mathematics

MDPI

Article

A New Motion Tracking Controller with Feedforward Compensation for Robot Manipulators Based on Sectorial Fuzzy Control and Adaptive Neural Networks

Andres Pizarro-Lerma [1], Victor Santibañez [2], Ramon Garcia-Hernandez [2,*], Jorge Villalobos-Chin [3] and Javier Moreno-Valenzuela [4]

[1] Instituto Tecnológico de Sonora, Ciudad Obregón C.P. 85000, Sonora, Mexico; andres.pizarro@itson.edu.mx
[2] Tecnológico Nacional de Mexico/Instituto Tecnológico de La Laguna, Torreón C.P. 27000, Coahuila, Mexico; vasantibanezd@lalaguna.tecnm.mx
[3] Facultad de Ingeniería, Ciencias y Arquitectura de la Universidad Juárez del Estado de Durango, Gómez Palacio C.P. 35070, Durango, Mexico; jorge.villaloboschin@ujed.mx
[4] Centro de Investigación y Desarrollo de Tecnología Digital, Instituto Politécnico Nacional, Tijuana C.P. 22435, Baja California, Mexico; moreno@citedi.mx
* Correspondence: rgarciah@lalaguna.tecnm.mx; Tel.: +52-871-705-1324

Abstract: A novel trajectory tracking control approach for robot manipulators that uses adaptive neural network feedforward compensation plus a sectorial fuzzy controller is presented. We conduct simulation and real-time experiments comparing it with two previously published control schemes: a Proportional–Derivative (PD) plus feedforward compensation controller model, and a sectorial fuzzy control plus feedforward compensation model. The proposed controller shows a faster transient response and better steady-state angular error performance than its counterparts, and it maintains its tolerance to parameter deviation, a main characteristic of fuzzy controllers; furthermore, it excludes the need for knowledge of the robot manipulator model to achieve excellent results. A formal stability analysis of the proposed controller in a closed loop with the robot manipulator guarantees that position and velocity errors converge to zero and all signals are uniformly bounded.

Keywords: sectorial fuzzy controller; adaptive neural feedforward compensation; robot manipulator; trajectory tracking

MSC: 93C42

check for updates

Academic Editor: Tongxing Li

Received: 14 February 2025
Revised: 7 March 2025
Accepted: 14 March 2025
Published: 16 March 2025

Citation: Pizarro-Lerma, A.; Santibañez, V.; Garcia-Hernandez, R.; Villalobos-Chin, J.; Moreno-Valenzuela, J. A New Motion Tracking Controller with Feedforward Compensation for Robot Manipulators Based on Sectorial Fuzzy Control and Adaptive Neural Networks. *Mathematics* **2025**, *13*, 977. https://doi.org/10.3390/math13060977

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

The dynamics of a robot manipulator is, in most cases, very difficult to model accurately due to its nonlinearities and the variations in its parameters, such as the friction occurring at the robot's joints. Consequently, model-based controllers do not always work correctly [1]. Various proposals have previously been explored to solve this problem, such as fuzzy control and adaptive control. Fuzzy controllers are highly recommended when obtaining a model of the plant or process to be controlled is not feasible but there exists empirical knowledge of its operation. Fuzzy control of nonlinear systems has demonstrated its effectiveness as a tool that integrates heuristic elements with analytical models. Since the 1990s, fuzzy modeling for control purposes, along with neural network (NN) modeling, has become one of the most extensively developed methods applied in practice. This is one of the most successful constructions of nonlinear controllers, especially from a practical and application-oriented point of view [2].

Since Calcev published his papers outlining a guideline for the design of fuzzy controllers to be input and output strictly as passive sectorial fuzzy controllers [3,4], several approaches using this type of controller applied to the motion control of robotic manipulators, both for regulation and trajectory tracking, have been presented in different instances, showing excellent results [5,6]. Furthermore, in [7], a sectorial fuzzy controller using non-Lipschitz membership functions was used in a computed torque control array to achieve finite-time convergence for joint angular position errors of a robotic arm, with the advantage of not using discontinuous functions. Fuzzy control has yielded very good results in the control of flexible joint robotic manipulators. The paper by [8] employs fuzzy descriptions to represent uncertainty bounds and system performance. The stability proof is carried out using Lyapunov theory, and an implementation example is given via simulations.

An important approach to the control of robot manipulators is model feedforward control. This simple strategy uses the inverse dynamics of a system, evaluated along reference signals, as the control law [9]. However, implementing this control strategy requires precise values of the robot model's parameters. An enhancement to this strategy is represented by the PD model plus a feedforward control compensation scheme, which was analyzed and tested in [1]. They reported a stability analysis and experimental results for robot motion control using a PD plus feedforward controller, achieving performance comparable to the popular computed-torque control law. Recently, this approach has been extended to its fuzzy implementation in [10], where Global Uniform Asymptotic Stability (GUAS) for n-link serial robot manipulators has been formally proven, showing its good performance both in simulation and real-time experiments. To effectively utilize the excellent performance of PD control along with a feedforward controller, feedforward compensation is often implemented in an adaptive manner to address model uncertainties or disturbances. Artificial neural networks (ANNs), both feedforward-fixed and recurrent (adaptive) types, are the so-called battle horse of all current applications in Machine Learning for Robotic Control Systems, where they are extensively used to implement these types of systems [11]. In the work presented in [12], a fixed NN is applied to optimize the parameters of a Proportional–Integral–Derivative (PID) controller used in the tracking motion control of a Two-Degrees-of-Freedom (DOF) robotic arm. The NN yielded better results than those obtained via Genetic Algorithms or the cuckoo search algorithm. Performance comparisons were carried out with the PID parameter values obtained by the different optimization algorithms, demonstrating that their proposal showed the best results in terms of transient response and disturbance rejection. Adaptive neural networks have been extensively utilized in robot manipulators since the first successful approach introduced in [13]. This approach is illustrated in the work of [14], where an NN is used adaptively to model the desired dynamics of a robotic manipulator. This method is used in conjunction with a nonlinear PD controller to manage the motion trajectories of robot manipulators. Similarly, ref. [15] introduced a wavelet NN to control a deicing robot manipulator used on farm transmission lines. In this controller, the neural network was responsible for estimating the disturbances and unmodeled dynamics of the robot. A radial-basis adaptive NN was used to control a robotic arm with a unified objective bound in [16], showing satisfactory performance results. This unified objective bound includes all type of constraints related to various robotic manipulation tasks, such as transient performances, speed limits, and power consumption. However, the robot used only moved parallel to the horizontal plane such that no gravity vector was involved. Also, the control law used some discontinuous functions whose nature was not considered in the stability proof. In recent years, this approach has been expanded upon, as demonstrated by [17], who proposed an adaptive controller based on neural networks to address uncertainties and input saturation in robotic manipulators. In a study by [18], an adaptive NN controller was

developed for n-serial robotic systems with time-varying state constraints. The researchers demonstrated the controller's stability and achieved promising results in simulations. Additionally, in [19], the authors introduced a controller for an n-link constrained robotic arm, utilizing a radial-basis adaptive NN. They considered actuator dynamics, along with state and input constraints, as well as unknown time-varying delays, all at the same time. Time-varying barrier Lyapunov functions were employed, proving Semiglobally Uniformly Ultimately Bounded (SGUUB) closed-loop stability. Saturation functions and Lyapunov–Krasovskii functionals were used to eliminate the effects of actuator saturation and time delays. They validated their proposal via simulation studies. Also, in [20], a radial-basis NN is used in two parts to model and control a 6-DOF robotic arm. The article shows the whole procedure followed to construct the robot, along with the kinematic modeling and the controller design. The controller design uses a previous model of the robot estimated using a fixed radial-basis NN, and then an adaptive radial basis to control the trajectory tracking of their robot. The stability analysis is developed by applying Lyapunov theory. The controller performance is assessed by both simulation and experimental experiments, yielding excellent results with almost zero angular position errors and very minimal adaption time. In [21], a very similar control scheme to [14] is applied, combining feedback linearization (FDL) with a multilayer NN-based observer. Its stability is proven using Lyapunov's second method. The authors carry out several simulations to test their proposed controller versus the FDL and PID approaches, considering both parametric uncertainties and strong external disturbances. An NN controller applied to the regulation of the position of a 3-DOF robot manipulator is described in [22]. In this work, a three-layer B-spline ANN is used to control the robotic arm using only the angular position errors at each joint as inputs to the NN. The paper includes both the development of the robot's kinematic model and a comparative assessment of the new proposal versus a PID controller, with and without parameter deviations. The new controller shows acceptable angular position errors.

In the current work, using the new properties of sectorial fuzzy controllers (SFCs) presented in [10], which led to the discovery of strict Lyapunov functions that allowed us to conclude that GUAS should be utilized for sectorial fuzzy control (SFC) plus model feedforward compensation control of robot manipulators, a controller composed of an SFC in the feedback loop plus an adaptive NN in a feedforward array is proposed. This type of control proposal compensates for disturbances and parameter deviation, guaranteeing global uniform asymptotic motion trajectory tracking for robot manipulators of n-link at the same time. The adaptive NN is made up of three actual layers that require the calculation of two independent sets of weights, which is an initial step to implement this adaptive NN as a deep NN composed of a larger number of layers, as shown in [23]. Furthermore, formal stability analysis through Lyapunov functions and a LaSalle–Yoshizawa corollary for systems with discontinuities [24], which were not used in previous approaches, demonstrate that the proposed controller leads to global uniform convergence of position and velocity errors to zero, keeping all of its signals bounded. To the best of the authors' knowledge, this proposed controller is the first motion trajectory controller for robot manipulators based on an SFC for a feedback loop with adaptive feedforward NN compensation that does not need knowledge of robot dynamics to achieve the control objective. Finally, intensive real-time experimental tests are carried out and compared with the angular position error performance achieved by two previously published motion controllers, where, in general, the best performance is achieved by the proposed SFC with an adaptive feedforward NN compensation controller.

The remainder of this paper is structured as follows: Section 2 presents a review of robot dynamics focused on its useful properties for stability analysis, along with a brief

introduction to ANNs for modeling approximation, and the PD control plus feedforward compensation that our new proposal improves. Section 3 presents the SFC, listing its main properties used for stability analysis. Section 4 introduces our new approach, the sectorial fuzzy controller with Adaptive NN compensation, its control law, and the adaptive dynamics for the NN weights. A stability analysis of the closed-loop error dynamics of our new control approach, applying a corollary of the LaSalle–Yoshizawa theorem for nonsmooth systems and Lyapunov stability theory, is presented in depth in Section 5. Section 6 shows the main characteristics of the 2-DOF robot that is used as the plant for all the controllers, and includes all the design information, numerical simulation information, and real-time experiments, as well as a performance comparison of our new proposed controller versus the other two previously published controllers. In Section 7, we present the main conclusions obtained from the comparative analysis of the simulation and real-time implementation of our proposal with respect to previous approaches.

2. Mathematical Preliminaries

2.1. Robotic Manipulator Dynamics

The dynamics of a serial n-link robot can be summarized by the Euler–Lagrange formulation [25,26] as follows:

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{f}(\dot{\boldsymbol{q}}) = \boldsymbol{\tau} + \boldsymbol{\eta} \tag{1}$$

where $\boldsymbol{M}(\boldsymbol{q})$ is the $n \times n$ symmetric positive-definite inertia matrix; $\boldsymbol{C}(\boldsymbol{q}\ \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$ is the $n \times 1$ vector of centrifugal and Coriolis torques; $\boldsymbol{g}(\boldsymbol{q})$ is the vector of gravitational torques; $\boldsymbol{\eta}$ is the n-vector of uncertainties, which includes external disturbances and all uncertainties in the parameters and non-modeled dynamics in the robot manipulator; and $\boldsymbol{f}(\dot{\boldsymbol{q}})$ is the $n \times 1$ vector of friction torques. If a static model is assumed, friction is to be modeled by a vector $\boldsymbol{f}(\dot{\boldsymbol{q}}) \in \mathbb{R}^n$ that depends only on the joint velocity $\dot{\boldsymbol{q}}$ and combines both viscous and Coulomb frictions such that it is defined as

$$\boldsymbol{f}(\dot{\boldsymbol{q}}) = \boldsymbol{F}_v\dot{\boldsymbol{q}} + \boldsymbol{F}_C\ \mathrm{sgn}(\dot{\boldsymbol{q}}) \tag{2}$$

where both $\boldsymbol{F}_v$ and $\boldsymbol{F}_C$ are $n \times n$ diagonal positive-definite matrices, and their diagonal elements are the viscous friction parameters and the Coulomb friction parameters, respectively. Finally, $\mathrm{sgn}(\dot{\boldsymbol{q}})$ is the vector sign function [1].

The dynamics of the n-link robot manipulator modeled by (1) has the following properties, which hold for robotic manipulators with only rigid-link revolving joints [1].

Property 1. *The inertia matrix $\boldsymbol{M}(\boldsymbol{q})$ is both positive-definite and symmetric, such that*

$$\lambda_{\min}\{\boldsymbol{M}\}\|\boldsymbol{q}\|^2 \leq \boldsymbol{q}^T\boldsymbol{M}(\boldsymbol{q})\boldsymbol{q} \leq \lambda_{\max}\{\boldsymbol{M}\}\|\boldsymbol{q}\|^2 \tag{3}$$

where $\lambda_{\min}\{\boldsymbol{M}\} = \inf_{q} \lambda_{\min}\{\boldsymbol{M}(\boldsymbol{q})\}$, and $\lambda_{\max}\{\boldsymbol{M}\}$ is $\sup_{q} \lambda_{\max}\{\boldsymbol{M}(\boldsymbol{q})\}$.

