frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 03 December 2020
doi: 10.3389/fnbot.2020.577749

PD Control Compensation Based on a Cascade Neural Network Applied to a Robot Manipulator

Luis Arturo Soriano[1]*, *Erik Zamora*[2], *J. M. Vazquez-Nicolas*[3], *Gerardo Hernández*[2], *José Antonio Barraza Madrigal*[4] and *David Balderas*[5]

[1] Departamento de Ingeniería Mecánica Agrícola, Universidad Autónoma Chapingo, Texcoco, Mexico, [2] Laboratorio de Robótica y Mecatrónica, Instituto Politécnico Nacional, CIC, Ciudad de México, Mexico, [3] Unidad Mixta Internacional, French-Mexican Laboratory of Informatics and Automatic Control, 3175 French National Research Council, Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, Deparment of Control of Dynamic Systems, Ciudad de México, Mexico, [4] Unidad Profesional Adolfo López Mateos, Escuela Superior de Ingeniería Química e Industrias Extractivas del Instituto Politécnico Nacional, Academia de Física, Ciudad de México, Mexico, [5] Tecnologico de Monterrey, Escuela de Ingeniería y Ciencias, Ciudad de México, Mexico

OPEN ACCESS

Edited by:
Yongping Pan,
National University of Singapore,
Singapore

Reviewed by:
Heng Liu,
Guangxi University for Nationalities,
China
Huiming Wang,
Chongqing University of Posts and
Telecommunications, China

Correspondence:
Luis Arturo Soriano
lsorianoa@chapingo.mx

Received: *29 June 2020*
Accepted: *14 September 2020*
Published: *03 December 2020*

Citation:
Soriano LA, Zamora E, Vazquez-Nicolas JM, Hernández G, Barraza Madrigal JA and Balderas D (2020) PD Control Compensation Based on a Cascade Neural Network Applied to a Robot Manipulator. Front. Neurorobot. 14:577749. doi: 10.3389/fnbot.2020.577749

A Proportional Integral Derivative (PID) controller is commonly used to carry out tasks like position tracking in the industrial robot manipulator controller; however, over time, the PID integral gain generates degradation within the controller, which then produces reduced stability and bandwidth. A proportional derivative (PD) controller has been proposed to deal with the increase in integral gain but is limited if gravity is not compensated for. In practice, the dynamic system non-linearities frequently are unknown or hard to obtain. Adaptive controllers are online schemes that are used to deal with systems that present non-linear and uncertainties dynamics. Adaptive controller use measured data of system trajectory in order to learn and compensate the uncertainties and external disturbances. However, these techniques can adopt more efficient learning methods in order to improve their performance. In this work, a nominal control law is used to achieve a sub-optimal performance, and a scheme based on a cascade neural network is implemented to act as a non-linear compensation whose task is to improve upon the performance of the nominal controller. The main contributions of this work are neural compensation based on a cascade neural networks and the function to update the weights of neural network used. The algorithm is implemented using radial basis function neural networks and a recompense function that leads longer traces for an identification problem. A two-degree-of-freedom robot manipulator is proposed to validate the proposed scheme and compare it with conventional PD control compensation.

Keywords: cascade neural networks, robot manipulator, PD control, radial basis function, control compensation

1. INTRODUCTION

An industrial robot manipulator frequently works at high velocities to reach its desired position. Common tasks performed by robot manipulators include trajectory tracking, reaching positions, and picking and dropping objects. These tasks need the robot controllers to maintain satisfactory dynamic behavior in spite of possible external perturbations, unknown dynamic parameters, and sensor information loss (Armendariz et al., 2014). Several controllers that are often implemented to manage these features are also mentioned (Luo and Kuo, 2016; Makarov et al., 2016; Nicolis et al., 2016; Pan et al., 2018; Hwang and Yu, 2020). Over time, the Proportional Integral Derivative

(PID) control has been used to design industrial robots due to their simple structure and simple hardware implementation. However, during operation, the PID integral gain provokes the controller to reduce its bandwidth and stability (Rahimi Nohooji, 2020). PD control with uncertainty compensation has been proposed to manage the increase of integral gain due to the steady-state error. The PD controllers are also limited without gravity compensation, which requires a dynamic model (Wen Yu and Rosen, 2013). In practice, the non-linearities of a dynamic robot system are generally unknown.

To solve this issue, different approaches have been developed in order to compensate for unmodeled uncertainties (i.e., noise, gravity, and friction). Intelligent compensation is a model free and it has been applied to well-known algorithms such as the neural networks (NNs) and fuzzy logic (FL) (Krishna and Vasu, 2018; Wang et al., 2019b). In Liu et al. (2020), the authors propose an adaptive NN backstepping control design for fractional-order non-linear systems with actuator faults whose parameters and patterns are fully unknown. Baek et al. (2016) present an adaptive sliding mode control scheme that implements the time-delay estimation. In Xu et al. (2018), a fuzzy NN sliding mode control is designed to improve controller performance against system uncertainty and external disturbances. Kumar et al. (2012) proposed a hybrid trajectory tracking controller for redundant robot manipulators. The adaptive controller is implemented to estimate unstructured uncertainties and error reconstruction. In He et al. (2018), one Radial Basis Function Neural Network (RBFNN) is used to estimate the unknown dynamics robotic manipulator. Jung and Hsia (2000) proposed two NN control schemes for a non-model-based robot manipulator, which show advantages over feedback error learning. In Zhang et al. (2018), the authors proposed a gravity compensation based on an RBFNN and robustness analysis, and the results were compared with a classic PID and PD with fixed gravity compensation. Gandolfo et al. (2019) propose a control scheme that combines a classical PD and a robust adaptive compensator based on NNs.

Although adaptive controllers are addressed for systems with non-linear and uncertainties dynamics, thus their slow convergence can lead to performance degradation or even affect operational safety. In Liu et al. (2019), an adaptive NN control with optimal number of hidden nodes and less computation is formulated for approximating the trajectory of robot manipulator. Similarly, Yang et al. (2018) develop a control and identification scheme in order to identify the unknown robot parameters with an enhanced convergence rate. Another approach is to relax the linear parameterized assumption and the requirements of system knowledge, thus, NNs have been used as function approximators. In time series modeling, RBFNN is commonly used for function approximation, since its value is different from zero in infinite space, and its approximation can avoid the local minimum (Wang et al., 2019a). An RBFNN uses a Gauss function as its activation function. In general, RBFNN controllers waste less computational resources in comparison to other NN controllers (He et al., 2018). In Wang et al. (2012), the authors proposed an RBFNN to compensate for non-linear dynamics of the robotic manipulator and a robust control designed to suppress the modeling error of NN.

