Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 3240210, 14 pages
https://doi.org/10.1155/2020/3240210

Research Article

Neural Dynamics Variations Observer Designed for Robot Manipulator Control Using a Novel Saturated Control Technique

Francisco G. Rossomando,[1] **Emanuel Serrano**,[2] **Carlos M. Soria**,[1] **and Gustavo Scaglia**[2]

[1]*CONICET, Instituto de Automática, Universidad Nacional de San Juan, San Juan CP5400, Argentina*
[2]*CONICET, Instituto de Ingeniería Química, Universidad Nacional de San Juan, San Juan CP5400, Argentina*

Correspondence should be addressed to Francisco G. Rossomando; frosoma@inaut.unsj.edu.ar and Gustavo Scaglia; gscaglia@unsj.edu.ar

Received 11 February 2020; Revised 16 April 2020; Accepted 27 April 2020; Published 21 May 2020

Guest Editor: Rongwei Guo

This work presents a novel controller for the dynamics of robots using a dynamic variations observer. The proposed controller uses a saturated control law based on $\sin(\mathrm{tg}^{-1}(.))$ function instead of $\tanh(.)$. Besides, this function is an alternative to the use of $\tanh(.)$ in saturation control, since it reaches its maximum value more gradually than the hyperbolic tangent function. Using this characteristic, the transition between states is smoother, with similar accuracy to $\tanh(.)$. The controller is designed using a saturated SMC (sliding mode controller) and a dynamic variations observer based on GRNN (general regression neural network). The originality of this work is the use of a combination of adaptive GRNN with a sliding mode controller (SMC) including a new saturation function. Finally, experiments based on trajectory tracking demonstrate the robustness and simplicity of this method.

1. Introduction

Several years ago, the work of many researchers resulted in the development of various controllers for robots and mechatronic systems [1–6]. These algorithms were the first in the literature whose stability was proved using different theories [7–9] or using finite-time stability [10–13].

Moreover, some of these algorithms were implemented on different robot arms [14–16] and subsequent authors proposed different variants of adaptive control for robot manipulators [17–20]. Only a few of those algorithms solve the particular problems present in industrial robots; in this research work, our main objective is to reduce the lack of fidelity present in robot's arms that could vitiate any real benefits from a model-based controller. Because of this specific situation, some authors use some type of estimators to identify such infidelities between models (plant controller) or disturbances to improve controller efficiency. In this case, Mohammadi et al. [21] designed a general systematic disturbance observer for robot manipulators, where the proposed estimator doesn't have restrictions on the number of DOF, configuration, and joints. The disturbance observer has been designed based on the Linear Matrix Inequality (LMI) theory. This method presents different types of convergence depending on the disturbance characteristics, and this technique was designed to minimize large and fast disturbances while the controller drives the output to the desired reference. For efficient operation, the disturbances must be bounded. Also, as stated by Feemster et al. [22], it is necessary that the external disturbance has to be bounded and its period known.

Na et al. [23] proposed an unknown system dynamic estimator (USDE) based on simple algebraic calculations. This estimator can be modified by an unknown disturbance estimator (UDE) for external disturbances. Both estimators can adjust their parameters to approximate the unknown dynamics and disturbances, respectively. This work proposes these estimators in the design of composite controllers for trajectory tracking, but the estimations do not consider the accelerations of the joints.

Xie et al. [24] proposed a kinematic controller based on global asymptotic stability, which is combined with the speed-based adaptive dynamic compensation controller design. This work uses the $\sin(\mathrm{tg}^{-1}(.))$ function only in its kinematic controller.

Bussola et al. [25] used a redundant SCARA architecture to execute different tasks and compared the performance with a classical SCARA in different situational tasks. This control proposes to use nonlinear control in both cases.

In another work, Freire et al. [26] proposed an adaptive PID control to control a manipulator robot in trajectory tracking. To adjust the PID gains, an ANN to retropropagate the trajectory error is applied. Similarly, Sharma et al. [27] used a recurrent neural network to implement an adaptive PID controller.

Al-Khedher and Alshamasin [28] used artificial neural networks (ANN) for the control of a SCARA robot, and its performance is compared with a classical PD controller. An alternative type of solution has been proposed for robot manipulators control problems in Rossomando and Soria [29]; their work is based on the design of an SMC adaptive controller in discrete time on a discretized model.

Jha and Biswal [30] utilized an ANN model to solve the inverse kinematics problem of a SCARA manipulator; the ANN is based on a multilayer perceptron using a gradient descendent rule to find the weights tuning law.

Yi and Zhai [31] showed a novel control technique for chattering-free in trajectory tracking for the robot's manipulators when external disturbances and different uncertainties are acting on its dynamics. To reduce these undesirable perturbations, a second-order fast nonsingular terminal sliding mode (SOFNTSM) control is designed. This proposal guarantees robustness and ensures convergence. Also, using a simulation technique demonstrates the effectiveness of the control method. More recently, Rosas et al. [32] developed an active compensation of disturbances (ADRC) where the effectiveness of this proposal is applied on a SCARA robot arm.

This paper addresses the role of the disturbance observers in nonlinear electromechanical systems, in a particular case of a robot manipulator. Prior information in dynamic information is manifested in terms of nonlinear matrices (inertia, friction, and Coriolis), and this information is applied to design the control action. However, there are differences between the real and known dynamics that can affect the robot performance in trajectory tracking. To overcome this problem, a neural GRNN estimator is implemented. This observation technique can obtain good estimations of the dynamical differences to reduce the control error. Also, the main contribution of this proposal is the implementation of the $\sin(\mathrm{tg}^{-1}(.))$ function as saturation function in the control law, which is smoother than $\tanh(.)$ or $\mathrm{sat}(.)$ functions. Due to this characteristic, the transition between states can improve the effectiveness of the controller, and similar error levels can be obtained. Also, this bounded function can prevent neural parameters from drifting due to control errors from values outside the saturation's maximum value. These problems sometimes are occasioned by communication errors between the robot's control unit and the PC. Being this last consideration the main novelty of this work. To probe the theoretical results, some experiments on the SCARA Bosch SR-800 arm system were performed. The convergence of the presented control technique and the demonstration of the closed loop system stability is another contribution of this work.

This research work is organized as follows: the section "Model Description" presents a brief description of a robot manipulator model SCARA type. The section "Controller Design" shows the design of the controller based on inverse nonlinear sliding mode control. An overview of GRNN and a function approximation of the different dynamics is presented in the section "GRNN Function Approximation and Adaptive Tuning Laws." Also, in this section, the adaptive tuning laws for the GRNN are obtained. Finally, realistic simulation results and conclusions are given in the sections "Results and Discussion" and "Conclusions," respectively.

2. Model Description

Consider the robot manipulator SCARA type model (Figure 1) described in studies by Rossomando et al. [29] and Freire et al. [26],

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{f}(\dot{\mathbf{q}}) = \mathbf{u}, \tag{1}$$

where $\mathbf{q} = [q_1, q_2]^T$ are the generalized coordinates of position of the robot arm and $\mathbf{u}$ are the normalized command signals arranged as components of the control signal torque vector $\mathbf{u} = [u_1, u_2]^T$. Moreover, $\mathbf{M}(\mathbf{q})$ is the inertia matrix, $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})$ is the Coriolis matrix and $\mathbf{f}(\dot{\mathbf{q}})$ is defined as the friction term.

Recalling the terms $\mathbf{C}(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{f}(\dot{\mathbf{q}}) = \mathbf{N}(\mathbf{q},\dot{\mathbf{q}})$ and replacing in the following equation:

$$\mathbf{M}(\mathbf{q})\ddot{\mathbf{q}} + \mathbf{N}(\mathbf{q},\dot{\mathbf{q}}) = \mathbf{u}. \tag{2}$$

Different robotic arms own intrinsic dynamic properties; these properties will be taken into account for the implementation of the control law. The structural properties of the robot arm are as follows [33]:

(1) $\mathbf{M}(\mathbf{q})$ is a symmetric and positive definite matrix

(2) $\dot{\mathbf{M}}(\mathbf{q}) - \mathbf{C}(\mathbf{q},\dot{\mathbf{q}}) \in \mathbb{R}^{n\times n}$ is a skew-symmetric matrix

The known manipulator dynamics is represented by

$$\widehat{\mathbf{M}}(\mathbf{q})\ddot{\mathbf{q}} + \widehat{\mathbf{N}}(\mathbf{q},\dot{\mathbf{q}}) = \mathbf{u}, \tag{3}$$

where equation (3) will be used in the controller design, and the estimated matrices are defined as follows:

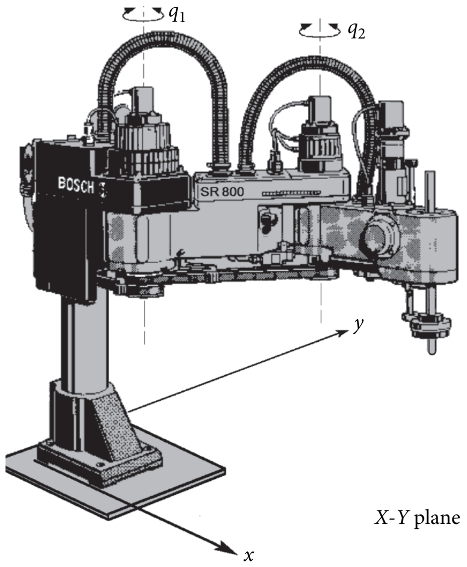


FIGURE 1: Representation of the 2-DOF robot arm.

