optimal trajectory planning algorithm for flexible-joint robots was introduced in [13], where the algorithm is used to move the robot along smooth parametrized paths. A control scheme for trajectory tracking of FJRs was proposed in [14], where a model-based controller provides a simultaneous solution for trajectory tracking and variable impedance control. In [15], authors proposed a method to design a family of virtual contraction-based controllers to solve the trajectory tracking problem of flexible-joint robots in the port-Hamiltonian framework. A continuous fuzzy non-singular terminal sliding mode control based on a non-linear finite-time observer to control a serial-chain n-link flexible joint electrically-driven robot manipulator was proposed in [16].

FJRs are complex and are considered as underactuated mechanical systems [17]. The underactuated mechanical systems are those that have more degrees of freedom than control inputs [18]. Next, some examples of underactuated systems will be cited. The inertia wheel pendulum is a mechanism which consists of a wheel attached at the end of a pendulum, the wheel is actuated, and the pendulum is not actuated. Some works related to inertia wheel pendulum are [19–21] and [22]. The pendubot, studied in [23–24], is a double pendulum actuated in the shoulder joint, while the elbow joint remains free. Similarly, the acrobot system is a double pendulum, but the actuated joint is the elbow, and the free joint is the shoulder [25]. Another underactuated mechanical system is the cart-pendulum, studied in [26–27]. The cart-pendulum consists of a passive pendulum mounted on a cart. The movement of the cart should keep the pendulum at the upward vertical position. The Furuta pendulum is another test bench used in the literature to test the performance of controllers, as can be seen in [28]. Besides, the control of other underactuated mechanical systems has been studied in the literature.

The feedback linearization technique has been widely used to design efficient control schemes. This technique cancels the non-linear terms of the dynamic model, leaving a linear system to stabilize [29]. A disadvantage of this technique is that the dynamic model and its parameters have to be known. Besides, a good acquisition of the full-state is needed. For those reasons, this technique is combined with other ones to robustify the performance of the control action. Some recent works related to feedback linearization will be mentioned. In [30], a sliding mode controller based on feedback linearization was developed for an induction motor drive. The feedback linearization technique was applied to a sensorless induction motor drive in [31]. A model reference adaptive system-sliding mode observer is also incorporated to the scheme to compensate for the lack of sensors. In [32], the authors proposed a structural feedback linearization technique with a robust linearization to make an effective rejection of the non-linearities of a specific class of functions. A robust controller based on the feedback linearization technique was developed in [33] for a permanent magnet synchronous generator based-wind energy conversion system. In [34], a feedback linearization-based distributed model predictive control was proposed for the secondary voltage and frequency of an islanded microgrid. The feedback linearization technique with fractional-order calculus was implemented in [35] for the maximum power extraction framework of a wind energy conversion system. In [36], a feedback controller with two gain-scaling factor was proposed for global stabilization of a class of approximately feedback linearized non-linear systems. A static and a dynamic output feedback controllers were designed for a class of switched non-linear time-delay systems in [37].

As was mentioned before, the feedback linearization technique needs the knowledge of the parameters of the dynamic model. In order to avoid this problem, adaptive controllers are a solution when the parameters are not precisely known, given that this technique allows estimating the parameters online. Adaptive control has been proven to show a good performance, as shown in [38–45]. Besides, the use of neural networks in control systems has been very popular. One of the most important properties of neural network is stated in the universal approximation theorem [46]. The weights of neural networks can be computed online, or can be computed offline with different training algorithms. In the literature, works where offline algorithms are employed to control complex systems have been given in [47–49].

On the other hand, adaptive neural network controllers are of low computational cost, with the advantages of not requiring an offline training process and the property that non-linear functions may be approximated. Adaptive neural networks-based controllers compensate the unknown dynamics, the unknown disturbances, and they do not demand the system parameters resulting in a robust controller. Nowadays, adaptive neural networks have been widely used. Let us mention some recent works. In [50], an inverse non-linear controller combined with an adaptive neural controller was proposed for mobile robots. An adaptive neural sliding mode controller in discrete-time was developed in [51], which was applied for trajectory tracking of a SCARA robot arm. In [52], a PID for a SCARA robot was proposed with the particularity that the PID gains are estimated online with an adaptive neural network. A trajectory tracking controller for an underactuated control moment gyroscope was proposed in [53]. The control goal was achieved utilizing an adaptive neural network. In [54], an adaptive neural controller was proposed for attitude tracking and attitude stabilization for a hexacopter with uncertain dynamics. In [55], an adaptive neural network-based controller with an event triggered mechanism was developed for a class of single-input single-output high-order non-linear systems. In [56], the authors proposed a trajectory tracking controller for a class of uncertain non-linear systems with time delay by using command filtered-based event-triggered adaptive neural network control. A filtering adaptive neural network-based controller with Gaussian radial basis function was developed in [57] to compensate mismatched uncertainties in multivariable non-linear systems. A backstepping-based controller with adaptive neural networks was proposed in [58] for non-strict-feedback systems with input delays. The consensus tracking control problem was addressed in [59], where an adaptive neural networks controller based on backstepping technique was proposed. In [60], a controller based on a double adaptive neural network was proposed to stabilize a Furuta pendulum in the presence of external disturbances. The problem of input saturation and state constraints for rigid robot manipulators was studied in [61] by using adaptive neural networks. In [62], an adaptive neural network controller for stochastic non-linear systems with full state constraints was developed. An adaptive tracking controller for a class of stochastic non-linear systems with input saturation based on multi-dimensional Taylor network via backstepping was proposed in [63]. In [64], an adaptive neural network finite-time dynamic surface controller was proposed for systems with an unknown dead zone.

Given the robustness of adaptive neural networks, this technique has been widely applied to FJRs for decades. For example, in [65], an adaptive neural networks controller was designed by using the feedback linearization technique developed in [3]. More recently, in [66], an adaptive dynamic surface control was designed for a FJR with unmodelled dynamics and time-varying output constraints. The authors used radial basis functions neural networks to approximate the unknown continuous functions. A neuro-adaptive observer-based controller was developed in [67], which used radial basis function neural networks to estimate state variables of one-link FJR. In [68], an adaptive neural network controller was proposed for trajectory tracking control of FJRs. The scheme used a multilayer neural network, in which output weights are adapted online based on the trajectory tracking error, while the input weights use an online algorithm based on backpropagation. An adaptive controller with actor-critic design for trajectory tracking was developed in [69]. The controller used a neural network to calculate the cost function used to judge the control performance, and another neural network is used to cope with system uncertainties. An adaptive hybrid impedance controller for electrically driven FJRs with input saturation was proposed in [70], where adaptive neural networks are used to approximate the saturation terms.

As can be seen in the literature review, many works on adaptive neural networks have been applied to FJRs. In all the works, the universal approximation property of neural networks is used to approximate unknown functions. While in [65], the state feedback linearization control technique is used to develop a controller, in this manuscript the input-output feedback linearization framework is used to develop an adaptive controller. In our design, the output function is inspired by the results given in [71]. Another important difference between the results in [65] and our approach is that in the input–output feedback linearization, an analysis for the internal dynamics should be carried out. In order to stabilize the internal dynamics, the desired rotor position is calculated as an adaptive regressor-based auxiliary controller. As it was mentioned in the literature review, previous works explored different ways to address the trajectory tracking control, but no one considered the approach introduced in this work. Besides, the controller in this work is developed so that a singularity-free scheme is synthesized.

In this document, a combined adaptive neural network and regressor-based trajectory tracking controller for FJRs is proposed. First, a model-based controller is developed by using the input-output feedback linearization technique. The controller is designed to achieve link trajectory tracking. The controller is designed in two parts, the control law that stabilizes the external dynamics, and the desired rotor position stabilizes the internal dynamics. Then, taking into account the model-based controller, an adaptive version is developed by using neural networks. It is important to remark that adaptive neural networks require a discontinuous function to compensate for the approximation error. Given that the controller $\boldsymbol{\tau}_m$ used to stabilize the external dynamics requires the desired rotor acceleration, the desired rotor position can not be developed with adaptive neural networks. Taking this into account, the desired rotor position required by the adaptive neural network controller is given in the form of a link dynamics adaptive regressor-based controller. The main contribution of this work is the development of a combined adaptive controller which uses adaptive neural networks and a link dynamics adaptive regressor. Combining these two techniques makes the controller not require any parameter knowledge, not even the elasticity constant. The use of neural networks compensates for non-modelled system dynamics. Besides, as was mentioned in the literature review, adaptive neural networks do not require a training process. In fact, the weights are estimated online like any adaptive regressor-based controller. Analysis of the closed-loop trajectories is provided. The theory is validated by means of real-time experiments in a flexible joint robot of two links. In order to provide a comparison, the model-based controller, a model-based controller with desired compensation proposed in [9], an adaptive controller based on joint torque feedback developed in [72], and an adaptive neural network-based controller designed in [65] are implemented experimentally and the results are analyzed. The proposed adaptive controller shows the best performance among all the controllers tested in this work.

The remaining of this document is organized as follows. In Section 2, the dynamic model and the control problem is presented. The model-based controller is developed in Section 3, which is inspired in the input-output feedback linearization technique. In Section 4, the design of the combined adaptive controller and the analysis of the closed-loop trajectories are presented. The experimental platform, the controllers used for the performance comparison, and the experimental results are in Section 5. Finally, the conclusions are presented in Section 6.

2 | DYNAMIC MODEL, CONTROL PROBLEM, AND MATHEMATICAL PRELIMINARIES

2.1 | Dynamic model

The dynamic model of a flexible joint robot in the horizontal plane, as the one shown in Figure 1, is given by [2], [73],

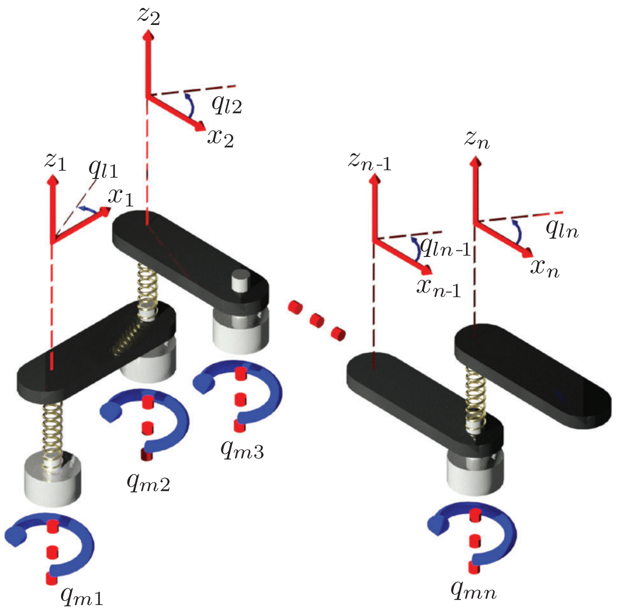

FIGURE 1 Representation of an n-link flexible joint robot

$$M(\boldsymbol{q_l})\ddot{\boldsymbol{q}}_l + C(\boldsymbol{q_l},\dot{\boldsymbol{q}}_l)\dot{\boldsymbol{q}}_l + F_{vl}\dot{\boldsymbol{q}}_l + K(\boldsymbol{q_l}-\boldsymbol{q_m}) = 0_{n\times 1}, \tag{1}$$

$$J\ddot{\boldsymbol{q}}_m + F_{vm}\dot{\boldsymbol{q}}_m + K(\boldsymbol{q_m}-\boldsymbol{q_l}) = \boldsymbol{\tau}_m, \tag{2}$$

with $M(\boldsymbol{q_l}) \in \mathbb{R}^{n\times n}$ the inertia matrix of the link side of the robot, $C(\boldsymbol{q_l},\dot{\boldsymbol{q}}_l) \in \mathbb{R}^{n\times n}$ the Coriolis and centrifugal forces matrix, $K \in \mathbb{R}^{n\times n}$ is the matrix associated with the stiffness of the springs, J is the constant diagonal positive definite matrix of $n \times n$, which contains the inertia of the gears and rotors. F_{vl} and $F_{vm} \in \mathbb{R}^{n\times n}$ are constant positive definite diagonal matrices which contain the viscous friction coefficients of the joints, before and after of the flexible elements, respectively, and $\boldsymbol{\tau}_m \in \mathbb{R}^n$ is the torque input vector. The coordinates $\boldsymbol{q_l}$ and $\boldsymbol{q_m} \in \mathbb{R}^n$ denote the position of the links and the rotors, respectively. Finally, the coordinates $\dot{\boldsymbol{q}}_l, \ddot{\boldsymbol{q}}_l \in \mathbb{R}^n$ and $\dot{\boldsymbol{q}}_m, \ddot{\boldsymbol{q}}_m \in \mathbb{R}^n$ are the velocities and accelerations of the link and rotor, respectively.

2.2 | Mathematical preliminaries and important properties

In this work, the Euclidean norm of a vector $\boldsymbol{x} \in \mathbb{R}^n$ is denoted by $\|\boldsymbol{x}\| = \sqrt{\boldsymbol{x}^T\boldsymbol{x}}$. The notation $\lambda_{\min}\{A\}$ and $\lambda_{\max}\{A\}$ represent the minimum and maximum eigenvalues of a matrix $A \in \mathbb{R}^{n\times n}$, respectively. The spectral norm of a matrix A is defined as $\|A\| = \sqrt{\lambda_{\max}A^TA}$.

The Rayleigh–Ritz theorem establishes that for all $\boldsymbol{x} \in \mathbb{R}^n$,

$$\lambda_{\max}\{A\}\|\boldsymbol{x}\|^2 \geq \boldsymbol{x}^TA\boldsymbol{x} \geq \lambda_{\min}\{A\}\|\boldsymbol{x}\|^2, \tag{3}$$

where $A = A^T \in \mathbb{R}^{n\times n}$ is a symmetric matrix.

For a symmetric and positive definite matrix A, the next inequality holds

$$|\boldsymbol{y}^TA\boldsymbol{x}| \leq \|A\|\|\boldsymbol{y}\|\|\boldsymbol{x}\|, \tag{4}$$

where $\|A\| = \lambda_{\max}\{A\}$ in this case, and $\boldsymbol{x},\boldsymbol{y} \in \mathbb{R}^n$.

The trace of a matrix A is denoted by $Tr(A)$. The trace identity relation of two vectors $\boldsymbol{a} \in \mathbb{R}^m$ and $\boldsymbol{b} \in \mathbb{R}^m$ is defined by

$$Tr(\boldsymbol{ab}^T) = \boldsymbol{b}^T\boldsymbol{a}. \tag{5}$$

In the case of a matrix $B \in \mathbb{R}^{n\times m}$, the Frobenius norm is defined by [74]

$$\|B\|_F^2 = \sum_{i,j} b_{i,j}^2 = Tr(B^TB). \tag{6}$$

The Frobenious inner product is defined as

$$\langle C, D\rangle_F = Tr(C^TD),$$

and according to the Schwartz inequality

$$|\langle C, D\rangle_F| = |Tr(C^TD)| \leq \|C\|_F\|D\|_F, \tag{7}$$

with $C \in \mathbb{R}^{n\times m}$ and $D \in \mathbb{R}^{n\times m}$ as two compatibility dimensioned matrices.

It is worth noting that the rigid part of a FJR, denoted by equation (1), holds the next properties [75]:

Property 1. *The inertia matrix $M(\boldsymbol{q_l})$ is symmetric positive definite, and*

$$\lambda_{\min}\{M(\boldsymbol{q_l})\}\|\boldsymbol{x}\|^2 \leq \boldsymbol{x}^TM(\boldsymbol{q_l})\boldsymbol{x} \leq \lambda_{\max}\{M(\boldsymbol{q_l})\}\|\boldsymbol{x}\|^2. \tag{8}$$

for all $\boldsymbol{x} \in \mathbb{R}^n$.

Property 2. *For the Coriolis and centrifugal matrix $C(\boldsymbol{q_l},\dot{\boldsymbol{q}}_l)$, there exists a number k_{C_1}, such that*

$$\|C(\boldsymbol{q_l},\boldsymbol{x})\boldsymbol{y}\| \leq k_{C_1}\|\boldsymbol{x}\|\|\boldsymbol{y}\| \tag{9}$$

for all $\boldsymbol{x},\boldsymbol{y} \in \mathbb{R}^n$.