Property 2. *The vector $\boldsymbol{C}(\boldsymbol{q},\boldsymbol{x})\boldsymbol{y}$ can be upper bounded as*

$$\|\boldsymbol{C}(\boldsymbol{q},\boldsymbol{x})\boldsymbol{y}\| \leq k_{C1}\|\boldsymbol{x}\|\|\boldsymbol{y}\|,\ \forall\ \boldsymbol{q},\boldsymbol{x},\boldsymbol{y} \in \mathbb{R}^n;\ \ k_{C1} > 0. \tag{4}$$

Property 3. *If the centrifugal and Coriolis torque matrices $\boldsymbol{C}(\boldsymbol{q},\ \dot{\boldsymbol{q}})\dot{\boldsymbol{q}}$ are assumed to be obtained using Christoffel symbols of the first kind, then we have*

$$\boldsymbol{q}^T\left[\dot{\boldsymbol{M}} - 2\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\right]\boldsymbol{q} = 0\ \ \forall\ \boldsymbol{q},\dot{\boldsymbol{q}} \tag{5}$$

and

$$\dot{\boldsymbol{M}} = \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}}) + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})^T. \tag{6}$$

Property 4. *The residual dynamics, $\boldsymbol{h}(\tilde{\boldsymbol{q}}, \dot{\tilde{\boldsymbol{q}}})$, ref. [27], is defined as*

$$\boldsymbol{h}(\tilde{\boldsymbol{q}}, \dot{\tilde{\boldsymbol{q}}}) = [\boldsymbol{M}(\boldsymbol{q}_d) - \boldsymbol{M}(\boldsymbol{q})]\ddot{\boldsymbol{q}}_d + \boldsymbol{g}(\boldsymbol{q}_d) - \boldsymbol{g}(\boldsymbol{q}) + [\boldsymbol{C}(\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d) - \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})]\dot{\boldsymbol{q}}_d \tag{7}$$

where $\boldsymbol{q}_d$ is the desired angular joint position, assumed to be bounded, and at least three times differentiable with bounded derivatives for all $t \geq 0$. In addition,

$$\tilde{\boldsymbol{q}} = \boldsymbol{q}_d - \boldsymbol{q} \tag{8}$$

is the angular joint position error.

The residual dynamics (7) has the property defined in (9) and satisfies the inequality (10) [1,10]

$$\boldsymbol{h}(\boldsymbol{0}, \boldsymbol{0}) = \boldsymbol{0}, \tag{9}$$

and

$$\|\boldsymbol{h}(\tilde{\boldsymbol{q}}, \dot{\tilde{\boldsymbol{q}}})\| \leq k_{h1}\|\dot{\tilde{\boldsymbol{q}}}\| + k_{h2}\|\mathbf{tanh}(\tilde{\boldsymbol{q}})\|, \tag{10}$$

where k_{h1} and k_{h2} are sufficiently large strictly positive constants that depend on the robot model parameters.

2.2. Artificial Neural Networks

The Universal Approximation Theorem states that ANNs with one hidden layer can approximate any smooth continuous function within a given range that defines a compact domain [28,29]. Therefore, a multilayer ANN with a single hidden layer and output weights can also be used to learn the desired dynamics of a robot manipulator uniformly over a given compact domain, as is exemplified in [14]. This approximation is achieved by adjusting the ANN's synaptic weights such that the error function between its output and the output of the function to be approximated (the desired robot dynamics, in this case) is minimized, as defined in (11):

$$\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{W}^T\boldsymbol{\sigma}\left(\boldsymbol{V}^T\boldsymbol{x}\right) + \boldsymbol{\epsilon}, \ \ \forall\, \boldsymbol{x} \in \mathbb{R}^{N+1} \tag{11}$$

where $\boldsymbol{f}(\boldsymbol{x}) \in \mathbb{R}^L$ is the function to be approximated by the ANN, $x \in \mathbb{R}^{N+1}$ is the vector of input signals to the ANN, $\boldsymbol{V} \in \mathbb{R}^{(N+1)\times L}$ is the optimal matrix of input weights, $\boldsymbol{W} \in \mathbb{R}^{L\times N+1}$ is the optimal matrix of output weights, L is the number of neurons in the hidden layer, $\boldsymbol{\sigma}(\cdot) \in \mathbb{R}^L$ is the activation function in the hidden layer, and $\boldsymbol{\epsilon} \in \mathbb{R}^L$ is the approximation error, where

$$|\epsilon_i| \leq E \tag{12}$$

holds $\forall\, i = 1, 2, \ldots, n$, with $E > 0$.

Since multilayered ANNs are nonlinear in their weights $\boldsymbol{V}$ due to their activation function, σ, a Taylor series can be applied to approximate it such that upon expanding σ at $\widehat{V}$, which is the estimate of the optimal input weights V, we have

$$\begin{aligned}\boldsymbol{\sigma}\left(\boldsymbol{V}^T\boldsymbol{x}\right) =& \boldsymbol{\sigma}\left(\widehat{\boldsymbol{V}}^T\boldsymbol{x}\right) + \boldsymbol{\sigma}'\left(\widehat{\boldsymbol{V}}^T\boldsymbol{x}\right)\left(\boldsymbol{V}^T\boldsymbol{x} - \widehat{\boldsymbol{V}}^T\boldsymbol{x}\right) \\ &+ \frac{1}{2}\boldsymbol{\sigma}''\left(\widehat{\boldsymbol{V}}^T\boldsymbol{x}\right)\left(\boldsymbol{V}^T\boldsymbol{x} - \widehat{\boldsymbol{V}}^T\boldsymbol{x}\right)^2 + \ldots\end{aligned} \tag{13}$$

where $\sigma'\left(\widehat{V}^T x\right)$ is the Jacobian of $\sigma\left(\widehat{V}^T x\right)$,

$$\sigma'\left(\widehat{V}^T x\right) = \frac{\partial \sigma(x)}{\partial x}; \;\; x = \widehat{V}^T x_d \tag{14}$$

We define the weight estimation errors $\widetilde{V}$ and $\widetilde{W}$ for both V and W, respectively, as

$$\widetilde{V} = V - \widehat{V} \tag{15}$$

$$\widetilde{W} = W - \widehat{W} \tag{16}$$

and the higher order terms as $O\left(\widetilde{V}^T x\right)$:

$$O\left(\widetilde{V}^T x\right) = \frac{1}{2}\sigma''\left(\widehat{V}^T x\right)\left(\widetilde{V}^T x\right)^2 + \ldots \tag{17}$$

We can rewrite (13) as follows:

$$\sigma\left(V^T x\right) = \sigma\left(\widehat{V}^T x\right) + \sigma'\left(\widehat{V}^T x\right)\widetilde{V} + O\left(\widetilde{V}^T x\right) \tag{18}$$

2.3. PD Control Plus Feedforward Controller

The original control proposal used for the trajectory control of robotic manipulators that we are trying to improve via a fuzzy controller and an ANN is the PD control plus feedforward controller. A block diagram of this control structure is shown in Figure 1.

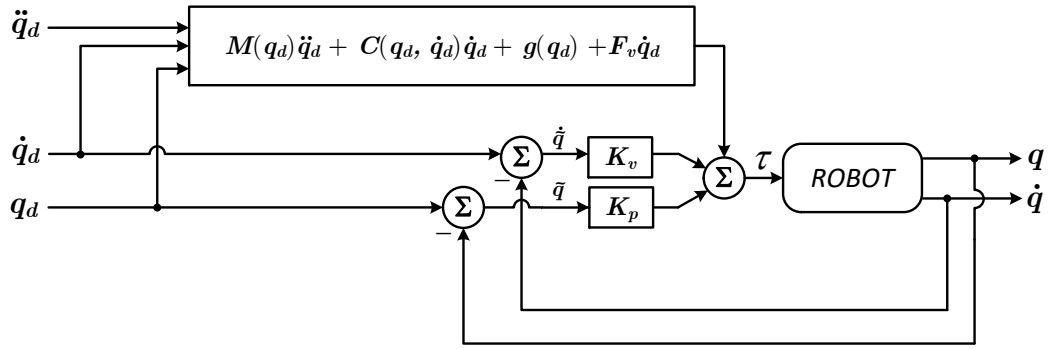


Figure 1. PD control plus feedforward diagram.

The PD control plus feedforward controller analyzed in [1] is defined as

$$\tau = K_p\tilde{q} + K_v\dot{\tilde{q}} + M(q_d)\ddot{q}_d + C(q_d, \dot{q}_d)\dot{q}_d + g(q_d) + F_v\dot{q}_d$$

where $K_p, K_v \in \mathbb{R}^{n\times n}$ are symmetric positive-definite matrices, called gains of position and velocity, respectively.

3. Sectorial Fuzzy Controller

An SFC $\Phi(\cdot,\cdot)$ [3] is a special type of fuzzy controller with two inputs and only one output, which form a nonlinear input–output static fuzzy mapping. $\Phi(\tilde{q}, \dot{\tilde{q}})$ [3,5] is an $n \times 1$ vector whose elements $\phi_i(\tilde{q}_i, \dot{\tilde{q}}_i)$, with $i = 1, 2, 3, \ldots, n$, are the real input–output mappings of the SFC,

$$\Phi(\tilde{q}, \dot{\tilde{q}}) = \begin{bmatrix} \phi_1(\tilde{q}_1, \dot{\tilde{q}}_1) \\ \phi_2(\tilde{q}_2, \dot{\tilde{q}}_2) \\ \vdots \\ \phi_n(\tilde{q}_n, \dot{\tilde{q}}_n) \end{bmatrix} \tag{19}$$

An SFC has useful sectorial properties that enable the analysis of its stability. These properties have been already proven in [3,5], and are listed below:

- Property 1: $\phi(0,0) = 0$.
- Property 2: $\phi_i(\tilde{q}_i, \dot{\tilde{q}}_i) = -\phi_i(-\tilde{q}_i, -\dot{\tilde{q}}_i)$.
- Property 3: there exists $\zeta_i, \rho_i > 0$, such that

$$0 < \tilde{q}_i[\phi_i(\tilde{q}_i, \dot{\tilde{q}}_i) - \phi_i(0, \dot{\tilde{q}}_i)] \leq \rho_i \tilde{q}_i^2 \tag{20}$$

$$0 < \dot{\tilde{q}}_i[\phi_i(\tilde{q}_i, \dot{\tilde{q}}_i) - \phi_i(\tilde{q}_i, 0)] \leq \zeta_i \dot{\tilde{q}}_i^2 \tag{21}$$

- Property 4: $\phi_i(\tilde{q}_i, 0) = 0 \Rightarrow \tilde{q}_i = 0$.
- Property 5: $|\phi_i(\tilde{q}_i, \dot{\tilde{q}}_i)| \leq \delta := \max_{l1,l2} \bar{y}^{l1,l2}$.
- Property 6: $\bar{y}^{k,0} \leq |\phi_i(\tilde{q}_i, 0)| \leq \bar{y}^{k+1,0}$; for $i = 1, 2, 3, \ldots, n$, where $\bar{y}^{l1,l2}, \bar{y}^{k,0}, \bar{y}^{k+1,0}$ define the centers of the corresponding output membership functions (MFs) that must be assigned by the control designer. Through the definition of the MFs, a piece-wise transient response can be given to the controlled system [30].

  New properties were found and used in [10]

- Property 7:

$$|\phi_i(\tilde{q}_i, \dot{\tilde{q}}_i) - \phi_i(0, \dot{\tilde{q}}_i)| \geq \alpha_i |\tanh(\tilde{q}_i)| \tag{22}$$

  with α_i defined as

$$\alpha_i = \varphi_i \left| \min_{-M<m<M} \{\bar{y}_i^{1,m} - \bar{y}_i^{0,m}, \bar{y}_i^{1,m+1} - \bar{y}_i^{0,m+1}, \bar{y}_i^{0,m} - \bar{y}_i^{-1,m}, \bar{y}_i^{0,m+1} - \bar{y}_i^{-1,m+1}\} \right| \tag{23}$$

  where M is the maximum value of the partitions considered within the domain of discourse of $\tilde{q}_i$ in this case, and $\varphi_i > 0$ is the upper bound of the derivative $|\mu_{i1}^{0\prime}(\tilde{q}_i)|$ of the MF $\mu_{i1}^0(\tilde{q}_i)$.

$$|\mu_1^{0\prime}(\tilde{q}_i)| \leq \varphi_i \tag{24}$$

- Property 8: given $\zeta_i > 0$,

$$|\phi_i(0, \dot{\tilde{q}}_i)| \leq \zeta_i |\dot{\tilde{q}}_i|. \tag{25}$$

An SFC will have the properties listed before in [10], "if it is designed using the following guidelines: Define a fuzzy system with one output computed as the fuzzy relationship of two inputs. Establish an odd number of input and output fuzzy sets covering the entire span of the fuzzified inputs considered, with all MFs symmetric with respect to zero. The MFs of adjacent input fuzzy sets must be defined to have complementary membership values for every crisp value. The definitions of fuzzy sets for the input MF are recommended to be convex in the sense given by [3]; therefore, around zero, only functions with one isolated maximum (like triangular MFs) can be used, where no trapezoidal or similar MFs are allowed. The consequents of the fuzzy rules table increase, logically speaking, from left to right, and from top to bottom, with a zero output when both inputs are zero. Due to the condition that must be met for Property 7 to exist, that is, $|\phi_i(\tilde{q}_i, \dot{\tilde{q}}_i) - \phi_i(0, \dot{\tilde{q}}_i)| \neq 0$, no column or row adjacent to a null input column or row, respectively, can have the same consequent. The output is computed by using the center average fuzzifier, applying the product inference method, and center average defuzzifier", which will turn the output computing into a convex combination of its inputs as, demonstrated previously in [5,10,31].