$$\widehat{\mathbf{M}}(\mathbf{q}) = \begin{pmatrix} 1.7277 + 0.1908\cos(q_2) & 0.0918 + 0.0954\cos(q_2) \\ 0.0918 + 0.0954\cos(q_2) & 0.0918 \end{pmatrix},$$
$$\widehat{\mathbf{C}}(\mathbf{q}, \dot{\mathbf{q}}) = \begin{pmatrix} 31.8192 - 0.0954\sin(q_2)\dot{q}_2 & -0.0954\sin(q_2)(\dot{q}_2 + \dot{q}_1) \\ 0.3418\sin(q_2)\dot{q}_1 & 12.5783 \end{pmatrix}, \tag{4}$$
$$\widehat{\mathbf{f}}(\dot{\mathbf{q}}) = \begin{pmatrix} 1.0256\,\mathrm{sign}(\dot{q}_1) \\ 1.7842\,\mathrm{sign}(\dot{q}_2) \end{pmatrix}.$$

3. Controller Design

3.1. Neuro Adaptive SMC Controller. In this section, the trajectory tracking control scheme is described. The control scheme is based on the generalized coordinates. First, the equation of motion is represented in terms of joint coordinates. Then, the trajectory tracking control scheme with adaptive neural SMC control is proposed. To start this analysis, the control error signal is defined as

$$\mathbf{e} = \mathbf{q} - \mathbf{q}_{\mathrm{ref}}. \tag{5}$$

Now, in order to apply the control law, a sliding surface is used for the control error signal:

$$\mathbf{r} = \mathrm{diag}\left[\left(\frac{\mathrm{d}}{\mathrm{d}t} + \alpha_1\right), \left(\frac{\mathrm{d}}{\mathrm{d}t} + \alpha_2\right)\right]\mathbf{e} = \dot{\mathbf{e}} + \boldsymbol{\alpha}\mathbf{e}, \tag{6}$$

where $\boldsymbol{\alpha}$ is a diagonal matrix with positive constants $(\alpha_i > 0)$ and each α_i is defined as a design parameter. Deriving equation (6), it leads to

$$\dot{\mathbf{r}} = \ddot{\mathbf{e}} + \boldsymbol{\alpha}\dot{e}. \tag{7}$$

Using equation (7) and replacing equation (5),

$$\ddot{\mathbf{q}} - \ddot{\mathbf{q}}_{\mathrm{ref}} = \dot{\mathbf{r}} - \boldsymbol{\alpha}\dot{e}. \tag{8}$$

From equation (6) and rearranging equation (8),

$$\ddot{\mathbf{q}} = \dot{\mathbf{r}} - \boldsymbol{\alpha}(\mathbf{r} - \boldsymbol{\alpha}\mathbf{e}) + \ddot{\mathbf{q}}_{\mathrm{ref}} = \dot{\mathbf{r}} - \boldsymbol{\alpha}\mathbf{r} + \boldsymbol{\alpha}^2\mathbf{e} + \ddot{\mathbf{q}}_{\mathrm{ref}}. \tag{9}$$

Replacing equation (9) in equation (2),

$$\dot{\mathbf{r}} - \boldsymbol{\alpha}\mathbf{r} + \boldsymbol{\alpha}^2\mathbf{e} + \ddot{\mathbf{q}}_{\mathrm{ref}} = \mathbf{M}^{-1}\mathbf{u} - \mathbf{M}^{-1}\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}). \tag{10}$$

Obtaining $\dot{\mathbf{r}}$ from equation (10),

$$\dot{\mathbf{r}} = -\mathbf{M}^{-1}\mathbf{N}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{M}^{-1}\mathbf{u} + \boldsymbol{\alpha}\mathbf{r} - \boldsymbol{\alpha}^2\mathbf{e} - \ddot{\mathbf{q}}_{\mathrm{ref}}. \tag{11}$$

3.2. Saturation Definition. In this subsection, an approximation of $\sin(\mathrm{tg}^{-1}(.))$ function is proposed. The main objective of this work is to propose an alternative of the $\tanh(.)$ function as a smooth saturation function. Figure 2 shows $\sin(\mathrm{tg}^{-1}(.))$, $\mathrm{sat}(.)$, and $\tanh(.)$ functions, as the $\sin(\mathrm{tg}^{-1}(.))$ function reaches its max value more gradually than the hyperbolic tangent function $\tanh(.)$, and since it presents a smoother transition, this characteristic can be used to take advantage of control applications. In [24], a $\sin(\mathrm{tg}^{-1}(.))$ function was used in a kinematics controller for mobile robot applications, being the tracking accuracy of the robot improved. Now, expressing the $\sin(\mathrm{tg}^{-1}(.))$ function as

$$\sin\left(\mathrm{tg}^{-1}(b_2 r_i)\right) = \frac{b_2 r_i}{\sqrt{1 + (b_2 r_i)^2}}. \tag{12}$$

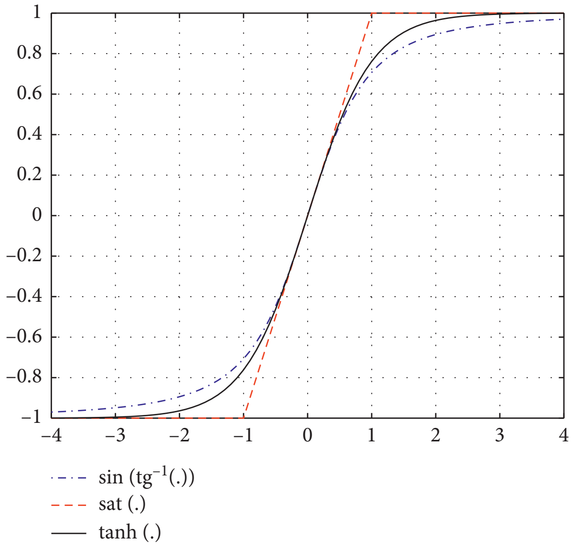


FIGURE 2: sat(.), tanh(.), and sin(tg^{-1}(.)) functions.

Equation (12) changes from trigonometric to an algebraic function. An algebraic function is simpler to implement in microcontrollers and industrial computers.

Assumption 1. It is assumed that the quantity $\sqrt{1+(b_2 r_i)^2}$ is nonnegative along the evolution of the system.

The control law is expressed as

$$\mathbf{u}=\widehat{\mathbf{M}}\left(\widehat{\mathbf{M}}^{-1}\widehat{\mathbf{N}}(\mathbf{q},\dot{\mathbf{q}})+\ddot{\mathbf{q}}_{\mathrm{ref}}-b_1\sin\left(\mathrm{tg}^{-1}(b_2\mathbf{r})\right)-\boldsymbol{\xi}_\varepsilon\operatorname{sign}(\mathbf{r})\right), \tag{13}$$

where $\xi_\varepsilon \in \mathfrak{R}^{2\times 2}$, and it is expressed as $\xi_\varepsilon=\operatorname{diag}(\xi_{\varepsilon 1},\xi_{\varepsilon 2})$. The proposed control structure is shown in Figure 3. Now, closing the control loop and replacing equation (13) in equation (11), we lead to

$$\dot{\mathbf{r}}=-b_1\sin\left(\mathrm{tg}^{-1}(b_2\mathbf{r})\right)+\boldsymbol{\alpha}\mathbf{r}-\boldsymbol{\alpha}^2\mathbf{e}-\boldsymbol{\xi}_\varepsilon\operatorname{sign}(\mathbf{r})+\boldsymbol{\delta}(t). \tag{14}$$

From equation (14), $\boldsymbol{\delta}(t)$ denotes the error due to the dynamic difference, and it is expressed as

$$\begin{aligned}\boldsymbol{\delta}(t)&=-\mathbf{M}^{-1}\mathbf{N}(\mathbf{q},\dot{\mathbf{q}})+\mathbf{M}^{-1}(\mathbf{q})\mathbf{u}-\left(-\widehat{\mathbf{M}}^{-1}\widehat{\mathbf{N}}(\mathbf{q},\dot{q})+\widehat{\mathbf{M}}^{-1}(\mathbf{q})\mathbf{u}\right)\\&=\left(-\mathbf{M}^{-1}\mathbf{N}(\mathbf{q},\dot{\mathbf{q}})+\widehat{\mathbf{M}}^{-1}\widehat{\mathbf{N}}(\mathbf{q},\dot{\mathbf{q}})\right)+\left(\mathbf{M}^{-1}(\mathbf{q})-\widehat{\mathbf{M}}^{-1}(\mathbf{q})\right)\mathbf{u},\end{aligned} \tag{15}$$

where $\boldsymbol{\delta}$ is considered bounded by

$$\boldsymbol{\delta}(t)\leq\boldsymbol{\delta}_0, \tag{16}$$

where $\boldsymbol{\delta}_0$ is a positive constant value.

3.3. GRNN Function Approximation and Adaptive Tuning Laws. In this subsection, a general regression neural network (GRNN) identification technique is presented, where the GRNN is a single-pass neural network which uses a Gaussian activation function in the hidden layer, and it can be used to approximate any continuous function [34].

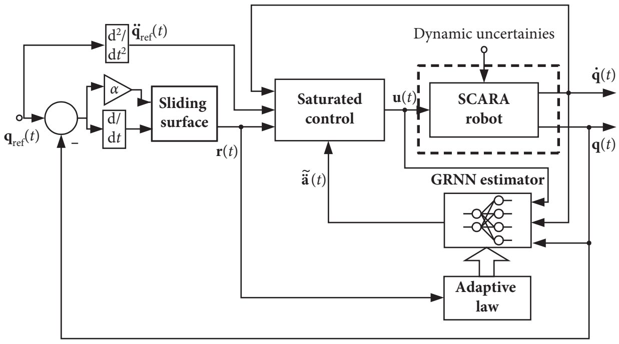


FIGURE 3: Control structure of the robot arm using a neural observer.