However, update laws commonly increase the weight magnitudes until the output error has been mitigated, without a robust design continued training can lead to excessive control effort. In order to avoid this, adaptive controls frequently update the neural weights according to robust adaptive laws, which are computed with Lyapunov methods (Razmi and Macnab, 2020). In this work, a robust adaptive control design to compensate a nominal controller for robot manipulator with uncertainties and external perturbations is formulated. Moreover, a scheme based on two RBFNN in cascade is proposed in order to improve the response of the nominal controller. In the scheme aforementioned, the first NN is used to estimate the error and the second uses the estimation error value to improve the output of the nominal controller. NN weights are online updated by developing new adaptive laws. The adaptive law based on the gradient is modified by introducing a recompense function of the online error in order to improve the convergence of the NN weights.

In this work, a PD control with a scheme based on NNs in cascade is designed to manage the compensation of uncertainties in a robot manipulator. The main contributions of this paper are summarized as follows:

- In order to improve the robustness of the system against external disturbance, and unknown system parameters, a scheme of cascade NNs is proposed.
- A recompense function for neural weights updates is proposed in order to improve the NNs' weights convergence.
- The response of the nominal controller is improved.

To validate the proposed scheme and compare it with conventional neural compensation, a two-degree-of-freedom robot manipulator (TDOFRM) is proposed. This paper is organized as follows: Section 2 presents the preliminary mathematical model of the TDOFRM, the conventional PD compensation with RBFNN. Section 3 describes the controller design based on NNs in cascade. Section 4 presents simulation experiments and compared with the conventional compensation, and finally, section 5 presents the conclusions.

2. ADAPTIVE CONTROL TO ROBOTIC MANIPULATOR

2.1. Dynamic Model of Robotic Manipulator

The dynamic model of an n degree of freedom robot manipulator can be described as follows (Spong and Vidyasagar, 1989):

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

$M(q)$ is a $n \times n$ inertia matrix, $C(q,\dot{q})$ is a $n \times n$ is a centrifugal and Coriolis matrix, and $g(q)$ is a $n \times 1$ vector of gravity. $q, \dot{q}, \ddot{q}$ are the position, velocity, and acceleration of each link, respectively. $\tau \in R^n$ is the control input and d denotes disturbances.

2.2. Proportional Derivative Control Scheme

In industrial application, the exact model is difficult to obtain and external disturbances are always present in practice. According to Liu (2013), a nominal model of robot manipulator can be computed as $M_0(q)$, $C_0(q,\dot{q})$, and $g_0(q)$. Considering $\Delta M = M_0 - M$, $\Delta C = C_0 - C$, and $\Delta g = g_0 - g$, Equation (1) is reordered as follows:

$$(M_0(q) - \Delta M)\ddot{q} + (C_0(q,\dot{q}) - \Delta C)\dot{q} + (g_0(q) - \Delta g) = \tau + d \quad (2)$$

Thus,

$$M_0(q)\ddot{q} + C_0(q,\dot{q})\dot{q} + g_0(q) = \tau + \Delta M\ddot{q} + \Delta C\dot{q} + \Delta g + d \quad (3)$$

Defining $f(\cdot) = \Delta M\ddot{q} + \Delta C\dot{q} + \Delta g + d$ and if $f(\cdot)$ is known, the control law is designed as

$$M_0(q)(\ddot{q_d} - \kappa_v\dot{e} - \kappa_p e) + C_0(q,\dot{q})\dot{q} + g_0(q) - f(\cdot) = \tau \quad (4)$$

Submitting Equation (4) into Equation (3), the close loop system can be expressed as follows:

$$\ddot{e} + \kappa_v\dot{e} + \kappa_p e = 0 \quad (5)$$

where $e = q - q_d$, $\dot{e} = \dot{q} - \dot{q}_d$, and $\ddot{e} = \ddot{q} - \ddot{q}_d$. Frequently, $f(\cdot)$ in industrial applications is unknown, hence, $f(\cdot)$ requires to be estimated and compensated.

2.3. Radial Basis Function Neural Network Approximation

The NNs approximates $M(q)$, $C(q)$, and $g(q)$ when they are unknown. The Radial Basis Function (RBF) algorithm can approximate a continuous function and it is defined as

$$\phi_i = exp^{\left(-\frac{\|x-c_i\|^2}{b_i^2}\right)}, i = 1,2,\ldots,n \quad (6)$$

$$y = W\phi \quad (7)$$

where x is the input vector, $\phi = [\phi_1, \phi_2, \ldots, \phi_n]$ is the output of the Gaussian function, y is the output of the NN, W is the weight values matrix, and c_i is the center and b_i is the width of the Gaussian function. In Gandolfo et al. (2019), Liu et al. (2019), it has been shown that an RBFNN can approximate a non-linear function $f(\cdot)$ under the following assumptions

1. The output $\hat{f}(x, W^*)$ is a continuous function.
2. Given a small positive constant ϵ_0 and a continuous function $f(\cdot)$, a weight vector W^* exists so that $\hat{f}(\cdot)$ satisfies.

$$max\|f(\cdot) - \hat{f}^*(\cdot)\| \leq \varepsilon_0 \quad (8)$$

and $W^* = \underset{W\in\beta(M_W)}{\text{argmin}}\left\{\underset{W\in\beta(M_W)}{\sup}\|f(\cdot) - \hat{f}^*(\cdot)\|\right\}$, where W^* is $n \times n$ matrix that denotes the optimal weigh values for $f(\cdot)$ approximation.