From the properties defined for an SFC, we can develop the following useful relationships, which will be applied in the stability analysis of our proposed control scheme in Section 5:

By extending (25) to the vector-matrix case, we can rewrite

$$\|\boldsymbol{\Phi}(\mathbf{0}, \dot{\tilde{\boldsymbol{q}}})\| \leq \lambda_{\max}\{\boldsymbol{Z}\}\|\dot{\tilde{\boldsymbol{q}}}\| \tag{26}$$

where $\boldsymbol{Z} = \text{diag}\{\zeta_i\}$ for $i = 1, 2, \ldots n$.

From Property 9, we can express (22) using the following general expression

$$\|\boldsymbol{\Phi}(\tilde{\boldsymbol{q}}, \dot{\tilde{\boldsymbol{q}}}) - \boldsymbol{\Phi}(\mathbf{0}, \dot{\tilde{\boldsymbol{q}}})\| \geq \lambda_{\min}\{\boldsymbol{A}\}\|\tanh(\tilde{\boldsymbol{q}})\| \tag{27}$$

with $\boldsymbol{A} = \text{diag}(\alpha_i)$.

And from (20) in Property 3, it can be reinterpreted as

$$\text{sign}(\Phi(\tilde{q}_i, \dot{\tilde{q}}_i) - \Phi(0, \dot{\tilde{q}}_i)) = \text{sign}(\tilde{q}_i) \tag{28}$$

Also, considering that $\text{sign}(\tilde{q}_i) = \text{sign}(\tanh(\tilde{q}_i))$, (20) can be rewritten in vector notation as

$$\tanh(\tilde{\boldsymbol{q}})^T\left[\boldsymbol{\Phi}(\tilde{\boldsymbol{q}}, \dot{\tilde{\boldsymbol{q}}}) - \boldsymbol{\Phi}(\mathbf{0}, \dot{\tilde{\boldsymbol{q}}})\right] \geq 0 \tag{29}$$

Applying (22) to (29), we have

$$\begin{aligned} \tanh(\tilde{\boldsymbol{q}})^T\left[\boldsymbol{\Phi}(\tilde{\boldsymbol{q}}, \dot{\tilde{\boldsymbol{q}}}) - \boldsymbol{\Phi}(\mathbf{0}, \dot{\tilde{\boldsymbol{q}}})\right] &\geq \tanh(\tilde{\boldsymbol{q}})^T\boldsymbol{A}\tanh(\tilde{\boldsymbol{q}}) \\ \tanh(\tilde{\boldsymbol{q}})^T\boldsymbol{A}\tanh(\tilde{\boldsymbol{q}}) &\geq \lambda_{\min}\{\boldsymbol{A}\}\|\tanh(\tilde{\boldsymbol{q}})\|^2 \\ \lambda_{\min}\{\boldsymbol{A}\}\|\tanh(\tilde{\boldsymbol{q}})\|^2 &> 0 \;\; \forall \tilde{\boldsymbol{q}} \neq \mathbf{0} \in \mathbb{R}^n \end{aligned} \tag{30}$$

Finally, if we substitute $\dot{\tilde{q}}_i = 0$ in (22), it yields

$$|\phi_i(\tilde{q}_i, 0)| \geq \alpha_i|\tanh(\tilde{q}_i)| \tag{31}$$

which, in matrix expression, can be written as

$$\|\boldsymbol{\Phi}(\tilde{\boldsymbol{q}}, \mathbf{0})\| \geq \lambda_{\min}\{\boldsymbol{A}\}\|\tanh(\tilde{\boldsymbol{q}})\|. \tag{32}$$

4. Sectorial Fuzzy Controller with Adaptive Neural Network Compensation

A block diagram of our proposal is shown in Figure 2.

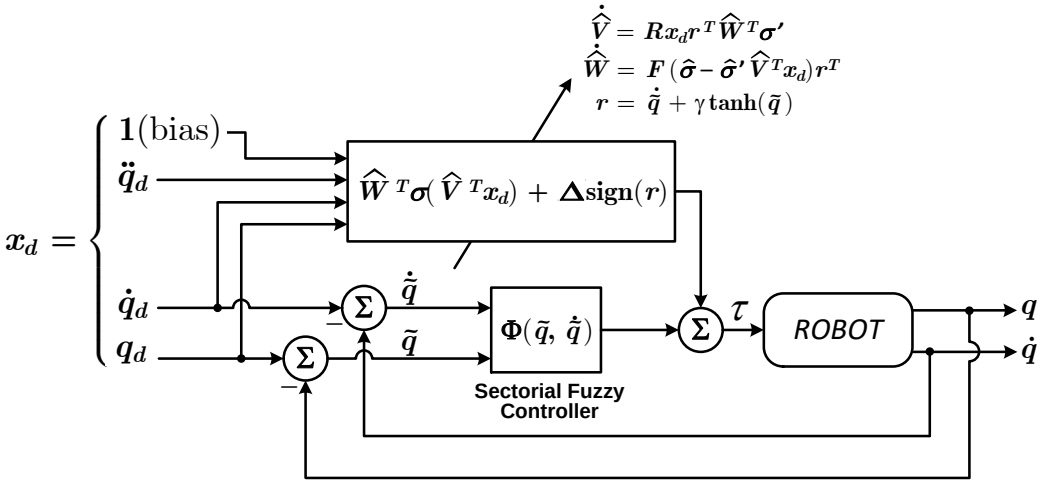


Figure 2. SFC plus adaptive NN compensation controller.

And its control law is given in (33):

$$\boldsymbol{\tau} = \widehat{\boldsymbol{W}}^{\boldsymbol{T}}\boldsymbol{\sigma}\left(\widehat{\boldsymbol{V}}^{\boldsymbol{T}}\boldsymbol{x}_{\boldsymbol{d}}\right) + \boldsymbol{\Phi}\left(\tilde{\boldsymbol{q}}, \dot{\tilde{\boldsymbol{q}}}\right) + \boldsymbol{\Delta}\text{sign}(\boldsymbol{r}) \tag{33}$$

where

$$\boldsymbol{x}_{\boldsymbol{d}} = \left[\boldsymbol{q}_{\boldsymbol{d}}^{T}\ \dot{\boldsymbol{q}}_{\boldsymbol{d}}^{T}\ \ddot{\boldsymbol{q}}_{\boldsymbol{d}}^{T}\ 1\right]^{T} \in \mathbb{R}^{N+1} \tag{34}$$

is the vector of input signals to the NN, consisting of the desired trajectories for the n joints of the robot, their two first derivatives (desired angular velocities and angular accelerations; therefore, $N = 3n$), and a normalized bias; $\boldsymbol{\Delta} \in \mathbb{R}^{n\times n}$ is a designed positive-definite matrix; L is the number of neurons to be applied; $\widehat{\boldsymbol{V}} \in \mathbb{R}^{N+1\times L}$ is the matrix of estimated input weights; and $\widehat{\boldsymbol{W}} \in \mathbb{R}^{L\times n}$ is the matrix of estimated output weights, where its update laws are defined in (35) and (36), respectively:

$$\dot{\widehat{\boldsymbol{V}}} = \boldsymbol{R}\boldsymbol{x}_{\boldsymbol{d}}\boldsymbol{r}^{\boldsymbol{T}}\widehat{\boldsymbol{W}}^{\boldsymbol{T}}\hat{\boldsymbol{\sigma}}' \tag{35}$$

$$\dot{\widehat{\boldsymbol{W}}} = \boldsymbol{F}\left(\hat{\boldsymbol{\sigma}} - \hat{\boldsymbol{\sigma}}'\widehat{\boldsymbol{V}}^{\boldsymbol{T}}\boldsymbol{x}_{\boldsymbol{d}}\right)\boldsymbol{r}^{\boldsymbol{T}} \tag{36}$$

$\boldsymbol{R} \in \mathbb{R}^{(N+1)\times(N+1)}$, $\boldsymbol{F} \in \mathbb{R}^{L\times L}$ are positive-definite matrices, $\hat{\boldsymbol{\sigma}} = \boldsymbol{\sigma}\left(\widehat{\boldsymbol{V}}^{\boldsymbol{T}}\boldsymbol{x}_{\boldsymbol{d}}\right)$, and $\hat{\boldsymbol{\sigma}}'$ is the Jacobian of $\hat{\boldsymbol{\sigma}}$, as defined previously in (14). In addition,

$$\boldsymbol{r} = \dot{\tilde{\boldsymbol{q}}} + \gamma\,\tanh(\tilde{\boldsymbol{q}}) \tag{37}$$

where γ is a positive scalar.

Also, assuming that $\boldsymbol{q}_{\boldsymbol{d}}$, $\dot{\boldsymbol{q}}_{\boldsymbol{d}}$, and $\ddot{\boldsymbol{q}}_{\boldsymbol{d}}$ are all bounded, the vector of input signals to the NN are also bounded:

$$\|\boldsymbol{x}_{\boldsymbol{d}}\| < k_{xd};\ \text{with}\ k_{xd} > 0 \tag{38}$$

5. Stability Analysis of SFC with Adaptive NN Compensation

5.1. LaSalle–Yoshizawa Corollary for Nonsmooth Systems

Since our control law (33) uses the sign function, sign$(\cdot)$, which is discontinuous at the origin, this creates a differential equation with a right-hand side discontinuity (discontinuous on x and continuous on t) in the closed-loop error dynamics (54), which is the main equation where the stability of our control proposal will be proven. If the differential equation defining any system includes a right-hand-side discontinuity, its solution in the classical sense cannot be guaranteed; neither of these differential equations is locally Lipschitz. Several approaches to overcome the aforementioned problem have been proposed, of which Filippov solutions are the most formally accepted. Filippov generalizes the concept of a differential equation to a differential inclusion (set of differential equations) and allows us to formalize the stability analysis for nonsmooth right-hand-side differential equations. A typical Lyapunov stability analysis that uses smooth Lyapunov functions cannot be used to prove the stability of discontinuous systems. The latter naturally leads to the need to use a generalized Lyapunov analysis for nonsmooth systems. Therefore, we have included, with the aim of this manuscript being self-contained, the second corollary of this theorem developed in [24], which extends the stability analysis capacities of the LaSalle–Yoshizawa theorem. It adapts the LaSalle–Yoshizawa Theorem to systems where the time derivatives of the system states are not locally Lipschitz, applying Filippov solutions for nonautonomous nonlinear systems with right-hand-side discontinuities utilizing Lipschitz-continuous and regular Lyapunov functions whose time derivatives (in the sense of Filippov) can be upper bounded by negative semi-definite functions to prove stability. This corollary also uses and extends the applicability of Barbalat's Lemma to prove stability.

Corollary 1 ([24]). *For the system*

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}(t), t) \tag{39}$$

where $\boldsymbol{x}(t) \in \mathcal{D} \subset \mathbb{R}^n$ *is the state vector, suppose* $\boldsymbol{f} : \mathcal{D} \times [0, \infty) \to \mathbb{R}^n$ *is Lebesgue-measurable and essentially locally bounded, uniformly in* t*, and let* $\mathcal{D}$ *be an open and connected set containing* $\boldsymbol{x} = 0$*, and* $U : \mathcal{D} \times [0, \infty) \to \mathbb{R}$ *be locally Lipschitz and regular such that*

$$G_1(\boldsymbol{x}) \leq U(\boldsymbol{x}, t) \leq G_2(\boldsymbol{x}), \tag{40}$$

$$\dot{\tilde{U}}(\boldsymbol{x}, t) \leq -G_3(\boldsymbol{x}) \tag{41}$$

$\forall t \geq 0, \forall \boldsymbol{x} \in \mathcal{D}$ *where* $G_1(\boldsymbol{x})$ *and* $G_2(\boldsymbol{x})$ *are continuous positive-definite functions, and* $G_3(\boldsymbol{x})$ *is a continuous positive semi-definite function on* $\mathcal{D}$*. Choose* $r > 0$ *and* $c > 0$ *such that* $B_r \subset \mathcal{D}$ *and* $c < \min_{\|\boldsymbol{x}\|=r} G_1(\boldsymbol{x})$*. Then, all Filippov solutions of (39) such that* $\boldsymbol{x}(t_0) \in \{\boldsymbol{x} \in B_r | G_2(\boldsymbol{x}) \leq c\}$ *are bounded and satisfy*

$$G_3(\boldsymbol{x}(t)) \to 0 \ \ as \ \ t \to \infty \tag{42}$$

In this case, $\boldsymbol{x}(t)$ is assumed to be a Filippov solution of (39), and $\dot{\tilde{U}}(\boldsymbol{x}, t)$ is the time derivative of the regular function $U(\boldsymbol{x}, t)$, obtained via Clarke's generalized gradient, computed as

$$\dot{\tilde{U}}(\boldsymbol{x}, t) = \bigcap_{\xi \in \partial U(x,t)} \xi^T \begin{pmatrix} K[f](\boldsymbol{x}(t), t) \\ 1 \end{pmatrix} \tag{43}$$

where $K[f](\boldsymbol{x}(t), t)$ is an upper semi-continuous, nonempty, compact and convex valued map on $\mathcal{D}$ defined as

$$K[f](\boldsymbol{x}(t), t) = \bigcap_{\delta > 0} \bigcap_{\mu(M=0)} \overline{co} f(B(\boldsymbol{x}(t), \delta) \setminus M, t) \tag{44}$$

with $\bigcap_{\mu(M=0)}$ being the intersection over sets M of Lebesgue measure zero, $\overline{co}$ being the convex closure, and

$$B(\boldsymbol{x}(t), \delta) = \{v \in \mathbb{R}^n \Big| \|x(t) - v\| < \delta\} \tag{45}$$

An arbitrary set of measure zero in $B(\boldsymbol{x}(t), \delta)$ will be excluded from the evaluation of $K[f](\boldsymbol{x}(t), t)$, so that the result is the same for two vector fields that differ on a set of Lebesgue measure zero, according to Filippov theory [32].