The variable $\boldsymbol{\delta}$(t) represents the unknown parameters due to mass variation of the payload and can be represented as

$$\boldsymbol{\delta}(\mathbf{t})-\left(\boldsymbol{\varphi}_{MN}^{*T}\boldsymbol{\Psi}(\overline{\mathbf{q}})+\boldsymbol{\varphi}_{M}^{*T}\boldsymbol{\chi}(\mathbf{q})\mathbf{u}\right)\leq\varepsilon<\infty, \tag{17}$$

or

$$\boldsymbol{\delta}(t)=\boldsymbol{\varphi}_{MN}^{*T}\boldsymbol{\Psi}(\overline{\mathbf{q}})+\boldsymbol{\varphi}_{M}^{*T}\boldsymbol{\chi}(\mathbf{q})\mathbf{u}+\boldsymbol{\varepsilon}, \tag{18}$$

where $\overline{\mathbf{q}}$ is the regressor vector and is defined as $\overline{\mathbf{q}}=[\mathbf{q},\dot{\mathbf{q}}]$. The neuron number is defined as m, and the vector $\Psi(.)\in\mathfrak{R}^{m\times 1}$ is a functions vector where each element is defined as $\Psi_i=\exp(\overline{\mathbf{q}}-\mathbf{z}/2\rho)/\sum_{j=1}^{m}\exp(\overline{\mathbf{q}}-\mathbf{z}_j/2\rho_j)$ and $\mathbf{z}\in\mathfrak{R}^{\dim(\overline{q})\times 1}$ and $\rho\in\mathfrak{R}^{m\times 1}$ define the centers and the spread of each exponential function, respectively. In the same way, the matrix $\boldsymbol{\chi}(.)\in\mathfrak{R}^{m\times n}$ is defined as $\chi_i=\exp(\mathbf{q}-\mathbf{z}/2\rho)/\sum_{j=1}^{m}\exp\ (\mathbf{q}-\mathbf{z}_j/2\rho_j)$ and $\mathbf{z}\in\mathfrak{R}^{\dim(\mathbf{q})\times 1}$ and $\rho\in\mathfrak{R}^{m\times 1}$ define the centers and the spread of each exponential function, respectively. Besides, $\boldsymbol{\varphi}_{MN}^{*}\in\mathbf{R}^{n\times m}$ and $\boldsymbol{\varphi}_{M}^{*}\in\mathfrak{R}^{n\times m}$I are the optimal parameter matrix; each element of $\boldsymbol{\varphi}_{MN}^{*}$ and $\boldsymbol{\varphi}_{M}^{*}$ is constant and unknown. The vector ε is the approximation error and $|\varepsilon_i|\leq\xi_{\varepsilon i}$ for $i=1,2$. Being $n=2$ the model output number, the estimation function is chosen as follows:

$$\widehat{\boldsymbol{\delta}}(t)=\widehat{\boldsymbol{\varphi}}_{MN}^{T}\boldsymbol{\Psi}(\overline{\mathbf{q}})+\widehat{\boldsymbol{\varphi}}_{M}^{T}\boldsymbol{\chi}(\mathbf{q})\mathbf{u}. \tag{19}$$

The GRNN structure has the capacity of approximating any real continuous vector function; in [34, 35], there are different references about this identification technique. Using the estimated output of GRNN equation (19) and taking the difference between equation (18) and equation (19), we obtain

$$\boldsymbol{\delta}(t)-\widehat{\boldsymbol{\delta}}(t)=\widetilde{\boldsymbol{\delta}}(t). \tag{20}$$

The approximation error $\widetilde{\boldsymbol{\delta}}(t)$ can be approximated by

$$\widetilde{\boldsymbol{\delta}}(t)=\left(\boldsymbol{\varphi}_{MN}^{*T}\boldsymbol{\Psi}(\overline{\mathbf{q}})+\boldsymbol{\varphi}_{M}^{*T}\boldsymbol{\chi}(\mathbf{q})\mathbf{u}+\boldsymbol{\varepsilon}\right)-\left(\widehat{\boldsymbol{\varphi}}_{MN}^{T}\boldsymbol{\Psi}(\overline{\mathbf{q}})+\widehat{\boldsymbol{\varphi}}_{M}^{T}\boldsymbol{\chi}(\mathbf{q})\mathbf{u}\right). \tag{21}$$

Taking into account that $\boldsymbol{\varphi}_{MN}^{*T}=\widetilde{\boldsymbol{\varphi}}_{MN}^{T}+\widehat{\boldsymbol{\varphi}}_{MN}^{T}$ and $\boldsymbol{\varphi}_{M}^{*T}=\widetilde{\boldsymbol{\varphi}}_{M}^{T}+\widehat{\boldsymbol{\varphi}}_{M}^{T}$ and replacing them in equation (21),

$$\widetilde{\boldsymbol{\delta}}(t)=\left(\widetilde{\boldsymbol{\varphi}}_{MN}^{T}\boldsymbol{\Psi}(\overline{\mathbf{q}})+\widetilde{\boldsymbol{\varphi}}_{M}^{T}\boldsymbol{\chi}(\mathbf{q})\mathbf{u}+\boldsymbol{\varepsilon}\right). \tag{22}$$

Replacing the proposed control law (13) in the robot dynamics (11) and also the neural approximation (22), it leads to

$$\begin{aligned}\dot{\mathbf{r}} &= -b_1 \sin\left(\mathrm{tg}^{-1}\left(b_2\mathbf{r}\right)\right) + \tilde{\boldsymbol{\delta}}(t) + \boldsymbol{\alpha}\mathbf{r} - \boldsymbol{\alpha}^2\mathbf{e} - \xi_{\varepsilon}\mathrm{sign}\left(\mathbf{r}\right) = \cdots \\ &= -b_1 \sin\left(tg^{-1}\left(b_2\mathbf{r}\right)\right) + \left(\tilde{\boldsymbol{\varphi}}_{MN}^T \boldsymbol{\Psi}\left(\overline{\mathbf{q}}\right) + \tilde{\boldsymbol{\varphi}}_{M}^T \boldsymbol{\chi}\left(\mathbf{q}\right)\mathbf{u} + \boldsymbol{\varepsilon}\right) + \boldsymbol{\alpha}\mathbf{r} \\ &\quad - \boldsymbol{\alpha}^2\mathbf{e} - \xi_{\boldsymbol{\varepsilon}}\mathbf{sign}\left(\mathbf{r}\right).\end{aligned} \tag{23}$$

Theorem: the closed-loop system (23) using the proposed control law (13) using the neural approx. (22) has asymptotic stability.

3.3.1. Demonstration. The first step is defining a definite positive Lyapunov's function candidate (LFC) to demonstrate the convergence of the proposed control method.

Defining the LFC as

$$\begin{aligned}\mathscr{L} = \sum_{i}^{2}\Bigg(&\frac{1}{b_2}\sqrt{1 + \left(b_2 r_i\right)^2} + \frac{1}{2}e_i^2 + \frac{1}{2b_3}\left(\tilde{\boldsymbol{\varphi}}_{MNi}^T \tilde{\boldsymbol{\varphi}}_{MNi}\right) \\ &+ \frac{1}{2b_4}\left(\tilde{\boldsymbol{\varphi}}_{Mi}^T \tilde{\boldsymbol{\varphi}}_{Mi}\right)\Bigg).\end{aligned} \tag{24}$$

Deriving equation (24),

$$\begin{aligned}\dot{\mathscr{L}} = \sum_{i}^{2}\Bigg(&\frac{b_2 r_i}{\sqrt{1 + \left(b_2 r_i\right)^2}}\dot{r}_i + e_i\dot{e}_i + \frac{1}{b_3}\left(\tilde{\boldsymbol{\varphi}}_{MNi}^T \dot{\tilde{\boldsymbol{\varphi}}}_{MNi}\right) \\ &+ \frac{1}{b_3}\left(\tilde{\boldsymbol{\varphi}}_{Mi}^T \dot{\tilde{\boldsymbol{\varphi}}}_{Mi}\right)\Bigg).\end{aligned} \tag{25}$$

Replacing equation (30) in equation (32),

$$\begin{aligned}\dot{\mathscr{L}} = \sum_{i}^{2} -\Bigg(&\mathrm{b}_1\left(\frac{b_2 r_i}{\sqrt{1 + \left(b_2 r_i\right)^2}}\right)^2 + \frac{b_2 r_i}{\sqrt{1 + \left(b_2 r_i\right)^2}}\left(\tilde{\boldsymbol{\varphi}}_{MNi}^T \boldsymbol{\Psi}\left(\overline{\mathbf{q}}\right) + \tilde{\boldsymbol{\varphi}}_{Mi}^T \boldsymbol{\chi}\left(\mathbf{q}\right)\mathbf{u} + \varepsilon_i\right) \\ &+ \cdots + \frac{b_2 r_i}{\sqrt{1 + \left(b_2 r_i\right)^2}}\left(\alpha_i r_i - \alpha_i^2 e_i\right) - \left(\frac{b_2 r_i}{\sqrt{1 + \left(b_2 r_i\right)^2}}\right)\xi_{\boldsymbol{\varepsilon}}\mathrm{sign}\left(r_i\right) + e_i\left(r_i - \alpha_i e_i\right) + \cdots + \frac{1}{b_3}\left(\tilde{\boldsymbol{\varphi}}_{MNi}^T \dot{\tilde{\boldsymbol{\varphi}}}_{MNi}\right) + \frac{1}{b_3}\left(\tilde{\boldsymbol{\varphi}}_{Mi}^T \dot{\tilde{\boldsymbol{\varphi}}}_{Mi}\right)\Bigg).\end{aligned} \tag{26}$$