Property 3. *The matrix $C(\boldsymbol{q_l},\dot{\boldsymbol{q}}_l)$ denotes the Coriolis and centrifugal forces matrix and is related to the inertia matrix, as follows*

$$\boldsymbol{x}^T\left[\frac{1}{2}\dot{M}(\boldsymbol{q_l}) - C(\boldsymbol{q_l},\dot{\boldsymbol{q}}_l)\right]\boldsymbol{x} = 0, \tag{10}$$

for all $\boldsymbol{x} \in \mathbb{R}^n$. Besides,

$$\dot{M}(\boldsymbol{q_l}) = C(\boldsymbol{q_l},\dot{\boldsymbol{q}}_l) + C(\boldsymbol{q_l},\dot{\boldsymbol{q}}_l)^T. \tag{11}$$

Assumption 1. The equation (1) is linear in the parameters, in the form:

$$K^{-1}[M(\boldsymbol{q}_l)\ddot{\boldsymbol{q}}_l + C(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l)\dot{\boldsymbol{q}}_l + F_{vl}\dot{\boldsymbol{q}}_l + K(\boldsymbol{q}_l - \boldsymbol{q}_m)] \\ = \Phi(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l, \ddot{\boldsymbol{q}}_l)\boldsymbol{\theta}, \qquad (12)$$

where $\Phi(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l, \ddot{\boldsymbol{q}}_l) \in \mathbb{R}^{n \times p}$ is the regression matrix, and $\boldsymbol{\theta} \in \mathbb{R}^p$ is the vector with all the lumped parameters.

Because the matrix $K \in \mathbb{R}^{n \times n}$ is diagonal and invertible, equation (12) resembles a property rather than an assumption.

2.3 | Control problem

Let us assume that $\boldsymbol{q}_l$, $\dot{\boldsymbol{q}}_l$, $\ddot{\boldsymbol{q}}_l$, $\dddot{\boldsymbol{q}}_l$, $\boldsymbol{q}_m$, and $\dot{\boldsymbol{q}}_m$ are available for measurement. The control problem consists of designing a control law $\boldsymbol{\tau}_m$ such that

$$\lim_{t \to \infty} \tilde{\boldsymbol{q}}_l(t) = 0_{n \times 1}, \qquad (13)$$

where

$$\tilde{\boldsymbol{q}}_l = \boldsymbol{q}_{ld} - \boldsymbol{q}_l, \qquad (14)$$

with $\boldsymbol{q}_{ld}(t) \in \mathbb{R}^n$ the desired position of the links, which is assumed to be four times differentiable at least.

Next, an output function will be presented, and the input–output feedback linearization technique will be used to develop a motion controller. Besides, the stability of the internal and external dynamics will be discussed.

3 | INPUT-OUTPUT FEEDBACK LINEARIZATION-BASED CONTROLLER

3.1 | Open-loop system

By using the definition of the link position error $\tilde{\boldsymbol{q}}_l$ in (14) and defining the rotor position error as

$$\tilde{\boldsymbol{q}}_m = \boldsymbol{q}_{md} - \boldsymbol{q}_m,$$

with $\boldsymbol{q}_{md}$ as the desired position of the rotor. Then, the equations (1) and (2) can be expressed in the state-space form as

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(t, \boldsymbol{x}) + G(t, \boldsymbol{x})\boldsymbol{\tau}_m, \qquad (15)$$

where

$$\boldsymbol{x} = \left[\tilde{\boldsymbol{q}}_l^{\,T} \; \dot{\tilde{\boldsymbol{q}}}_l^{\,T} \; \tilde{\boldsymbol{q}}_m^{\,T} \; \dot{\tilde{\boldsymbol{q}}}_m^{\,T}\right]^T \in \mathbb{R}^{4n}, \qquad (16)$$

$$\boldsymbol{f}(t, \boldsymbol{x}) = \begin{bmatrix} \dot{\tilde{\boldsymbol{q}}}_l \\ \ddot{\boldsymbol{q}}_{ld} + M(\boldsymbol{q}_{ld} - \tilde{\boldsymbol{q}}_l)^{-1}[C(\boldsymbol{q}_{ld} - \tilde{\boldsymbol{q}}_l, \dot{\boldsymbol{q}}_{ld} - \dot{\tilde{\boldsymbol{q}}}_l) \\ (\dot{\boldsymbol{q}}_{ld} - \dot{\tilde{\boldsymbol{q}}}_l) + F_{vl}(\dot{\boldsymbol{q}}_{ld} - \dot{\tilde{\boldsymbol{q}}}_l) \\ +K(\boldsymbol{q}_{ld} - \tilde{\boldsymbol{q}}_l - \boldsymbol{q}_{md} + \tilde{\boldsymbol{q}}_m)] \\ \dot{\tilde{\boldsymbol{q}}}_m \\ \ddot{\boldsymbol{q}}_{md} + J^{-1}[F_{vm}(\dot{\boldsymbol{q}}_{md} - \tilde{\boldsymbol{q}}_m) \\ +K(\boldsymbol{q}_{md} - \tilde{\boldsymbol{q}}_m - \boldsymbol{q}_{ld} + \tilde{\boldsymbol{q}}_l)] \end{bmatrix}, \qquad (17)$$

and

$$G(t, \boldsymbol{x}) = \begin{bmatrix} 0_{n \times n} \\ 0_{n \times n} \\ 0_{n \times n} \\ -J^{-1} \end{bmatrix}.$$

Notice that an FJR is an underactuated system of $4n$ dimension represented by the states $\boldsymbol{x}$ in (16). Thus, the control goal in (13) is achieved under a non-conventional design controller [17].

The output function is inspired by the feedback linearization controllers developed in [71], the output function consists of a linear combination of the states $\boldsymbol{x}$, as follows:

$$\boldsymbol{y} = \alpha_1\tilde{\boldsymbol{q}}_l + \alpha_2\tilde{\boldsymbol{q}}_m + \beta_1\dot{\tilde{\boldsymbol{q}}}_l + \beta_2\dot{\tilde{\boldsymbol{q}}}_m,$$

where $\boldsymbol{y} \in \mathbb{R}^n$ and α_1, α_2, β_1, and β_2 are constant matrices of $n \times n$ dimension to be defined.

3.2 | Controller development

Next, the feedback linearization technique will be used to develop the control input $\boldsymbol{\tau}_m$ for the system (15).

Computing the time derivative of the output function $\boldsymbol{y}$, we get

$$\dot{\boldsymbol{y}} = \alpha_1\dot{\tilde{\boldsymbol{q}}}_l + \alpha_2\dot{\tilde{\boldsymbol{q}}}_m + \beta_1\ddot{\tilde{\boldsymbol{q}}}_l + \beta_2\ddot{\tilde{\boldsymbol{q}}}_m \\ = \alpha_1\dot{\tilde{\boldsymbol{q}}}_l + \alpha_2\dot{\tilde{\boldsymbol{q}}}_m + \beta_1\boldsymbol{f}_2 + \beta_2\boldsymbol{f}_4 - \beta_2 J^{-1}\boldsymbol{\tau}_m, \qquad (18)$$

where $\boldsymbol{f}_2$ and $\boldsymbol{f}_4 \in \mathbb{R}^n$ are the second and the fourth components of the vector $\boldsymbol{f} \in \mathbb{R}^{4n}$ in (17), respectively. The relative degree is $r = n$, given the output function has n dimension, and after differentiating once with respect to time, the control input appears.

Multiplying both sides of the equation (18) by $J\beta_2^{-1}$, we obtain

$$J\beta_2^{-1}\dot{\boldsymbol{y}} = J\beta_2^{-1}\alpha_1\dot{\tilde{\boldsymbol{q}}}_l + J\beta_2^{-1}\alpha_2\dot{\tilde{\boldsymbol{q}}}_m + J\beta_2^{-1}\beta_1\boldsymbol{f}_2 + J\boldsymbol{f}_4 - \boldsymbol{\tau}_m. \qquad (19)$$

By defining β_2 as a diagonal and positive definite matrix, the product $J\beta_2^{-1}$ is positive definite. The control law can then be designed as

$$\boldsymbol{\tau}_m = J\beta_2^{-1}\alpha_1\dot{\tilde{\boldsymbol{q}}}_l + J\beta_2^{-1}\alpha_2\dot{\tilde{\boldsymbol{q}}}_m + J\beta_2^{-1}\beta_1\boldsymbol{f}_2 + J\boldsymbol{f}_4 + K_{po}\boldsymbol{y}, \tag{20}$$

where K_{po} is a positive definite diagonal matrix of $n \times n$ dimension. Under (20), the system becomes the next linear system:

$$J\beta_2^{-1}\dot{\boldsymbol{y}} = -K_{po}\boldsymbol{y}, \tag{21}$$

which is globally exponentially stable if K_{po} is selected as a diagonal positive definite matrix, as is proven in the following.

3.2.1 | Analysis of the closed-loop output dynamics

Notice that (21) can be written as

$$\dot{\boldsymbol{y}} = A\boldsymbol{y}, \tag{22}$$

where $A = -\beta_2 J^{-1}K_{po}$. It is important to remark that $J \in \mathbb{R}^{n\times n}$ and $\beta_2 \in \mathbb{R}^{n\times n}$ are diagonal and positive definite matrices. If K_{po} is a diagonal positive definite matrix, all the eigenvalues of A are negative and real. The solution of (22) is

$$\boldsymbol{y}(t) = e^{At}\boldsymbol{y}(0).$$

According to [76], if A is diagonal, and $-A = -A^T > 0$, then

$$e^{At} = \begin{bmatrix} e^{a_1 t} & 0 & \dots & 0 \\ 0 & e^{a_2 t} & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & e^{a_n t} \end{bmatrix},$$

which means that $\boldsymbol{y}(t)$ converges to zero as time t goes to infinity.

3.2.2 | Internal dynamics and desired rotor position $\boldsymbol{q}_{md}$

In order to obtain the normal form of the closed-loop systems, a change of variables should be computed [29].

The quantity of independent coordinates $\eta_i(\boldsymbol{x})$ which express the internal dynamics is

$$i = 4n - r = 4n - n = 3n.$$

These coordinates should satisfy [29]

$$\frac{\partial \eta_i(\boldsymbol{x})}{\partial \boldsymbol{x}} G(t, \boldsymbol{x}) = 0_{1\times 4n}.$$

Specifically, the following three states are used:

$$\boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{\eta}_1 \\ \boldsymbol{\eta}_2 \\ \boldsymbol{\eta}_3 \end{bmatrix} = \begin{bmatrix} \tilde{\boldsymbol{q}}_l \\ \dot{\tilde{\boldsymbol{q}}}_l \\ \tilde{\boldsymbol{q}}_m \end{bmatrix} \in \mathbb{R}^{3n},$$

which stand as the independent coordinates of the internal dynamics. Hence,

$$\boldsymbol{z} = \begin{bmatrix} \boldsymbol{\eta}_1 \\ \boldsymbol{\eta}_2 \\ \boldsymbol{\eta}_3 \\ \boldsymbol{y} \end{bmatrix} = T\boldsymbol{x} = \begin{bmatrix} I_n & 0 & 0 & 0 \\ 0 & I_n & 0 & 0 \\ 0 & 0 & I_n & 0 \\ \alpha_1 & \beta_1 & \alpha_2 & \beta_2 \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{q}}_l \\ \dot{\tilde{\boldsymbol{q}}}_l \\ \tilde{\boldsymbol{q}}_m \\ \dot{\tilde{\boldsymbol{q}}}_m \end{bmatrix} \tag{23}$$

is the transformation to obtain the closed-loop system.

The inverse transformation of (23) is given by

$$\boldsymbol{x} = T^{-1}\boldsymbol{z} = \begin{bmatrix} I_n & 0 & 0 & 0 \\ 0 & I_n & 0 & 0 \\ 0 & 0 & I_n & 0 \\ -\beta_2^{-1}\alpha_1 & -\beta_2^{-1}\beta_1 & -\beta_2^{-1}\alpha_2 & \beta_2^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_1 \\ \boldsymbol{\eta}_2 \\ \boldsymbol{\eta}_3 \\ \boldsymbol{y} \end{bmatrix}. \tag{24}$$

Equations (23) and (24) are useful to obtain the internal dynamics, which is given by

$$\dot{\boldsymbol{\eta}}_1 = \boldsymbol{\eta}_2, \tag{25}$$

$$\begin{aligned} \dot{\boldsymbol{\eta}}_2 = \ddot{\boldsymbol{q}}_{ld} + M(\boldsymbol{q}_l)^{-1}[C(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l)\dot{\boldsymbol{q}}_l \\ + F_{vl}\dot{\boldsymbol{q}}_l + K(\boldsymbol{q}_l - \boldsymbol{q}_{md}) + K\boldsymbol{\eta}_3], \end{aligned} \tag{26}$$

$$\dot{\boldsymbol{\eta}}_3 = -\beta_2^{-1}[\alpha_1\boldsymbol{\eta}_1 + \beta_1\boldsymbol{\eta}_2 + \alpha_2\boldsymbol{\eta}_3 - \boldsymbol{y}]. \tag{27}$$

Equation (26) is left-multiplied by $K^{-1}M(\boldsymbol{q}_l)$, and therefore (25)–(27) as

$$\dot{\boldsymbol{\eta}}_1 = \boldsymbol{\eta}_2, \tag{28}$$

$$\begin{aligned} K^{-1}M(\boldsymbol{q}_l)\dot{\boldsymbol{\eta}}_2 = K^{-1}M(\boldsymbol{q}_l)\ddot{\boldsymbol{q}}_{ld} + K^{-1}C(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l)\dot{\boldsymbol{q}}_l + K^{-1}F_{vl}\dot{\boldsymbol{q}}_l \\ + \boldsymbol{q}_l - \boldsymbol{q}_{md} + \boldsymbol{\eta}_3, \end{aligned} \tag{29}$$

$$\dot{\boldsymbol{\eta}}_3 = -\beta_2^{-1}[\alpha_1\boldsymbol{\eta}_1 + \beta_1\boldsymbol{\eta}_2 + \alpha_2\boldsymbol{\eta}_3 - \boldsymbol{y}]. \tag{30}$$

Notice that (29) has a similar structure to a rigid joint robot manipulator, taking into account $\boldsymbol{q}_{md}$ as the control input. The next desired rotor position $\boldsymbol{q}_{md}$ is proposed in order to stabilize the internal dynamics:

$$\begin{aligned} \boldsymbol{q}_{md} = K^{-1}M(\boldsymbol{q}_l)\ddot{\boldsymbol{q}}_{ld} + K^{-1}C(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l)\dot{\boldsymbol{q}}_l + K^{-1}F_{vl}\dot{\boldsymbol{q}}_l \\ + \boldsymbol{q}_l + K_d\boldsymbol{\eta}_2 + K_p\boldsymbol{\eta}_1, \end{aligned} \tag{31}$$

with K_p and $K_d \in \mathbb{R}^{n\times n}$ are diagonal positive definite matrices.

It is worth mentioning that the signal $\boldsymbol{q}_{\boldsymbol{md}}$ can be designed as any signal which stabilizes the internal dynamics (28) and (29).

The closed-loop system (normal form) can be rewritten by replacing (20) and (31) into (19) and (30), resulting in

$$\begin{bmatrix} \dot{\boldsymbol{\eta}}_1 \\ K^{-1}M(\boldsymbol{q}_l)\dot{\boldsymbol{\eta}}_2 \\ \dot{\boldsymbol{\eta}}_3 \end{bmatrix} = \begin{bmatrix} 0_{n\times n} & I_n & 0_{n\times n} \\ -K_p & -K_d & I_n \\ -\beta_2^{-1}\alpha_1 & -\beta_2^{-1}\beta_1 & -\beta_2^{-1}\alpha_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{\eta}_1 \\ \boldsymbol{\eta}_2 \\ \boldsymbol{\eta}_3 \end{bmatrix} + \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \beta_2^{-1} \end{bmatrix}\boldsymbol{y}, \quad (32)$$

$$J\beta_2^{-1}\dot{\boldsymbol{y}} = -K_{po}\boldsymbol{y}, \quad (33)$$

It is possible to demonstrate that there are sufficient conditions for the closed-loop system (32) and (33) to be locally exponentially stable, which consist of the matrix K_{po} being a symmetric positive definite, and of the matrix

$$A(t) = \begin{bmatrix} 0_{n\times n} & I_n & 0_{n\times n} \\ -M(\boldsymbol{q}_l)^{-1}KK_p & -M(\boldsymbol{q}_l)^{-1}KK_d & M(\boldsymbol{q}_l)^{-1}K \\ -\beta_2^{-1}\alpha_1 & -\beta_2^{-1}\beta_1 & -\beta_2^{-1}\alpha_2 \end{bmatrix}$$

satisfying the Lyapunov equation

$$-\dot{P}(t) = P(t)A(t) + A(t)^T P(t) + Q(t), \quad (34)$$

with $P(t)$ and $Q(t)$ being positive definite matrices, continuously differentiable, symmetric and bounded for all $t \geq 0$.