Thus, the Lyapunov function, $U(\boldsymbol{x}, t)$, to be used in order to apply this corollary of the LaSalle–Yoshizawa Theorem, must be both decrescent and radially unbounded, as stated in (40); and its time derivative must be negative semidefinite as in (41).

5.2. Closed-Loop Error Dynamics

The closed-loop equivalent of the system represented in the diagram shown in Figure 2 is obtained by first considering the Coulomb friction terms as part of the uncertainties in the robot dynamics model defined in (1); since they are unknown, we have

$$\boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} + \boldsymbol{g}(\boldsymbol{q}) + \boldsymbol{F}_v\dot{\boldsymbol{q}} \tag{46}$$

We define the desired torque function, $\boldsymbol{\tau_d}$ from (46) as follows:

$$\boldsymbol{\tau_d} = \boldsymbol{M}(\boldsymbol{q_d})\ddot{\boldsymbol{q}}_d + \boldsymbol{C}(\boldsymbol{q_d}, \dot{\boldsymbol{q}}_d)\dot{\boldsymbol{q}}_d + \boldsymbol{g}(\boldsymbol{q_d}) + \boldsymbol{F}_v\dot{\boldsymbol{q}}_d \tag{47}$$

Subtracting (46) from (47), we have

$$\begin{aligned}\boldsymbol{\tau}_d - \boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q}_d)\ddot{\boldsymbol{q}}_d + \boldsymbol{C}(\boldsymbol{q}_d, \dot{\boldsymbol{q}}_d)\dot{\boldsymbol{q}}_d + \boldsymbol{g}(\boldsymbol{q}_d) + \boldsymbol{F}_v\dot{\boldsymbol{q}}_d \\ - \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} - \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\boldsymbol{q}} - \boldsymbol{g}(\boldsymbol{q}) - \boldsymbol{F}_v\dot{\boldsymbol{q}}\end{aligned}$$

which leads to

$$\boldsymbol{\tau}_d - \boldsymbol{\tau} = \boldsymbol{M}(\boldsymbol{q})\ddot{\tilde{\boldsymbol{q}}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{F}_v\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{h}(\tilde{\boldsymbol{q}}, \dot{\tilde{\boldsymbol{q}}}), \tag{48}$$

where the residual dynamics, $\boldsymbol{h}(\tilde{\boldsymbol{q}}, \dot{\tilde{\boldsymbol{q}}})$ was used as defined in (7).

Now, we approximate the desired torque, $\boldsymbol{\tau}_d$, in (47) by means of an NN:

$$\boldsymbol{\tau}_d = \boldsymbol{W}^T\sigma\left(\boldsymbol{V}^T\boldsymbol{x}_d\right) + \boldsymbol{\epsilon} \tag{49}$$

with $\boldsymbol{x}_d$ defined as in (34).

Substituting the control law (33) and the approximation of τ_d (49) in (48) yields

$$\begin{aligned}\boldsymbol{M}(\boldsymbol{q})\ddot{\tilde{\boldsymbol{q}}} + \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{F}_v\dot{\tilde{\boldsymbol{q}}} + \boldsymbol{h}(\tilde{\boldsymbol{q}}, \dot{\tilde{\boldsymbol{q}}}) =& \boldsymbol{W}^T\sigma\left(\boldsymbol{V}^T\boldsymbol{x}_d\right) - \widehat{\boldsymbol{W}}^T\sigma\left(\widehat{\boldsymbol{V}}^T\boldsymbol{x}_d\right) \\ &+ \boldsymbol{\epsilon} - \boldsymbol{\Phi}\left(\tilde{\boldsymbol{q}}, \dot{\tilde{\boldsymbol{q}}}\right) - \boldsymbol{\Delta}\mathrm{sign}(\boldsymbol{r})\end{aligned} \tag{50}$$

We apply Taylor series expansion to the activation function, σ, on (50), and simplify

$$\begin{aligned}\boldsymbol{M}(\boldsymbol{q})\ddot{\tilde{\boldsymbol{q}}} =& - \boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\tilde{\boldsymbol{q}}} - \boldsymbol{F}_v\dot{\tilde{\boldsymbol{q}}} - \boldsymbol{h}(\tilde{\boldsymbol{q}}, \dot{\tilde{\boldsymbol{q}}}) + \widetilde{\boldsymbol{W}}^T\widehat{\sigma} - \widetilde{\boldsymbol{W}}^T\widehat{\sigma}'\widehat{\boldsymbol{V}}^T\boldsymbol{x}_d \\ &+ \widehat{\boldsymbol{W}}^T\widehat{\sigma}'\widetilde{\boldsymbol{V}}^T\boldsymbol{x}_d + \boldsymbol{\omega}(\boldsymbol{t}) - \boldsymbol{\Phi}\left(\tilde{\boldsymbol{q}}, \dot{\tilde{\boldsymbol{q}}}\right) - \boldsymbol{\Delta}\mathrm{sign}(\boldsymbol{r})\end{aligned} \tag{51}$$

where $\boldsymbol{\omega}(\boldsymbol{t})$ is defined as

$$\boldsymbol{\omega}(\boldsymbol{t}) = \widetilde{\boldsymbol{W}}^T\widehat{\sigma}'\boldsymbol{V}^T\boldsymbol{x}_d + \boldsymbol{W}^T\boldsymbol{O}\left(\widetilde{\boldsymbol{V}}^T\boldsymbol{x}_d\right) + \boldsymbol{\epsilon} \tag{52}$$

If the activation function being used is analytical, $\boldsymbol{\omega}(\boldsymbol{t})$ is assumed to be bounded [14]:

$$\|\boldsymbol{\omega}(\boldsymbol{t})\| \leq k_\omega \tag{53}$$

Including both the output and input weight update dynamics, $\boldsymbol{V}$ and $\boldsymbol{W}$, given in (36) and (35), respectively, and expressing them in terms of $\widetilde{\boldsymbol{V}}$, $\widetilde{\boldsymbol{W}}$, the whole closed-loop errors dynamics can be expressed in matrix notation as

$$\frac{d}{dt}\begin{bmatrix}\tilde{\boldsymbol{q}} \\ \dot{\tilde{\boldsymbol{q}}} \\ \widetilde{\boldsymbol{V}} \\ \widetilde{\boldsymbol{W}}\end{bmatrix} = \begin{bmatrix}\dot{\tilde{\boldsymbol{q}}} \\ \boldsymbol{M}(\boldsymbol{q})^{-1}[-\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})\dot{\tilde{\boldsymbol{q}}} - \boldsymbol{F}_v\dot{\tilde{\boldsymbol{q}}} - \boldsymbol{h}(\tilde{\boldsymbol{q}}, \dot{\tilde{\boldsymbol{q}}}) + \widetilde{\boldsymbol{W}}^T\widehat{\sigma} - \widetilde{\boldsymbol{W}}^T\widehat{\sigma}'\widehat{\boldsymbol{V}}^T\boldsymbol{x}_d \\ \quad + \widetilde{\boldsymbol{W}}^T\widehat{\sigma}'\widetilde{\boldsymbol{V}}^T\boldsymbol{x}_d + \boldsymbol{\omega}(\boldsymbol{t}) - \boldsymbol{\Phi}\left(\tilde{\boldsymbol{q}}, \dot{\tilde{\boldsymbol{q}}}\right) - \boldsymbol{\Delta}\mathrm{sign}(\boldsymbol{r})] \\ -\boldsymbol{R}\boldsymbol{x}_d\boldsymbol{r}^T\left(\boldsymbol{W}^T - \widetilde{\boldsymbol{W}}^T\right)\hat{\sigma}' \\ -\boldsymbol{F}\left(\widehat{\sigma} - \widehat{\sigma}'\boldsymbol{V}^T\boldsymbol{x}_d + \widehat{\sigma}'\widetilde{\boldsymbol{V}}^T\boldsymbol{x}_d\right)\boldsymbol{r}^T\end{bmatrix} \tag{54}$$

5.3. Lyapunov Function Candidate

The stability analysis for the closed-loop system is formalized in the following paragraphs. The proposed Lyapunov function candidate (LFC) is

$$U(\tilde{\boldsymbol{q}},\dot{\tilde{\boldsymbol{q}}},\widetilde{\boldsymbol{V}},\widetilde{\boldsymbol{W}},t)=\frac{1}{2}\dot{\tilde{\boldsymbol{q}}}^{T}\boldsymbol{M}(\boldsymbol{q})\dot{\tilde{\boldsymbol{q}}}+\sum_{i=1}^{n}\int_{0}^{\tilde{q}_i}\phi(\xi_i,0)d\xi_i+\frac{1}{2}\mathrm{tr}(\widetilde{\boldsymbol{V}}^{T}\boldsymbol{R}^{-1}\widetilde{\boldsymbol{V}})$$
$$+\frac{1}{2}\mathrm{tr}(\widetilde{\boldsymbol{W}}^{T}\boldsymbol{F}^{-1}\widetilde{\boldsymbol{W}})+\gamma\,\mathbf{tanh}(\tilde{\boldsymbol{q}})^{T}\boldsymbol{M}(\boldsymbol{q})\dot{\tilde{\boldsymbol{q}}}. \tag{55}$$

To be able to apply the corollary presented in (41), our LFC must be proven to be both radially unbounded and decrescent. To work out the radially unbounded property, first, we focus on the fuzzy related term of (55), and by applying Property 7 of an SFC given in (22), the next inequalities can be developed:

$$\sum_{i=1}^{n}\int_{0}^{\tilde{q}_i}\phi(\xi_i,0)d\xi_i \geq \sum_{i=1}^{n}\alpha_i\int_{0}^{\tilde{q}_i}\tanh(\xi_i)d\xi_i \tag{56}$$
$$=\sum_{i=1}^{n}\alpha_i|\ln(\cosh(\tilde{q}_i))| \geq \sum_{i=1}^{n}\alpha_i\beta_i|\tanh(\tilde{q}_i)|^2 \tag{57}$$
$$\geq \lambda_{\min}\{\boldsymbol{AB}\}\|\mathbf{tanh}(\tilde{\boldsymbol{q}})\|^2. \tag{58}$$

In (57), we use $\ln\{\cosh(\tilde{q}_i)\}\geq\beta_i\tanh(\tilde{q}_i)^2$, with $0\leq\beta_i\leq\frac{1}{2}$, and $\boldsymbol{B}=\mathrm{diag}\{\beta_i\}$, for $i=1,2,\ldots n$.