2.4. Adaptive Law to Compensation Control

In the controller scheme proposed (Feng, 1995), the close loop system is given by the following equation:

$$\tau = M_0(q)(\ddot{q_d} - \kappa_v\dot{e} - \kappa_p e) + C_0(q,\dot{q})\dot{q} + g_0(q) - \hat{f}(\cdot) \quad (9)$$

where $\hat{f}(\cdot) = \hat{W}\phi(x)$ and $\hat{W}$ is an estimation of W^*. Equations (1) and (9) have the same term, and the substitution result is shown in the following equation:

$$\begin{aligned} &M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) - d \\ &\quad = M_0(q)(\ddot{q_d} - \kappa_v\dot{e} - \kappa_p e) + C_0(q,\dot{q})\dot{q} + g_0(q) - \hat{f}(\cdot) \end{aligned} \quad (10)$$

Then, the equation $M_0(q)\ddot{q} + C_0(q,\dot{q})\dot{q} + g_0(q)$ is subtracted with Equation (10) in both sides as follows:

$$\begin{aligned} &M_0(q)\ddot{q} + C_0(q,\dot{q})\dot{q} + g_0(q) - \left[M(q)\ddot{q} + C(q,\dot{q})\dot{q} + g(q) - d\right] \\ &\quad = M_0(q)\ddot{q} + C_0(q,\dot{q})\dot{q} + g_0(q) - \left[M_0(q)(\ddot{q_d} - \kappa_v\dot{e} - \kappa_p e)\right. \\ &\quad \left. + C_0(q,\dot{q})\dot{q} + g_0(q) - \hat{f}(\cdot)\right] \end{aligned} \quad (11)$$

The result is given as follows:

$$\begin{aligned} &M_0^{-1}(q)\left[\Delta M\ddot{q} + \Delta C\dot{q} + \Delta g(q) + d\right] \\ &\quad = \ddot{e} + \kappa_v\dot{e} + \kappa_p e + M_0^{-1}(q)\left[\hat{f}(\cdot)\right] \end{aligned} \quad (12)$$

Equation (12) can be rewritten as follows:

$$\ddot{e} + \kappa_v\dot{e} + \kappa_p e = M_0^{-1}(q)\left[f(\cdot) - \hat{f}(\cdot)\right] \quad (13)$$

Select to $x = (\, e \;\; \dot{e} \,)^T$. Equation (13) turns into

$$\dot{x} = Ax + B(f(\cdot) - \hat{f}(\cdot)) \quad (14)$$

where

$$A = \begin{pmatrix} 0 & I \\ -\kappa_p & -\kappa_v \end{pmatrix}; B = \begin{pmatrix} 0 \\ M_0^{-1}(q) \end{pmatrix} \quad (15)$$

Setting $f(\cdot) - \hat{f}(\cdot) = f(\cdot) - \hat{f}^*(\cdot) + \hat{f}^*(\cdot) - \hat{f}(\cdot) = \zeta + W^{*T}\phi - \hat{W}^T\phi = \zeta - \tilde{W}^T\phi$ and $\tilde{W} = \hat{W} - W^*, \zeta = f(\cdot) - \hat{f}^*(\cdot)$, where ζ denotes the modeling error due to the use of the NN. The modeling error ζ is bounded by a finite constant ζ_0, where $\zeta_0 = \underset{t\geq 0}{\sup}\|f(\cdot) - \hat{f}^*(\cdot)\|$. Finally,

$$\dot{x} = Ax + B(\zeta - \tilde{W}^T\phi) \quad (16)$$

The Lyapunov function is given by the following equation:

$$L = \frac{1}{2}x^T Px + \frac{1}{2\varphi}\|\tilde{W}\|^2; \varphi > 0 \quad (17)$$

where $\tilde{W} = \hat{W} - W^*$ is a definition that describes the estimation error. In Equation (17), P is a positive definite matrix that satisfies the Lyapunov equation

$$PA + A^T P = -Q; Q \geq 0 \quad (18)$$

Equation (17) can be rewritten in terms of the next definition

$$\| \tilde{W} \|^2 = \sum_{i,j} |w_{ij}|^2 = tr(\tilde{W}\tilde{W}^T) = tr(\tilde{W}^T\tilde{W}) \tag{19}$$

Thus, the derivative of V is given as follows:

$$\dot{L} = \frac{1}{2}[x^T P\dot{x} + \dot{x}^T Px] + \frac{1}{\varphi} tr(\dot{\tilde{W}}^T \tilde{W}) \tag{20}$$

Substituting Equation (16) for (20), the results is given by

$$\begin{aligned}
\dot{L} &= \frac{1}{2}[x^T P(Ax + B(\zeta - \tilde{W}^T\phi)) \\
&\quad +(Ax + B(\zeta - \tilde{W}^T\phi))^T Px] + \frac{1}{\varphi} tr(\dot{\tilde{W}}^T\tilde{W}) \\
&= \frac{1}{2}[x^T PAx + x^T PB(\zeta - \tilde{W}^T\phi) + x^T A^T Px) \\
&\quad +(\zeta - \tilde{W}^T\phi)^T B^T Px] + \frac{1}{\varphi} tr(\dot{\tilde{W}}^T\tilde{W}) \\
&= \frac{1}{2}[x^T PAx + x^T A^T Px + x^T PB(\zeta - \tilde{W}^T\phi)) \\
&\quad +(\zeta - \tilde{W}^T\phi)^T B^T Px] + \frac{1}{\varphi} tr(\dot{\tilde{W}}^T\tilde{W}) \\
&= \frac{1}{2}[x^T(PA + A^T P)x + x^T PB\zeta - x^T PB\tilde{W}^T\phi) \\
&\quad +\zeta^T B^T Px - \phi^T\tilde{W}B^T Px] + \frac{1}{\varphi} tr(\dot{\tilde{W}}^T\tilde{W}) \\
&= -\frac{1}{2}[x^T Qx] + \zeta^T B^T Px - B^T\tilde{W}\phi^T Px + \frac{1}{\varphi} tr(\dot{\tilde{W}}^T\tilde{W}) (21)
\end{aligned}$$

Considering that $x^T PB\zeta = \zeta^T B^T Px$, $x^T PB\tilde{W}^T\phi = \phi^T\tilde{W}B^T Px$ and $\phi^T\tilde{W}B^T Px = tr[B^T Px\phi^T\tilde{W}]$ in Equation (21), this results in

$$\dot{L} = -\frac{1}{2}[x^T Qx] + \zeta^T B^T Px + \frac{1}{\varphi} tr(-\varphi B^T Px\phi^T\tilde{W} + \dot{\tilde{W}}^T\tilde{W}) \tag{22}$$

2.5. Adaptive Control Based on Cascade Neural Network

The PD control has been widely implemented in robot control to deal with the drawbacks presented by the integer gain in PID control. However, the PD control can have similar deficiencies if the derivative gain has high values (Wen Yu and Rosen, 2013). The PD control with compensation presents positive results to avoid high derivative gains and identify uncertainties that occur in the real operation of robot manipulators. In this work, a compensation of the nominal controller is proposed. The adaptive control scheme is shown in **Figure 1**. Two RBFNN in cascade are proposed to deal with the tracking error and the NN weight estimation error.