The conditions for the gains K_p, K_d and the parameters α_1, α_2, β_1, and β_2 for $A(t)$ to meet (34) will be developed in the next section.

3.3 | Comments on the designed controller

From equations (20) and (31), the overall controller is established as follows:

$$\boldsymbol{\tau}_m = J\beta_2^{-1}\alpha_1\dot{\tilde{\boldsymbol{q}}}_{\boldsymbol{l}} + J\beta_2^{-1}\alpha_2\dot{\tilde{\boldsymbol{q}}}_{\boldsymbol{m}} + J\beta_2^{-1}\beta_1\boldsymbol{f}_2 + J\boldsymbol{f}_4 + K_{po}\boldsymbol{y}, \quad (35)$$

$$\begin{aligned} \boldsymbol{q}_{\boldsymbol{md}} = {} & K^{-1}M(\boldsymbol{q}_l)\ddot{\boldsymbol{q}}_{\boldsymbol{ld}} + K^{-1}C(\boldsymbol{q}_l,\dot{\boldsymbol{q}}_l)\dot{\boldsymbol{q}}_l + K^{-1}F_{vl}\dot{\boldsymbol{q}}_l + \boldsymbol{q}_l \\ & + K_p\tilde{\boldsymbol{q}}_l + K_d\dot{\tilde{\boldsymbol{q}}}_l. \end{aligned} \quad (36)$$

with

$$\begin{aligned} \boldsymbol{f}_2 = {} & \ddot{\boldsymbol{q}}_{\boldsymbol{ld}} + M(\boldsymbol{q}_{\boldsymbol{ld}} - \tilde{\boldsymbol{q}}_l)^{-1}[C(\boldsymbol{q}_{\boldsymbol{ld}} - \tilde{\boldsymbol{q}}_l, \dot{\boldsymbol{q}}_{\boldsymbol{ld}} - \dot{\tilde{\boldsymbol{q}}}_l)(\dot{\boldsymbol{q}}_{\boldsymbol{ld}} - \dot{\tilde{\boldsymbol{q}}}_l) \\ & + F_{vl}(\dot{\boldsymbol{q}}_{\boldsymbol{ld}} - \dot{\tilde{\boldsymbol{q}}}_l) + K(\boldsymbol{q}_{\boldsymbol{ld}} - \tilde{\boldsymbol{q}}_l - \boldsymbol{q}_{\boldsymbol{md}} + \tilde{\boldsymbol{q}}_{\boldsymbol{m}})], \\ \boldsymbol{f}_4 = {} & \ddot{\boldsymbol{q}}_{\boldsymbol{md}} + J^{-1}[F_{vm}(\dot{\boldsymbol{q}}_{\boldsymbol{md}} - \tilde{\boldsymbol{q}}_{\boldsymbol{m}}) + K(\boldsymbol{q}_{\boldsymbol{md}} - \tilde{\boldsymbol{q}}_{\boldsymbol{m}} - \boldsymbol{q}_{\boldsymbol{ld}} + \tilde{\boldsymbol{q}}_l)]. \end{aligned}$$

Notice that the computing of the desired rotor velocity $\dot{\boldsymbol{q}}_{\boldsymbol{md}}$ requires the link acceleration $\ddot{\boldsymbol{q}}_l$, while the computing of the desired rotor acceleration $\ddot{\boldsymbol{q}}_{\boldsymbol{md}}$ requires the link jerk $\dddot{\boldsymbol{q}}_l$. Nevertheless, under the assumption that the model parameters are known exactly $\ddot{\boldsymbol{q}}_l$ can be obtained from equation (1) given the matrix $M(\boldsymbol{q}_l)$ is invertible. At the same time, $\dddot{\boldsymbol{q}}_l$ is obtained by calculating the time derivative of $\ddot{\boldsymbol{q}}_l$, as follows

$$\begin{aligned} \dddot{\boldsymbol{q}}_l = {} & -\dot{M}^{-1}[C(\boldsymbol{q}_l,\dot{\boldsymbol{q}}_l)\dot{\boldsymbol{q}}_l + F_{vl}\dot{\boldsymbol{q}}_l + K(\boldsymbol{q}_l - \boldsymbol{q}_{\boldsymbol{m}})] \\ & - M(\boldsymbol{q}_l)^{-1}[\dot{C}\dot{\boldsymbol{q}}_l + C(\boldsymbol{q}_l,\dot{\boldsymbol{q}}_l)\ddot{\boldsymbol{q}}_l + F_{vl}\ddot{\boldsymbol{q}}_l + K(\dot{\boldsymbol{q}}_l - \dot{\boldsymbol{q}}_{\boldsymbol{m}})], \end{aligned}$$

with $\dot{M}^{-1} = \frac{dM(\boldsymbol{q}_l)^{-1}}{dt}$ and $\dot{C} = \frac{dC(\boldsymbol{q}_l,\dot{\boldsymbol{q}}_l)}{dt}$. This is possible because the design assumes all the robot manipulator parameters to be known.

An important observation is that the implementation of the feedback linearization-based controller (35), with $\boldsymbol{q}_{\boldsymbol{md}}$ in (36), requires that the desired link position $\boldsymbol{q}_{\boldsymbol{ld}}(t)$ to be at least four times continuously differentiable. Another observation is that the desired rotor velocity $\dot{\boldsymbol{q}}_{\boldsymbol{md}}$ and the desired rotor acceleration $\ddot{\boldsymbol{q}}_{\boldsymbol{md}}$ cannot be estimated as discussed above if the model parameters are unknown or uncertain. In this case, other methods should be used, such as numerical differentiation [65], [77].

4 | DESIGN OF THE COMBINED ADAPTIVE CONTROLLER

Notice that the controller in equations (35) and (36) is a model-based controller that requires the knowledge of the parameters of the flexible joint robot, which in some cases, are not precisely known. A combined adaptive controller is developed in this section to avoid this problem. Note that two adaptive techniques are used, adaptive neural networks to stabilize the output dynamics and adaptive regressor-based controller to compute the desired rotor position. This new controller is based on the structure of the model-based controller (35) and (36).

4.1 | Adaptive neural network controller

Notice that the output dynamics can be expressed from (19) as

$$J\beta_2^{-1}\dot{\boldsymbol{y}} = F(x) - \boldsymbol{\tau}_m, \quad (37)$$

where

$$F(\boldsymbol{x}) = J\beta_2^{-1}(\alpha_1\dot{\tilde{\boldsymbol{q}}}_l + \alpha_2\dot{\tilde{\boldsymbol{q}}}_{\boldsymbol{m}} + \beta_1\boldsymbol{f}_2 + \beta_2\boldsymbol{f}_4).$$

By using the universal approximation theorem [46], the function $F(\boldsymbol{x})$ can be approximated with a neural network as

$$F(\boldsymbol{x}) = W^T\boldsymbol{\sigma}(V^T\bar{\boldsymbol{x}}_f) + \epsilon$$

where $W \in \mathbb{R}^{L\times n}$ is the matrix of ideal output weights of the neural network, $V \in \mathbb{R}^{L\times k}$ is the constant input weights matrix, whose values are assigned randomly. The augmented input vector $\bar{\boldsymbol{x}}_f \in \mathbb{R}^k$ is formed by $[\boldsymbol{x}_f^T \; 1]^T$, with $\boldsymbol{x}_f = [\tilde{\boldsymbol{q}}_l{}^T \; \tilde{\boldsymbol{q}}_m{}^T \; \dot{\tilde{\boldsymbol{q}}}_l{}^T \; \dot{\tilde{\boldsymbol{q}}}_m{}^T \; \ddot{\boldsymbol{q}}_{ld}{}^T \; \ddot{\boldsymbol{q}}_{md}{}^T]^T \in \mathbb{R}^{k-1}$ as the input vector. The vector $\epsilon \in \mathbb{R}^n$ is the approximation error of the neural network, the number of elements of the input vector is $k-1=6n$, $L \in \mathbb{N}$ is the number of neurons, and $\boldsymbol{\sigma} \in \mathbb{R}^L$ is the vector of activation functions.

The adaptive neural network controller is

$$\boldsymbol{\tau}_m = \hat{F}(\boldsymbol{x}) + K_{po}\boldsymbol{y} + \Delta\mathbf{sign}(\boldsymbol{y}), \tag{38}$$

where

$$\hat{F}(\boldsymbol{x}) = \hat{W}^T\boldsymbol{\sigma}(V^T\bar{\boldsymbol{x}}_f) = \hat{W}^T\mathbf{tanh}(V^T\bar{\boldsymbol{x}}_f),$$

with $K_{po} \in \mathbb{R}^{n\times n}$ and $\Delta \in \mathbb{R}^{n\times n}$ diagonal positive definite matrices, $\hat{W} \in \mathbb{R}^{L\times n}$ the estimated output weights, and the function

$$\mathbf{sign}(\boldsymbol{y}) = \begin{bmatrix} \mathrm{sign}(y_1) \\ \vdots \\ \mathrm{sign}(y_n) \end{bmatrix},$$

being

$$\mathrm{sign}(x) = \begin{cases} 1, & \text{for} \quad x > 0, \\ 0, & \text{for} \quad x = 0, \\ -1, & \text{for} \quad x < 0. \end{cases}$$

with $x \in \mathbb{R}$.

The adaptation law for updating $\hat{W} \in \mathbb{R}^{L\times n}$ is defined as

$$\dot{\hat{W}} = N\boldsymbol{\sigma}(V^T\bar{\boldsymbol{x}}_f)\boldsymbol{y}^T\alpha - N\hat{W}\kappa\|\boldsymbol{y}\|, \tag{39}$$

with $N \in \mathbb{R}^{L\times L}$ and $\alpha \in \mathbb{R}^{n\times n}$ diagonal positive definite matrices, and $\kappa \in \mathbb{R}$ a positive constant.

4.2 | Desired rotor position (adaptive internal dynamics stabilizer)

Notice that the internal dynamics (28)-(30) can be expressed as

$$\dot{\boldsymbol{\eta}}_1 = \boldsymbol{\eta}_2, \tag{40}$$

$$K^{-1}M(\boldsymbol{q}_l)\dot{\boldsymbol{\eta}}_2 = \Phi(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l, \ddot{\boldsymbol{q}}_{ld})\boldsymbol{\theta} + \boldsymbol{q}_l - \boldsymbol{q}_{md} + \boldsymbol{\eta}_3, \tag{41}$$

$$\dot{\boldsymbol{\eta}}_3 = -\beta_2^{-1}[\alpha_1\boldsymbol{\eta}_1 + \beta_1\boldsymbol{\eta}_2 + \alpha_2\boldsymbol{\eta}_3 - \boldsymbol{y}], \tag{42}$$

with $\Phi(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l, \ddot{\boldsymbol{q}}_{ld}) \in \mathbb{R}^{n\times p}$ the system regressor, $\boldsymbol{\theta} \in \mathbb{R}^p$ is the lumped parameter vector, n is the number of links, and p the number of parameters. Notice that equation (41) is obtained by using the linear in the parameter assumption (12).

The desired rotor position $\boldsymbol{q}_{md}$ proposed to stabilize the internal dynamics (40)–(42) is

$$\boldsymbol{q}_{md} = \Phi(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l, \ddot{\boldsymbol{q}}_{ld})\hat{\boldsymbol{\theta}} + \boldsymbol{q}_l + K_p\tilde{\boldsymbol{q}}_l + K_d\dot{\tilde{\boldsymbol{q}}}_l, \tag{43}$$

where $\hat{\boldsymbol{\theta}} \in \mathbb{R}^p$ is the vector of estimated parameters, which are obtained with the adaptation law

$$\dot{\hat{\boldsymbol{\theta}}} = \Gamma\Phi(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l, \ddot{\boldsymbol{q}}_{ld})[\gamma K\tilde{\boldsymbol{q}}_l + K\dot{\tilde{\boldsymbol{q}}}_l], \tag{44}$$

with $\Gamma \in \mathbb{R}^{p\times p}$ a diagonal positive definite matrix. It is worthwhile to notice that the implementation of the adaptive neural network $\boldsymbol{\tau}_m$ in (38), relies on the assumption that the signals

$$\dot{\boldsymbol{q}}_{md} = \Phi(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l, \ddot{\boldsymbol{q}}_{ld})\dot{\hat{\boldsymbol{\theta}}} + \dot{\Phi}(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l, \ddot{\boldsymbol{q}}_l, \dddot{\boldsymbol{q}}_{ld})\hat{\boldsymbol{\theta}} + \dot{\boldsymbol{q}}_l + K_p\dot{\tilde{\boldsymbol{q}}}_l + K_d\ddot{\tilde{\boldsymbol{q}}}_l,$$

and

$$\begin{aligned} \ddot{\boldsymbol{q}}_{md} = {} & \Phi(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l, \ddot{\boldsymbol{q}}_{ld})\ddot{\hat{\boldsymbol{\theta}}} + 2\dot{\Phi}(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l, \ddot{\boldsymbol{q}}_l, \dddot{\boldsymbol{q}}_{ld})\dot{\hat{\boldsymbol{\theta}}} \\ & + \ddot{\Phi}\big(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l, \ddot{\boldsymbol{q}}_l, \dddot{\boldsymbol{q}}_l, \boldsymbol{q}_{ld}^{(4)}\big)\hat{\boldsymbol{\theta}} + \ddot{\boldsymbol{q}}_l + K_p\ddot{\tilde{\boldsymbol{q}}}_l + K_d\dddot{\tilde{\boldsymbol{q}}}_l, \end{aligned}$$

where

$$\begin{aligned} \ddot{\hat{\boldsymbol{\theta}}} = {} & \Gamma\dot{\Phi}\big(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l, \ddot{\boldsymbol{q}}_l, \dddot{\boldsymbol{q}}_{ld}\big)[\gamma K\tilde{\boldsymbol{q}}_l + K\dot{\tilde{\boldsymbol{q}}}_l] \\ & + \Gamma\Phi(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l, \ddot{\boldsymbol{q}}_{ld})[\gamma K\dot{\tilde{\boldsymbol{q}}}_l + K\ddot{\tilde{\boldsymbol{q}}}_l], \end{aligned}$$

are available. These signals can be indirectly estimated by numerical differentiation. The adaptation laws (39) and (44) are designed to match the analysis of the closed-loop trajectories to be developed.