Upon partitioning the integral term in (55) into two halves, using the first part of the result of (57) in the first half of this partition and the result of (58) in the other one, and applying the bounds defined in Property A of the robot dynamic model (3), we have

$$U(\cdot)\geq\frac{1}{2}\lambda_{\min}\{\boldsymbol{M}\}\|\dot{\tilde{\boldsymbol{q}}}\|^2+\frac{1}{2}\Big(\sum_{i=1}^{n}\alpha_i|\ln(\cosh(\tilde{q}_i))|+\lambda_{\min}\{\boldsymbol{AB}\}\|\mathbf{tanh}(\tilde{\boldsymbol{q}})\|^2\Big)$$
$$+\frac{1}{2}\mathrm{tr}(\widetilde{\boldsymbol{V}}^{T}\boldsymbol{R}^{-1}\widetilde{\boldsymbol{V}})+\frac{1}{2}\mathrm{tr}(\widetilde{\boldsymbol{W}}^{T}\boldsymbol{F}^{-1}\widetilde{\boldsymbol{W}})-\gamma\lambda_{\max}\{\boldsymbol{M}\}\|\mathbf{tanh}(\tilde{\boldsymbol{q}})\|\|\dot{\tilde{\boldsymbol{q}}}\|. \tag{59}$$

We organize (59)'s quadratic form,

$$U(\cdot)\geq\frac{1}{2}\Bigg(\begin{bmatrix}\|\mathbf{tanh}(\tilde{\boldsymbol{q}})\|\\ \|\dot{\tilde{\boldsymbol{q}}}\|\end{bmatrix}^{T}\boldsymbol{q_1}\begin{bmatrix}\|\mathbf{tanh}(\tilde{\boldsymbol{q}})\|\\ \|\dot{\tilde{\boldsymbol{q}}}\|\end{bmatrix}$$
$$+\sum_{i=1}^{n}\alpha_i|\ln(\cosh(\tilde{q}_i))|+\mathrm{tr}(\widetilde{\boldsymbol{V}}^{T}\boldsymbol{R}^{-1}\widetilde{\boldsymbol{V}})+\mathrm{tr}(\widetilde{\boldsymbol{W}}^{T}\boldsymbol{F}^{-1}\widetilde{\boldsymbol{W}})\Bigg) \tag{60}$$

with

$$\boldsymbol{q_1}=\begin{bmatrix}\lambda_{\min}\{\boldsymbol{AB}\} & -\gamma\lambda_{\max}\{\boldsymbol{M}\}\\ -\gamma\lambda_{\max}\{\boldsymbol{M}\} & \lambda_{\min}\{\boldsymbol{M}\}\end{bmatrix} \tag{61}$$

where $\boldsymbol{q_1}$ is positive-definite if, first, $\lambda_{\min}\{\boldsymbol{AB}\}>0$, which is already fulfilled, since $\boldsymbol{A},\boldsymbol{B}>0$, and second,

$$\det(\boldsymbol{q_1})=\lambda_{\min}\{\boldsymbol{M}\}\lambda_{\min}\{\boldsymbol{AB}\}-\gamma^2\lambda_{\max}\{\boldsymbol{M}\}^2>0 \tag{62}$$

holds. Calculating γ from (62), we have

$$0<\gamma<\frac{\sqrt{\lambda_{\min}\{\boldsymbol{M}\}\lambda_{\min}\{\boldsymbol{AB}\}}}{\lambda_{\max}\{\boldsymbol{M}\}} \tag{63}$$

The existence of this value of γ ensures that our proposed LFC, given in (55), is both globally positive-definite and radially unbounded. To prove that $U(\tilde{\boldsymbol{q}},\dot{\tilde{\boldsymbol{q}}},\widetilde{\boldsymbol{V}},\widetilde{\boldsymbol{W}},t)$ is also a

decrescent function, we now use Property 3 of an SFC given in (20) with $\dot{\tilde{q}}_i = 0$, with the integral term of (55):

$$\int_0^{\tilde{q}_i} \phi(\xi_i, 0) d\xi_i \leq \rho_i \int_0^{\tilde{q}_i} \xi_i d\xi_i \leq \frac{\rho_i}{2} \tilde{q}_i^2 \tag{64}$$

Applying this result to (55), the upper bound of the LFC is given by

$$U(\cdot) \leq \frac{1}{2} \Bigg[\lambda_{\max}\{\boldsymbol{M}\} \|\dot{\tilde{\boldsymbol{q}}}\|^2 + \lambda_{\max}\{\boldsymbol{R_o}\} \|\tilde{\boldsymbol{q}}\|^2 + \operatorname{tr}\left(\widetilde{\boldsymbol{V}}^{\boldsymbol{T}} \boldsymbol{R}^{-\mathbf{1}} \widetilde{\boldsymbol{V}}\right) + \operatorname{tr}\left(\widetilde{\boldsymbol{W}}^{\boldsymbol{T}} \boldsymbol{F}^{-\mathbf{1}} \widetilde{\boldsymbol{W}}\right) \Bigg] + \gamma \lambda_{\max}\{\boldsymbol{M}\} \|\dot{\tilde{\boldsymbol{q}}}\| \tag{65}$$

with $\boldsymbol{R_o} = \operatorname{diag}\{\rho_i\}, i = 1 \ldots n$.

Thus, our proposed LFC, given in (55), is a globally positive-definite radially unbounded decrescent function that can be used with the corollary of the LaSalle–Yoshizawa stability theorem as a valid Lyapunov Function (LF) in the stability analysis of our control scheme proposal.

5.4. Time Derivative of the Lyapunov Function

In order to apply the corollary for the LaSalle–Yoshizawa stability theorem, it is necessary to calculate the time derivative of the LF in the trajectories of the dynamics of the system given in (54). Since our LF is fully differentiable, the generalized Clarke gradient reduces to the standard gradient. Now, applying Leibniz's rule to the integral term of (55), we have

$$\begin{aligned} \dot{U}(\tilde{\boldsymbol{q}}, \dot{\tilde{\boldsymbol{q}}}, \widetilde{\boldsymbol{V}}, \widetilde{\boldsymbol{W}}, t) \subset\ & \dot{\tilde{\boldsymbol{q}}}^{\boldsymbol{T}} \boldsymbol{M}(\boldsymbol{q}) \ddot{\tilde{\boldsymbol{q}}} + \frac{1}{2} \dot{\tilde{\boldsymbol{q}}}^{\boldsymbol{T}} \dot{\boldsymbol{M}}(\boldsymbol{q}) \dot{\tilde{\boldsymbol{q}}} + \boldsymbol{\Phi}(\tilde{\boldsymbol{q}}, \boldsymbol{0})^{\boldsymbol{T}} \dot{\tilde{\boldsymbol{q}}} \\ & + \operatorname{tr}\left(\widetilde{\boldsymbol{V}}^{\boldsymbol{T}} \boldsymbol{R}^{-\mathbf{1}} \dot{\widetilde{\boldsymbol{V}}}\right) + \operatorname{tr}\left(\widetilde{\boldsymbol{W}}^{\boldsymbol{T}} \boldsymbol{F}^{-\mathbf{1}} \dot{\widetilde{\boldsymbol{W}}}\right) \\ & + \gamma \left[\mathbf{sech}^{\mathbf{2}}(\tilde{\boldsymbol{q}}) \dot{\tilde{\boldsymbol{q}}}\right]^{\boldsymbol{T}} \boldsymbol{M}(\boldsymbol{q}) \dot{\tilde{\boldsymbol{q}}} \\ & + \gamma\, \mathbf{tanh}(\tilde{\boldsymbol{q}})^{\boldsymbol{T}} \dot{\boldsymbol{M}}(\boldsymbol{q}) \dot{\tilde{\boldsymbol{q}}} + \gamma\, \mathbf{tanh}(\tilde{\boldsymbol{q}})^{\boldsymbol{T}} \boldsymbol{M}(\boldsymbol{q}) \ddot{\tilde{\boldsymbol{q}}} \end{aligned} \tag{66}$$

We substitute $\ddot{\tilde{\boldsymbol{q}}}$ from the closed-loop equation in (51), apply the properties of the robot model dynamics defined in (5) and (6), and simplify

$$\begin{aligned} \dot{U}(\cdot) \subset\ & -\dot{\tilde{\boldsymbol{q}}}^{\boldsymbol{T}} \left[\boldsymbol{\Phi}(\tilde{\boldsymbol{q}}, \dot{\tilde{\boldsymbol{q}}}) - \boldsymbol{\Phi}(\tilde{\boldsymbol{q}}, \boldsymbol{0})\right] - \dot{\tilde{\boldsymbol{q}}}^{\boldsymbol{T}} \boldsymbol{F}_{\boldsymbol{v}} \dot{\tilde{\boldsymbol{q}}} - \gamma\, \mathbf{tanh}(\tilde{\boldsymbol{q}})^{\boldsymbol{T}} \boldsymbol{F}_{\boldsymbol{v}} \dot{\tilde{\boldsymbol{q}}} \\ & - \dot{\tilde{\boldsymbol{q}}}^{\boldsymbol{T}} \boldsymbol{h}(\tilde{\boldsymbol{q}}, \dot{\tilde{\boldsymbol{q}}}) - \gamma\, \mathbf{tanh}(\tilde{\boldsymbol{q}})^{\boldsymbol{T}} \boldsymbol{h}(\tilde{\boldsymbol{q}}, \dot{\tilde{\boldsymbol{q}}}) + \gamma\, \mathbf{tanh}(\tilde{\boldsymbol{q}})^{\boldsymbol{T}} \boldsymbol{C}^{\boldsymbol{T}}(\tilde{\boldsymbol{q}}, \dot{\tilde{\boldsymbol{q}}}) \dot{\tilde{\boldsymbol{q}}} \\ & + \gamma \dot{\tilde{\boldsymbol{q}}}^{\boldsymbol{T}} \mathbf{sech}^{\mathbf{2}}(\tilde{\boldsymbol{q}}) \boldsymbol{M}(\boldsymbol{q}) \dot{\tilde{\boldsymbol{q}}} - \gamma\, \mathbf{tanh}(\tilde{\boldsymbol{q}})^{\boldsymbol{T}} \boldsymbol{\Phi}(\tilde{\boldsymbol{q}}, \dot{\tilde{\boldsymbol{q}}}) - \operatorname{tr}\left(\widetilde{\boldsymbol{V}}^{\boldsymbol{T}} \boldsymbol{R}^{-\mathbf{1}} \dot{\widetilde{\boldsymbol{V}}}\right) \\ & - \operatorname{tr}\left(\widetilde{\boldsymbol{W}}^{\boldsymbol{T}} \boldsymbol{F}^{-\mathbf{1}} \dot{\widehat{\boldsymbol{W}}}\right) + \boldsymbol{r}^{\boldsymbol{T}} \left(\widetilde{\boldsymbol{W}}^{\boldsymbol{T}} \widehat{\boldsymbol{\sigma}} - \widetilde{\boldsymbol{W}}^{\boldsymbol{T}} \widehat{\boldsymbol{\sigma}}' \widehat{\boldsymbol{V}}^{\boldsymbol{T}} \boldsymbol{x}_{\boldsymbol{d}} + \widehat{\boldsymbol{W}}^{\boldsymbol{T}} \widehat{\boldsymbol{\sigma}}' \widetilde{\boldsymbol{V}}^{\boldsymbol{T}} \boldsymbol{x}_{\boldsymbol{d}}\right) \\ & + \boldsymbol{r}^{\boldsymbol{T}} [\boldsymbol{\omega}(t) - \boldsymbol{\Delta} \mathbf{SIGN}(\boldsymbol{r})] \end{aligned} \tag{67}$$

with $\boldsymbol{r}$ defined as in (37), and

$$\mathbf{SIGN}(\boldsymbol{r}) = K[\mathbf{sign}](\boldsymbol{r}) = \begin{bmatrix} \operatorname{SIGN}(r_1) \\ \operatorname{SIGN}(r_2) \\ \operatorname{SIGN}(r_3) \\ \vdots \\ \operatorname{SIGN}(r_n) \end{bmatrix} \tag{68}$$

where

$$\text{SIGN}(r_j) = \begin{cases} 1 & ; r_j > 0 \\ [1 \ -1] & ; r_j = 0 \\ -1 & ; r_j < 0 \end{cases} \tag{69}$$

Using the properties between the matrix trace and the transposition to the multiplication of vectors, substituting the functions for $\dot{\hat{V}}^T$ and $\dot{\hat{W}}^T$, defined in (35) and (36), respectively, yields

$$\begin{aligned} \dot{U}(\cdot) \subset & -\dot{\tilde{\boldsymbol{q}}}^T\left[\boldsymbol{\Phi}(\tilde{\boldsymbol{q}},\dot{\tilde{\boldsymbol{q}}}) - \boldsymbol{\Phi}(\tilde{\boldsymbol{q}},\boldsymbol{0})\right] - \dot{\tilde{\boldsymbol{q}}}^T\boldsymbol{F}_v\dot{\tilde{\boldsymbol{q}}} - \gamma\,\mathbf{tanh}(\tilde{\boldsymbol{q}})^T\boldsymbol{F}_v\dot{\tilde{\boldsymbol{q}}} - \dot{\tilde{\boldsymbol{q}}}^T\boldsymbol{h}(\tilde{\boldsymbol{q}},\dot{\tilde{\boldsymbol{q}}}) \\ & - \gamma\,\mathbf{tanh}(\tilde{\boldsymbol{q}})^T\boldsymbol{h}(\tilde{\boldsymbol{q}},\dot{\tilde{\boldsymbol{q}}}) + \gamma\,\mathbf{tanh}(\tilde{\boldsymbol{q}})^T\boldsymbol{C}^T(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\tilde{\boldsymbol{q}}} \\ & + \gamma\dot{\tilde{\boldsymbol{q}}}^T\mathbf{sech}^2(\tilde{\boldsymbol{q}})\boldsymbol{M}(\boldsymbol{q})\dot{\tilde{\boldsymbol{q}}} - \gamma\,\mathbf{tanh}(\tilde{\boldsymbol{q}})^T\boldsymbol{\Phi}(\tilde{\boldsymbol{q}},\dot{\tilde{\boldsymbol{q}}}) \\ & + \boldsymbol{r}^T[\boldsymbol{\omega}(t) - \boldsymbol{\Delta}\mathbf{SIGN}(\boldsymbol{r})] \end{aligned} \tag{70}$$