Rearranging equation (33),

$$\begin{aligned}\dot{\mathscr{L}} \leq \sum_{i}^{2}\Bigg(&-\mathrm{b}_1\left(\frac{b_2 r_i}{\sqrt{1 + \left(b_2 r_i\right)^2}}\right)^2 + \frac{b_2 r_i}{\sqrt{1 + \left(b_2 r_i\right)^2}}\left(\tilde{\boldsymbol{\varphi}}_{MNi}^T \boldsymbol{\Psi}\left(\overline{q}\right)\right) + \cdots + \frac{b_2 r_i}{\sqrt{1 + \left(b_2 r_i\right)^2}}\left(\tilde{\boldsymbol{\varphi}}_{Mi}^T \boldsymbol{\chi}\left(\mathbf{q}\right)\mathbf{u}\right) \\ &+ \frac{b_2 r_i}{\sqrt{1 + \left(b_2 r_i\right)^2}}\left(\alpha_i r_i - \alpha_i^2 e_i\right) - \cdots - \left(\frac{b_2 r_i}{\sqrt{1 + \left(b_2 r_i\right)^2}}\right)\xi_{\varepsilon i}\mathrm{sign}\left(r_i\right) + \frac{b_2 r_i}{\sqrt{1 + \left(b_2 r_i\right)^2}}\left(\varepsilon_i\right) + \cdots + e_i\left(r_i - \alpha_i e_i\right) \\ &+ \frac{1}{b_3}\left(\tilde{\boldsymbol{\varphi}}_{MNi}^T \dot{\tilde{\boldsymbol{\varphi}}}_{MNi}\right) + \frac{1}{b_3}\left(\tilde{\boldsymbol{\varphi}}_{Mi}^T \dot{\tilde{\boldsymbol{\varphi}}}_{Mi}\right)\Bigg).\end{aligned} \tag{27}$$

Rearranging equation (34),

$$\dot{\mathscr{L}} \leq \sum_{i}^{2} -\left(\mathrm{b}_1 \left(\frac{b_2 r_i}{\sqrt{1+(b_2 r_i)^2}}\right)^2 + \frac{b_2 r_i}{\sqrt{1+(b_2 r_i)^2}}(\varepsilon_i) + \cdots + \frac{b_2 r_i}{\sqrt{1+(b_2 r_i)^2}}\left(\alpha_i r_i - \alpha_i^2 e_i\right) - \xi_{\varepsilon i}\left(\frac{b_2|r_i|}{\sqrt{1+(b_2 r_i)^2}}\right) + e_i\left(r_i - \alpha_i e_i\right)\right)$$
$$+ \sum_{i}^{2}\left(\tilde{\boldsymbol{\varphi}}_{MNi}^{T}(t)\left[\frac{b_2 r_i}{\sqrt{1+(b_2 r_i)^2}}\boldsymbol{\Psi}(\overline{q}) + \frac{1}{b_3}\dot{\tilde{\boldsymbol{\varphi}}}_{MNi}\right]\right) + \cdots + \sum_{i}^{2}\left(\tilde{\boldsymbol{\varphi}}_{Mi}^{T}(t)\left[\frac{b_2 r_i}{\sqrt{1+(b_2 r_i)^2}}\boldsymbol{\chi}(\mathbf{q})\mathbf{u} + \frac{1}{b_4}\dot{\tilde{\boldsymbol{\varphi}}}_{Mi}\right]\right). \tag{28}$$

Taking into account the last terms between brackets in equation (28) and defining $\dot{\tilde{\boldsymbol{\varphi}}}_{MNi}$ as

$$\dot{\tilde{\boldsymbol{\varphi}}}_{MNi} = -\frac{b_3 b_2 r_i}{\sqrt{1+(b_2 r_i)^2}}\boldsymbol{\Psi}(\overline{q}). \tag{29}$$

In the same way, for $\dot{\tilde{\boldsymbol{\varphi}}}_{Mi}$ is obtained in the next equation:

$$\dot{\tilde{\boldsymbol{\varphi}}}_{Mi} = -b_4\frac{b_2 r_i}{\sqrt{1+(b_2 r_i)^2}}\boldsymbol{\chi}(\mathbf{q})\mathbf{u}. \tag{30}$$

The subindex i in equation (29) and equation (30) denotes the vector weights adjusted by $\mathbf{r}_i$; replacing the same equation in (28) leads to

$$\dot{\mathscr{L}}_i = -b_1\left(\frac{b_2 r_i}{\sqrt{1+(b_2 r_i)^2}}\right)^2 + \frac{b_2 r_i}{\sqrt{1+(b_2 r_i)^2}}\left(\alpha_i r_i - \alpha_i^2 e_i\right)$$
$$+ e_i\left(r_i - \alpha_i e_i\right) + \Delta_i, \tag{31}$$

where $\Delta_i = (\varepsilon_i b_2 r_i - \xi_{\varepsilon i}|b_2 r_i|)(1/(\sqrt{1+(b_2 r_i)^2})) \leq 0$.

Now, it is necessary to analyze two cases, when $r_i = 0$ and $r_i \neq 0$.

Analyzing the first case $r_i = 0$, all the terms with $b_2 r_i/(\sqrt{1+(b_2 r_i)^2})$ are equal to zero. Thus, $\dot{\mathscr{L}}_i$ can be reduced to

$$\dot{\mathscr{L}}_i = -\alpha_i e_i^2 \leq 0. \tag{32}$$

This expression denotes that the control error is bounded when $r_i = 0$. Analyzing the second case $(r_i \neq 0)$, in this instance, $b_2 r_i/(\sqrt{1+(b_2 r_i)^2}) \neq 0$; rewriting equation (31),

$$\dot{\mathscr{L}}_i = -\left(b_1 - \frac{\alpha_i\sqrt{1+(b_2 r_i)^2}}{b_2}\right)\left(\frac{b_2 r_i}{\sqrt{1+(b_2 r_i)^2}}\right)^2$$
$$+\left(-\alpha_i^2 e_i\frac{b_2 r_i}{\sqrt{1+(b_2 r_i)^2}} + e_i r_i\right) - \alpha_i e_i^2 + \Delta_i. \tag{33}$$

Doing the same operation for the second term,

$$\dot{\mathscr{L}}_i = -\left(b_1 - \frac{\alpha_i\sqrt{1+(b_2 r_i)^2}}{b_2}\right)\left(\frac{b_2 r_i}{\sqrt{1+(b_2 r_i)^2}}\right)^2$$
$$- \cdots - \left(\alpha_i^2 - \frac{\sqrt{1+(b_2 r_i)^2}}{b_2}\right)e_i\frac{b_2 r_i}{\sqrt{1+(b_2 r_i)^2}} - \alpha_i e_i^2 + \Delta_i. \tag{34}$$

Defining the auxiliary vector $E_i = \left(e_i \;\; b_2 r_i/(\sqrt{1+(b_2 r_i)^2})\right)^T$ and rewriting (34) as

$$\dot{\mathscr{L}}_i = -\mathbf{E}_i^T\boldsymbol{\Theta}_i\mathbf{E}_i - \alpha_i e_i^2 + \Delta_i, \tag{35}$$

where $\boldsymbol{\Theta}_i$ is expressed as

$$\boldsymbol{\Theta}_i = \begin{pmatrix} \dfrac{\alpha_i}{2} & \dfrac{\alpha_i^2}{2} - \dfrac{\sqrt{1+(b_2 r_i)^2}}{2b_2} \\ \dfrac{\alpha_i^2}{2} - \dfrac{\sqrt{1+(b_2 r_i)^2}}{2b_2} & b_1 - \dfrac{\alpha_i\sqrt{1+(b_2 r_i)^2}}{b_2} \end{pmatrix}. \tag{36}$$

It is necessary to demonstrate that the matrix $\boldsymbol{\Theta}_i$ is a positive definite to demonstrate the convergence of the presented control technique. Taking into account the Sylvester's criterion, which is a simple way to determine if $\boldsymbol{\Theta}_i$ is a definite positive matrix.