By substituting the equations (38) and (39) into (37) and the equations (43) and (44) into (41), the overall closed-loop system is obtained as

$$\dot{\boldsymbol{\eta}}_1 = \boldsymbol{\eta}_2, \tag{45}$$

$$\dot{\boldsymbol{\eta}}_2 = M(\boldsymbol{q}_{ld} - \boldsymbol{\eta}_1)^{-1}K(-K_p\boldsymbol{\eta}_1 - K_d\boldsymbol{\eta}_2 + \boldsymbol{\eta}_3 + \Phi(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l, \ddot{\boldsymbol{q}}_{ld})\tilde{\boldsymbol{\theta}}), \tag{46}$$

$$\dot{\boldsymbol{\eta}}_3 = -\beta_2^{-1}\alpha_1\boldsymbol{\eta}_1 - \beta_2^{-1}\beta_1\boldsymbol{\eta}_2 - \beta_2^{-1}\alpha_2\boldsymbol{\eta}_3 + \beta_2^{-1}\boldsymbol{y}, \tag{47}$$

$$\dot{\tilde{\boldsymbol{\theta}}} = -\Gamma\Phi(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l, \ddot{\boldsymbol{q}}_{ld})[\gamma K\tilde{\boldsymbol{q}}_l + K\dot{\tilde{\boldsymbol{q}}}_l], \tag{48}$$

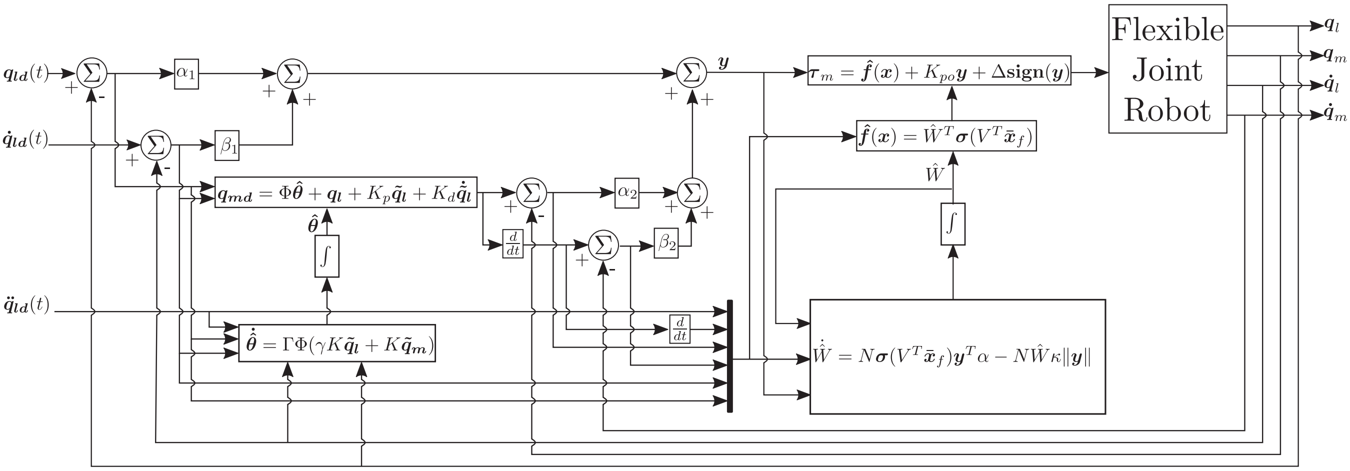
$q_{ld}(t)$
$\dot{q}_{ld}(t)$
$\ddot{q}_{ld}(t)$
Σ
$+$
$-$
α_1
β_1
α_2
β_2
$\boldsymbol{y}$
$\tau_m = \hat{\boldsymbol{f}}(\boldsymbol{x}) + K_{po}\boldsymbol{y} + \Delta\mathbf{sign}(\boldsymbol{y})$
$\hat{\boldsymbol{f}}(\boldsymbol{x}) = \hat{W}^T\boldsymbol{\sigma}(V^T\bar{\boldsymbol{x}}_f)$
$\hat{W}$
$\int$
$\dot{\hat{W}} = N\boldsymbol{\sigma}(V^T\bar{\boldsymbol{x}}_f)\boldsymbol{y}^T\alpha - N\hat{W}\kappa\|\boldsymbol{y}\|$
$\boldsymbol{q_{md}} = \Phi\hat{\boldsymbol{\theta}} + \boldsymbol{q_l} + K_p\tilde{\boldsymbol{q}}_{\boldsymbol{l}} + K_d\dot{\tilde{\boldsymbol{q}}}_{\boldsymbol{l}}$
$\hat{\boldsymbol{\theta}}$
$\dot{\hat{\boldsymbol{\theta}}} = \Gamma\Phi(\gamma K\tilde{\boldsymbol{q}}_{\boldsymbol{l}} + K\tilde{\boldsymbol{q}}_{\boldsymbol{m}})$
$\frac{d}{dt}$
Flexible Joint Robot
$\boldsymbol{q}_l$
$\boldsymbol{q}_m$
$\dot{\boldsymbol{q}}_l$
$\dot{\boldsymbol{q}}_m$

FIGURE 2 Block diagram of the novel adaptive controller (38–44)

$$J\beta_2^{-1}\dot{\boldsymbol{y}} = \tilde{W}\boldsymbol{\sigma}(V^T\bar{x}_f) - K_{po}\boldsymbol{y} - \Delta\mathbf{sign}(\boldsymbol{y}), \tag{49}$$

$$\dot{\tilde{W}} = -N\boldsymbol{\sigma}(V^T\bar{x}_f)\boldsymbol{y}^T\alpha + N\hat{W}\kappa\|\boldsymbol{y}\|, \tag{50}$$

where $\tilde{W} = W - \hat{W}$ is the matrix of estimated output weight errors, and $\tilde{\boldsymbol{\theta}} = \boldsymbol{\theta} - \hat{\boldsymbol{\theta}}$ is the vector of estimated parameter errors.

The block diagram of the novel adaptive controller (38), (39), (43), and (44) is presented in Figure 2.

4.3 | Analysis of the closed-loop trajectories

Taking into account the controller derived in this section, a theorem can be proposed:

Theorem 1. *Consider the dynamic model of an FJR defined in (1) and (2). The combined adaptive controller, expressed in equations (38), (39), (43), and (44), guarantees that the trajectories $\tilde{\boldsymbol{q}}_{\boldsymbol{l}}(t)$, $\dot{\tilde{\boldsymbol{q}}}_{\boldsymbol{l}}(t)$, $\tilde{\boldsymbol{q}}_{\boldsymbol{m}}(t)$, and $\dot{\tilde{\boldsymbol{q}}}_{\boldsymbol{m}}(t)$ tend to zero as time t increases for a compact set of initial conditions contained inside a local ball B_r of radius $r > 0$.*

Proof. In order to prove the theorem, the closed-loop system trajectories must be analyzed. The analysis consists of two parts. First, the output dynamics are analyzed. Then, the zero dynamics must be studied, which consists in substituting $\boldsymbol{y} = 0$ in the internal dynamics (45)–(48) and performing a Lyapunov-like analysis.

For the output dynamics, represented by equations (49) and (50), the next positive definite function is proposed:

$$V_1 = \frac{1}{2}\boldsymbol{y}^T\alpha J\beta_2^{-1}\boldsymbol{y} + \frac{1}{2}Tr(\tilde{W}^T N^{-1}\tilde{W}), \tag{51}$$

where $Tr(A)$ is the matrix trace. Notice that $J \in \mathbb{R}^{n\times n}$ is a diagonal positive definite matrix. Therefore, to guarantee that (51) be positive definite, it is sufficient to choose $\beta_2 \in \mathbb{R}^{n\times n}$ and $\alpha \in \mathbb{R}^{n\times n}$ as diagonal positive definite matrices.

Computing the time derivative of (51) along the trajectories, we get

$$\begin{aligned}\dot{V}_1 &= \boldsymbol{y}^T\alpha J\beta_2^{-1}\dot{\boldsymbol{y}} + Tr(\tilde{W}^T N^{-1}\dot{\tilde{W}})\\ &= \boldsymbol{y}^T\alpha(\tilde{W}^T\boldsymbol{\sigma}(V^T\bar{\boldsymbol{x}}_f) - K_{po}\boldsymbol{y} - \Delta\mathbf{sign}(\boldsymbol{y}) + \epsilon)\\ &\quad + Tr(-\tilde{W}^T\boldsymbol{\sigma}(V^T\bar{\boldsymbol{x}}_f)\boldsymbol{y}^T\alpha + \tilde{W}^T\hat{W}\kappa\|\boldsymbol{y}\|).\end{aligned}$$

By using the property (5)

$$\underbrace{\boldsymbol{y}^T\alpha}_{\boldsymbol{b}^T}\ \underbrace{\tilde{W}^T\boldsymbol{\sigma}(V^T\bar{\boldsymbol{x}}_f)}_{\boldsymbol{a}} = Tr(\underbrace{\tilde{W}^T\boldsymbol{\sigma}(V^T\bar{\boldsymbol{x}}_f)}_{\boldsymbol{a}}\ \underbrace{\boldsymbol{y}^T\alpha}_{\boldsymbol{b}^T}),$$

we get

$$\begin{aligned}\dot{V}_1 &= -\boldsymbol{y}^T\alpha K_{po}\boldsymbol{y} - \boldsymbol{y}^T\alpha\Delta\mathbf{sign}(\boldsymbol{y}) + \boldsymbol{y}^T\alpha\epsilon\\ &\quad + \kappa\|\boldsymbol{y}\|\,Tr(\tilde{W}^T(\tilde{W} - \tilde{W})).\end{aligned}$$

By using the Frobenious inner product defined in (7), we get

$$\begin{aligned}Tr(\tilde{W}^T(W - \tilde{W})) &= \langle\tilde{W}, W\rangle_F - \|\tilde{W}\|_F^2\\ &\leq \|\tilde{W}\|_F\|W\|_F - \|\tilde{W}\|_F^2,\end{aligned}$$

so that $\dot{V}_1$ can be upper bounded as

$$\begin{aligned}\dot{V}_1 \leq{}& -\lambda_{\max}\{\alpha\}\lambda_{\max}\{K_{po}\}\|\boldsymbol{y}\|^2\\ &- \kappa\|\boldsymbol{y}\|\left(\|\tilde{W}\|_F^2 - \|\tilde{W}\|_F^2\|W\|_F^2\right)\\ &+ (\lambda_{\max}\{\alpha\}(\lambda_{\max}\{\epsilon\} - \lambda_{\max}\{\Delta\}))\|\boldsymbol{y}\|\end{aligned}$$

$$\le -\lambda_{\max}\{\alpha\}\lambda_{\max}\{K_{po}\}\|\boldsymbol{y}\|^2 - \kappa\|\boldsymbol{y}\|\left(\|\tilde{W}\|_F - \|W\|_F\right)^2$$
$$-\left(\lambda_{\max}\{\alpha\}\lambda_{\max}\{\Delta\} - \lambda_{\max}\{\alpha\}\lambda_{\max}\{\epsilon\} - \kappa\frac{\|W\|_F^2}{4}\right)\|\boldsymbol{y}\|.$$

Therefore, to guarantee that $\dot{V}_1 \le 0$, the next conditions have to be met

$$\alpha = \alpha^T > 0, \tag{52}$$

$$K_{po} = K_{po}^T > 0, \tag{53}$$

$$\lambda_{\max}\{\Delta\} > \frac{\lambda_{\max}\{\alpha\}\lambda_{\max}\{\epsilon\} + \kappa\frac{\|W\|_F^2}{4}}{\lambda_{\max}\{\alpha\}}. \tag{54}$$

Because $\dot{V}_1 \le 0$, the states $\boldsymbol{y}(t)$ and $\tilde{W}(t)$ are bounded. In consequence, $\dot{\boldsymbol{y}}(t)$ is also bounded. Notice that the next inequality holds:

$$\dot{V}_1 \le -\lambda_{\max}\{\alpha\}\lambda_{\max}\{K_{po}\}\|\boldsymbol{y}\|^2. \tag{55}$$

By integrating both sides of (55) from $t = 0$ to $t = \infty$, we have that

$$-V_1(0) \le -\lambda_{\max}\{\alpha\}\lambda_{\max}\{K_{po}\}\int_0^\infty \|\boldsymbol{y}(t)\|^2 dt - V_1(\infty).$$

Given that V_1 is positive definite, $V_1(\infty)$ is either positive or zero, and we finally show that

$$-V_1(0) \le -\lambda_{\max}\{\alpha\}\lambda_{\max}\{K_{po}\}\int_0^\infty \|\boldsymbol{y}(t)\|^2,$$

$$\int_0^\infty \|\boldsymbol{y}(t)\|^2 dt \le \frac{V_1(0)}{\lambda_{\max}\{\alpha\}\lambda_{\max}\{K_{po}\}}.$$

There are sufficient conditions to invoke the Barbalat's lemma [78] to conclude that

$$\lim_{t\to\infty} \boldsymbol{y}(t) = 0.$$

Now, to analyze the internal dynamics, it is necessary to obtain the zero dynamics, by making $\boldsymbol{y} = 0$, giving as result

$$\dot{\boldsymbol{\eta}}_1 = \boldsymbol{\eta}_2, \tag{56}$$

$$M(\boldsymbol{q}_{ld} - \boldsymbol{\eta}_1)\dot{\boldsymbol{\eta}}_2 = -KK_p\boldsymbol{\eta}_1 - KK_d\boldsymbol{\eta}_2 + K\boldsymbol{\eta}_3 + K\Phi(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l, \ddot{\boldsymbol{q}}_{ld})\tilde{\boldsymbol{\theta}}, \tag{57}$$

$$\dot{\boldsymbol{\eta}}_3 = -\beta_2^{-1}\alpha_1\boldsymbol{\eta}_1 - \beta_2^{-1}\beta_1\boldsymbol{\eta}_2 - \beta_2^{-1}\alpha_2\boldsymbol{\eta}_3. \tag{58}$$

The next positive definite function is proposed:

$$V_2 = \frac{1}{2}\begin{bmatrix}\boldsymbol{\eta}_1\\ \boldsymbol{\eta}_2\\ \boldsymbol{\eta}_3\end{bmatrix}^T \begin{bmatrix} KK_p + \gamma KK_d & \gamma M & 0\\ \gamma M & M & 0\\ 0 & 0 & K\beta_1^{-1}\beta_2\end{bmatrix}\begin{bmatrix}\boldsymbol{\eta}_1\\ \boldsymbol{\eta}_2\\ \boldsymbol{\eta}_3\end{bmatrix}$$
$$+ \frac{1}{2}\tilde{\boldsymbol{\theta}}^T\Gamma^{-1}\tilde{\boldsymbol{\theta}}, \tag{59}$$

where $M = M(\boldsymbol{q}_{ld} - \boldsymbol{\eta}_1)$ by simplicity.

Notice that (59) satisfies the next inequality

$$V_2 \ge \frac{1}{2}\begin{bmatrix}\|\boldsymbol{\eta}_1\|\\ \|\boldsymbol{\eta}_2\|\\ \|\boldsymbol{\eta}_3\|\end{bmatrix}^T P \begin{bmatrix}\|\boldsymbol{\eta}_1\|\\ \|\boldsymbol{\eta}_2\|\\ \|\boldsymbol{\eta}_3\|\end{bmatrix} + \frac{1}{2}\tilde{\boldsymbol{\theta}}^T\Gamma^{-1}\tilde{\boldsymbol{\theta}}, \tag{60}$$

where

$$P = \begin{bmatrix}\lambda_{\min}\{K\}\lambda_{\min}\{K_p\} + \gamma\lambda_{\min}\{K\}\lambda_{\min}\{K_d\} & -\gamma\lambda_{\max}\{M\} & 0\\ -\gamma\lambda_{\max}\{M\} & \lambda_{\min}\{M\} & 0\\ 0 & 0 & \lambda_{\min}\{K\}\lambda_{\min}\{\beta_2\}/\lambda_{\min}\{\beta_1\}\end{bmatrix}.$$

Notice that to get the lower bound expressed in equation (60), the properties (3), (4), (8) are used.