For the last term in (53), by applying the Cauchy–Schwarz inequality, given that the norms $\|\boldsymbol{r}\| \leq \|\boldsymbol{r}\|_1$ are equivalent, and also the bound for $\boldsymbol{\omega(t)}$ given in (52), we have

$$\boldsymbol{r}^T[\boldsymbol{\omega}(t) - \boldsymbol{\Delta}\mathbf{SIGN}(\boldsymbol{r})] \leq -[\lambda_{\min}\{\boldsymbol{\Delta}\} - k_\omega]\sum_{j=0}^{n}|r_j| \tag{71}$$

Applying the relationships defined in (4), (10), and (53), and the bounds in (26), (30), and (71), we have

$$\begin{aligned} \dot{U}(\cdot) \leq & -\dot{\tilde{\boldsymbol{q}}}^T\left[\boldsymbol{\Phi}(\tilde{\boldsymbol{q}},\dot{\tilde{\boldsymbol{q}}}) - \boldsymbol{\Phi}(\tilde{\boldsymbol{q}},\boldsymbol{0})\right] - \gamma(\lambda_{\min}\{\boldsymbol{A}\} - k_{h2})\|\mathbf{tanh}(\tilde{\boldsymbol{q}})^2\| \\ & + \gamma\left(\lambda_{\max}\{\boldsymbol{Z}\} + \lambda_{\max}\{\boldsymbol{F}_v\} + \frac{k_{h2}}{\gamma} + k_{h1} + k_{C1}\|\dot{\boldsymbol{q}}_M\|\right)\|\mathbf{tanh}(\tilde{\boldsymbol{q}})\|\|\dot{\tilde{\boldsymbol{q}}}\| \\ & - \gamma\left(\frac{\lambda_{\min}\{\boldsymbol{F}_v\} - k_{h1}}{\gamma} - \sqrt{n}k_{C1} - \lambda_{\max}\{\boldsymbol{M}\}\right)\|\dot{\tilde{\boldsymbol{q}}}\|^2 \\ & - [\lambda_{\min}\{\boldsymbol{\Delta}\} - k_\omega]\sum_{j=0}^{n}|r_j|. \end{aligned} \tag{72}$$

Defining the terms

$$a = \lambda_{\min}\{\boldsymbol{A}\} - k_{h2}, \ \ b = k_{h2} \tag{73}$$

$$c = \lambda_{\max}\{\boldsymbol{Z}\} + \lambda_{\max}\{\boldsymbol{F}_v\} + k_{h1} + k_{C1}\|\dot{\boldsymbol{q}}_M\| \tag{74}$$

$$d = \lambda_{\min}\{\boldsymbol{F}_v\} - k_{h1}, \ \ e = \sqrt{n}k_{C1} + \lambda_{\max}\{\boldsymbol{M}\} \tag{75}$$

and substituting them in (72), we have

$$\begin{aligned} \dot{U}(\cdot) \leq & -\dot{\tilde{\boldsymbol{q}}}^T\left[\boldsymbol{\Phi}(\tilde{\boldsymbol{q}},\dot{\tilde{\boldsymbol{q}}}) - \boldsymbol{\Phi}(\tilde{\boldsymbol{q}},\boldsymbol{0})\right] \\ & - \gamma\begin{bmatrix}\|\mathbf{tanh}(\tilde{\boldsymbol{q}})\| \\ \|\dot{\tilde{\boldsymbol{q}}}\|\end{bmatrix}^T\boldsymbol{q_3}\begin{bmatrix}\|\mathbf{tanh}(\tilde{\boldsymbol{q}})\| \\ \|\dot{\tilde{\boldsymbol{q}}}\|\end{bmatrix} - [\lambda_{\min}\{\boldsymbol{\Delta}\} - k_\omega]\sum_{j=0}^{n}|r_j| \end{aligned} \tag{76}$$

with

$$\boldsymbol{q_3} = \begin{bmatrix} a & -\frac{\frac{b}{\gamma}+c}{2} \\ -\frac{\frac{b}{\gamma}+c}{2} & \frac{d}{\gamma} - e \end{bmatrix} \tag{77}$$

Since Property 3 of SFCs holds for the first term of (76),

$$\lambda_{\min}\{\boldsymbol{\Delta}\} \geq k_\omega, \ Q_3 > 0 \implies \dot{U}(\cdot) < 0.$$

Thus, we can derive the following relationships

$$\begin{aligned} \lambda_{\min}\{\boldsymbol{A}\} &> k_{h2} \implies a > 0 \\ \lambda_{\min}\{\boldsymbol{F_v}\} &> \gamma(\sqrt{n}k_{C1} + \lambda_{\max}\{\boldsymbol{M}\}) + k_{h1} \implies \frac{d}{\gamma} - e > 0 \\ \det(Q_3) &> 0 \end{aligned}$$

where

$$\det(Q_3) = a\left(\frac{d}{\gamma} - e\right) - \frac{\left(\frac{b}{\gamma} + c\right)^2}{4} > 0. \tag{78}$$

Obtaining $\lambda_{\min}\{\boldsymbol{A}\}$ from (73) and (78), we obtain:

$$\lambda_{\min}\{\boldsymbol{A}\} > k_{h2} + \frac{(k_{h2} + \gamma c)^2}{4\gamma\left(\lambda_{\min}\{\boldsymbol{F_v}\} - \gamma\left[\sqrt{n}k_{C1} - \lambda_{\max}\{\boldsymbol{M}\}\right] - k_{h1}\right)}. \tag{79}$$

$$\dot{\tilde{U}}(\cdot) \leq \underbrace{-\gamma \begin{bmatrix} \|\tanh(\tilde{\boldsymbol{q}})\| \\ \|\dot{\tilde{\boldsymbol{q}}}\| \end{bmatrix}^T \boldsymbol{q_3} \begin{bmatrix} \|\tanh(\tilde{\boldsymbol{q}})\| \\ \|\dot{\tilde{\boldsymbol{q}}}\| \end{bmatrix}}_{G_3(\cdot)}. \tag{80}$$

Since $\boldsymbol{q_3}$ is positive-definite, the function $G_3(\cdot)$ that must be found, according to the the LaSalle–Yoshizawa corollary for nonsmooth systems, is also positive-definite. Therefore, it is possible to state that the angular position and velocity errors globally converge to zero:

$$\lim_{t\to\infty} \|\tilde{\boldsymbol{q}}\| = 0,$$
$$\lim_{t\to\infty} \|\dot{\tilde{\boldsymbol{q}}}\| = 0.$$

Remark 1. *The value of $\lambda_{min}\{\boldsymbol{\Delta}\}$ is determined during the experimental stage and is used as a tuning parameter. A gain that is smaller than necessary would theoretically result in input-to-state stability [33], which implies that a smaller-than-necessary gain would not make the system's solutions grow unbounded under the proposed control action. This allows for appropriate implementation of the control law. Also, both input and output weights, $\boldsymbol{V}$ and $\boldsymbol{W}$, must be initialized, and in most cases, the whole ANN will be pretrained off-line to learn the desired dynamics; therefore, $\tilde{\boldsymbol{W}}$ in $t = t_0$ will be very small. If $\boldsymbol{W}$ were initialized in zero, as is the case in many off-line training algorithms, the maximum value of the spectral norm of $\tilde{\boldsymbol{W}}$ would be the same as the upper bound of the whole robot norm.*

6. Controller Design, Simulation, and Experimentation

To asses the performance of our new proposal, we implemented it on a 2-DOF robot manipulator in a comparative fashion versus two other controllers, the PD plus feedforward controller and the SFC plus feedforward controller, and its design is described in the following subsections along with a description of the robot used.

6.1. 2-DOF Robot Manipulator Description

All control schemes to be compared were implemented for a 2-DOF robot manipulator that operates only vertically, as shown in Figure 3 [1,10,34,35]. The robot manipulator is composed of two rigid links. Every joint is driven by high-torque brushless direct-drive servos with no gear reduction, which reduces backlash and significantly diminishes joint friction compared to using gear drives. The maximum torques that can be applied, according to the manufacturer, is 150 [N–m] for joint 1 and 15 [N–m] for joint 2. The values for all the mechanical parameters of this robot are given in Table 1.

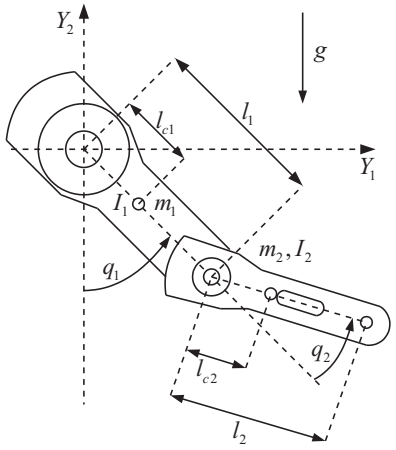


Figure 3. Diagram of 2-DOF robot manipulator.

Table 1. The parameter values for the 2-DOF robot manipulator.

Notation	Value	Units
l_1	0.450	m
l_2	0.450	m
l_{c1}	0.091	m
l_{c2}	0.091	m
m_1	23.902	Kg
m_2	3.880	Kg
I_1	1.266	Kg m^2
I_2	0.093	Kg m^2
fv_1	2.288	N-m s
fv_2	0.175	N-m s
g	9.81	m/s^2

The dynamical model of the robot manipulator shown in Figure 3 can be expressed as

$$\underbrace{\begin{bmatrix} M_{11}(q) & M_{12}(q) \\ M_{21}(q) & M_{22}(q) \end{bmatrix}}_{\boldsymbol{M(q)}} \ddot{\boldsymbol{q}} + \underbrace{\begin{bmatrix} C_{11}(q,\dot{q}) & C_{12}(q,\dot{q}) \\ C_{21}(q,\dot{q}) & C_{22}(q,\dot{q}) \end{bmatrix}}_{\boldsymbol{C(q,\dot{q})}} \dot{\boldsymbol{q}} + \underbrace{\begin{bmatrix} g_1(q) \\ g_2(q) \end{bmatrix}}_{\boldsymbol{g(q)}} + \underbrace{\begin{bmatrix} fv_1 & 0 \\ 0 & fv_2 \end{bmatrix}}_{\boldsymbol{F_v}} \dot{\boldsymbol{q}} = \boldsymbol{\tau} \quad (81)$$

with

$$\begin{aligned}
M_{11}(q) &= m_1 l_{c1}^2 + m_2[l_1^2 + l_{c2}^2 + 2l_1 l_{c2}\cos(q_2)] + I_1 + I_2, \\
M_{12}(q) &= m_2[l_{c2}^2 + l_1 l_{c2}\cos(q_2)] + I_2, \\
M_{21}(q) &= m_2[l_{c2}^2 + l_1 l_{c2}\cos(q_2)] + I_2, \\
M_{22}(q) &= m_2 l_{c2}^2 + I_2, \\
C_{11}(q,\dot{q}) &= -m_2 l_1 l_{c2}\sin(q_2)\dot{q}_2, \\
C_{12}(q,\dot{q}) &= -m_2 l_1 l_{c2}\sin(q_2)[\dot{q}_1 + \dot{q}_2] \\
C_{21}(q,\dot{q}) &= m_2 l_1 l_{c2}\sin(q_2)\dot{q}_1, \\
C_{22}(q,\dot{q}) &= 0, \\
g_1(q) &= [m_1 l_{c1} + m_2 l_1]g\sin(q_1) + m_2 l_{c2}\, g\sin(q_1 + q_2), \\
g_2(q) &= m_2 l_{c2} g\sin(q_1 + q_2),
\end{aligned}$$

In this case, the Coulomb friction terms were neglected and will be considered as a disturbance within the robot model that the controllers to be tested must overcome.

6.2. PD Plus Feedforward Controller Design

For this controller, its gain matrices $\boldsymbol{K_p}, \boldsymbol{K_v} \in \mathbb{R}^{2\times 2}$ were assigned the values

$$\boldsymbol{K_p} = \begin{bmatrix} 225 & 0 \\ 0 & 10 \end{bmatrix} \quad \boldsymbol{K_v} = \begin{bmatrix} 70 & 0 \\ 0 & 4 \end{bmatrix} \tag{82}$$

These values were proposed based on the minima defined in [34] so that the torques applied to joints 1 and 2 of the robot manipulator do not exceed their limit values of 150 N-m and 15 N-m, respectively.

6.3. Proposal of New Tracking Controller with Feedforward Compensation Based on SFC and Adaptive NN

For this controller, we used the design already carried out in our previous manuscript [10,35]; the second reference was only supported by experimental tests; in contrast, the current manuscript presents a formal stability analysis guaranteeing that position and velocity errors converge to zero and all signals are uniformly bounded. The MFs for each input of each joint and for the output were defined as shown in Figures 4, 5, and 6, respectively. The output MF was defined by singletons to expedite the complete output calculation in real time. In the definition of fuzzy sets, the acronyms used for each MF, for both inputs and the output, are as follows: NB = Negative Big; NS = Negative Small; Z = Zero; PS = Positive Small; and PB = Positive Big.