Now computing if the upper left 1-by-1 corner of $\boldsymbol{\Theta}_i$ has a positive determinant. In this case,

$$\frac{\alpha_i}{2} > 0. \tag{37}$$

Consequently, if the upper left 2-by-2 corner of $\boldsymbol{\Theta}_i$ has a positive determinant, where

$$\frac{\alpha_i}{2}\left(\mathrm{b}_1 - \frac{\alpha_i\sqrt{1+(b_2 r_i)^2}}{b_2}\right) - \left(\frac{\alpha_i^2}{2} - \frac{\sqrt{1+(b_2 r_i)^2}}{2b_2}\right)^2 > 0. \tag{38}$$

Expanding the abovementioned equation (38) and rearranging,

$$\left(\frac{\alpha_i}{2}b_1 - \frac{\alpha_i^2\sqrt{1+(b_2r_i)^2}}{2b_2}\right) - \left(\frac{\alpha_i^4}{4} - \frac{\alpha_i^2}{2}\,\frac{\sqrt{1+(b_2r_i)^2}}{b_2} + \frac{1+(b_2r_i)^2}{4(b_2)^2}\right)$$
$$= \cdots = \frac{\alpha_i}{2}b_1 - \frac{\alpha_i^4}{4} - \frac{1}{4(b_2)^2} - \frac{(r_i)^2}{4} > 0. \tag{39}$$

Then, equation (39) can be expressed as

$$\alpha_i\left(2b_1 - \alpha_i^3\right) > \left(\frac{1}{(b_2)^2} + (r_i)^2\right). \tag{40}$$

From definition of $\mathscr{L}_i$ (equation (24)),

$$\mathscr{L}_i > \frac{1}{b_2}\sqrt{1+(b_2r_i)^2}. \tag{41}$$

Obtaining r_i from the abovementioned equation (41),

$$\frac{\sqrt{(b_2\mathscr{L}_i)^2 - 1}}{b_2} > r_i. \tag{42}$$

Replacing in equation (40),

$$\alpha_i\left(2b_1 - \alpha_i^3\right) > \frac{(b_2\mathscr{L}_i)^2 - 1}{(b_2)^2} + \frac{1}{(b_2)^2}, \tag{43}$$

with condition (i) and condition (ii) and the matrix $\mathbf{\Theta}_i$ which is a definite positive; thus, $\dot{\mathscr{L}}_i$ can be the upper bound by the following expression:

$$\dot{\mathscr{L}}_i \leq -\alpha_i e_i^2, \quad \text{for } \alpha_i\left(2b_1 - \alpha_i^3\right) > (\mathscr{L}_i)^2. \tag{44}$$

From equation (44) $\dot{\mathscr{L}}_i \leq 0$, for all $t \in [0, \infty]$, the sufficient condition for equation (45) can be obtained:

$$\dot{\mathscr{L}}_i \leq -\alpha_i e_i^2, \quad \text{for } \alpha_i\left(2b_1 - \alpha_i^3\right) > (\mathscr{L}_i(0))^2. \tag{45}$$

From the definition of LFC, equation (24), $V_i(0)$ is expressed as

$$\mathscr{L}_i(0) = \frac{1}{b_2}\sqrt{1+(b_2r_i(0))^2} + \frac{1}{2}e_i^2(0) \leq \left(\frac{1}{b_2} + \left|r_i(0)\right|\right) + \frac{1}{2}e_i^2(0). \tag{46}$$

Using the abovementioned expression, equation (46) can be rewritten as

$$\dot{\mathscr{L}}_i \leq -\alpha_i e_i^2,$$
$$\text{for } \alpha_i\left(2b_1 - \alpha_i^3\right) > \left(\left(\frac{1}{b_2} + \left|\dot{e}_i(0)\right| + \alpha_i\left|e_i(0)\right|\right) + \frac{1}{2}e_i^2(0)\right)^2. \tag{47}$$

Based on previous results and considering equations (24), (37), (39), and (47) and Barbalat's lemma [36], these results guarantee the boundedness of the error control signal e_i during the closed loop operation.

4. Results and Discussion

4.1. Experiments. The experiments were performed on the SCARA Bosch SR-800 (Figure 1) arm system. A CPU Intel Dual Core, with 2.6 GHz and 4 GB RAM memory to control the robot arm, is used. Debian Linux has been installed in the CPU and it uses RTAI (Real Time Application Interface). The approach proposed in this paper is applied to the robotic manipulator Bosch SR-800 by controlling the power unit that applies the calculated torques (control actions) to the joints of the robot arm. The first experiment shows different experiments with different neural networks. For performance comparison purposes, in addition to the use of the different saturated controllers to control the robotic manipulator, a nonadaptive SMC controller was also applied to the same robot for reference. In the second set of experiments, different saturated functions were used.

4.2. Adaptive Neural Control Comparison. The first set of experiments was made with different neural networks to compensate for the dynamic variations.

The objective of this approach is to contribute to the use of a new saturation function and prove its efficiency in an adaptive control technique. In the first set of experiments, the basic form of Lissajous "8" is considered as a reference trajectory. The experiment was carried out using three different adaptive techniques using the single saturation function $\sin(\mathrm{tg}^{-1}(.))$, where these controllers were applied on the robot arm to demonstrate their effectiveness. The controllers are based on saturated SMC, the first one uses a nonadaptive SMC, which does not use any on-line adjustment, and the second one, an MLP adaptive controller and the last one, a GRNN adaptive controller based on SMC.

Figure 4 shows the trajectories followed by the robot arm using each one of the control strategies. Figure 5 shows the reference trajectory for each joint (q_1 and q_2) and the followed trajectory. Figure 6 shows the square norm of the control errors generated by each control strategy (the error norm is defined by $\mathbf{e} = \mathrm{sqrt}(e_{q1}^2 + e_{q2}^2)$). The experimental results prove that the highest error was obtained by a nonadaptive SMC controller, which has no on-line calibration. In this case, the effects of the model uncertainties on the error can be observed clearly. The methods that compensate the model uncertainties have a lower error than the previous case. Finally, the lowest error was obtained by the GRNN adaptive SMC strategy, which reduces the trajectory error caused by the model uncertainties. In Figure 7, the control actions of the GRNN adaptive SMC are shown, and the chattering phenomenon can be seen. This phenomenon can be reduced using a low pass filter previous to the electromechanical actuator.

Figure 8 presents the distribution of the errors obtained along with the experiment. In the 1st row, it shows the error distribution using nonadaptive SMC control. The 2nd row presents the error distribution using adaptive GRNN-SMC control. Finally, in the 3rd row, the error distribution using adaptive MLP-SMC control is shown. In Table 1, the performance indicators IAE and $ITAE$ are presented

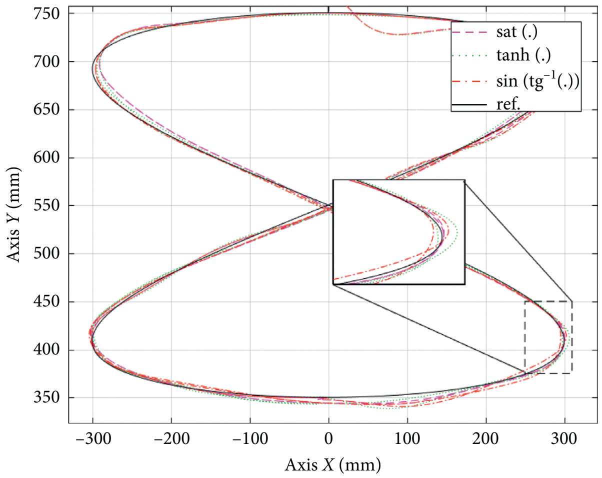
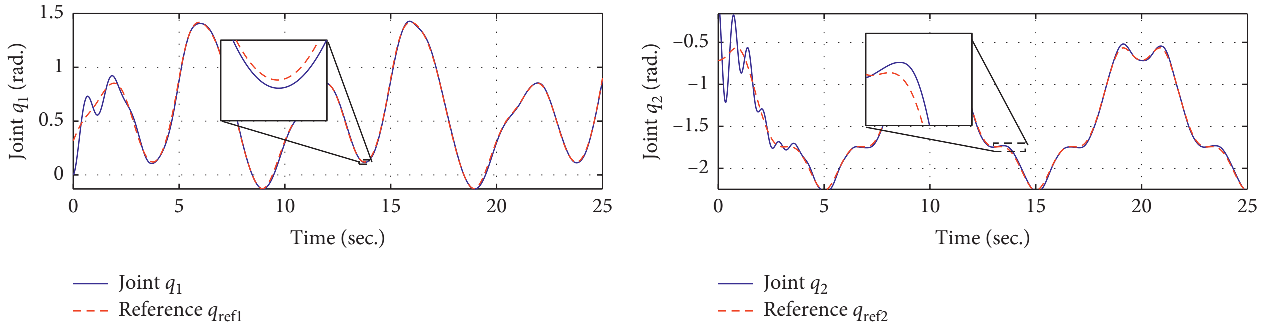


Figure 4: Trajectory followed for the different strategies.



(a)

(b)

Figure 5: Reference (dashed line) and trajectory followed (solid line).

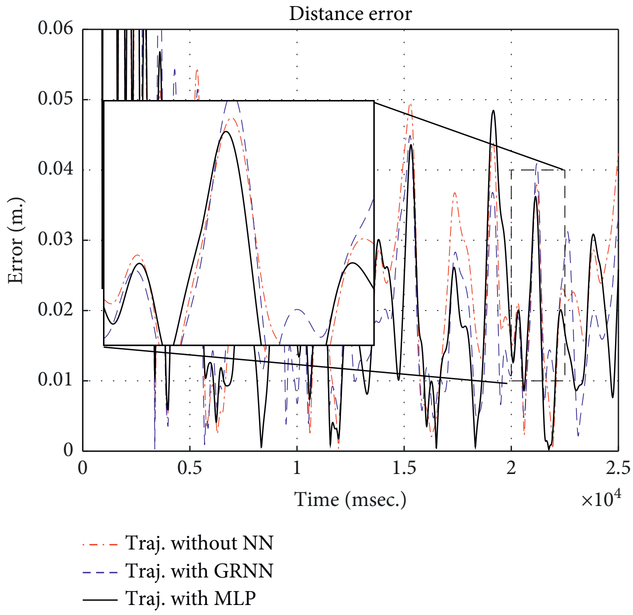


Figure 6: Square norm of the trajectory errors generated by each control strategy.