To guarantee that P be positive definite, the next condition has to be met

$$\sigma_1 - \sigma_2 < \gamma < \sigma_1 + \sigma_2. \tag{61}$$

where

$$\sigma_1 = \frac{\lambda_{\min}\{K\}\lambda_{\min}\{K_d\}\lambda_{\min}\{M\}}{2\lambda_{\max}\{M\}^2},$$

$$\sigma_2 = \frac{\sqrt{\lambda_{\min}\{K\}^2\lambda_{\min}\{K_d\}^2\lambda_{\min}\{M\}^2 + 4\lambda_{\max}\{M\}^2\lambda_{\min}\{M\}\lambda_{\min}\{K\}\lambda_{\min}\{K_p\}}}{2\lambda_{\max}\{M\}^2}.$$

Computing the time derivative of (59) along the trajectories (56)-(58), we get

$$\dot{V}_2 = \boldsymbol{\eta}_1^T(KK_p + \gamma K_d)\boldsymbol{\eta}_2 + \boldsymbol{\eta}_2^T\dot{M}\boldsymbol{\eta}_2 + \boldsymbol{\eta}_2^T(-KK_p\boldsymbol{\eta}_1 - KK_d\boldsymbol{\eta}_2$$
$$+ K\boldsymbol{\eta}_3 + K\Phi(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l, \ddot{\boldsymbol{q}}_{ld})\tilde{\boldsymbol{\theta}}) + \gamma\boldsymbol{\eta}_1^T\dot{M}\boldsymbol{\eta}_2$$
$$+ \boldsymbol{\eta}_3^T K\beta_1^{-1}(-\alpha_1\boldsymbol{\eta}_1 - \beta_1\boldsymbol{\eta}_2 - \alpha_2\boldsymbol{\eta}_3) + \gamma\boldsymbol{\eta}_1^T(-KK_p\boldsymbol{\eta}_1$$
$$- KK_d\boldsymbol{\eta}_2 + K\boldsymbol{\eta}_3 + K\Phi(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l, \ddot{\boldsymbol{q}}_{ld})\tilde{\boldsymbol{\theta}}) + \dot{\tilde{\boldsymbol{\theta}}}^T\Gamma\tilde{\boldsymbol{\theta}}$$
$$+\gamma\boldsymbol{\eta}_2^T M\boldsymbol{\eta}_2 \tag{62}$$

By substituting the adaptation law (48) into (62), defining $\alpha_1 = \gamma\beta_1$, and simplifying, we get

$$\dot{V}_2 = -\gamma\boldsymbol{\eta}_1^T K K_p \boldsymbol{\eta}_1 - \boldsymbol{\eta}_2^T \left(K K_d - \gamma M - \frac{1}{2}\dot{M}\right)\boldsymbol{\eta}_2 + \boldsymbol{\eta}_3^T K \beta_2^{-1}\alpha_2 \boldsymbol{\eta}_3 + \gamma\boldsymbol{\eta}_1^T \dot{M}\boldsymbol{\eta}_2. \tag{63}$$

Notice that (63) can be upper bounded by

$$\dot{V}_2 \leq -\begin{bmatrix} \|\boldsymbol{\eta}_1\| \\ \|\boldsymbol{\eta}_2\| \\ \|\boldsymbol{\eta}_3\| \end{bmatrix}^T Q \begin{bmatrix} \|\boldsymbol{\eta}_1\| \\ \|\boldsymbol{\eta}_2\| \\ \|\boldsymbol{\eta}_3\| \end{bmatrix}, \tag{64}$$

with

$$Q = \begin{bmatrix} \gamma\lambda_{\min}\{K\}\lambda_{\min}\{K_p\} & -\gamma k_{C_1}\|\dot{\boldsymbol{q}}_{ld}\| & 0 \\ -\gamma k_{C_1}\|\dot{\boldsymbol{q}}_{ld}\| & \sigma_3 & 0 \\ 0 & 0 & \dfrac{\lambda_{\min}\{K\}\lambda_{\min}\{\alpha_2\}}{\lambda_{\max}\{\beta_2\}} \end{bmatrix}, \tag{65}$$

and

$$\sigma_3 = \lambda_{\min}\{K\}\lambda_{\min}\{K_d\} - \gamma\lambda_{\max}\{M\} - k_{C_1}\|\dot{\boldsymbol{q}}_{ld}\| - (k_{C_1} + 2\gamma k_{C_1})\|\boldsymbol{\eta}\|$$

where the properties (9), (10), (11), the inequality

$$k_{C_1}\|\boldsymbol{\eta}_2\| + 2\gamma k_{C_1}\|\boldsymbol{\eta}_1\| \leq (k_{C_1} + 2\gamma k_{C_1})\|\boldsymbol{\eta}\|,$$

and the triangle inequality

$$k_{C_1}\|\boldsymbol{\eta}_2\|\|\dot{\boldsymbol{q}}_{ld} - \boldsymbol{\eta}_2\| \leq k_{C_1}\|\boldsymbol{\eta}_2\|\|\dot{\boldsymbol{q}}_{ld}\| + k_{C_1}\|\boldsymbol{\eta}_2\|^2,$$

with

$$\|\boldsymbol{\eta}\| = \left\|\begin{matrix} \boldsymbol{\eta}_1 \\ \boldsymbol{\eta}_2 \\ \boldsymbol{\eta}_3 \end{matrix}\right\|,$$

are used.

Notice that there exists a ball B_r of radius $r > 0$ defined as

$$B_r = \left\{ \boldsymbol{\eta}_1, \boldsymbol{\eta}_2, \boldsymbol{\eta}_3 \in \mathbb{R}^n : \left\|\begin{matrix} \boldsymbol{\eta}_1 \\ \boldsymbol{\eta}_2 \\ \boldsymbol{\eta}_3 \end{matrix}\right\| < r \right\}, \tag{66}$$

on which $\dot{V}_2 \leq 0$.

The radius r can be defined as

$$r := \frac{\lambda_{\min}\{K\}\lambda_{\min}\{K_d\} - \gamma\lambda_{\max}\{M\} - k_{C_1}\|\dot{\boldsymbol{q}}_{ld}\|}{k_{C_1} + 2\gamma k_{C_1}} - \frac{\gamma k_{C_1}^2 \|\dot{\boldsymbol{q}}_{ld}\|^2}{(k_{C_1} + 2\gamma k_{C_1})\lambda_{\min}\{K\}\lambda_{\min}\{K_p\}}. \tag{67}$$

If the conditions

$$\gamma < \frac{\lambda_{\min}\{K\}^2\lambda_{\min}\{K_d\}\lambda_{\min}\{K_p\} - k_{C_1}\lambda_{\min}\{K\}\lambda_{\min}\{K_p\}\|\dot{\boldsymbol{q}}_{ld}\|}{\lambda_{\max}\{M\}\lambda_{\min}\{K\}\lambda_{\min}\{K_p\} + k_{C_1}^2\|\dot{\boldsymbol{q}}_{ld}\|^2}, \tag{68}$$

$$\lambda_{\min}\{K_d\} > \frac{k_{C_1}^2\|\dot{\boldsymbol{q}}_{ld}\|}{\lambda_{\min}\{K\}}, \tag{69}$$

are satisfied, the radius r in (67) is ensured to be positive and Q in (65) is positive definite.

The existence of gains such that V_2 in (59) is positive definite and $\dot{V}_2$ in (63) is locally negative semi-definite is guaranteed by satisfying the conditions (61), (68), and (69). By using the theorem of Rayleigh–Ritz [75], the upper bound (64) can be upper bounded by

$$\dot{V}_2 \leq -\lambda_{\min}\{Q\}\|\boldsymbol{\eta}\|^2. \tag{70}$$

Then, we obtain the conclusion that $\boldsymbol{\eta}(t)$, $\dot{\boldsymbol{\eta}}(t)$ and $\tilde{\boldsymbol{\theta}}(t)$ are bounded. By integrating from $t = 0$ to $t = \infty$ both sides of (70), we obtain

$$-V_2(0) \leq -\lambda_{\min}\{Q\}\int_0^\infty \|\boldsymbol{\eta}(t)\|^2 dt - V_2(\infty).$$

Given that V_2 is positive definite, $V_2(\infty)$ is either positive or zero, and we finally get

$$\int_0^\infty \|\boldsymbol{\eta}(t)\|^2 dt \leq \frac{V_2(0)}{\lambda_{\min}\{Q\}}.$$

Then, there are sufficient conditions to invoke Barbalat's lemma [78] to conclude that, for a set of initial conditions $[\boldsymbol{\eta}_1(0)^T\ \boldsymbol{\eta}_2(0)^T\ \boldsymbol{\eta}_3(0)^T]^T$ inside of the ball B_r, the limit

$$\lim_{t\to\infty}\begin{bmatrix} \boldsymbol{\eta}_1(t) \\ \boldsymbol{\eta}_2(t) \\ \boldsymbol{\eta}_3(t) \end{bmatrix} = 0$$

is satisfied.

Since $\boldsymbol{y}(t)$ converges to zero and the continuity of (45)-(48), the solution of the internal dynamics (45), (46), and (47) will tend to zero as time t goes to infinity.

Since the solutions $\boldsymbol{y}(t)$ and $[\boldsymbol{\eta}_1^T(t)\ \boldsymbol{\eta}_2^T(t)\ \boldsymbol{\eta}_3^T(t)]^T$ tend to zero as time t increases, and the transformation (23) is invert-

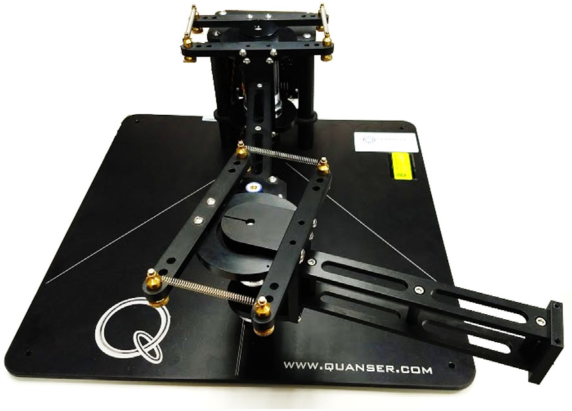

FIGURE 3 Flexible joint robot used in the real-time experiments

ible, the limit

$$\lim_{t\to\infty}\begin{bmatrix}\tilde{\boldsymbol{q}}_{\boldsymbol{l}}(t)\\ \dot{\tilde{\boldsymbol{q}}}_{\boldsymbol{l}}(t)\\ \tilde{\boldsymbol{q}}_{\boldsymbol{m}}(t)\\ \dot{\tilde{\boldsymbol{q}}}_{\boldsymbol{m}}(t)\end{bmatrix}=0$$

is concluded for a compact set of initial conditions $[\tilde{\boldsymbol{q}}_{\boldsymbol{l}}(0)^T\ \dot{\tilde{\boldsymbol{q}}}_{\boldsymbol{l}}(0)^T\ \tilde{\boldsymbol{q}}_{\boldsymbol{m}}(0)^T\ \dot{\tilde{\boldsymbol{q}}}_{\boldsymbol{m}}(0)^T]^T$ inside of the ball B_r defined in (66). □

5 | EXPERIMENTAL RESULTS

5.1 | Experimental platform

The two-link flexible joint robot of Quanser [79] shown in Figure 3 is used to carry out the real-time experiments. The platform has four optical encoders of 4096 counts per revolution in quadrature. The system is equipped with a DC motor (shoulder) Maxon 273759 connected to a harmonic drive with zero-backlash and a gear ratio of 100:1, and a DC motor (elbow) Maxon 118752 connected to a harmonic drive with zero-backlash and a gear ratio of 50:1. The first and second links have a gear ratio with respect to the encoders of 6.4:1. Each link has a workspace limitation of ±90 [deg] and counts with limit switches to protect the platform if the position limit is exceeded. The motors are powered by the Quanser AMPAQ current amplifier, which is connected to the data acquisition device Q8-USB. The signals are sent from a computer with Matlab 2012 with Simulink toolbox. The corresponding elements of the dynamic model (1) and (2) are

$$M(\mathbf{q})=\begin{bmatrix}\theta_1+2\theta_2\cos(q_2) & \theta_3+\theta_2\cos(q_2)\\ \theta_3+\theta_2\cos(q_2) & \theta_3\end{bmatrix},$$

$$C(\mathbf{q},\dot{\mathbf{q}})=\begin{bmatrix}-\theta_2\sin(q_2)\dot{q}_2 & -\theta_2\sin(q_2)(\dot{q}_1+\dot{q}_2)\\ \theta_2\sin(q_2)\dot{q}_1 & 0\end{bmatrix},$$

$$F_{vl}=\begin{bmatrix}\theta_6 & 0\\ 0 & \theta_7\end{bmatrix}, J=\begin{bmatrix}\theta_4 & 0\\ 0 & \theta_5\end{bmatrix}, F_{vm}=\begin{bmatrix}\theta_8 & 0\\ 0 & \theta_9\end{bmatrix},$$

$$K=\begin{bmatrix}\theta_{10} & 0\\ 0 & \theta_{11}\end{bmatrix}.$$

TABLE 1 Parameters of the experimental flexible joint robot reported in [73]

Symbol	Value
θ_1	0.207184
θ_2	0.017580
θ_3	0.013163
θ_4	0.216776
θ_5	0.216776
θ_6	0.006842
θ_7	0.037675
θ_8	0.002959
θ_9	0.135564
θ_{10}	9.358730
θ_{11}	4.212811

The parameters are shown in Table 1, which were obtained from [73].

Joint and rotor velocities are calculated by using an average filter described as follows. First, the average position is calculated by

$$\bar{\boldsymbol{q}}(kT)=\frac{Q(z^{-1})}{p+1}\boldsymbol{q}(kT), \tag{71}$$

where k is the integer time index, p is a natural number, T is the sampling period, $\boldsymbol{q}$ represents either $\boldsymbol{q_l}$ or $\boldsymbol{q_m}$, and

$$Q(z^{-1})=1+z^{-1}+z^{-2}+\cdots+z^{-p}$$

is a polynomial in terms of z-transform. Then, the numerical derivative is expressed as

$$\dot{\bar{\boldsymbol{q}}}(kT)=\frac{1-z^{-1}}{T}\bar{\boldsymbol{q}}(kT). \tag{72}$$

Finally, an average velocity filter is given as follows

$$\dot{\boldsymbol{q}}(kT)\approx\frac{Q(z^{-1})}{p+1}\dot{\bar{\boldsymbol{q}}}(kT). \tag{73}$$

In this work, the parameters used to calculate the velocities are $T=0.001$ [ms] and $p=4$. The method described by equations (71)–(73) has been used in past works, as [73] and [80], and has shown to be a good algorithm to estimate the velocity from encoder measurements. A low-pass filter is used to compute the time derivative of the desired rotor position $\boldsymbol{q_{md}}$, denoted as

$\dot{\boldsymbol{q}}_{md}$:

$$\dot{\boldsymbol{q}}_{md} = \frac{\lambda S}{S+\lambda}\boldsymbol{q}_{md}, \tag{74}$$

where S is the Laplace operator. To compute $\ddot{\boldsymbol{q}}_{md}$, a second order low-pass filter is connected in cascade so that

$$\ddot{\boldsymbol{q}}_{md} = \left[\frac{\lambda S}{S+\lambda}\right]^2 \boldsymbol{q}_{md}, \tag{75}$$

where $\lambda = 10$ [rad/s] (see [77]).

The adaptive controller (38) is used with the adaptation law (39) and the desired rotor position (43) with the adaptive law (44). Notice that the model regressor of (44) and the parameter vector result in

$$\Phi(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l, \ddot{\boldsymbol{q}}_{ld}) = \begin{bmatrix} \ddot{q}_{ld1} & \phi_{12} & \ddot{q}_{ld2} & \dot{q}_{l1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddot{q}_{ld1} + \ddot{q}_{ld2} & \cos(q_{l2})\ddot{q}_{ld1} + \sin(q_{l2})\dot{q}_{l1}^2 & \dot{q}_{l2} \end{bmatrix},$$

$$\boldsymbol{\theta} = \begin{bmatrix} \frac{\theta_1}{\theta_{10}} & \frac{\theta_2}{\theta_{10}} & \frac{\theta_3}{\theta_{10}} & \frac{\theta_6}{\theta_{10}} & \frac{\theta_3}{\theta_{11}} & \frac{\theta_2}{\theta_{11}} & \frac{\theta_7}{\theta_{11}} \end{bmatrix},$$

with

$$\phi_{12} = 2\cos(q_{l2})\ddot{q}_{ld1} + \cos(q_{l2})\ddot{q}_{ld2} - \sin(q_{l2})\dot{q}_{l2}(\dot{q}_{l1} + \dot{q}_{l2}) \\ - \sin(q_{l2})\dot{q}_{l1}\dot{q}_{l2},$$

and the parameters θ_i in Table 1.

Besides, the low-pass filters (74) and (75) are used to compute the first and second time derivatives of the desired rotor position $\boldsymbol{q}_{md}(t)$, respectively.

The desired trajectory used for all the experiments is

$$q_{ld1} = b_1(1 - e^{-2t^3}) + c_1(1 - e^{-2t^3})\sin(w_1 t), \tag{76}$$

$$q_{ld2} = b_2(1 - e^{-2t^3}) + c_2(1 - e^{-2t^3})\sin(w_2 t), \tag{77}$$

with the parameters $b_1 = \pi/5$ [rad], $c_1 = \pi/9$ [rad], $w_1 = 1.5$ [rad/s], $b_2 = \pi/5$ [rad], $c_2 = \pi/9$ [rad], $w_2 = 2$ [rad/s].

Different controllers are tested in order to make a comparison with the proposed controller. First, the model-based controller (20) and (31) is tested by using model parameters reported in [73]. An implementation is carried out with the model-based controller denominated as desired compensation controller with backward difference reported in [9] is implemented. This controller is described by

$$\boldsymbol{\tau}_m = K^{-1}(K + K_{pz})\boldsymbol{\tau}_d - K_{pz}\boldsymbol{z} - K_{dz}\dot{\boldsymbol{z}}, \tag{78}$$

where $\boldsymbol{z} = \boldsymbol{q}_m - \boldsymbol{q}_l$, and

$$\boldsymbol{\tau}_d = M(\boldsymbol{q}_{ld})(\ddot{\boldsymbol{q}}_{ld} + \Lambda\dot{\tilde{\boldsymbol{q}}}_l) + C(\boldsymbol{q}_{ld}, \dot{\boldsymbol{q}}_{ld})(\dot{\boldsymbol{q}}_{ld} + \Lambda\tilde{\boldsymbol{q}}_l) + F_{vl}\dot{\boldsymbol{q}}_{ld} \\ + K(K + K_{pz})^{-1}(J(\ddot{\boldsymbol{q}}_{ld} + \Lambda\dot{\tilde{\boldsymbol{q}}}_l) + F_{vm}\dot{\boldsymbol{q}}_m) + K_s(\dot{\tilde{\boldsymbol{q}}}_l + \Lambda\tilde{\boldsymbol{q}}_l), \tag{79}$$

with K_{pz}, K_{dz}, K_s, and $\Lambda \in \mathbb{R}^{2\times2}$ diagonal positive definite matrices.