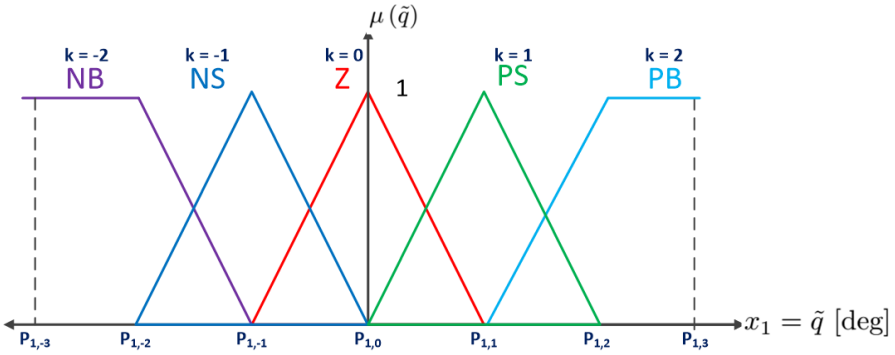


Figure 4. Fuzzy sets and their partitions for input $x_1 = \tilde{q}_i$.

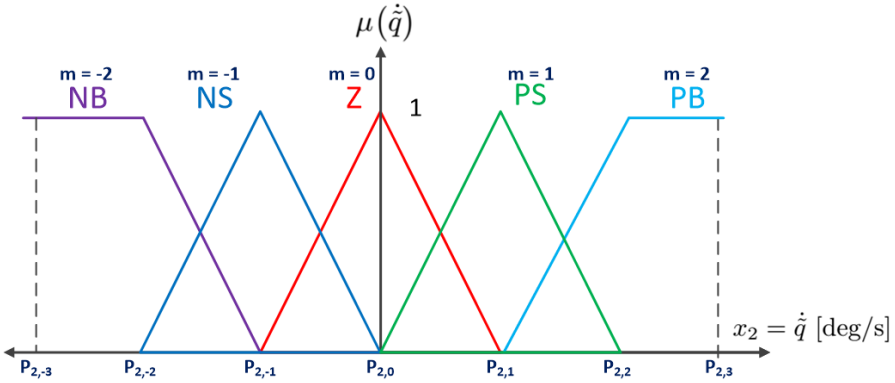


Figure 5. Fuzzy sets and their partitions for input $x_2 = \dot{\tilde{q}}_i$.

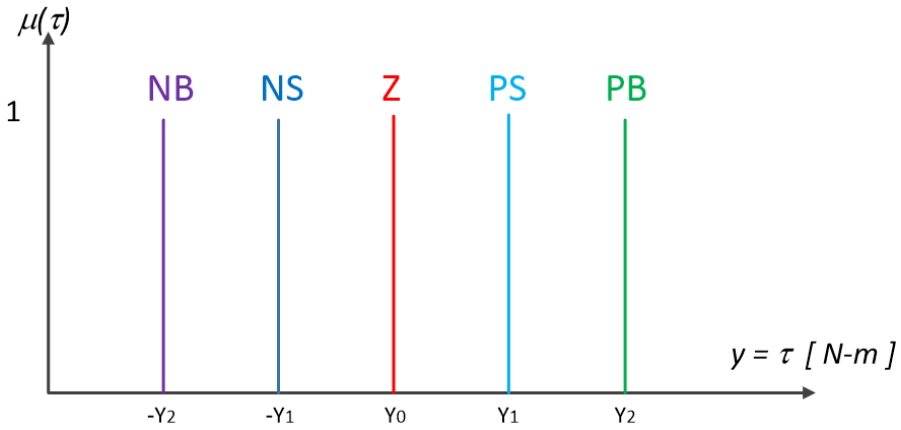


Figure 6. Fuzzy sets and their partitions for the output τ.

The fuzzy rules were defined as shown in Table 2 so that they followed the guidelines indicated in Section 3 to define an SFC.

Table 2. Fuzzy rules look-up table.

$\dot{\tilde{q}}/\tilde{q}$	NB	NS	Z	PS	PB
NB	NB	NB	NS	Z	Z
NS	NB	NB	NS	Z	Z
Z	NS	NS	Z	PS	PS
PS	Z	Z	PS	PB	PB
PB	Z	Z	PS	PB	PB

The base values for the fuzzy sets were calculated using Genetic Algorithms (GAs), similar to those previously shown in [36]. The final support values for the fuzzy sets depicted in Figures 4–6 are given in Table 3 for joint 1 and in Table 4 for joint 2.

Table 3. Fuzzy set partition values for joint 1.

Partition	Value	Units
$P_{1,0}$	0	degrees
$P_{1,1}$	6.518	degrees
$P_{1,2}$	53.770	degrees
$P_{1,3}$	125.5	degrees
$P_{2,0}$	0	degrees/s
$P_{2,1}$	122.2	degrees/s
$P_{2,2}$	138.5	degrees/s
$P_{2,3}$	871.8	degrees/s
Y_0	0	N-m
Y_1	82.29	N-m
Y_2	204.5	N-m

Table 4. Fuzzy set partition values for joint 2.

Partition	Value	Units
$P_{1,0}$	0	degrees
$P_{1,1}$	5.982	degrees
$P_{1,2}$	36.67	degrees
$P_{1,3}$	163.5	degrees
$P_{2,0}$	0	degrees/s
$P_{2,1}$	153.8	degrees/s
$P_{2,2}$	318.7	degrees/s
$P_{2,3}$	1016	degrees/s
Y_0	0	N-m
Y_1	15	N-m
Y_2	180	N-m

6.4. SFC with Adaptive NN Compensation Controller Design

For this control scheme, we first used all MFs that were previously designed for the SFC plus feedforward controller, along with the fuzzy rules table and MFs for the outputs. For the adaptive NN compensation, $N = 3n = 3(2) = 6$, with $n = 2$, since the robot manipulator is composed of two links. We tried different activation functions, and finally selected $\sigma(\boldsymbol{x}) = \tanh(\boldsymbol{x})$ because it was the one that performed the best. Also, we tested our controller in a simulation with different numbers of neurons, and set the number of neurons to be used to 20; therefore, $L = 20$. The NN was first pre-trained in the simulation several times using the whole controller with the SFC block designed in the previous section until we obtained suitable values for the weights that avoided both start-up overshoots and surpassing the maximum allowed values for torques that can be applied to the two joints; in addition, they resulted in a fast response time and lower steady-state errors. The other design matrices and scalars that were part of the control law or adaptive dynamics of our proposed controller were given the following values:

$$\alpha = 7.7,\ \gamma = 0.4,\ \boldsymbol{\Delta} = \text{diag}\{0.69\}, \boldsymbol{R} = \text{diag}(2.6),\ \ \boldsymbol{F} = \text{diag}(2.2) \tag{83}$$

6.5. Simulation of the Controllers

We used the model of the 2-DOF robot manipulator described in Section 6.1, including both viscous and Coulomb frictions. The desired angular position, angular velocity, and joint acceleration trajectories, $\boldsymbol{q_d}(t)$, $\dot{\boldsymbol{q}}_{\boldsymbol{d}}(t)$, and $\ddot{\boldsymbol{q}}_{\boldsymbol{d}}(t)$, are given in the following equations, according to the values and functions proposed in [1], in order to achieve the maximum allowable performance for this specific robot manipulator:

$$q_{1\mathrm{d}}(t) = a_1 + b_1(1 - e^{-d_1 t^3}) + c_1(1 - e^{-d_1 t^3})\sin(\omega_1 t)\ \text{[rad]}$$
$$q_{2\mathrm{d}}(t) = a_2 + b_2(1 - e^{-d_2 t^3}) + c_2(1 - e^{-d_2 t^3})\sin(\omega_2 t)\ \text{[rad]}$$

where $a_1 = \pi/2$ [rad], $b_1 = \pi/4$ [rad], $c_1 = \pi/18$ [rad], $d_1 = 2$, $\omega_1 = 15$ [rad/s], $a_2 = \pi/2$ [rad], $b_2 = \pi/3$ [rad], $c_2 = 25\pi/36$ [rad], $d_2 = 1.8$, and $\omega_2 = 3.5$ [rad/s]. The desired angular velocities and accelerations were analytically computed by calculating the direct derivative of the desired angular positions.

All controllers were implemented and simulated in the Simulink software of MATLAB R2015a environment ® (MATLAB is a registered trademark of MathWorks, Inc., Natick, MA, USA). and their angular positions, angular position errors, and applied torques in each

joint were obtained numerically and graphically. A comparison of the Root Mean Square (RMS) of every angular position error, computed using the function *rms* from Matlab for each joint, is shown in a comparative fashion for every controller tested in Table 5. All values are the RMS; the suffix "ss" represents steady-state values, calculated from 5 s to the end of the simulation time window of 10 s.

Table 5. Position error RMS comparison.

Controller	$\tilde{q}_1$	$\tilde{q}_{1,ss}$	$\tilde{q}_2$	$\tilde{q}_{2,ss}$
PD + ff	20.8706°	0.6086°	20.4219°	3.7940°
Sectorial	14.3812°	0.6217°	15.8117°	0.5126°
Neuro + Sectorial	16.8950°	0.5724°	16.2417°	0.3278°

Both the SFC plus feedforward controller and the SFC plus adaptive NN compensation controllers show better position angular error performance than the original controller they are derived from, the PD plus feedforward controller. The SFC plus feedforward controller presents smaller values of angular position errors on both joints in comparison to the other controllers for the whole simulation, whereas the SFC plus adaptive NN compensation is the controller that has the smallest steady-state angular position errors thanks to the adaptive NN feedforward compensation. The SFC plus adaptive NN compensation has slightly larger total angular position errors than its SFC plus feedforward counterpart because the adaptive NN takes a moment to adapt, while the feedforward compensation in the second controller already has the complete model from the start.

6.6. Comparison of the Experimental Performance of the Controllers

All previously designed and simulated controllers were implemented on a PC-based real-time platform for real-time simulation and control of the 2-DOF robot manipulator (see Figure 7) in a real-time Microsoft Windows application called WinMechLab running on a old 800 MHz dual-core desktop personal computer. Both $W(t)$ and $V(t)$ were computed by integrating the weight update dynamics $\dot{W}(t)$ and $\dot{V}(t)$ defined in (35) and (36), respectively, through typical rectangular numeric integration. The sample period used was 2.5 ms (sampling frequency of 400 Hz). A MultiQ-PCI data acquisition board from Quanser Consulting, Inc, was used as the digital input/output hardware interface. The angular position error responses were obtained for each joint, as were the applied torques, as shown in Figures 8–11.



Figure 7. Physical 2-DOF robot manipulator.

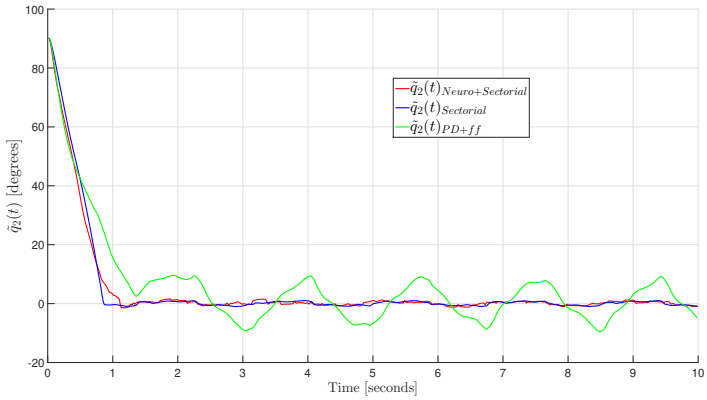


Figure 8. A comparison of the angular position errors for all controllers on joint 1.

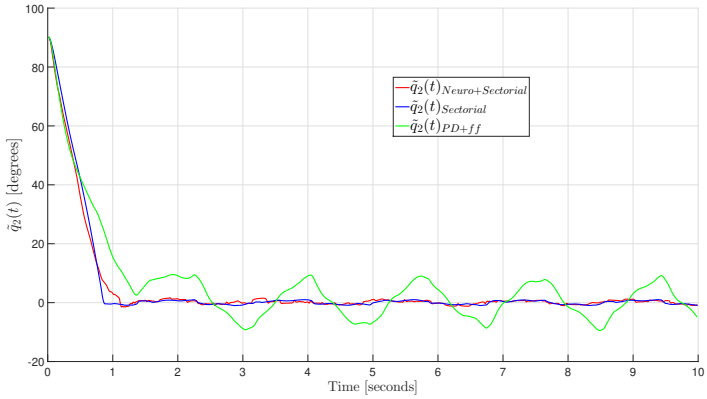


Figure 9. A comparison of the angular position errors for all controllers on joint 2.

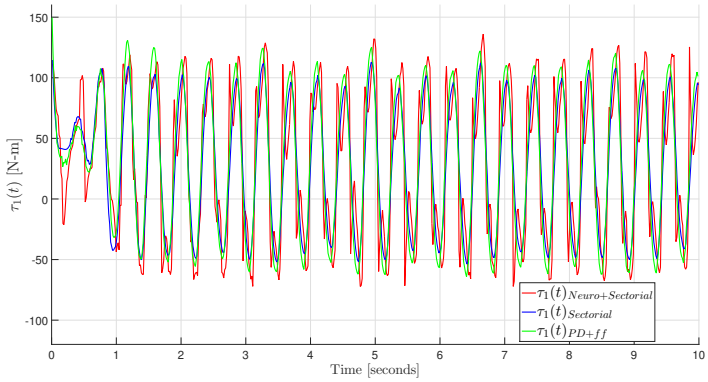


Figure 10. A comparison of the torque applied to joint 1 for all of the controllers.