2.5.1. NN Weight Estimation Error

The input of NN is an error vector defined as follows:

$$x = \begin{bmatrix} e \\ \dot{e} \end{bmatrix} = \begin{bmatrix} q_i - q_{\vartheta i} \\ \dot{q}_i - \dot{q}_{\vartheta i} \end{bmatrix} \tag{23}$$

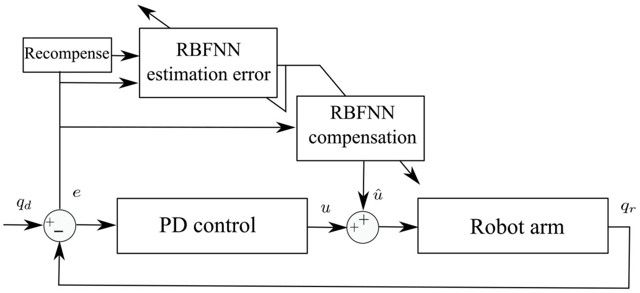

FIGURE 1 | Proposed adaptive control scheme.

where i and ϑ denote the real and desired position and velocity of the n-link of the robot. The NN estimation error anticipates which action to take in order to improve the output of the nominal control. Thus, the prediction of estimation error is also given as the output of RBFNN as follows:

$$\hat{x} = \sum_{j=1}^{J} W_e\phi_e \tag{24}$$

where ϕ_e is the Gaussian function given by Equation (6). The criterion for the weight update is proposed in the following equation:

$$\dot{W}_e = \gamma_e r\phi_e(x)^T PB \tag{25}$$

where $r = \| e^{-x_e^2} \|$ is a function that represents the recompense signal. The goal of the recompense function is to lead longer traces for an identification problem.

2.5.2. Adaptive Law to Compensate the Nominal Controller

A novel adaptive control scheme is proposed to ensure that the output of the nominal controller for the system defined in Equation (1) reaches the position desired, and the estimated NN can converge an ideal weight. An RBFNN is selected to approximate the system dynamics and deals with uncertainties and external disturbances, which is given as follows:

$$\hat{f}^* = \sum_{j=1}^{J} \hat{W}\phi_c \tag{26}$$

where ϕ_c is a Gaussian function given by the equation form 6. The NN actor provides compensation to the PD controller to, that is, $\hat{f}(\cdot) = \hat{f}^*$. The weigh update of the NN actor is proposed according to the following equation:

$$\dot{\hat{W}} = \gamma_c\phi_c(\hat{x} + (\hat{x} - x))^T PB \tag{27}$$

If we select the parameter update law as $\dot{\tilde{W}} = \dot{\hat{W}}$, we assume that the value $x \approx \hat{x} + (\hat{x} - x)$. Substituting Equation (27) into Equation (22), the result is given as

$$\dot{L} = -\frac{1}{2}[x^T Qx] + \zeta^T B^T Px \tag{28}$$

As it is well-known, $\|\zeta^T\| \leq \|\zeta_0\|$ and from Equation (15), $\|B\| = \|M_0^{-1}(q)\|$, and Equation (28) can be written as follows:

$$\dot{L} \leq -\tfrac{1}{2}\lambda_{min}(Q)\|x\|^2 + \|\zeta_0\|\|M_0^{-1}(q)\|\lambda_{max}(P)\|x\| \quad (29)$$

$$\dot{L} = -\tfrac{1}{2}\|x\|\left[\lambda_{min}(Q)\|x\| - 2\|\zeta_0\|\|M_0^{-1}(q)\|\lambda_{max}(P)\right] \quad (30)$$

where λ_{min} denotes the minimum eigenvalues of matrix Q and λ_{max} denotes the maximum eigenvalues of matrix P. In order to satisfy $\dot{L} \leq 0$, the value of $\|x\|$ should be satisfied as follows:

$$\|x\| \leq \frac{2\|M_0^{-1}(q)\|\lambda_{max}(P)}{\lambda_{min}Q}\|\zeta_0\| \quad (31)$$

According to Equation (28), $\dot{L}$ is negative semidefinite, that is $L(x, W, t) \leq L(x, W, 0)$. It implies that x, and W are bounded. Let function $\Omega = \dot{L}$ and integrate Ω with respect to time as follows:

$$\int_0^t \Omega(s)ds \leq L(x, W, t) \leq L(x, W, 0) \quad (32)$$

Due to $L(x, W, 0)$ is bounded, and $L(x, W, t)$ is non-increasing and bounded, the following result can be computed:

$$\lim_{t\to\infty}\int_0^t \Omega(s)ds < \infty \quad (33)$$

Since $\dot{\Omega}(s)$ is bounded, by Barbalat's lemma (Slotine and Li, 1991), $\lim_{t\to\infty}\Omega(s) = 0$, that is $x \to 0$ as $t \to \infty$.

3. RESULTS

In order to validate the proposed scheme of control, a set of simulations was carried out. Two simulations are proposed, the first considers an adaptive control using an RBFNN, and the second is based on the proposed scheme using two RBFNNs in cascade. The controllers were also implemented in a TDOFRM, which is shown in **Figure 2**. The main objective of controllers is the position tracking in the presence of external disturbances.

$$\begin{bmatrix}\boldsymbol{\tau}_1\\ \boldsymbol{\tau}_2\end{bmatrix} = \begin{bmatrix}M_{11}(q) & M_{12}(q)\\ M_{21}(q) & M_{22}(q)\end{bmatrix}\begin{bmatrix}\ddot{q}_1\\ \ddot{q}_2\end{bmatrix} + \begin{bmatrix}C_{11}(q,\dot{q}) & C_{12}(q,\dot{q})\\ C_{21}(q,\dot{q}) & C_{22}(q,\dot{q})\end{bmatrix}\begin{bmatrix}\dot{q}_1\\ \dot{q}_2\end{bmatrix} + \begin{bmatrix}g_1(\dot{q})\\ g_2(\dot{q})\end{bmatrix} \quad (34)$$

where:

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

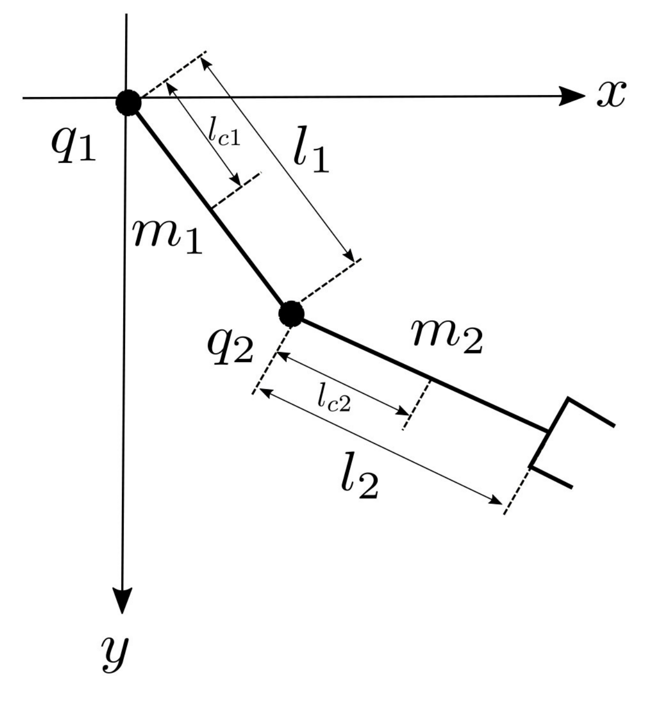

FIGURE 2 | Model of two degree-of-freedom (DOF) robot manipulator.

TABLE 1 | Robot manipulator parameters.

Parameter	Value	Units
l_1	0.45	m
l_2	0.45	m
l_{c1}	0.091	m
l_{c2}	0.048	m
m_1	23.902	kg
m_2	3.880	kg
I_1	1.266	kgm^2
I_1	0.093	kgm^2
g	9.81	$\frac{m}{s^2}$

$$\begin{aligned}
g_1(q) &= (m_1 l_{c1} + m_2 l_1) g sin(q_1) + m_2 l_{c2} g sin(q_1 + q_2)\\
g_2(q) &= m_2 l_{c2} g sin(q_1 + q_2)
\end{aligned} \quad (35)$$

The parameters came from Kelly and Santibáñez (2003) and presented in **Table 1**.

The desired position vector is defined as follows:

$$\boldsymbol{q}_d = \begin{bmatrix}\boldsymbol{q}_{1d}\\ \boldsymbol{q}_{2d}\end{bmatrix} = \begin{bmatrix}\frac{\pi}{6}sin(\frac{1}{4}\pi t)\\ \frac{\pi}{6}sin(\frac{2}{3}\pi t)\end{bmatrix} \quad (36)$$

The initial positions are given by $q_0 = \begin{bmatrix}\frac{\pi}{12} & \frac{\pi}{12}\end{bmatrix}^T$. The uncertainties and unknown disturbances are defined as follows:

$$d = \begin{bmatrix}sin(2\pi t) + cos(\frac{1}{4}\pi t)\\ sin(2\pi t) + cos(\frac{1}{4}\pi t)\end{bmatrix} \quad (37)$$

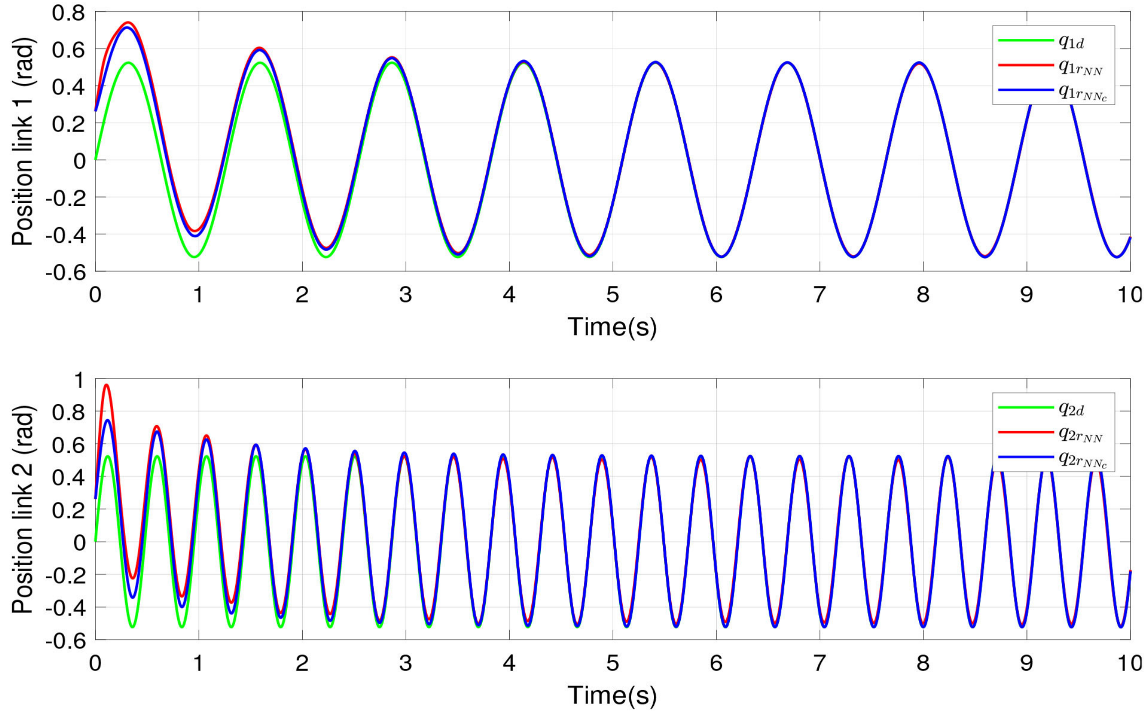

FIGURE 3 | Real and desired links positions of the robot manipulator.