Besides, the adaptive controller with joint torque feedback developed in [72] was tested experimentally. In this case, the authors define

$$\boldsymbol{\tau}_s = K(\boldsymbol{q}_m - \boldsymbol{q}_l),$$

and the tracking errors $\boldsymbol{e}_l = \boldsymbol{q}_l - \boldsymbol{q}_{ld}$ and $\boldsymbol{e}_t = \boldsymbol{\tau}_s - \boldsymbol{\tau}_{sd}$, where

$$\boldsymbol{\tau}_{sd} = \Phi_{\tau s}(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l, \boldsymbol{v}_l, \dot{\boldsymbol{v}}_l)\hat{\boldsymbol{\theta}}_{\tau s} - K_{lD}\boldsymbol{r}_l, \tag{80}$$

with

$$\Phi_{\tau s}(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l, \boldsymbol{v}_l, \dot{\boldsymbol{v}}_l) \\ = \begin{bmatrix} \dot{v}_{l1} & \phi_{\tau s_{12}} & \dot{v}_{l2} & v_{l1} & 0 \\ 0 & \sin(q_{l2})\dot{q}_{l1}v_{l1} + \cos(q_{l2})\dot{v}_{l1} & \dot{v}_{l1} + \dot{v}_{l2} & 0 & v_{l2} \end{bmatrix},$$

being

$$\phi_{\tau s_{12}} = 2\cos(q_{l2})\dot{v}_{l1} + \cos(q_{l2})\dot{v}_{l2} - \sin(q_{l2})v_{l2}(\dot{q}_{l1} + \dot{q}_{l2}) \\ - \sin(q_{l2})v_{l1}\dot{q}_2,$$

and v_{li} with $i = 1, 2$ are the component vector of

$$\boldsymbol{v}_l = \dot{\boldsymbol{q}}_{ld} - \Lambda\boldsymbol{e}_l,$$

where Λ is a diagonal matrix of 2×2. The adaptation law is given by

$$\dot{\hat{\boldsymbol{\theta}}}_{\tau s} = -\Gamma_2\Phi_{\tau s}(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l, \boldsymbol{v}_l, \dot{\boldsymbol{v}}_l)\boldsymbol{r}_l \tag{81}$$

with $\boldsymbol{r}_l = \dot{\boldsymbol{q}}_l - \boldsymbol{v}_l$. The signal $\dot{\hat{\boldsymbol{\theta}}}_{\tau s}(t)$ is limited by

$$l_i < \hat{\theta}_{\tau si} < h_i. \tag{82}$$

Then, the adaptive parameters result in

$$\hat{\theta}_{\tau si}(t^+) = \begin{cases} l_i, & \text{if } \hat{\theta}_{\tau si}(t) < l_i, \\ h_i, & \text{if } \hat{\theta}_{\tau si}(t) > h_i, \\ \hat{\theta}_{\tau si}(t), & \text{otherwise,} \end{cases}$$

where t^+ is the resetting time. The adaptive controller in [72] with joint torque feedback is given by

$$\boldsymbol{\tau}_m = \Phi_\tau(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l, \boldsymbol{\tau}_s, \boldsymbol{v}_\tau, \dot{\boldsymbol{v}}_\tau)[\boldsymbol{\theta}_\tau - \Delta_u] - K_{\tau D}\boldsymbol{r}_\tau, \tag{83}$$

where $K_{\tau D}$ is a 2×2 diagonal matrix, $\boldsymbol{r}_\tau = \dot{\boldsymbol{\tau}}_s - \boldsymbol{v}_\tau$, with $\boldsymbol{v}_\tau = \dot{\boldsymbol{\tau}}_{sd} - \Lambda_\tau \boldsymbol{e}_t$, and $\Lambda_\tau \in \mathbb{R}^{2\times 2}$ is a diagonal constant matrix. The parameters vector $\boldsymbol{\theta}_\tau$ and the regressor vector $\Phi_\tau(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l, \boldsymbol{\tau}_s, \boldsymbol{v}_\tau, \dot{\boldsymbol{v}}_\tau)$ in (83) are given by

$$\boldsymbol{\theta}_\tau = \begin{bmatrix}\theta_4 & \theta_5 & \theta_8 & \theta_9 & 1 & 1 & \theta_8/\theta_{10} & \theta_9/\theta_{11} & \theta_4/\theta_{10} & \theta_5/\theta_{11}\end{bmatrix}^T,$$

$$\Phi_\tau(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l, \boldsymbol{\tau}_s, \boldsymbol{v}_\tau, \dot{\boldsymbol{v}}_\tau) = \begin{bmatrix} \ddot{q}_{l1} & 0 & \dot{q}_{l1} & 0 & \tau_{s1} & 0 & v_{\tau 1} & 0 & \dot{v}_{\tau 1} & 0 \\ 0 & \ddot{q}_{l2} & 0 & \dot{q}_{l2} & 0 & \tau_{s2} & 0 & v_{\tau 2} & 0 & \dot{v}_{\tau 2} \end{bmatrix},$$

and

$$\Delta_u = \begin{cases} -\rho \dfrac{\Phi_\tau^T r_\tau}{\|\Phi_\tau^T r_\tau\|} & \text{if } \|\Phi_\tau(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l, \boldsymbol{\tau}_s, \boldsymbol{v}_\tau, \dot{\boldsymbol{v}}_\tau)^T r_\tau\| > \epsilon_2 \\ -\dfrac{\rho}{\epsilon_2}\Phi_\tau^T r_\tau & \text{if } \|\Phi_\tau(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l, \boldsymbol{\tau}_s, \boldsymbol{v}_\tau, \dot{\boldsymbol{v}}_\tau)^T r_\tau\| \le \epsilon_2, \end{cases}$$

where $\rho > 0$ is a constant gain and $\epsilon_2 > 0$ is a constant parameter to choose. To get the accelerations $\ddot{q}_{li}$, with $i = 1, 2$, a filter with the form of (75) is used with $\lambda = 10$.

Finally, the adaptive neural network proposed in [65] is used for comparison purpose. The controller has the form of

$$\boldsymbol{\tau}_m = u_{nn} + u_{rb}, \tag{84}$$

with

$$u_{nn} = \hat{M}_y^{-1}(-\hat{F}_y + \dot{\boldsymbol{y}}_{4d} + K_1(\boldsymbol{y}_{1d} - \boldsymbol{y}_1) + K_2(\boldsymbol{y}_{2d} - \boldsymbol{y}_2)$$
$$+ K_3(\boldsymbol{y}_{3d} - \boldsymbol{y}_3) + K_4(\boldsymbol{y}_{4d} - \boldsymbol{y}_4))$$

$$u_{rb} = -\delta \frac{B_T^T P \tilde{\boldsymbol{q}}_l}{\|B_T^T P \tilde{\boldsymbol{q}}_l\|},$$

$\delta > 0 \in \mathbb{R}$, $\boldsymbol{x}$ defined in (16),

$$\begin{bmatrix} \mathbf{y}_{1d}(t) \\ \mathbf{y}_{2d}(t) \\ \mathbf{y}_{3d}(t) \\ \mathbf{y}_{4d}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{q}_{\mathbf{ld}}(t) \\ \dot{\mathbf{q}}_{\mathbf{ld}}(t) \\ \mathbf{q}_{\mathbf{ld}}(t) \\ \mathbf{q}_{\mathbf{ld}}(t) \end{bmatrix}, \begin{bmatrix} \mathbf{y}_1(t) \\ \mathbf{y}_2(t) \\ \mathbf{y}_3(t) \\ \mathbf{y}_4(t) \end{bmatrix} = \begin{bmatrix} \mathbf{q}_{\mathbf{l}}(t) \\ \dot{\mathbf{q}}_{\mathbf{l}}(t) \\ \mathbf{q}_{\mathbf{l}}(t) \\ \mathbf{q}_{\mathbf{l}}(t) \end{bmatrix},$$

$$B_T = \begin{bmatrix} 0_{2\times 2} \\ 0_{2\times 2} \\ 0_{2\times 2} \\ I_{2\times 2} \end{bmatrix}, A_T = \begin{bmatrix} 0_{2\times 2} & I_{2\times 2} & 0_{2\times 2} & 0_{2\times 2} \\ 0_{2\times 2} & 0_{2\times 2} & I_{2\times 2} & 0_{2\times 2} \\ 0_{2\times 2} & 0_{2\times 2} & 0_{2\times 2} & I_{2\times 2} \\ -K_1 & -K_2 & -K_3 & -K_4 \end{bmatrix}.$$

The matrix $P \in \mathbb{R}^{8\times 8}$ must satisfy the Lyapunov equation $A_T^T P + PA_T = -Q$, with $Q \in \mathbb{R}^{8\times 8}$ a positive definite matrix, $K_i \in \mathbb{R}^{2\times 2}$ as positive definite matrix, with $i = 1, 2, 3, 4$,

$$\hat{M}_y = M(\boldsymbol{q}_l)^{-1}KJ^{-1} + \hat{\Delta}M_0, \tag{85}$$

$$\hat{F}_y = \boldsymbol{f}_y + \hat{\Delta}\boldsymbol{f}_y,$$

$$\boldsymbol{f}_y = -M^{-1}(C\dddot{\boldsymbol{q}}_l + 2\dot{C}\ddot{\boldsymbol{q}}_l + \ddot{C}\dot{\boldsymbol{q}}_l + K\ddot{\boldsymbol{q}}_l$$
$$+ KJ^{-1}\big(F_{vm}\dot{\boldsymbol{q}}_m + K(\boldsymbol{q}_m - \boldsymbol{q}_l) + \boldsymbol{f}_{cm}(\dot{\boldsymbol{q}}_m)\big))$$
$$- 2\dot{M}^{-1}\big(C\ddot{\boldsymbol{q}}_l + \dot{C}\ddot{\boldsymbol{q}}_l + K(\dot{\boldsymbol{q}}_l - \dot{\boldsymbol{q}}_m)\big)$$
$$- \ddot{M}^{-1}\big(C\dot{\boldsymbol{q}}_l + K(\boldsymbol{q}_l - \boldsymbol{q}_m)\big), \tag{86}$$

where $M = M(\boldsymbol{q}_l)$, $\dot{M}^{-1} = \frac{dM(\boldsymbol{q}_l)^{-1}}{dt}$, $\ddot{M}^{-1} = \frac{d^2M(\boldsymbol{q}_l)^{-1}}{dt^2}$, $C = C(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l)$, $\dot{C} = \frac{dC(\boldsymbol{q}_l)}{dt}$, $\ddot{C} = \frac{d^2C(\boldsymbol{q}_l, \dot{\boldsymbol{q}}_l)}{dt^2}$.

In equations (85) and (86), the functions $\hat{\Delta}M_0$ and $\hat{\Delta}\boldsymbol{f}_y$ are adaptive neural networks with radial basis functions defined as

$$\hat{\Delta}M_0 = \hat{\boldsymbol{\phi}}^T E(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{\phi}_1^T \boldsymbol{\eta}_1(\boldsymbol{x}) \\ \boldsymbol{\phi}_2^T \boldsymbol{\eta}_2(\boldsymbol{x}) \end{bmatrix},$$

$$\hat{\Delta}\boldsymbol{f}_y = [\hat{\Psi}^T \cdot \Xi(\boldsymbol{q}_l)] = \begin{bmatrix} \hat{\boldsymbol{\psi}}_{11}^T \boldsymbol{\xi}_{11}(\boldsymbol{q}_l) & \hat{\boldsymbol{\psi}}_{12}^T \boldsymbol{\xi}_{12}(\boldsymbol{q}_l) \\ \hat{\boldsymbol{\psi}}_{21}^T \boldsymbol{\xi}_{21}(\boldsymbol{q}_l) & \hat{\boldsymbol{\psi}}_{22}^T \boldsymbol{\xi}_{22}(\boldsymbol{q}_l) \end{bmatrix},$$

$$\hat{\boldsymbol{\phi}} = \begin{bmatrix} \hat{\boldsymbol{\phi}}_1 \\ \hat{\boldsymbol{\phi}}_2 \end{bmatrix},$$

$$\hat{\Psi} = \begin{bmatrix} \hat{\boldsymbol{\psi}}_{11} & \hat{\boldsymbol{\psi}}_{12} \\ \hat{\boldsymbol{\psi}}_{21} & \hat{\boldsymbol{\psi}}_{22} \end{bmatrix},$$

where $\hat{\boldsymbol{\phi}}_i \in \mathbb{R}^{L_y}$ and $\hat{\boldsymbol{\psi}}_{ik}(\boldsymbol{q}_l) \in \mathbb{R}^{L_y}$, with $i = k = 1, 2$, are the estimated weights of the neural network, L_y are the number of neurons use for each neural network, in this case $L_y = 10$. The radial basis functions are defined as

$$\boldsymbol{a}(\boldsymbol{x}_{nn}) = \exp\left(-\frac{\|\boldsymbol{x}_{nn} - \mu_i\|_2^2}{\sigma^2}\right),$$

where $\boldsymbol{a}(\boldsymbol{x}_{nn}) \in \mathbb{R}^{L_y}$ is either $\hat{\boldsymbol{\psi}}_{ik}(\boldsymbol{q}_l)$ or $\boldsymbol{\eta}_i(\boldsymbol{x})$. The adaptive laws are defined as

$$\dot{\hat{\boldsymbol{\phi}}}_i = \gamma_{i0}\eta_i\tilde{\boldsymbol{q}}_l^{\,T} PB_{Ti}, \tag{87}$$

$$\dot{\hat{\boldsymbol{\psi}}}_{ik} = \gamma_{ij}\xi_{ik}u_{nni}\tilde{\boldsymbol{q}}_l^{\,T} PB_{Ti}, \tag{88}$$

with γ_{i0} and γ_{ij} positive constants.

The gains and parameters used for the model-based controller (35) and (36) and the adaptive controller (38), (39), (43) and (44), are

$$K_{po} = \begin{bmatrix} 1.1 & 0 \\ 0 & 0.08 \end{bmatrix}, K_p = \begin{bmatrix} 2 & 0 \\ 0 & 1.5 \end{bmatrix}, K_v = \begin{bmatrix} 0.1 & 0 \\ 0 & 0.05 \end{bmatrix},$$

$$\alpha_2 = \begin{bmatrix} 80 & 0 \\ 0 & 10 \end{bmatrix}, \beta_1 = \begin{bmatrix} 2.4 & 0 \\ 0 & 0.5 \end{bmatrix}, \beta_2 = \begin{bmatrix} 3 & 0 \\ 0 & 0.8 \end{bmatrix},$$

$$\alpha = \begin{bmatrix} 45 & 0 \\ 0 & 40 \end{bmatrix}, \Delta = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}, \kappa = 15,$$

$\alpha_1 = \gamma\beta_1, \gamma = 0.24, N = \mathrm{diag}_{10}\{0.22\}$, and

$$\Gamma = \mathrm{diag}\{0.005, 0.0004, 0.004, 0.005, 0.0009, 0.0008, 0.0008\},$$

and the initial conditions are $\hat{W}(0) = 0_{10\times 2}$ and $\hat{\boldsymbol{\theta}}(0) = 0_{7\times 1}$. Notice that all the gains and parameters satisfy all the conditions (52), (53), (54), (61), (68) and (69), these conditions are satisfied with the parameters given in Table 1, the desired trajectory (76) and (77), and the constants

$$\|\dot{\boldsymbol{q}}_{ld}\| \leq 2.1259,$$

$$\lambda_{\max}\{M\} = 0.2464,$$

$$\lambda_{\min}\{M\} = 0.0091,$$

$$k_{C1} = 0.0703,$$

$$\lambda_{\min}\{K_d\} > 0.0355.$$

The gains for the model-based controller described in (78) and (79) are

$$\Lambda = \begin{bmatrix} 0.500 & 0 \\ 0 & 4.333 \end{bmatrix}, K_{pz} = \begin{bmatrix} 7.125 & 0 \\ 0 & 1.825 \end{bmatrix},$$

$$K_{dz} = \begin{bmatrix} 7.800 & 0 \\ 0 & 0.550 \end{bmatrix}, K_s = \begin{bmatrix} 12 & 0 \\ 0 & 0.7 \end{bmatrix}.$$

The gains used for the adaptive controller with joint torque feedback (80), (81), and (83) are

$$K_{lD} = \begin{bmatrix} 1.4 & 0 \\ 0 & 0.5 \end{bmatrix}, K_{\tau D} = \begin{bmatrix} 0.7 & 0 \\ 0 & 0.08 \end{bmatrix},$$

$$\Lambda = \begin{bmatrix} 8 & 0 \\ 0 & 10 \end{bmatrix}, \Lambda_\tau = \begin{bmatrix} 9 & 0 \\ 0 & 13 \end{bmatrix},$$

$$\Gamma_2 = \mathrm{diag}\{0.03, 0.03, 0.01, 0.02, 0.03\}.$$

The upper and lower bounds, described in (82), are

$$\boldsymbol{h} = \begin{bmatrix} 0.05 & 0.005 & 0.005 & 0.02 & 0.001 \end{bmatrix}^T,$$

$$\boldsymbol{l} = -\boldsymbol{h},$$

while for Δ_u the gains $\rho = 0.01$ and $\epsilon_2 = 0.001$ were used. Finally, the gains for the adaptive neural network controller (84), (87), and (88), are set to

$$K_1 = \begin{bmatrix} 58320 & 0 \\ 0 & 233280 \end{bmatrix}, K_2 = \begin{bmatrix} 15336 & 0 \\ 0 & 30672 \end{bmatrix},$$

$$K_3 = \begin{bmatrix} 1908 & 0 \\ 0 & 1908 \end{bmatrix}, K_4 = \begin{bmatrix} 78 & 0 \\ 0 & 78 \end{bmatrix},$$

$Q = \mathrm{diag}\{1000, 1000, 10\ 10\ 10\ 10\ 10\ 10\}$, $\gamma_{10} = 5$, $\gamma_{20} = 10$, $\gamma_{11} = 0.01$, $\gamma_{12} = 0.05$, $\gamma_{21} = 0.05$, $\gamma_{22} = 0.01$, $\mu = 0$, $\sigma^2 = 10$, and the initial conditions of $\hat{\boldsymbol{\phi}}_i(0) = 0_{10}$ and $\hat{\boldsymbol{\psi}}_{ik}(0) = 0_{10}$.