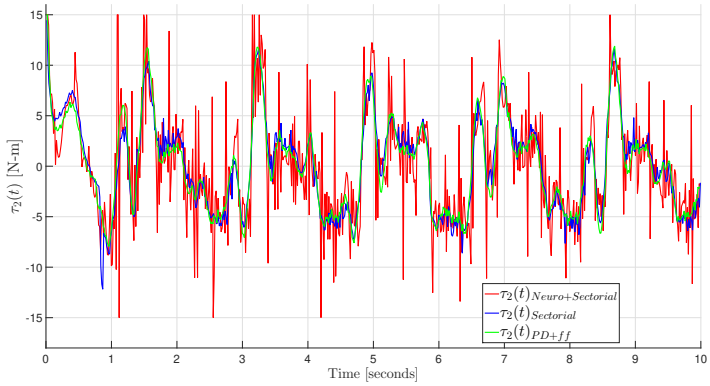


Figure 11. A comparison of the torque applied to joint 2 for all of the the controllers.

In Table 6, a comparison of the position error RMS in each joint is shown for each controller. Again, all values are the RMS, obtained using the function *rms* in Matlab, and "ss" stands for steady-state values, which were computed from 5 s to the end of the time window of 10 s for the experiment.

Table 6. Position error RMS comparison.

Controller	$\tilde{q}_1$	$\tilde{q}_{1,ss}$	$\tilde{q}_2$	$\tilde{q}_{2,ss}$
PD + ff	17.1754	1.4025	17.1683	5.3153
Sectorial	15.218	1.1870	15.9701	0.6231
Neuro + Sectorial	16.7607	2.1181	15.1606	0.6647

Again, both the angular position errors and torques applied to joints 1 and 2 have smaller values for the SFC plus feedforward controller and the SFC plus adaptive NN compensation compared to the PD plus feedforward controller. In the experimental case, the SFC plus adaptive NN yielded larger errors in both joints for both the complete and the steady-state analyses compared to the SFC plus feedforward controller, since it was affected by the bit-resolution available in the real-time platform we had available to compute the values of the NN weights. In the simulation, we were using a 64-bit double real-number definition, while in the real-time implementation, only single 32-bit real number resolution was available.

7. Concluding Remarks

An SFC plus adaptive NN feedforward controller applied to the trajectory tracking control of robotic manipulators is presented along with a formal stability proof applying the LaSalle–Yoshizawa corollaries for nonsmooth systems. We successfully evaluated our new controller proposal both in simulation and in real-time experiments, comparing it with a previous approach (SFC plus feedforward compensation of the robot model) and the original classic scheme (PD plus feedforward compensation of the robot model). As a result of the stability analysis and real-time experiments, we can highlight that our new control proposal excludes the need for knowledge of the robotic manipulator model to achieve excellent results. And in the case of the real-time experimental implementation of SFC plus feedforward compensation, there was no need to retune any control parameters obtained via simulation, unlike classical PD plus feedforward compensation, which required several changes to its control parameters in order to obtain decent experimental performance results.

Author Contributions: Conceptualization, A.P.-L., V.S. and R.G.-H.; methodology, A.P.-L., V.S. and R.G.-H.; software, A.P.-L., R.G.-H. and J.V.-C.; validation, A.P.-L., V.S., R.G.-H., J.V.-C. and J.M.-V.; formal analysis, A.P.-L., V.S. and J.V.-C.; investigation, A.P.-L., V.S., R.G.-H., J.V.-C. and J.M.-V.; resources, V.S. and R.G.-H.; data curation, A.P.-L.; writing—original draft preparation, A.P.-L., V.S. and R.G.-H.; writing—review and editing, A.P.-L., V.S., R.G.-H. and J.V.-C.; visualization, A.P.-L. and J.V.-C.; supervision, V.S. and R.G.-H.; project administration, V.S. and R.G.-H.; funding acquisition, A.P.-L., V.S., R.G.-H., J.V.-C. and J.M.-V. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Programa para el Desarrollo Profesional Docente (PRODEP-México) under grant ITSON-126, Tecnológico Nacional de México (TecNM) projects, and the RICCA of TecNM.

Data Availability Statement: The experimental data generated during the current study are available from the corresponding author on reasonable request. All simulation files for Simulink and Matlab

used in this work are available at https://github.com/elothron/SFC-adaptive-NN-for-robot-arms (accessed on 3 March 2025).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ANN	Artificial neural network
DOF	Degree of Freedom
FDL	Feedback linearization
GA	Genetic Algorithm
GUAS	Global Uniform Asymptotic Stability
LFC	Lyapunov function candidate
MF	Membership function
NN	Neural network
PD	Proportional–Derivative
PID	Proportional–Integral–Derivative
RMS	Root Mean Square
SFC	Sectorial fuzzy controller
SGUUB	Semiglobally Uniformly Ultimately Bounded

References

1. Kelly, R.; Santibanez, V.; Loria, A. *Control of Robot Manipulators in Joint Space*; Springer-Verlag London Limited: Berlin/Heidelberg, Germany, 2005.
2. Tatjewski, P. *Advanced Control of Industrial Processes: Structures and Algorithms*; Springer-Verlag London Limited: London, UK, 2007.
3. Calcev, G. Some remarks on the stability of Mamdani fuzzy control systems. *IEEE Trans. Fuzzy Syst.* **1998**, *6*, 436–442. [CrossRef]
4. Calcev, G.; Gorez, R.; Neyer, M.D. Passivity approach to fuzzy control systems. *Automatica* **1998**, *34*, 339–344. [CrossRef]
5. Santibanez, V.; Kelly, R.; Llama, M. Global asymptotic stability of a tracking sectorial fuzzy controller for robot manipulators. *IEEE Trans. Syst. Man, Cybern. Part B Cybern.* **2004**, *34*, 710–718. [CrossRef] [PubMed]
6. Zheng, C.; Su, Y.; Mercorelli, P. A Simple Fuzzy Controller for Robot Manipulators with Bounded Inputs. In Proceedings of the Proc. IEEE International Conference on Advanced Intelligent Mechatronics (AIM2017), Munich, Germany, 3–7 July 2017; pp. 1737–1742.
7. Villalobos-Chin, J.; Pizarro-Lerma, A.; Santibáñez, V.; García-Hernández, R.; Zavala-Río, A. Continuous finite-time fuzzy control of mechanical systems via non-Lipschitz membership functions. *Eng. Appl. Artif. Intell.* **2023**, *126*, 106804. [CrossRef]
8. Jiang, H.; Ye-Hwa, C.; Xiaomin, Z.; Fangfang, D. Optimal design for robust control of uncertain flexible joint manipulators: A fuzzy dynamical system approach. *Int. J. Control* **2018**, *91*, 937–951. [CrossRef]
9. Yoshikawa, T. *Foundations of Robotics: Analysis and Control*; MIT Press: Cambridge, MA, USA, 1990.
10. Pizarro-Lerma, A.; Santibáñez, V.; Garcia-Hernandez, R.; Villalobos-Chin, J. Sectorial Fuzzy Controller Plus Feedforward for the Trajectory Tracking of Robotic Arms in Joint Space. *Mathematics* **2021**, *9*, 616. . [CrossRef]
11. Knights, V.A.; Petrovska, O.; Kljusurić, J.G. Nonlinear Dynamics and Machine Learning for Robotic Control Systems in IoT Applications. *Future Internet* **2024**, *16*, 435. [CrossRef]
12. Elsisi, M.; Mahmoud, K.; Lehtonen, M.; Darwish, M.M.F. An Improved Neural Network Algorithm to Efficiently Track Various Trajectories of Robot Manipulator Arms. *IEEE Access* **2021**, *9*, 11911–11920. [CrossRef]
13. Lewis, F.; Yesildirek, A.; Liu, K. Multilayer neural-net robot controller with guaranteed tracking performance. *IEEE Trans. Neural Netw.* **1996**, *7*, 388–399. [CrossRef]
14. Puga-Guzmán, S.; Moreno-Valenzuela, J.; Santibanez, V. Adaptive neural network motion control of manipulators with experimental evaluations. *Sci. World J.* **2014**, *2014*, 1–13. [CrossRef]
15. Wei, S.; Wang, Y.; Zuo, Y. Wavelet neural networks robust control of farm transmission line deicing robot manipulators. *Comput. Stand. Interfaces* **2012**, *34*, 327–333. [CrossRef]
16. Li, X.; Cheah, C.C. Adaptive Neural Network Control of Robot Based on a Unified Objective Bound. *IEEE Trans. Control Syst. Technol.* **2014**, *22*, 1032–1043. [CrossRef]
17. He, W.; Dong, Y.; Sun, C. Adaptive Neural Impedance Control of a Robotic Manipulator with Input Saturation. *IEEE Trans. Syst. Man Cybern. Syst.* **2016**, *46*, 334–344. [CrossRef]

18. Lu, S.M.; Li, D.P.; Liu, Y.J. Adaptive Neural Network Control for Uncertain Time-Varying State Constrained Robotics Systems. *IEEE Trans. Syst. Man Cybern. Syst.* **2019**, *49*, 2511–2518. [CrossRef]
19. Sun, W.; Wu, Y.; Lv, X. Adaptive Neural Network Control for Full-State Constrained Robotic Manipulator With Actuator Saturation and Time-Varying Delays. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 3331–3342. [CrossRef] [PubMed]
20. Song, Q.; Li, S.; Bai, Q.; Yang, J.; Zhang, A.; Zhang, X.; Zhe, L. Trajectory Planning of Robot Manipulator Based on RBF Neural Network. *Entropy* **2021**, *23*, 1207. [CrossRef]
21. Hu, J.; Wang, P.; Xu, C.; Zhou, H.; Yao, J. High accuracy adaptive motion control for a robotic manipulator with model uncertainties based on multilayer neural network. *Asian J. Control* **2022**, *24*, 1503–1514. [CrossRef]
22. Galvan-Perez, D.; Yañez-Badillo, H.; Beltran-Carbajal, F.; Rivas-Cambero, I.; Favela-Contreras, A.; Tapia-Olvera, R. Neural Adaptive Robust Motion-Tracking Control for Robotic Manipulator Systems. *Actuators* **2022**, *11*, 255. [CrossRef]
23. Patil, O.S.; Le, D.M.; Greene, M.L.; Dixon, W.E. Lyapunov-Derived Control and Adaptive Update Laws for Inner and Outer Layer Weights of a Deep Neural Network. *IEEE Control Syst. Lett.* **2022**, *6*, 1855–1860. [CrossRef]
24. Fischer, N.; Kamalapurkar, R.; Dixon, W.E. Lasalle-yoshizawa corollaries for nonsmooth systems. *IEEE Trans. Autom. Control* **2013**, *58*, 2333–2338. [CrossRef]
25. Lewis, F.L.; Dawson, D.M.; Abdallah, C.T. *Robot Manipulator Control, Theory and Practice*, 2nd ed.; Revised and Expanded; Marcel Dekker, Inc.: New York, NY, USA; Basel, Switzerland, 2004.
26. Merabet, A.; Gu, J. *Advanced Nonlinear Control of Robot Manipulators, Robot Manipulators New Achievements*; InTech: Rijeka, Croatia, 2010.
27. Arimoto, S. Fundamental problems of robot control: Part I, Innovations in the realm of robot servo-loops. *Robotica* **1995**, *13*, 19–27. [CrossRef]
28. Funahashi, K.I. On the approximate realization of continuous mappings by neural networks. *Neural Netw.* **1989**, *2*, 183–192. [CrossRef]
29. Cybenko, G. Approximations by superpositions of a sigmoidal function. *Math. Control Signals Syst.* **1989**, *2*, 303–314. [CrossRef]
30. Pizarro-Lerma, A.O.; Santibanez, V.; Garcia-Hernandez, R.; Chin, J.V. Sectorial Fuzzy Controller Plus Feedforward applied to the Trajectory Tracking of Robot Manipulators. *IFAC-PapersOnLine* **2020**, *53*, 9918–9923. . [CrossRef]
31. Santibanez, V.; Kelly, R.; Llama, M.A. A novel global asymptotic stable set-point fuzzy controller with bounded torques for robot manipulators. *IEEE Trans. Fuzzy Syst.* **2005**, *13*, 362–372. [CrossRef]
32. Cortes, J. Discontinuous dynamical systems. *IEEE Control Syst. Mag.* **2008**, *28*, 36–73. [CrossRef]
33. Haddad, W.M.; Chellaboina, V. *Nonlinear Dynamical Systems and Control: A Lyapunov-Based Approach*; Mathematics, Princeton University Press: Princeton, NJ, USA, 2008.
34. Kelly, R.; Santibanez, V. *Control de Movimiento de Robots Manipuladores*; Pearson, Prentice Hall: Madrid, Spain, 2003.
35. Pizarro-Lerma, A.O.; Garcia-Hernandez, R.; Santibanez, V.; Chin, J.V. Experimental Evaluation of a Sectorial Fuzzy Controller Plus Adaptive Neural Network Compensation Applied to a 2-DOF Robot Manipulator. *IFAC-PapersOnLine* **2019**, *52*, 233–238.
36. Mester, G. Design of the fuzzy control systems based on genetic algorithm for intelligent robots. *Interdiscip. Descr. Complex Syst.* **2014**, *12*, 245–254. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.