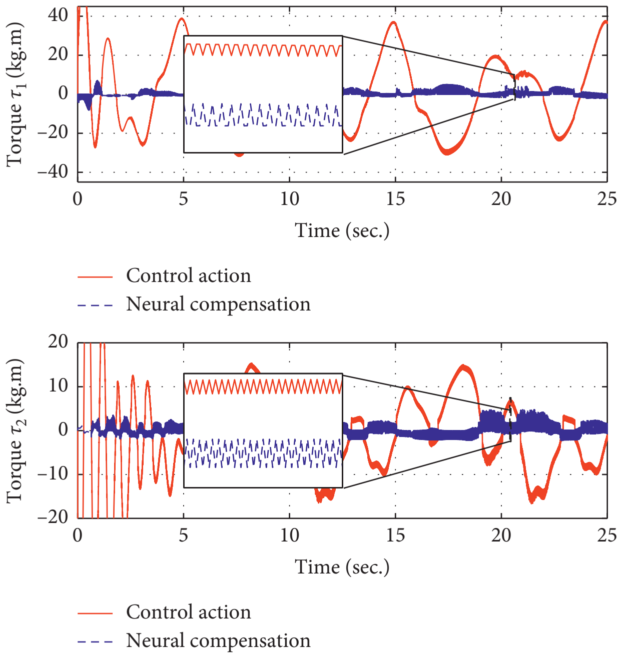
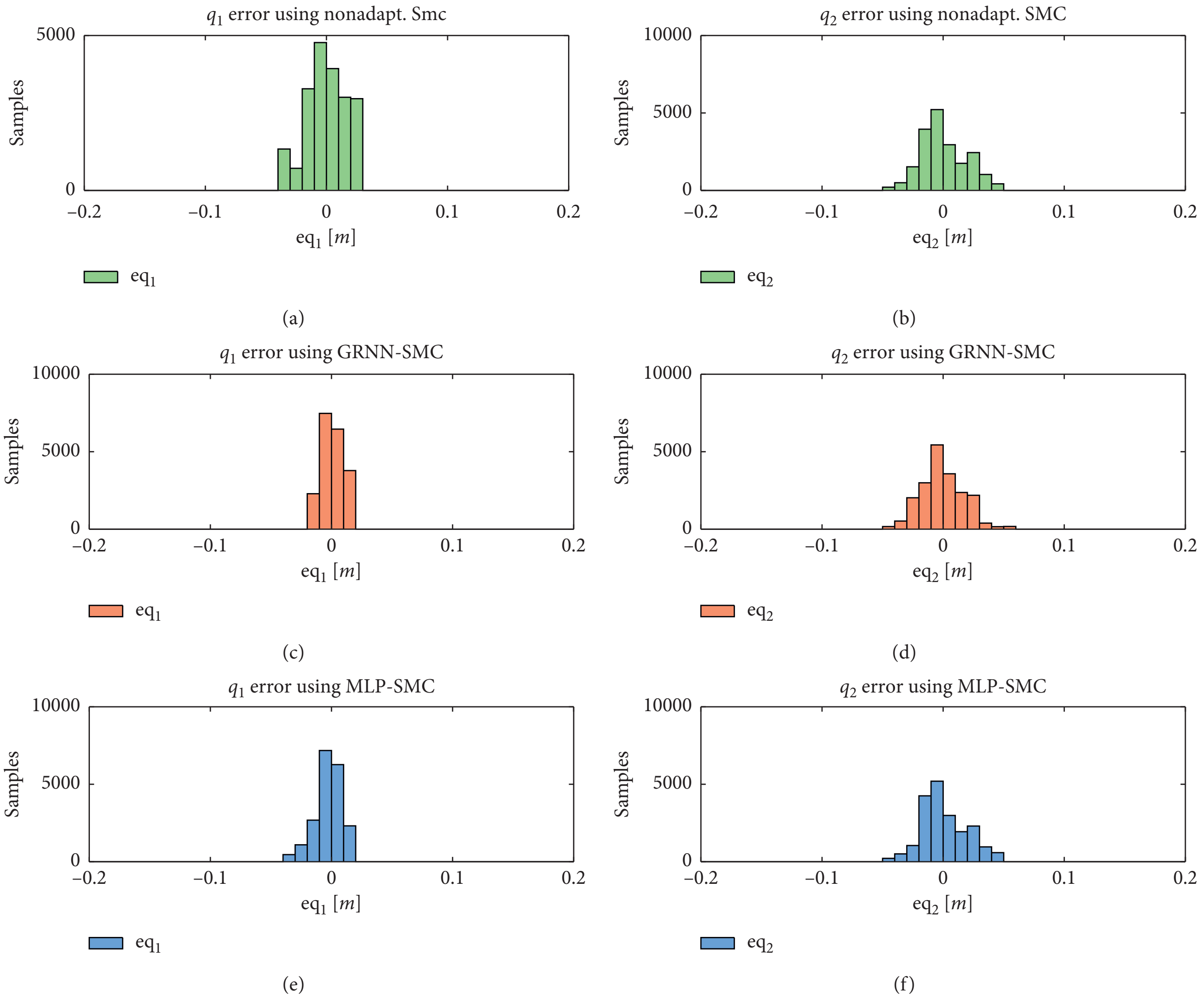


FIGURE 7: Control actions (adaptive + neural) and neural compensation using adaptive GRNN.



(a)

(b)

(c)

(d)

(e)

(f)

FIGURE 8: Distribution of the errors obtained along the experiment. (a, b) The error distribution using a nonadaptive SMC controller. (c, d) The error distribution using an adaptive GRNN-SMC controller. (e, f) The error distribution using an adaptive MLP-SMC controller.

Table 1: The performance indexes IAE and ITAE of different controllers using $\sin(\mathrm{tg}^{-1}(.))$.

Controller	IAE	ITAE
Nonadaptive SMC	569.3324	$1.4233e+04$
Adapt. GRNN-SMC	424.7610	$1.0619e+04$
Adapt. MLP-SMC	473.8136	$1.1845e+04$

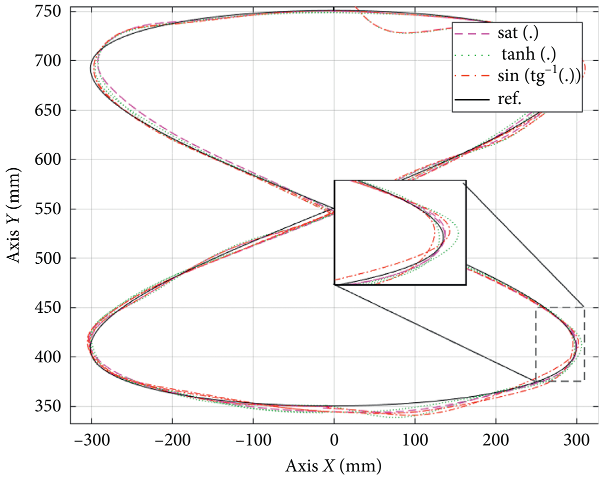
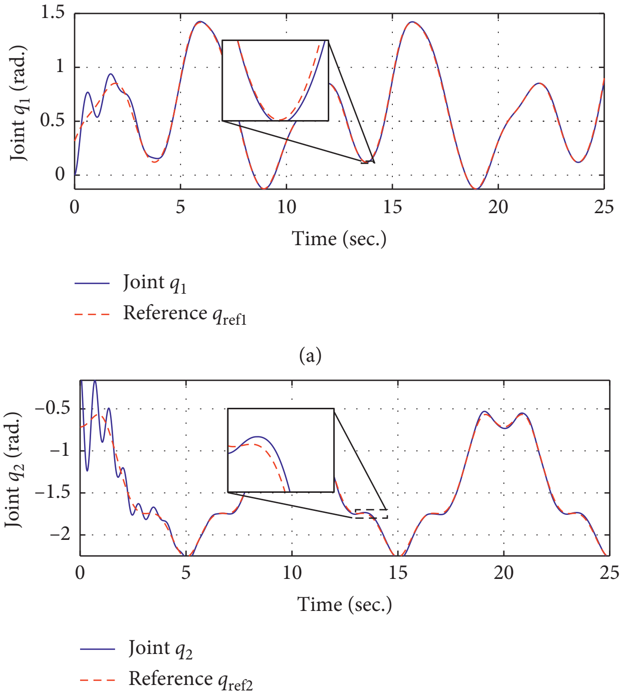


Figure 9: Trajectory followed for the different strategies.



(a)

(b)

Figure 10: (a) Joint q1. (b) Joint q2. Reference (dashed line) and trajectory followed (solid line).