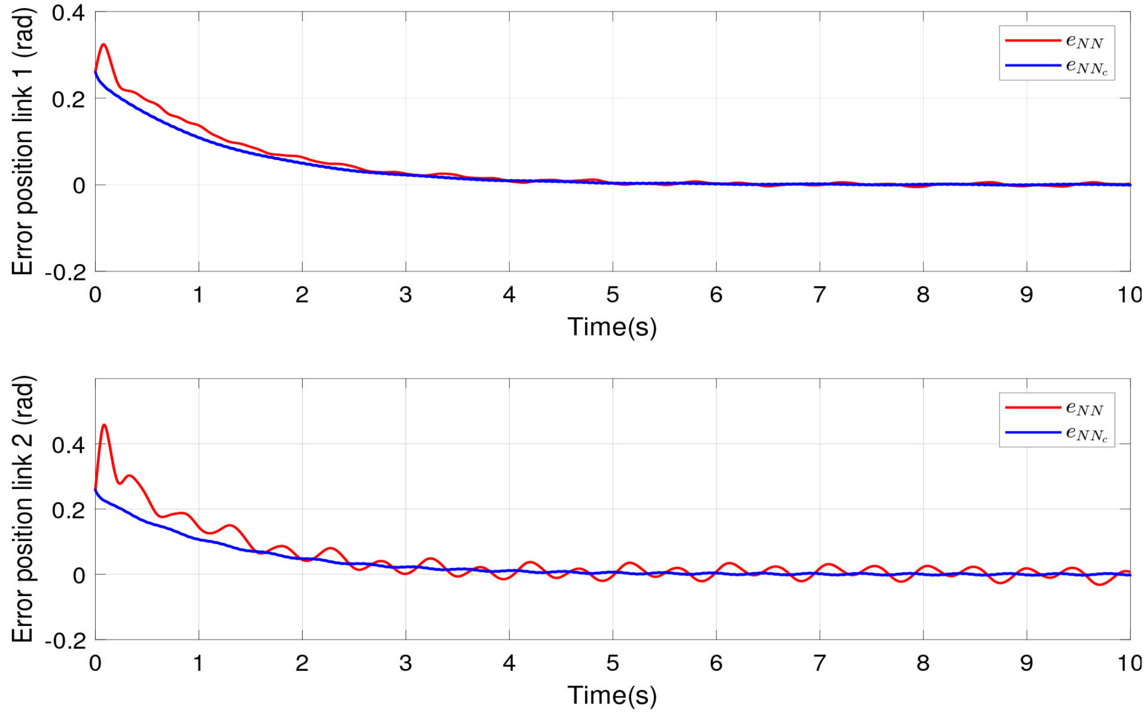

FIGURE 4 | Tracking error of the robot manipulator joints by scheme proposed and conventional compensation.

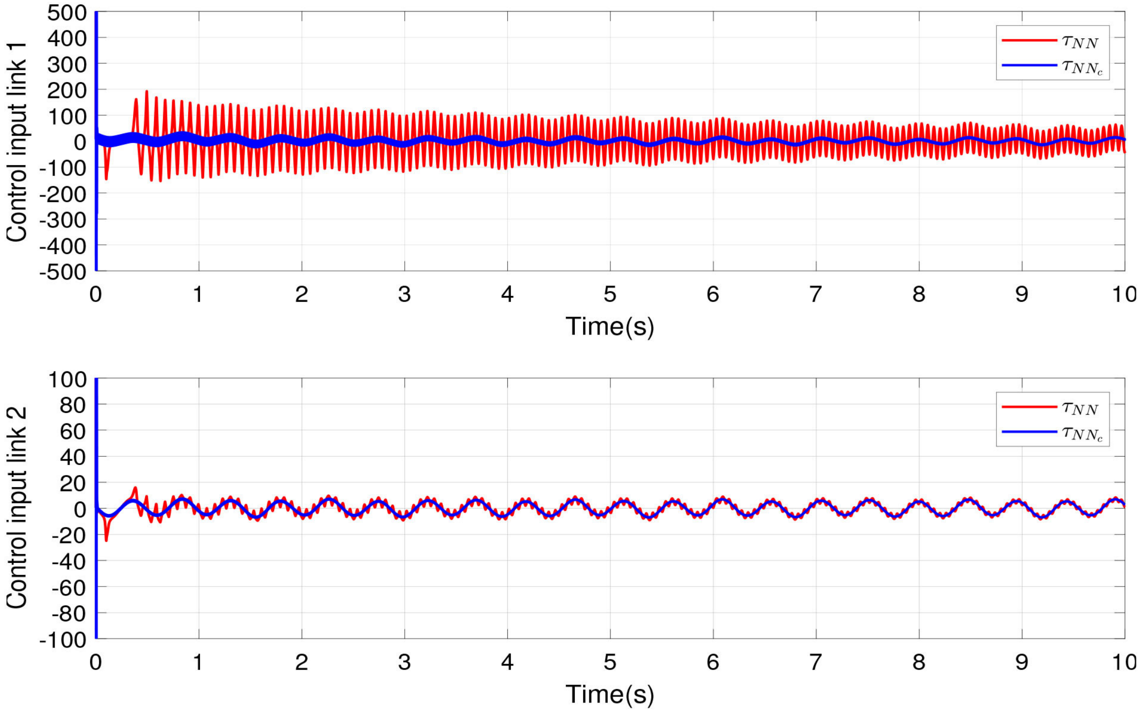

FIGURE 5 | Control inputs of links 1 and 2.

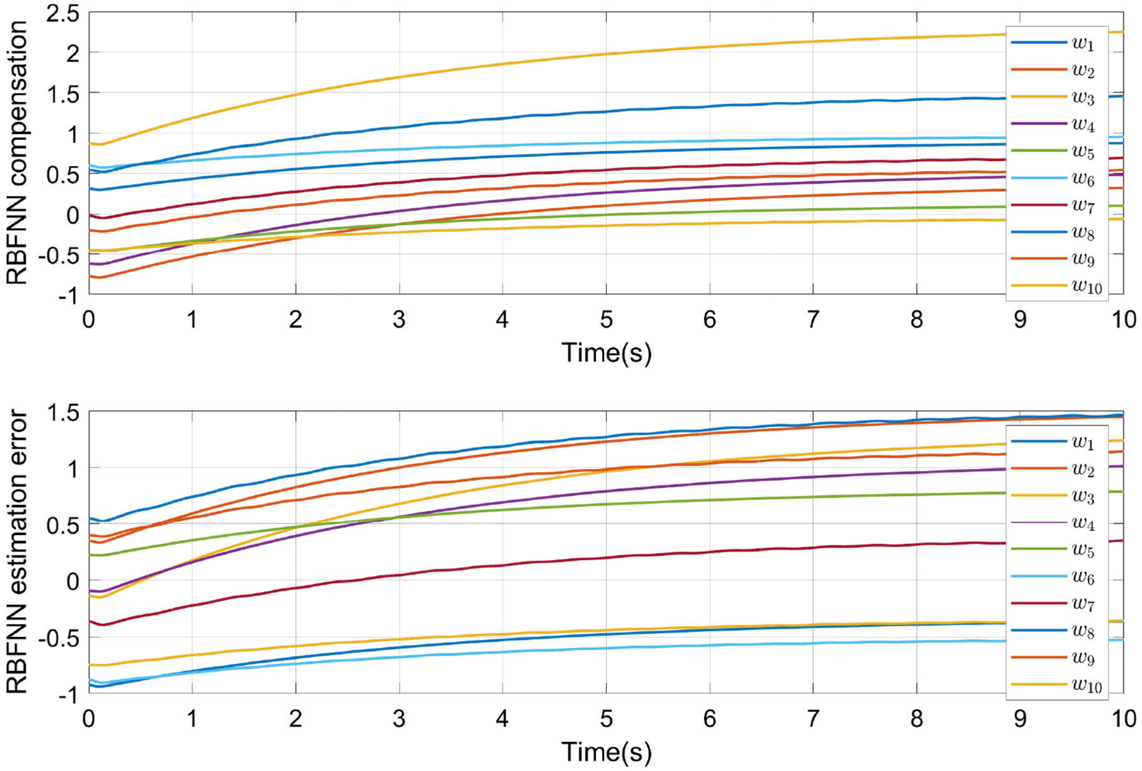

FIGURE 6 | Convergence of neural networks parameters.