5.2 | Experimental results

Figures 4–8 show the results obtained from the experiments with the controllers presented previously. The label MB denotes the model-based controller (35) and (36). The model-based controller (78) and (79) is presented with the label DC-BD. For the adaptive controller with joint torque feedback (80), (81), and (81), the label AJTF is used. The label ANN stands for the adaptive neural network (84), (87), and (88). Finally, the novel adaptive controller (38), (39), (43), and (44) is represented with the label NAC.

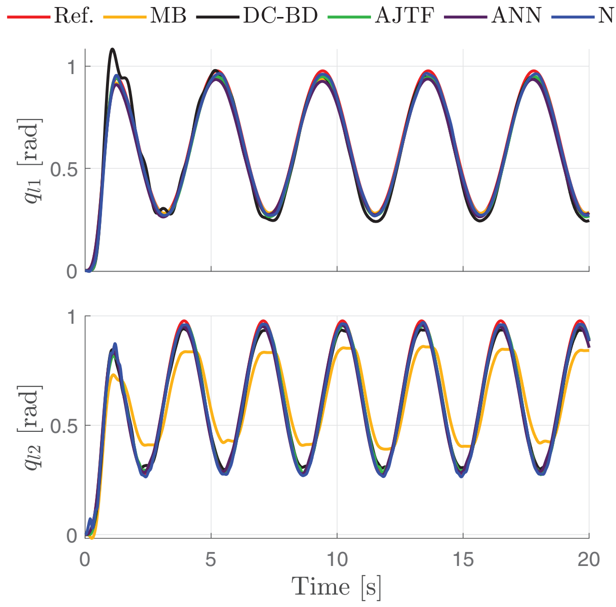

FIGURE 4 Experiment: Time evolution of the link position $\boldsymbol{q}_l(t)$

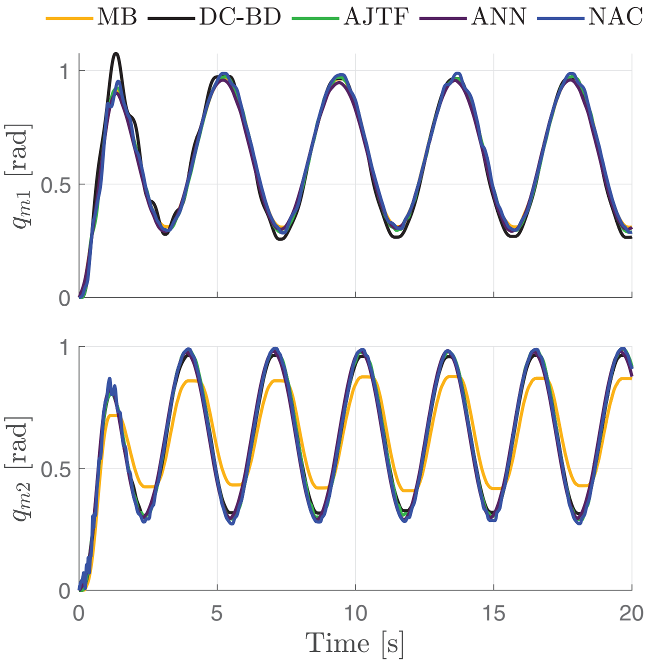

FIGURE 5 Experiment: Time evolution of the rotor position $\boldsymbol{q}_m(t)$

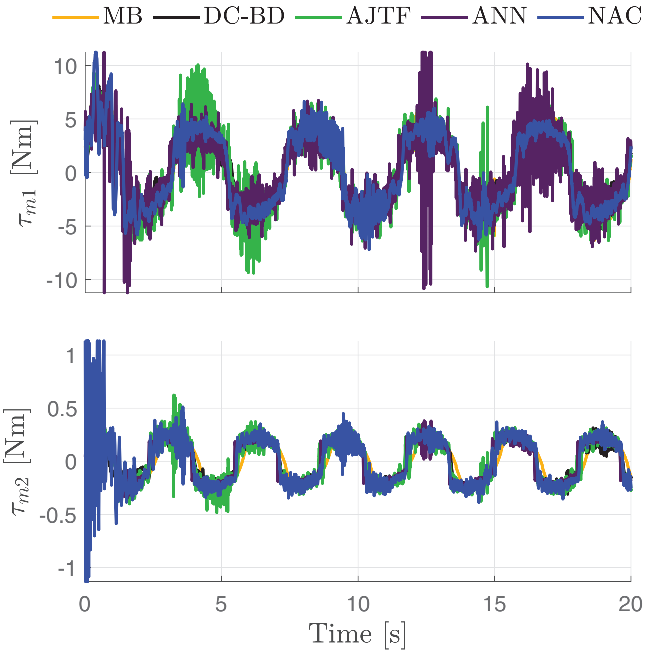

FIGURE 6 Experiment: Time evolution of the control input $\boldsymbol{\tau}_m(t)$

TABLE 2 RMS value of the trajectory tracking error $\tilde{\boldsymbol{q}}_l(t)$ in the time interval 20 to 50 [s]

q_{li}	MB	DC-BD	AJTF	ANN	NAC
1	0.0176	0.0359	0.0198	0.0300	**0.0142**
2	0.1022	0.0252	0.0141	0.0169	**0.0134**

In Figure 4, the link position $\boldsymbol{q}_l(t)$ is shown; it can be seen that the novel adaptive controller (38), (39), (43), and (44) is the controller which best follows the reference. In Figure 5, the rotor position $\boldsymbol{q}_m(t)$ is shown, given that the reference for $\boldsymbol{q}_{md}(t)$ is computed in a different form for each algorithm. To save space, the references are not presented in Figure 5. The control input $\boldsymbol{\tau}_m(t)$ is depicted in Figure 6.

Concerning the novel adaptive controller, the time evolution of the estimated parameters $\hat{\boldsymbol{\theta}}(t)$ and the real parameters $\boldsymbol{\theta}$ are shown in Figure 7. Notice that the estimated parameters remain bounded for all time $t \geq 0$ and do not converge to the real parameters; however, the adaptive control guarantees that the control goal is accomplished. In Figure 8, the time evolution of the estimated output weights $\hat{W}(t)$ is presented. Notice that the output weights were separated into two vectors, $\hat{\boldsymbol{w}}_1$ and $\hat{\boldsymbol{w}}_2$, which correspond to the control inputs τ_{m1} and τ_{m2}, respectively. The estimated output weights $\hat{W}(t)$ remain bounded for all time as expected.

Finally, the root mean square (RMS) values of the positions link error $\tilde{\boldsymbol{q}}_l(t)$ in the time interval 20 to 50 [s] are shown in Table 2. The bold numbers remark the results of the best performance. It can be noticed that the novel adaptive controller (38), (39), (43), and (44) is the controller with the lower position error RMS value. Therefore, the advantage of the non-linear compensation is clear, as can be seen in the performance of the model-based controller. Notice that the adaptive controllers, the AJTF and NAC schemes, present the best performance in comparison to the MB and DC-BD controllers. Finally, we can conclude that the proposed adaptive controller presents better tracking accuracy.

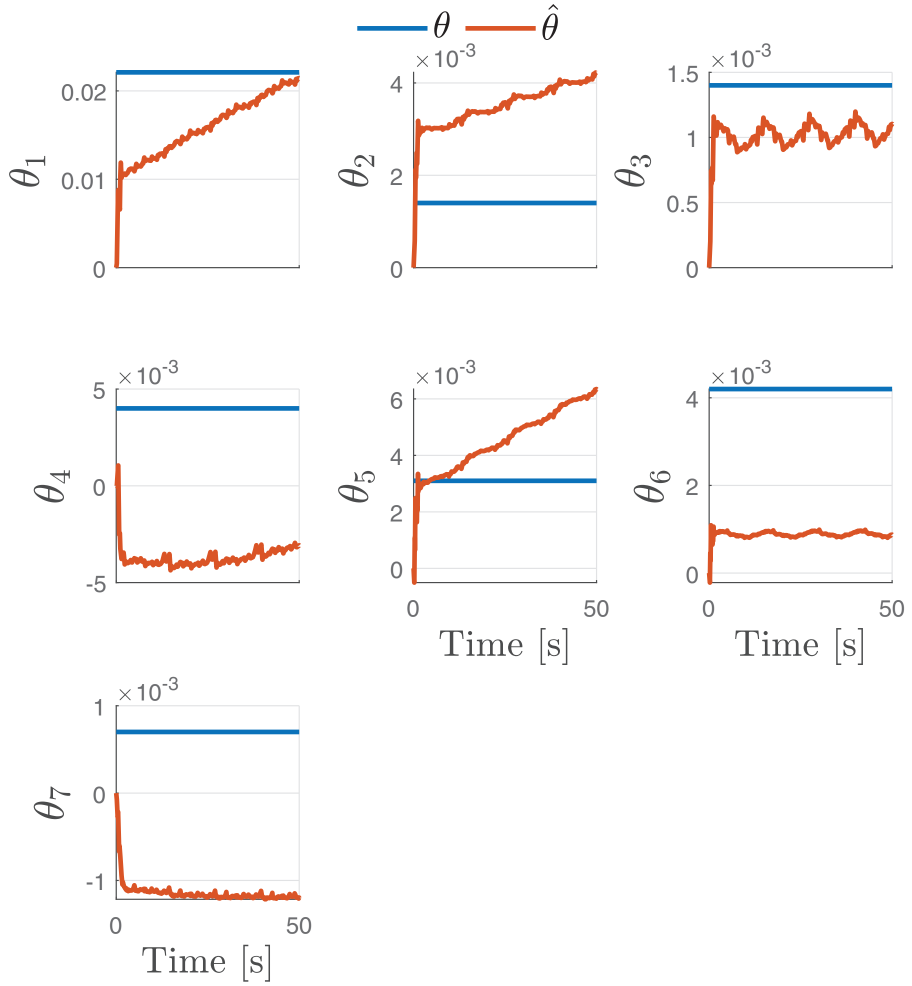

FIGURE 7 Experiment: Time evolution of the estimated parameters $\hat{\theta}(t)$ obtained with the proposed controller (38), (39), (43), and (44)

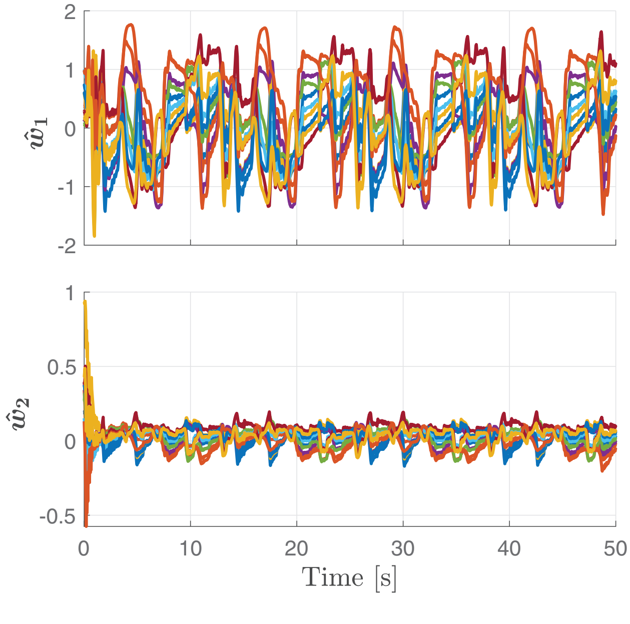

FIGURE 8 Experiment: Time evolution of the estimated output weights $\hat{W}(t)$ obtained with the proposed controller (38), (39), (43), and (44)

6 | CONCLUSIONS

In this document, a novel combined adaptive neural network and model regressor controller based on input–output feedback linearization has been proposed for flexible joint robots. It is well known that neural networks require a robustifying term to compensate for the approximation error of the neural network. Hence, a neural network cannot be used to generate the commands for the rotor position since it is required to be two-times differentiable. For this reason, the desired rotor position has been proposed as an adaptive link dynamics model regressor, which has the purpose of stabilizing the internal dynamics produced by the input–output feedback linearization.

The experimental tests were carried out in a two-link flexible joint robot, which has high flexibility. An experimental comparison among different controllers previously reported was carried out. The proposed model-based controller, a model-based controller with desired compensation, an adaptive controller based on joint torque feedback, and an adaptive neural network-based controller were used in the comparison, showing that the combined adaptive controller presented the best performance.

The use of adaptive neural network and model regressor control showed the advantage of compensating either for the unknown dynamic effects, the friction terms, or error in the parameter values. Notice that the proposed controller works for flexible joint robots with low and high spring stiffness values.

DATA AVAILABILITY STATEMENT

Data available on request from the authors

CONFLICT OF INTEREST

The authors have declared no conflict of interest.

ACKNOWLEDGMENT

This work was supported in part by the Consejo Nacional de Ciencia y Tecnología, CONACYT-Fondo Sectorial de Investigación para la Educación under Project A1-S-24762, and in part by CONACYT Project 134534, Secretaría de Investigación y Posgrado-Instituto Politécnico Nacional, México. Proyecto Apoyado por el Fondo Sectorial de Investigación para la Educación. Work partially supported by TecNM projects.