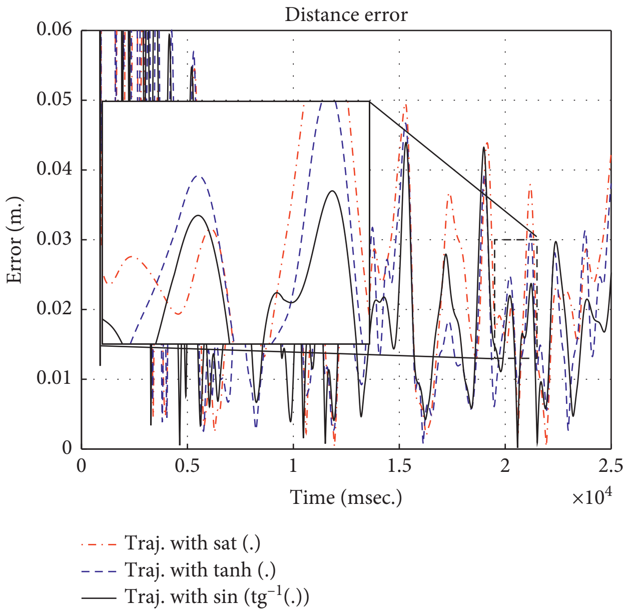
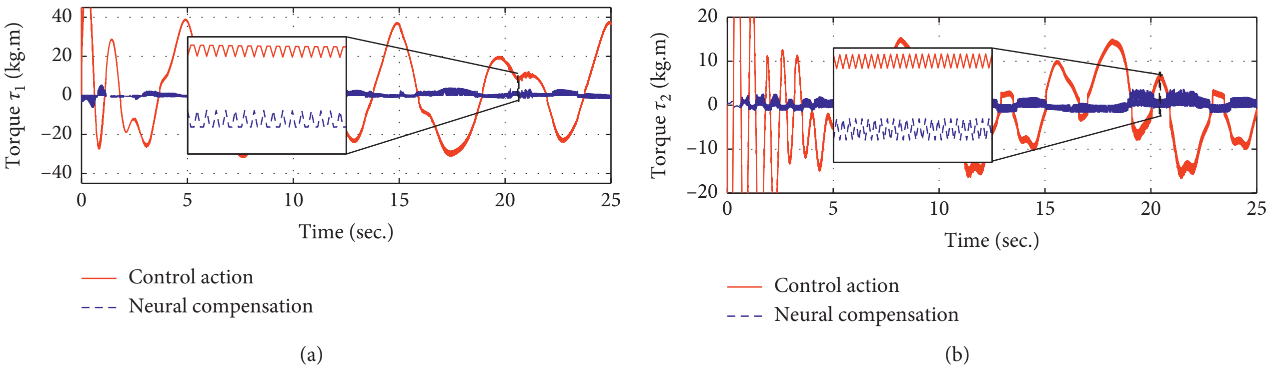


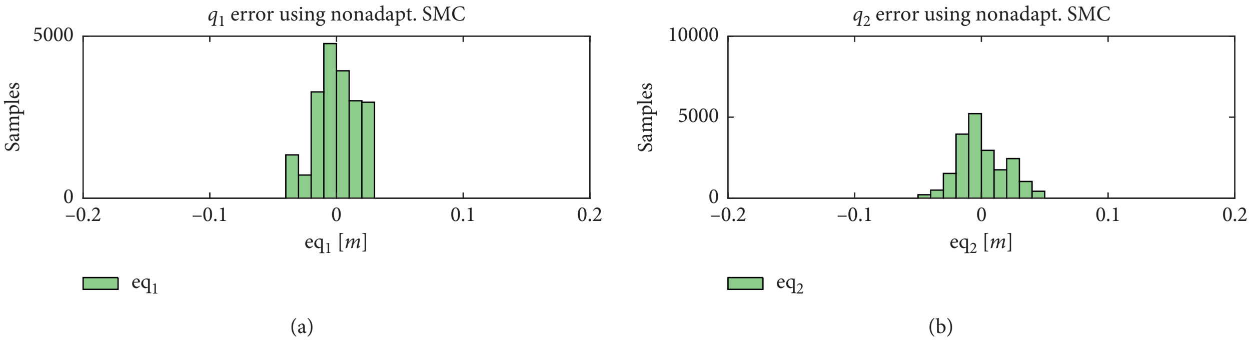
FIGURE 11: Square norm of the trajectory errors generated by each control strategy.



(a)

(b)

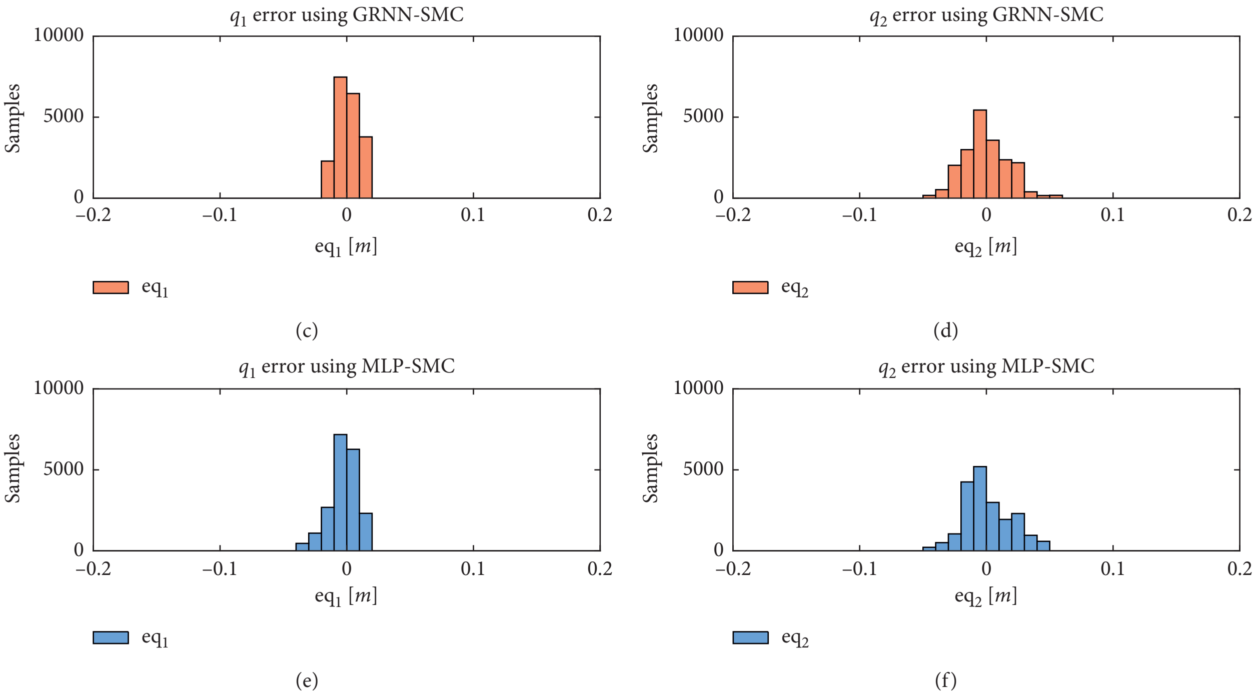
FIGURE 12: (a) Torque 1. (b) Torque 2. Control actions (adaptive + neural) and neural compensation using adaptive GRNN.



(a)

(b)

FIGURE 13: Continued.



(c)

(d)

(e)

(f)

Figure 13: Distribution of the errors obtained along the experiment. (a-b) The error distribution using the sat(.) function. (c-d) The error distribution using the tanh(.) function. (e-f) The error distribution using the $\sin(\text{tg}^{-1}(.))$ function.

Table 2: Performance indexes IAE and the ITAE of different controllers based on adaptive GRNN-SMC using sat(.), tanh(.), and $\sin(\text{tg}^{-1}(.))$.

Controller	IAE	ITAE
sat(.)	482.6992	$1.2067e+04$
tanh(.)	424.7610	$1.0619e+04$
$\sin(\text{tg}^{-1}(.))$	442.6008	$1.1065e+04$

(see equation (48) where T is the total time of the experiments). These indicators are applied on the proposed controllers based on using only $\sin(\text{tg}^{-1}(.))$ function, where the indicators show that the GRNN-SMC controller gets the smallest error as a result:

$$\begin{aligned}\text{IAE} &= \int_0^T \left(\left|e_{q1}\right| + \left|e_{q2}\right|\right) dt,\\ \text{ITAE} &= \int_0^T t\left(\left|e_{q1}\right| + \left|e_{q2}\right|\right) dt.\end{aligned} \tag{48}$$

4.3. Dynamical Comparison between Functions. In the second set of the experiment, a dynamical comparison between functions sat(.), tanh(.), and $\sin(\text{tg}^{-1}(.))$ was made. Figure 9 shows the trajectories performed by the robot arm using each one of the control strategies. Figure 10 shows the reference (dashed line) and trajectory followed (solid line), being the maximum error during the initial time. Figure 11 shows the square norm of the control errors generated by each saturation type applied in the control law. In this figure, it is appreciated that two last strategies have got similar errors. In Figure 12, the control actions based on the adaptive GRNN-SMC controller are shown. Both signals are polluted by the chattering effect, but in a percentage of about 10 with respect to the maximum value of the control signal, higher values are dangerous for the electromechanical systems. Figure 13 presents the distribution of the errors obtained along with the experiment. Figures 13(a) and 13(b) show the error distribution using the sat(.) function. Figures 13(c) and 13(d) present the error distribution using the tanh(.) function. Figures 13(e) and 13(f) show the error distribution using the $\sin(\text{tg}^{-1}(.))$ function. Finally, in Table 2, the performance indicators IAE and $ITAE$ are presented. These indicators are applied on alternative controllers based on adaptive GRNN-SMC using different saturation types sat(.), tanh(.), and $\sin(\text{tg}^{-1}(.))$. The denoted indicators show similar results as saturation functions in control. Thus, such functions could be used in different applications when a variable must not exceed a certain maximum value.

Based on these results, the viability of the proposed controller is demonstrated. The good results obtained, despite the nonlinearities and uncertainties of the model, together with the variations in the dynamics of the robot, show the level of robustness of the proposed GRNN-SMC controller. In addition, the stability of the closed loop system was demonstrated analytically and the adjustment laws were obtained using Lyapunov's theory. Another interesting observation is that the selection of the neural net structure of the estimator can improve the accuracy of the proposed control.

In this proposal for intelligent control, it can be considered as a general solution for the control of nonlinear systems. Its uses can be highlighting in robotic systems, in situations where the dynamics are variable or when there are uncertainties associated with the dynamic model.

5. Conclusions

This work presents a saturated control of two techniques using neural networks based on the classical $\text{sat}(.)$,$\tanh(.)$, and $\sin(\text{tg}^{-1}(.))$ functions and also provides the stability proof for a saturated control applied to a robot manipulator.