The matrix Q, A, and B are

$$Q = \begin{bmatrix} 50 & 0 & 0 & 0 \\ 0 & 50 & 0 & 0 \\ 0 & 0 & 50 & 0 \\ 0 & 0 & 0 & 50 \end{bmatrix};$$

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -0.25 & 0 & -1 & 0 \\ 0 & -0.25 & 0 & -1 \end{bmatrix}; B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{38}$$

Two RBFNNs are proposed for critic and actor agents. Here, γ_e is the learning rate of the RBFNN estimation error and γ_c is the learning rate of the RBFNN tracking error; c_i is the center vector of neural net i and b is the width value of Gaussian function for neural net i. The next values are proposed due to the optimal weight for the actor NN and critic NN could take arbitrary large values, but in order to avoid any numerical problems, this work considers $\gamma_e = 0.5$, $\gamma_c = 0.4$, $c_i = \begin{bmatrix} -2 & -1 & 0 & 1 & 2 \end{bmatrix}$, $b_i = 0.5$, and $i = 4$.

Figure 3 shows the tracking position of links 1 and 2, where *NN* and NN_c indicate an RBFNN compensation and a compensation based on two RBFNN in cascade, respectively. The green lines show the desired tracking position. The red lines indicate the tracking position of NN compensation. The blue lines show the tracking position of the proposed controller. The uncertainties and disturbances were added to the controller. According to the RBFNN in cascade, an RBFNN predicts the NN estimation error, and this value is included in adaptive laws to update the RBFNN compensation in order to take adequate action for the disturbances and guarantee the convergence of tracking error. The proposed recompense function helps to maintain longer traces for the identification task over time.

The compensation of proposed algorithm is compared with an RBFNN compensation. **Figure 4** shows tracking errors for links 1 and 2. The red line represents the tracking error of an RBFNN for compensation, which presents overshoot to reach the desired positions and oscillations in steady state. The blue line indicates the tracking error of the proposed algorithm, which present robustness against uncertainties and disturbances. The green line indicates the desired tracking positions for links 1 and 2.

The simulations were proposed in order to show the difference between adaptive conventional control and the proposed scheme. In this sense, two desired tracking signal was proposed that goes at different velocities, and link 1 follows a slow signal and link 2 follows a fast signal. **Figure 5** shows the control inputs to links 1 and 2, and it also exhibits the improvement of the nominal controller under our scheme proposed in comparison with adaptive conventional control.

In **Figure 6**, the NN weights convergence process of the two RBFNN in cascade are shown. **Figure 6** also denoted as the identification process in order to deal with the uncertainties and external disturbances is reached.

The important factors that usually must be considered together are time and error. A performance index is a measure that indicates those features of the response that are regarded to be important. In order to evaluate the performance of the proposed controller, a comparison of different performance indices is shown in **Table 2**. Hence, **Table 2** is based on the next four equations, integral absolute error (IAE), integral square error (ISE), integral time absolute error (ITAE), and integral time square error (ITSE).

TABLE 2 | Comparison of different errors, ITAE, ITSE, IAE, and ISE as performance indices.

Controller	Indices	Link 1	Link 2
PD+NN	IAE	0.3758	0.4992
	ISE	0.0554	0.0869
	ITAE	0.5162	1.0030
	ITSE	0.0340	0.0578
PD+NN$_C$	IAE	0.3057	0.3077
	ISE	0.0373	0.0370
	ITAE	0.3898	0.4148
	ITSE	0.0232	0.0231

$$ISE = \int e(t)^2 dt \tag{39}$$

$$IAE = \int \mid e(t) \mid dt \tag{40}$$

$$ITSE = \int t\, e(t)^2 dt \tag{41}$$

$$ITAE = \int t \mid e(t) \mid dt \tag{42}$$

Table 2 shows that the proposed controller presents a better response than the conventional PD control compensation based on an RBFNN. According to the performance indices of links 1 and 2, the proposed scheme presents an adequate response against external disturbance. Moreover, it presents less oscillation in steady state and less time in the transient response than the conventional compensation based on an RBFNN.

4. CONCLUSIONS

The algorithm proposed has been implemented to compensate for the PD control of a TDOFRM. A PD control was selected because the common knowledge that if designed with gravity compensation, it can reach asymptotic stability. The cascade scheme was implemented by two RBFNN in cascade, which compensates for the control input in order to deal with the estimation error, uncertainties, and external disturbances. The proposed algorithm was validated using a simulation of a TDOFRM. Two adaptive algorithms for compensating for the controller of robot manipulators were implemented. The first was based on a conventional RBFNN, and the second is the proposed algorithm that uses two RBFNN. An adaptive law is proposed to deal with the simultaneous convergence of both NNs, which are used to estimate for tracking error and estimation error. The results showed that the proposed compensation scheme based on RBFNN presents robustness against external uncertainties and disturbances, also an adequate convergence for the NN weights. Position tracking has been reached without overshoots and oscillations in steady state in comparison to compensations with a conventional RBFNN scheme.

DATA AVAILABILITY STATEMENT

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

AUTHOR CONTRIBUTIONS

JB and GH proposed a two-degree-of freedom robot manipulator to validate the scheme proposed and they also design the robot manipulator simulation. JV-N and DB contributed to the analysis and discussion of results. LS and EZ contributed with the theoretical analysis, simulations, and also written the main text. All authors participated and contributed to the final version manuscript.