ORCID

Jorge Montoya-Cháirez https://orcid.org/0000-0002-6951-9797
Javier Moreno-Valenzuela https://orcid.org/0000-0003-0670-5979
Víctor Santibáñez https://orcid.org/0000-0002-0870-8615
Ricardo Carelli https://orcid.org/0000-0003-0688-7020
Fracisco G. Rossomando https://orcid.org/0000-0002-7792-8101
Ricardo Pérez-Alcocer https://orcid.org/0000-0002-1543-3701

REFERENCES

1. Huang, A.C., Chen, Y.C.: Adaptive sliding control for single-link flexible-joint robot with mismatched uncertainties. IEEE Trans. Control Syst. Technol. 12(5), 770–775 (2004)
2. Canudas de Wit, C., Siciliano, B., Bastin, G. (eds.) : Theory of Robot Control. Springer–Verlag, London (1996)
3. Spong, M.W.: Modeling and control of elastic joint robots. ASME J. Dyn. Syst., Meas. Control 13, 310–318 (1987)
4. Dawson, D.M., Qu, Z., Bridges, M.M.: Hybrid adaptive control for the tracking of rigid-link flexible-joint robots. IEE Proc. D (Control Theory Appl.) 140(3), 155–159 (1993)
5. Bridges, M.M., Dawson, D.M., Qu, Z., Martindale, S.C.: Robust control of rigid-link flexible-joint robots with redundant actuators. IEEE Trans. Syst., Man, Cybern. 24(7), 961–970 (1994)
6. Bridges, M.M., Dawson, D.M., Abdallah, C.A.: Control of rigid-link flexible-joint robots: a survey of backstepping approaches. J. Rob. Syst. 12(3), 199–216 (1995)
7. Lim, S., Dawson, D., Queiroz, M.: A partial state feedback controller for trajectory tracking of rigid-link flexible-joint robots using an observed backstepping approach. J. Rob. Syst. 12(11), 727–746 (1995)
8. Lim, S., Dawson, D., Hu, J., Queiroz, M.: An adaptive link position tracking controller for rigid-link flexible-joint robots without velocity measurements. IEEE Trans. Syst., Man, Cybern. 27(3), 412–427 (1997)
9. de Queiroz, M.S., Donepudi, S., Burg, T., Dawson, D.M.: Model-based control of rigid-link flexible-joint robots: an experimental evaluation. Robotica 16, 11–21 (1998)
10. Ling, S., Wang, H., Liu, P.X.: Adaptive fuzzy tracking control of flexible-joint robots based on command filtering. IEEE Trans. Ind. Electron. 67(5), 4046–4055 (2019)
11. Zaare, S., Soltanpour, M.R., Moattari, M.L: Adaptive sliding mode control of *n* flexible-joint robot manipulators in the presence of structured and unstructured uncertainties. Multibody Syst. Dyn. 47(4), 397–434 (2019)
12. Kheirkhahan, P., Izadbakhsh, A.: Observer-based adaptive fractional-order control of flexible-joint robots using the Fourier series expansion: theory and experiment. J. Brazilian Soc. Mech. Sci. Eng. 42(10), 1–10 (2020)
13. Palleschi, A., Mengacci, R., Angelini, F., Caporale, D., Pallottino, L., De Luca, A.:, Garabini, M.: Time–optimal trajectory planning for flexible joint robots. IEEE Rob. Autom. Lett. 5(2), 938–945 (2020)
14. Papastavridis, E.S., Dai, J.S.: Minimally model-based trajectory tracking and variable impedance control of flexible-joint robots. IEEE Trans. Ind. Electron. 68(7), 6031–6041 (2020)
15. Reyes–Báez, R., van der Schaft, A., Jayawardhana, B., Pan, L.: A family of virtual contraction based controllers for tracking of flexible-joints port-Hamiltonian robots: Theory and experiments. Int. J. Robust Nonlinear Control 30(8), 3269–3295 (2020)
16. Zaare, S., Soltanpour, M.R.: Continuous fuzzy nonsingular terminal sliding mode control of flexible joints robot manipulators based on nonlinear finite time observer in the presence of matched and mismatched uncertainties. J. Franklin Inst. 357(11), 6539–6570 (2020)
17. Ott, C., Albu-Schaffer, A., Kugi, A., Hirzinger, y G.: On the passivity-based impedance control of flexible joint robots. IEEE Trans. Rob. 24(2), 416–429 (2008)
18. Fantoni, I., Lozano, R.: Non-Linear Control for Underactuated Mechanical Systems. Springer-Verlag, London (2002)
19. Puga–Guzmán, S.A., Moreno–Valenzuela, J., Santibáñez, V.: Neural controller for the trajectory tracking control of an inertia wheel pendulum (in Spanish). Rev. int. de métodos numéricos para cálculo y diseño en ingeniería 32(4), 204–211 (2016)
20. Moreno-Valenzuela, J., Aguilar-Avelar, C., Puga-Guzmán, S., Santibáñez, V.: Two adaptive control strategies for trajectory tracking of the inertia wheel pendulum: neural networks vis à vis model regressor. Intell. Autom. Soft Comput. 23(1), 63–73 (2017)
21. Rigatos, G., Abbaszadeh, M., Hamida, M.A.: Nonlinear optimal control for the inertia wheel inverted pendulum. Cyber-Phys. Syst. 6(2), 55–75 (2020)
22. Montoya, O.D., Gil-González, W., Dominguez-Jimenez, J.A., Molina-Cabrera, A., Giral-Ramírez, D.A.: Global stabilization of a reaction wheel pendulum: A discrete-inverse optimal formulation approach via a control Lyapunov function. Symmetry 12(11), 1–13 (2020)
23. Soto, I., Campa, R., Sanchez-Mazuca, S.: Modelado y control con compensación de fricción de un sistema pendubot. Rev. Iberoam. de Automática e Informática ind. 18, 39–47 (2020)
24. Turrisi, G., Carlos, B.B., Cefalo, M., Modugno, V., Lanari, L., Oriolo, G.: Enforcing constraints over learned policies via nonlinear MPC: application to the pendubot. In: Proceedings of the 21st IFAC World Congress 2020, Germany (2020)
25. Baek, I., Kim, H., Lee, S., Hwang, S., Shin, K.: Swing-up control design for spring attatched passive joint acrobot. Int. J. Precision Eng. Manuf. 21(10), 1865–1874 (2020)
26. Akshay, P., Vrushabh, D., Sonam, K., Wagh, S., Singh, N.: Hamiltonian monte carlo based path integral for stochastic optimal control. In: Proceedings of the 28th Mediterranean Conference on Control and Automation (MED), pp. 254–259. France (2020)
27. Kolathaya, S.: PD tracking for a class of underactuated robotic systems with kinetic symmetry. IEEE Control Syst. Lett. 5(3), 809–814 (2021)
28. Moreno-Valenzuela, J., Aguilar-Avelar, C., Puga-Guzmán, S.A., Santibáñez, V.: Adaptive neural network control for the trajectory tracking of the Furuta pendulum. IEEE Trans. Cybern. 46(12), 3439–3452 (2016)
29. Isidori, A.: Nonlinear Control Systems. Springer-Verlag, London (2000)

30. Devanshu, A., Singh, A., Kumar, N.: Sliding mode control of induction motor drive based on feedback linearization. IETE J. Res. 66(2), 256–269 (2020)
31. Ammar, A., Kheldoun, A., Metidji, B., Ameid, T., Azzoug, Y.: Feedback linearization based sensorless direct torque control using stator flux MRAS-sliding mode observer for induction motor drive. ISA Trans. 98, 382–392 (2020)
32. Bonilla, M., Blas, L.A., Azhmyakov, V., Malabre, M., Salazar, S.: Robust structural feedback linearization based on the nonlinearities rejection. J. Franklin Inst. 357(4), 2232–2262 (2020)
33. Cheikh, R., Menacer, A., Chrifi-Alaoui, L., Drid, S.: Robust nonlinear control via feedback linearization and Lyapunov theory for permanent magnet synchronous generator-based wind energy conversion system. Front. Energy 14(1), 180–191 (2020)
34. Guo, Z., Li, S., Zheng, Y.: Feedback linearization based distributed model predictive control for secondary control of islanded microgrid. Asian J. Control 22(1), 460–473 (2020)
35. Kahla, S., Bechouat, M., Amieur, T., Sedraoui, M., Babes, B., Hamouda, N.: Maximum power extraction framework using robust fractional-order feedback linearization control and GM-CPSO for PMSG-based WECS. Wind Eng. 45(4), 1040–1054 (2021)
36. Oh, S.Y., Choi, H.L.: On robust approximate feedback linearization: control with two gain-scaling factors. Int. J. Control, Autom. Syst. 19(2), 1151–1157 (2021)
37. Gholami, H., Shafiei, M.H.: Finite-time H_∞ static and dynamic output feedback control for a class of switched nonlinear time-delay systems. Appl. Math. Comput. 389, 1–23 (2021)
38. Ahmadian, N., Khosravi, A., Sarhadi, P.: Integrated model reference adaptive control to coordinate active front steering and direct yaw moment control. ISA Trans. 106, 85–96 (2020)
39. Fan, D.D., Nguyen, J., Thakker, R., Alatur, N., Agha-mohammadi, A.A., Theodorou, E.A.: Bayesian learning-based adaptive control for safety critical systems. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), pp. 4093–4099. France (2020)
40. Fei, J., Fang, Y., Yuan, Z.: Adaptive fuzzy sliding mode control for a micro gyroscope with backstepping controller. Micromachines 11(11), 968 (2020)
41. Liu, L., Liu, Y.J., Chen, A., Tong, S., Chen, C.P.: Integral barrier Lyapunov function-based adaptive control for switched nonlinear systems. Sci. China Inf. Sci. 63(3), 1–14 (2020)
42. Niu, B., Zhao, P., Liu, J.D., Ma, H.J., Liu, Y.J.: Global adaptive control of switched uncertain nonlinear systems: an improved MDADT method. Automatica 115, 1–10 (2020)
43. Tohidi, S.S., Yildiz, Y., Kolmanovsky, I.: Adaptive control allocation for constrained systems. Automatica 121, 1–11 (2020)
44. Fei, J., Feng, Z.: Adaptive super-twisting sliding mode control for micro gyroscope based on double loop fuzzy neural network structure. Int. J. Mach. Learn. Cybern. 12(3), 611–624 (2021)
45. Lopez, B.T., Slotine, J.J.E., How, J.P.: Robust adaptive control barrier functions: An adaptive and data-driven approach to safety. IEEE Control Syst. Lett. 5(3), 1031–1036 (2021)
46. Hornik, K., Stinchcombe, M., White, H.: Multilayer feedforward networks are universal approximators. Neural Networks 2, 359–366 (1989)
47. Dewangan, S., Vadhera, S.: Performance enhancement of a stand-alone induction generator-based wind energy system using neural network controller. Int. J. Green Energy 17(4), 274–289 (2020)
48. Zhang, C., Xue, W., Li, K.: Fuzzy wavelet neural network controller design for vehicle active suspensions based on genetic algorithm. In: Proceedings of the 39th Chinese Control Conference, pp. 5607–5612. China (2020)
49. Jape, V.M., Suryawanshi, H.M., Modak, J.P.: An efficient grasshopper optimization with recurrent neural network controller-based induction motor to replace flywheel of the process machine. Trans. Inst. Meas. Control 43(1), 151–166 (2021)
50. Rossomando, F.G., Soria, C., Carelli, R.: Neural network-based compensation control of mobile robots with partially known structure. IET Control Theory Appl. 6(12), 1851–1860 (2012)
51. F. G. Rossomando and C. M. Soria, Adaptive neural sliding mode control in discrete time for a SCARA robot arm. IEEE Lat. Am. Trans. 14(6), 2556–2564 (2016)
52. Freire, E.O., Rossomando, F.G., Soria, C.M.: Self-tuning of a neuro-adaptive PID controller for a SCARA robot based on neural network. IEEE Lat. Am. Trans. 16(5), 1364–1374 (2018)
53. Montoya–Cháirez, J., Santibáñez, V., Moreno–Valenzuela, J.: Adaptive control schemes applied to a control moment gyroscope of 2 degrees of freedom. Mechatronics 57, 73–85 (2019)
54. Al-Mahasneh, A.J., Anavatti, S.G., Ferdaus, M., Garratt, M.A.: Adaptive neural altitude control and attitude stabilization of a hexacopter with uncertain dynamics. In: Proceedings of the 2019 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT), pp. 44–49. Bali, Indonesia (2019)
55. Gao, H., Song, Y., Wen, C.: Event-triggered adaptive neural network controller for uncertain nonlinear system. Inf. Sci. 506, 148–160 (2020)
56. Li, B., Xia, J., Sun, W., Park, J.H., Sun, Z.Y.: Command filter-based event-triggered adaptive neural network control for uncertain nonlinear time-delay systems. Int. J. Robust Nonlinear Control 30(16), 6363–6382 (2020)
57. Ma, T.: Filtering adaptive neural network controller for multivariable nonlinear systems with mismatched uncertainties. Int. J. Robust Nonlinear Control 30(12), 4565–4583 (2020)
58. Wang, H., Liu, S., Yang, X.: Adaptive neural control for non-strict-feedback nonlinear systems with input delay. Inf. Sci. 514, 605–616 (2020)
59. Yao, D., Dou, C., Yue, D., Zhao, N., Zhang, T.: Adaptive neural network consensus tracking control for uncertain multi-agent systems with predefined accuracy. Nonlinear Dyn. 101(4), 2249–2262 (2020)
60. Zabihifar, S.H., Yushchenko, A.S., Navvabi, H.: Robust control based on adaptive neural network for Rotary inverted pendulum with oscillation compensation. Neural Comp. Appl. 32, 14667–14679 (2020)
61. Ma, Z., Huang, P.: Adaptive neural-network controller for an uncertain rigid manipulator with input saturation and full-order state constraint. IEEE Trans. Cybern. 1–9 (2020)
62. Zhu, Q., Liu, Y., Wen, G.: Adaptive neural network output feedback control for stochastic nonlinear systems with full state constraints. ISA Trans. 101, 60–68 (2020)
63. Han, Y.Q.: Adaptive tracking control for a class of stochastic non-linear systems with input saturation constraint using multi-dimensional Taylor network. IET Control Theory Appl. 14(9), 1193–1199 (2020)
64. Chen, L., Wang, Q.: Finite-time adaptive neural dynamic surface control for non-linear systems with unknown dead zone. IET Control Theory Appl. 15(1), 40–50 (2021)
65. Ge, S.S., Lee, T.H., Tan, E.G.: Adaptive neural network control of flexible joint robots based on feedback linearization. Int. J. Syst. Sci. 29(6), 623–635 (1998)
66. Bei, X., Tianping, Z., Yuequan, Y.: Adaptive neural network control of flexible robotic with unmodeled dynamics and time-varying output constraints. In: Proceedings of the 2017 36th Chinese Control Conference, pp. 3331–3336. China (2017)
67. Liu, X., Yang, C., Chen, Z., Wang, M., Su, C.Y.: Neuro-adaptive observer based control of flexible joint robot. Neurocomputing 275, 73–82 (2018)
68. Chen, S., Wen, J.T.: Adaptive neural trajectory tracking control for flexible-joint robots with online learning. In: Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), pp. 2358–2364. (2020)
69. Ouyang, Y., Dong, L., Wei, Y., Sun, C.: Neural network based tracking control for an elastic joint robot with input constraint via actor-critic design. Neurocomputing 409, 286–295 (2020)
70. Ding, S., Peng, J., Zhang, H., Wang, Y.: Neural network-based adaptive hybrid impedance control for electrically driven flexible-joint robotic manipulators with input saturation. Neurocomputing 458, 99–111 (2021)
71. Moreno–Valenzuela, J., Aguilar–Avelar, C.: Motion Control of Underactuated Mechanical Systems. Springer-Verlag, London (2018)

72. Tian, L., Goldenberg, A.A.: Robust adaptive control of flexible joint robots with joint torque feedback. In: Proceedings of the IEEE International Conference on Robotics and Automation, vol. 1, pp. 1229–1234 (1995)
73. Miranda–Colorado, R., Moreno–Valenzuela, y J.: Experimental parameter identification of flexible joint robot manipulators. Robotica 36, 313–332 (2018)
74. Lewis, F.L., Yesildirak, A., Jagannathan, S.: Neural Network Control of Robot Manipulators and Nonlinear Systems. Taylor & Francis, London (1998)
75. Kelly, R., Santibáñez, V., Loría, A.: Control of Robot Manipulators in Joint Space. Springer-Verlag, London (2005)
76. Ogata, K.: Modern Control Engineering. Prentice Hall, Upper Saddle River (2010)
77. Moreno-Valenzuela, J., Pérez-Alcocer, R., Guerrero-Medina, M., Dzul, A.: Nonlinear PID-type controller for quadrotor trajectory tracking. IEEE/ASME Trans. Mech. 23(5), 2436–2447 (2018)
78. Khalil, H.K.: Nonlinear Syst., 3rd edn. Prentice-Hall, Upper Saddle River (2002)
79. https://www.quanser.com/products/2-dof-serial-flexible-joint/
80. Moreno-Valenzuela, J., Miranda-Colorado, R., Aguilar-Avelar, C.: A Matlab-based identification procedure applied to a two-degrees-of-freedom robot manipulator for engineering students. Int. J. Electr. Eng. Educ. 54(4), 319–34 (2017)

How to cite this article: Montoya-Cháirez, J., Moreno-Valenzuela, J., Santibáñez, V., Carelli, R., Rossomando, F.G., Pérez-Alcocer, R.: Combined adaptive neural network and regressor-based trajectory tracking control of flexible joint robots. IET Control Theory Appl. 1–20 (2021). https://doi.org/10.1049/cth2.12202