Comparative experiments of all these control variants have been performed on a standard industrial SCARA manipulator; in all cases, the proposed control gets satisfactory results; in the third case, it has an average error and can be used as an alternative of $\tanh(.)$ or $\text{sat}(.)$ function as saturated control.

The feasibility of applying an uncertainty estimator with a control based on sliding mode control, including its asymptotic stability, was also demonstrated.

The experiments showed that the trajectory followed by the manipulator can be controlled precisely by the use of the proposed method and, at the same time, guarantee the asymptotical convergence.

Data Availability

The MATLAB code and sampled data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was partially funded by the following Argentine institutions: the National Council of Scientific and Technological Research (CONICET) and Universidad Nacional de San Juan (UNSJ).

References

[1] E. Özgür and Y. Mezouar, "Kinematic modeling and control of a robot arm using unit dual quaternions," *Robotics and Autonomous Systems*, vol. 77, pp. 66–73, 2016.

[2] Y. Pititeeraphab, P. Choitkunnan, N. Thongpance, K. Kullathum, and C. Pintavirooj, "Robot-arm control system using leap motion controller," in *Proceedings of the 2016 International Conference on Biomedical Engineering (BME-HUST) IEEE*, Hanoi, Vietnam, 2016.

[3] Y. Pan, H. Wang, X. Li, and H. Yu, "Adaptive command-filtered backstepping control of robot arms with compliant actuators," *IEEE Transactions on Control Systems Technology*, vol. 26, no. 3, pp. 1149–1156, 2017.

[4] F. Capraro, F. G. Rossomando, C. Soria, and G. Scaglia, "Cascade sliding control for trajectory tracking of a nonholonomic mobile robot with adaptive neural compensator," *Mathematical Problems in Engineering*, vol. 2017, Article ID 8501098, 13 pages, 2017.

[5] A. Rosales, G. Scaglia, V. Mut, and F. D. Sciascio, "Controller designed by means of numeric methods for a benchmark problem: RTAC (rotational translational actuator)," in *Proceedings of the Electronics, Robotics and Automotive Mechanics Conference (CERMA'06)*, pp. 97–104, Cuernavaca, Mexico, 2006.

[6] G. Scaglia, V. Mut, A. Rosales, and O. Quintero, "Tracking control of a mobile robot using linear interpolation,"in *Proceedings of the IMAACA07*, Buenos Aires, Argentina, vol. 1, pp. 11–15, 2007.

[7] G. Scaglia, "Tracking control design in nonlinear multivariable systems: robotic applications," *Mathematical Problems in Engineering*, vol. 2019, Article ID 8643515, 15 pages, 2019.

[8] Q. Han, H. Zhang, and J. Liu, "Nonlinear dynamics of controlled synchronizations of manipulator system," *Mathematical Problems in Engineering*, vol. 2014, Article ID 691464, 9 pages, 2014.

[9] C. W. De Silva, *Intelligent Control: Fuzzy Logic Applications*, CRC Press, Boca Raton, FL, USA, 2018.

[10] R. Guo, "Projective synchronization of a class of chaotic systems by dynamic feedback control method," *Nonlinear Dynamics*, vol. 90, no. 1, pp. 53–64, 2017.

[11] Z. Yan, Y. Song, and J. H. Park, "Quantitative mean square exponential stability and stabilization of stochastic systems with Markovian switching," *Journal of the Franklin Institute*, vol. 355, no. 8, pp. 3438–3454, 2018.

[12] Z. Yan, J. H. Park, and W. Zhang, "A unified framework for asymptotic and transient behavior of linear stochastic systems," *Applied Mathematics and Computation*, vol. 325, pp. 31–40, 2018.

[13] Z. Yan, W. Zhang, and G. Zhang, "Finite-time stability and stabilization of Itô stochastic systems with Markovian switching: mode-dependent parameter approach," *IEEE Transactions on Automatic Control*, vol. 60, no. 9, pp. 2428–2433, 2014.

[14] A. Coronel-Escamilla, F. Torres, J. F. Gómez-Aguilar, R. F. Escobar-Jiménez, and G. V. Guerrero-Ramírez, "On the trajectory tracking control for an SCARA robot manipulator in a fractional model driven by induction motors with PSO tuning," *Multibody System Dynamics*, vol. 43, no. 3, pp. 257–277, 2018.

[15] J. Wu, D. Wang, and L. Wang, "A control strategy of a two degrees-of-freedom heavy duty parallel manipulator," *Journal of Dynamic Systems, Measurement, and Control*, vol. 137, no. 6, Article ID 061007, 2015.

[16] A. Remazeilles, A. Fernandez Iribar, and A. Dominguez Garcia, "Inverse kinematics methods for flexible arm control,"in *Proceedings of the Soft and Stiffness-controllable Robotics Solutions for Minimally Invasive Surgery: The STIFF-LOP Approach*, Pisa, Italy, p. 151, 2018..

[17] H. Wang, "Adaptive control of robot manipulators with uncertain kinematics and dynamics," *IEEE Transactions on Automatic Control*, vol. 62, no. 2, pp. 948–954, 2017.

[18] J. Baek, M. Jin, and S. Han, "A new adaptive sliding-mode control scheme for application to robot manipulators," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 6, pp. 3628–3637, 2016.

[19] S. Fateh and M. M. Fateh, "Adaptive fuzzy control of robot manipulators with asymptotic tracking performance," *Journal of Control, Automation and Electrical Systems*, vol. 31, no. 1, pp. 52–61, 2020.

[20] M. Tong, W. Lin, X. Huo, Z. Jin, and C. Miao, “A model-free fuzzy adaptive trajectory tracking control algorithm based on dynamic surface control,” *International Journal of Advanced Robotic Systems*, vol. 17, no. 1, 2020.

[21] A. Mohammadi, M. Tavakoli, H. J. Marquez, and F. Hashemzadeh, “Nonlinear disturbance observer design for robotic manipulators,” *Control Engineering Practice*, vol. 21, no. 3, pp. 253–267, 2013.

[22] M. G. Feemster, Y. Fang, and D. M. Dawson, “Disturbance rejection for a magnetic levitation system,” *IEEE/ASME Transactions on Mechatronics*, vol. 11, no. 6, pp. 709–717, 2006.

[23] J. Na, B. Jing, Y. Huang, G. Gao, and C. Zhang, “Unknown system dynamics estimator for motion control of nonlinear robotic systems,” *IEEE Transactions on Industrial Electronics*, vol. 67, no. 5, pp. 3850–3859, 2019.

[24] D. Xie, S. Wang, and Y. Wang, “Trajectory tracking control of differential drive mobile robot based on improved kinematics controller algorithm,” in *Proceedings of the 2018 Chinese Automation Congress (CAC)*, pp. 2675–2680, Xi’an, China, 2018.

[25] R. Bussola, G. Legnani, M. Callegari, G. Palmieri, and M.-C. Palpacelli, “Simulation assessment of the performance of a redundant SCARA,” *Robotics*, vol. 8, no. 2, p. 45, 2019.

[26] E. O. Freire, F. G. Rossomando, and C. M. Soria, “Self-tuning of a neuro-adaptive PID controller for a SCARA robot based on neural network,” *IEEE Latin America Transactions*, vol. 16, no. 5, pp. 1364–1374, 2018.

[27] R. Sharma, V. Kumar, P. Gaur, and A. P. Mittal, “An adaptive PID like controller using mix locally recurrent neural network for robotic manipulator with variable payload,” *ISA Transactions*, vol. 62, pp. 258–267, 2016.

[28] M. A. Al-Khedher and M. S. Alshamasin, “SCARA robot control using neural networks,” in *Proceedings of the 2012 4th International Conference on Intelligent and Advanced Systems (ICIAS2012)*, pp. 126–130, Kuala Lumpur, Malaysia, 2012.

[29] F. G. Rossomando and C. M. Soria, “Discrete-time sliding mode neuro-adaptive controller for SCARA robot arm,” *Neural Computing and Applications*, vol. 28, no. 12, pp. 3837–3850, 2017.

[30] P. Jha and B. B. Biswal, “A neural network approach for inverse kinematic of a SCARA manipulator,” *IAES International Journal of Robotics and Automation*, vol. 3, no. 1, p. 52, 2014.

[31] S. Yi and J. Zhai, “Adaptive second-order fast nonsingular terminal sliding mode control for robotic manipulators,” *ISA Transactions*, vol. 90, pp. 41–51, 2019.

[32] A. D. Rosas, V. K. Velazquez, F. L. Olivares, T. A. Camacho, and I. Williams, “Methodology to assess quality of estimated disturbances in active disturbance rejection control structure for mechanical system,” *ISA Transactions*, vol. 70, pp. 238–247, 2017.

[33] R. Kelly, V. Santibáñez, and A. Loria, *Control of Robot Manipulators in Joint Space*, Springer Science and Business Media, Leipzig, Alemania, 2016.

[34] S. Samarasinghe, *Neural Networks for Applied Sciences and Engineering: From Fundamentals to Complex Pattern Recognition*, Taylor & Francis Group, Boca Raton, FL, USA, 2016.

[35] Al-MahasnehJ. Ahmad et al., “Applications of general regression neural networks in dynamic systems,” *Digital Systems*, vol. 133, 2018.

[36] M. Hou, G. Duan, and M. Guo, “New versions of Barbalat’s lemma with applications,” *Journal of Control Theory and Applications*, vol. 8, no. 4, pp. 545–547, 2010.