REFERENCES

Armendariz, J., Parra-Vega, V., Garcia-Rodriguez, R., and Rosales, S. (2014). Neuro-fuzzy self-tuning of PID control for semiglobal exponential tracking of robot arms. *Appl. Soft Comput*. 25, 139–148. doi: 10.1016/j.asoc.2014.08.037

Baek, J., Jin, M., and Han, S. (2016). A new adaptive sliding-mode control scheme for application to robot manipulators. *IEEE Trans. Indus. Electron*. 63, 3628–3637. doi: 10.1109/TIE.2016.2522386

Feng, G. (1995). A compensating scheme for robot tracking based on neural networks. *Robot. Auton. Syst*. 15, 199–206. doi: 10.1016/0921-8890(95)00023-9

Gandolfo, D. C., Rossomando, F. G., Soria, C. M., and Carelli, R. O. (2019). Adaptive neural compensator for robotic systems control. *IEEE Latin Am. Trans*. 17, 670–676. doi: 10.1109/TLA.2019.8891932

He, W., Huang, B., Dong, Y., Li, Z., and Su, C.-Y. (2018). Adaptive neural network control for robotic manipulators with unknown deadzone. *IEEE Trans. Cybernet*. 48, 2670–2682. doi: 10.1109/TCYB.2017.2748418

Hwang, C.-L., and Yu, W.-S. (2020). Tracking and cooperative designs of robot manipulators using adaptive fixed-time fault-tolerant constraint control. *IEEE Access* 8, 56415–56428. doi: 10.1109/ACCESS.2020.2979795

Jung, S., and Hsia, T. (2000). Neural network inverse control techniques for PD controlled robot manipulator. *Robotica* 18, 305–314. doi: 10.1017/S0263574799002064

Kelly, R., and Santibáñez, V. (2003). *Control de Movimiento de Robots Manipuladores*. Madrid: Pearson Educación; Prentice Hall.

Krishna, S., and Vasu, S. (2018). Fuzzy PID based adaptive control on industrial robot system. *Mater. Tdy Proc*. 5(5 Pt 2), 13055–13060. doi: 10.1016/j.matpr.2018.02.292

Kumar, N., Borm, J.-H., Panwar, V., and Chai, J. (2012). Tracking control of redundant robot manipulators using RBF neural network and an adaptive bound on disturbances. *Int. J. Precis. Eng. Manufactur*. 13, 1377–1386. doi: 10.1007/s12541-012-0181-5

Liu, C., Zhao, Z., and Wen, G. (2019). Adaptive neural network control with optimal number of hidden nodes for trajectory tracking of robot manipulators. *Neurocomputing* 350, 136–145. doi: 10.1016/j.neucom.2019.03.043

Liu, H., Pan, Y., Cao, J., Wang, H., and Zhou, Y. (2020). Adaptive neural network backstepping control of fractional-order nonlinear systems with actuator faults. *IEEE Trans. Neural Netw. Learn. Syst*. doi: 10.1109/TNNLS.2020.3027335. [Epub ahead of print].

Liu, J. (2013). *Radial Basis Function (RBF) Neural Network Control for Mechanical Systems*. Berlin; Heidelberg: Springer. doi: 10.1007/978-3-642-34816-7

Luo, R. C., and Kuo, C.-W. (2016). Intelligent seven-DoF robot with dynamic obstacle avoidance and 3-D object recognition for industrial cyber-physical systems in manufacturing automation. *Proc. IEEE* 104, 1102–1113. doi: 10.1109/JPROC.2015.2508598

Makarov, M., Grossard, M., Rodriguez-Ayerbe, P., and Dumur, D. (2016). Modeling and preview H_∞ control design for motion control of elastic-joint robots with uncertainties. *IEEE Trans. Indus. Electron*. 63, 6429–6438. doi: 10.1109/TIE.2016.2583406

Nicolis, D., Zanchettin, A. M., and Rocco, P. (2016). Constraint-based and sensorless force control with an application to a lightweight dual-arm robot. *IEEE Robot. Automat. Lett*. 1, 340–347. doi: 10.1109/LRA.2016.2517206

Pan, Y., Wang, H., Li, X., and Yu, H. (2018). Adaptive command-filtered backstepping control of robot arms with compliant actuators. *IEEE Trans. Control Syst. Technol*. 26, 1149–1156. doi: 10.1109/TCST.2017.2695600

Rahimi Nohooji, H. (2020). Constrained neural adaptive PID control for robot manipulators. *J. Franklin Instit*. 357, 3907–3923. doi: 10.1016/j.jfranklin.2019.12.042

Razmi, M., and Macnab, C. (2020). Near-optimal neural-network robot control with adaptive gravity compensation. *Neurocomputing* 389, 83–92. doi: 10.1016/j.neucom.2020.01.026

Slotine, J.-J. E., and Li, W. (1991). *Applied Nonlinear Control*. Englewood Cliffs, NJ: Prentice Hall.

Spong, M. W., and Vidyasagar, M. (1989). *Robot Dynamics and Control*. New York, NY: Wiley.

Wang, F., Chao, Z.-Q., Huang, L.-B., Li, H.-Y., and Zhang, C.-Q. (2019a). Trajectory tracking control of robot manipulator based on RBF neural network and fuzzy sliding mode. *Cluster Comput*. 22, 5799–5809. doi: 10.1007/s10586-017-1538-4

Wang, H., Lei, Z., Zhang, X., Zhou, B., and Peng, J. (2019b). A review of deep learning for renewable energy forecasting. *Energy Convers. Manage*. 198, 1–16. doi: 10.1016/j.enconman.2019.111799

Wang, L., Chai, T., and Yang, C. (2012). Neural-network-based contouring control for robotic manipulators in operational space. *IEEE Trans. Control Syst. Technol*. 20, 1073–1080. doi: 10.1109/TCST.2011.2147316

Xu, J., Wang, Q., and Lin, Q. (2018). Parallel robot with fuzzy neural network sliding mode control. *Adv. Mech. Eng*. 10, 1–8. doi: 10.1177/1687814018801261

Yang, C., Jiang, Y., He, W., Na, J., Li, Z., and Xu, B. (2018). Adaptive parameter estimation and control design for robot manipulators with finite-time convergence. *IEEE Trans. Indus. Electron*. 65, 8112–8123. doi: 10.1109/TIE.2018.2803773

Yu, W., and Rosen, J. (2013). Neural PID control of robot manipulators with application to an upper limb exoskeleton. *IEEE Trans. Cybernet*. 43, 673–684. doi: 10.1109/TSMCB.2012.2214381

Zhang, H., Du, M., Wu, G., and Bu, W. (2018). PD control with RBF neural network gravity compensation for manipulator. *Eng. Lett*. 26, 236–244.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Soriano, Zamora, Vazquez-Nicolas, Hernández, Barraza Madrigal and Balderas. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.